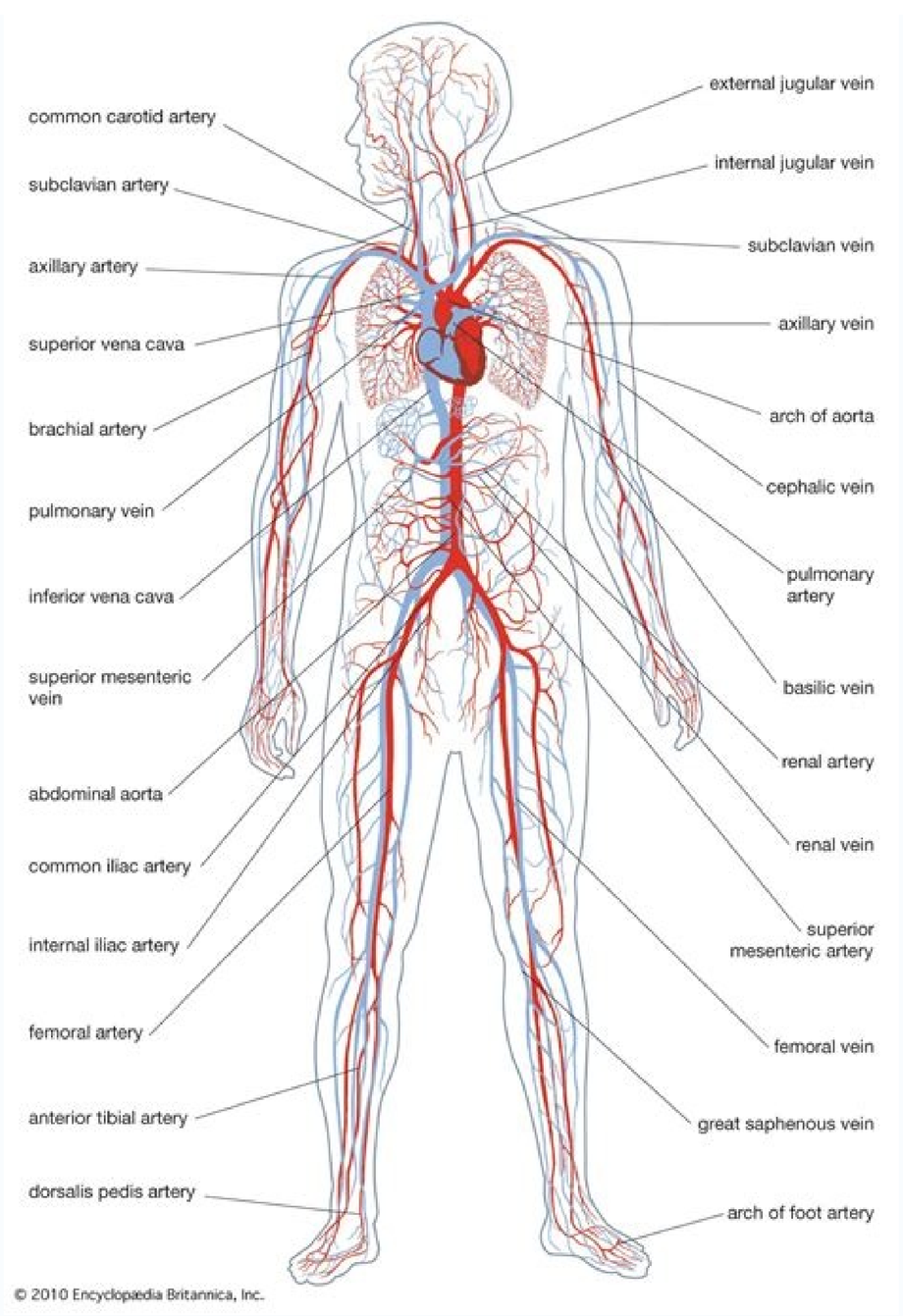
Embedded systems questions and answers



Embedded systems objective type questions and answers. Embedded systems questions and answers pdf. Embedded systems exam questions and answers pdf. Embedded systems objective questions and answers pdf. Embedded systems mcq questions and answers. Embedded systems lab viva questions and answers. Technical interview questions and answers on embedded systems pdf. Embedded and real time systems mcq questions and answers pdf.

In this moderately long page, I'll list down the most common embedded systems interview questions. I've been answering too many questions over the past few years and many readers have asked for a compiled list of questions and answers for embedded systems and embedded c interview questions. So here it is!Whether you're a student searching for an embedded systems internship, a graduate seeking an embedded software position, or a full-time embedded systems engineer seeking to switch your position or the company. In all cases, you'll need to go through one or more technical interviews to get where you want.Please, be advised that the interview questions depend heavily on the specialty of the company you're going after (automotive, security, robotics, etc), the position you're applying for (internship, testing, embedded software, tooling, etc), and also your CV. Yea, your CV somehow dictates some of the interview questions. For me personally, I was asked in (USB, CapTouch, etc) just because it's all written in my CV,

however, these topics are not mandatory at all for most embedded systems positions. So, only write in your CV what you can actually demonstrate and be ready to get questions about it.Regardless of the position you're seeking, the company specialty, or your previous experience, there are many general questions that you'll most probably get. And this is what I'm going to mainly focus on this page. To help you quickly revise and get ready for your next embedded systems interview.We can categorize these questions as follows.Embedded Systems Interview QuestionsEmbedded C: some questions about C programming, structs, typedef, pointers, the C build process, multi-file projects, memory sections, bootloader vs startup code, arrays, strings manipulations, and things like that.Computer Architecture: some questions about memory types, buses, 8-bit and 32-bit microcontrollers, Harvard vs von Neuman, ARM, instruction sets, Endianness, and other questions in this area.Microcontroller Peripherals: some questions about ADC, Timers, Interrupts, PWM, WDT, Com Protocols like UART, SPI, I2C, and others.Data Structures & Algorithms: some questions about basic data structures like the stack, queue, linked list, and implementation in C programming language. As well as some algorithms questions for sorting, searching, and things like that.RTOS & OS Concepts: some questions about real-time operating systems, FreeRTOS-based implementations, Mutex, Semaphores, DeadLock, Priority inversion and inheritance, and other OS concepts.Problem Solving & IQ: some questions for IQ assessment and to check how do you think while solving a problem and how well you can communicate this with the interviewer. You don't have to actually solve all the problems as long as you can show logical thinking and communicate this in a good manner.Automotive: (optional in most cases) questions about CAN, LIN, Ethernet, AUTOSAR software architecture, etc.Embedded Software Testing: (optional in most cases) questions about Static testing, Dynamic testing, V-Model, etc.Embedded Hardware: some questions about general embedded hardware knowledge. Things like debuggers, JTAG, using DSO, logic analyzers, emulators, and troubleshooting.Note: The questions on this page are answered in 3 different ways. Short answers are collapsable by clicking the question text you can view/hide the answer. Long answers are found on dedicated pages that redirect you back to here, after reading the answer with a couple of links there. And previously answered questions will be linked-to in such a way so that it opens up in a new tab without redirection. I hope it's a convenient way of handling this. Note: You should expect more or less questions on certain topics depending on the 2 factors I've stated earlier. The company's specialty and the position you're applying for. You may not be asked about automotive protocols at all or barely get a question related to software testing unless you're seeking certain positions. The same goes for RTOS and data structures and so.Embedded C Interview QuestionsWhat's The Startup Code?The Startup Code in embedded systems is used to set up data memory sections such as global data variables. It also zero initializes part of the data memory for variables that are uninitialized at load time. The startup code copies 5 from flash to variable y in the RAM as it's an initialized global. And it also initialized the x variable with 0 value.For applications that use dynamic memory allocation C functions like malloc, the C startup code also needs to initialize the data variables controlling the heap memory.After this initialization, the C startup code branches to the beginning of the main program, and your application starts running. The C startup code is inserted by the compiler at the linker stage automatically and is toolchain specific.What's The Bootloader?A Bootloader in embedded systems is a standalone application that gets developed, if needed, aside from the target application itself to provide the following functionalities. The bootloader is the first software that runs after board power up and it does hardware check, and initializations for the processor, peripherals, and the OS if used. Then it starts to run your main application.The bootloader application can also be used to upgrade the firmware on the target microcontroller via serial communication ports like UART, SPI, I2C, USB. So, it does control the hardware peripherals to do this task. Then, it does release everything and initializes the peripherals to be used by your main application after the bootloader has done its work.Local, Static, Global variables in CGlobal Vs Static Global In CA Global variable in C has a global scope across your whole project files. Not only the .c file in which this variable is declared. For any source code .c file to be able to use that variable it'll only need to extern it and it has access to it.However, on the other hand, if a global variable is declared to be static, it'll be only global across the source code .c file it's been declared into. No other files in the project can extern and access this variable.What's a CallBack Function? Usage Examples?A Callback Function is a reference to executable code that is passed as an argument to other code that allows a lower-level software layer to call a function defined in a higher-level layer. A callback allows a driver or library developer to specify a behavior at a lower layer but leave the implementation definition to the application layer. Which makes the software easily reusable and more portable.For example, if you're developing a driver code for a Timer module and it has a function to do whenever an overflow interrupt is fired. The functionality doesn't need to be fixed but it has to be defined by the application higher-level. So, you can use a call back function that points to nothing. And in the application code, the user will write the interrupt handler function's code and set the timer call back to point to this handler routine. So, whenever an interrupt is fired, the ISR will call the call back function which points to the user handler code and it'll get executed.And yes, this means that you as a driver developer will have to provide the call back registering "setting" function that the user can use to set the call back function and make it point to the implementation in the higher-level software layer.What's Void Pointer in C?A Void Pointer is a pointer that has no associated data type with it. A void pointer can hold the address of any type and can be type-casted to any type. This means you can use it to point to any variable type you want as shown in the code snippet below.void* ptr = &x; // void pointer holds address of int 'x' variableptr = &y; // void pointer holds address of char 'y' variableDynamic memory allocation c functions like malloc and calloc return (void *) type and this allows these functions to be used to allocate memory of any data type, as you can type-cast it to fit your variable data type.Void pointers in C are used to implement generic functions. And note that void pointers cannot be dereferenced. And C standards don't allow pointer arithmetic with void pointers.Null, Void, Dangling, Wild Pointers in CState All Memory Sections & Their UsageRAM Memory Sections: [.stack – .heap – .data – .bss].stack: This memory section is used for local variables.heap: For dynamically allocated variables (not used in most embedded applications).data: For global and static initialized variables.bss: For global and static uninitialized variablesFlash Memory (ROM) Sections: [.sdata – .rodata – .text – .cstartup].sdata: For initialized globals, the startup code copies it to .data section at system start.rodata: For const variables (read-only).text: The code of your application.cstartup: The startup code that initialized the system and branches to the main applicationWhat's A Segmentation Fault?A Segmentation Fault occurs when the system tries to access a memory location that is not allowed to be accessed or attempts to access a memory location in a way that is not allowed (i.e. attempting to write to a read-only location).At the hardware level, the fault is initially raised by the memory management unit (MMU) on illegal access (if the referenced memory exists), as part of its memory protection feature.At the operating system level, this fault is caught and a signal is passed on to the offending process, activating the process's handler for that signal. Different operating systems have different signal names to indicate that a segmentation fault has occurred.Startup Code Vs BootLoaderExplain The Usage of extern Keyword in Embedded CExtern is a keyword in the C programming language that is being used to declare global variables and functions. The typical use case for the extern keyword is to get a global variable from another .c source file. We always use extern in header files only to bring in global variables and functions from other .c files.All variables and functions in header files should be explicitly extern. Otherwise, you should make your global variables static to restrict access to those global variables and/or functions to everything within the same .c file but nothing outside. And if you want any function and/or global variable to be seen and accessed in other .c files outside their .c file, then you shouldn't make them static and extern them in the header .h file.What's the use of the "Static" keyword in embedded C?What's the use of the "Volatile" keyword in embedded C?When to use "volatile" variables in code?What's the difference between array and pointer in C?Constant pointer Vs pointer to a constantWhat are Macros in C? Pros and Cons?What does it mean to have a static function?What's an inline function in C?What's A Reentrant Function in C?A Function or Routine is described as reentrant if it can be safely called again before its previous invocation has been completed (i.e it can be safely executed concurrently). This type of function is used in different cases like recursion, hardware interrupt handling (ISR). To be reentrant, a computer program or function:Must hold no static (or global) non-constant data.Must not return the address to static (or global) non-constant data.Must work only on the data provided to it by the caller.Must not rely on locks to singleton resources.Must not modify its own code (unless executing in its own unique thread storage)Must not call non-reentrant computer programs or routines.What's memory padding and alignment?What's the C Preprocessor's Function?Functions of #define, #if, #pragmaHow to prevent multiple inclusion of header files? Why?Explain The Embedded C Build Process (Detailed Steps)What do you know about MISRA C rules?Do you think we can write recursive code in embedded C? why yes or no?Static vs Dynamic memory allocation in CWhat's Memory Fragmentation?Memory Fragmentation is an issue that arises when using dynamic memory allocation. If you keep allocating and releasing memory spaces over the time, you'll end up having non-contiguous memory blocks that are actually free and your in-use variables are scattered everywhere in the RAM.This is called memory fragmentation, and can potentially lead to dynamic memory allocation failure. If you'd like to allocate an array of 100 integers and there is no contiguous block of memory with that space. This causes a runtime error and it's one of the strongest reasons why we don't use dynamic memory allocation in embedded systems firmware in the first place.What's dynamic memory allocation? And why it's avoided in embedded C?Can you include a .c File into another .c file?The short answer is yes, however you shouldn't. It's a very bad practice to do this, you should only include header files that have declarations and functions prototypes, not their implementation.So, it depends on the contents of the .c file you're willing to include as #include is a preprocessor directive that works as a text "code" replacement tool. Which copies the code inside a .c file into your target .c file. Now, it's up to the compiler to look at the code and see if there is a C rule violation or not.Another thing to note is an important issue that arises from this bad practice is that you perform the inclusion process with preprocessor text replacement in mind. However, there is another important stage in the compilation process which is the linker stage which will link the compiled .c object file .o with the file in which you included it maybe main.c which is main.o at linker stage.Now, the linker will find duplicate implementations for certain functions if it's the case and this will fire linker errors and stop the build process. Therefore, it's another issue to care for and you'd better avoid this practice altogether in the first place.How to set, clear, and toggle a single bit of a register in C?Computer Architecture QuestionsDiscuss RISC vs CISC processorsWhat's pipelining? How does it affect CPU performance?What are the differences between Harvard & Von Neumann Architectures?What are the differences between a Microcontroller & a Microprocessor?What are the differences between RAM & ROM?What are the differences between FLASH & EEPROM?What is Endianness? Little and BigWhat's Bit Banging?What's Bit Banding?Differences between 8-Bit and 32-Bit architectures?What's DMA? Give example applications for DMAMicrocontroller Peripherals Questions Timers Using a timer, measure the execution time of a C functionUsing a timer, generate software PWM signalUsing a timer, measure a digital pulse widthUsing a timer, measure a digital signal's frequencyWrite the equations and select the parameters to generate a 1ms timer interruptWhat's the watchdog timer (WDT)? When and Why it's used?When should you kick the WDT? Interrupts What's an ISR routine?Can we send and return data from ISR?Can we call other functions inside ISR?Explain interrupt execution sequenceState the types of interruptsVectored VS Non-Vectored interruptsExplain Interrupts late arrival – ARMExplain Interrupts tail-chaining – ARMWhat's NVIC in ARM Cortex?Explain interrupts nestingWhat's interrupt latency? What causes it?How can you measure interrupt latency?How can we reduce the interrupt latency?What's the interrupt vector table? Where it's stored? Can you change its location? ICU & PWM What's PWM? How does it work? And What are the parameters of a PWM Signal?How to generate multiple software PWM signals with 1 timer?How to measure pulse width with ICU (input capture unit)?How to measure the frequency of a signal with ICU?How to measure the duty cycle of a signal with ICU?What's PWM resolution? How to control it?How to implement a software ICU?Using PWM, create a DAC systemGenerate 2 software PWM signals with controllable duty cycle and phase difference ADC & DAC How does ADC work? What are the ADC types?Explain ADC sampling time and acquisition timeHow to pick the proper sampling rate for ADC?What's the effect of analog input impedance on the ADC's reading?What's ADC channels cross-coupling? How to eliminate it?Write code to read temperature with ADCUsing 8-Bit DAC, write code to generate a sawtooth waveform @ 50HzUsing 8-Bit DAC, write code to generate a triangular waveform @ 50HzUsing 8-Bit DAC, write code to generate a sine waveform @ 50Hz Serial Communication Compare between SPI and I2CWhich offers the highest data rate among SPI, I2C, UART protocols?Which port to use for onboard high-speed communication? Why?At the same baud rate which would send more data SPI or UART? Why?1. UARTHow does UART communication work?What's the UART frame structure?What's Baud Rate and Bit Rate?Differences between UART and USARTWhat's parity bit in UART?is UART synchronous or asynchronous protocol?is UART Full-duplex or Half-duplex communication?What's the pin sampling rate for the UART receiver?Can UART be a multi-slave protocol?Give some example applications in which UART is a good choice as a communication portWrite code to send 1000 bytes of data with UART2. SPIHow does SPI communication work?Define the SPI clock phase and polarityWhat are SPI mode numbers 0,1, 2, 3How to daisy-chain some SPI devices?How many slaves can SPI master communicate to?Can an SPI slave initiate a communication?Is SPI a synchronous or asynchronous protocol?Give some example applications in which SPI is a good choice as a communication portWhat do you think sets the upper limit for SPI speed?3. I2CHow does I2C communication work?What are the basic elements of I2C transactions?How many slaves can be addressed on the I2C bus?Is I2C a synchronous or asynchronous protocol?What are start and stop conditions?What is arbitration in I2C? Why it's useful?Give some example applications in which I2C is a good choice as a communication portWhat would happen if two masters wanted to send data to the same slave at the exact same moment?Data Structures & Algorithms QuestionsImplement stack in cImplement a linked list in cImplement a double linked list in cImplement a queue in cSearch for a number in an unsorted arraySearch for a number in a sorted arrayReverse an arrayRTOS & OS QuestionsWhat's a kernel?What's a task and a process?What's The CPU Load?What's the scheduler? List down some scheduling algorithmsWhat's a deadlock and when does it occur?Differences between Mutex and SemaphoresWhat's the priority ceiling?What's the priority inversion?What's the priority inheritance?What does RTOS mean? Give some examplesWrite a FreeRTOS task implementation in cPreemptive vs Non-Preemptive kernelsAutomotive Embedded Interview QuestionsWhat's the speed range for the CAN bus?What's the speed range for LIN?Applications For CAN and LINDifferences between CAN and LINWhy CAN bus has 2 resistors at both ends?What happens when 2 CAN nodes transmit data with the same id at the same time?What do you know about AUTOSAR?What's SWC?What's the RTE? It's functions?What do you know about iso26262 and functional safety?Embedded Hardware QuestionsWhat's JTAG?Simulator VS EmulatorLocal, Static, Global variables in CProblem Solving QuestionsWrite C code to count the 1's in an integer numberWrite C code to count the 0's in an integer numberWrite C code to check for prime numberWrite C code to check if an int is a palindrome number or not (like 1243421)Write C code to draw X pattern with stars '*'Write C code to draw Z pattern with stars '*'Write C code to draw triangle pattern with stars '*'Write C code to draw char A pattern with stars '*'Write C code to draw a diamond pattern with stars '*'Write C code to reverse the digits of an integer numberWrite C code to find the greatest common divisor (GCD) of 3 numbersWrite C code to check for the endianness of the systemWrite C code to send 4-Byte integers from a big-endian machine to little-endian machine via UARTThe embedded systems interview questions mentioned on this page are general questions and most commonly seen in embedded systems interviews. I'll do my best to keep this page updated with more questions and to answer each of them on a separate page and update its link here, so you have a single page resource to do a quick revision before your next interview. I wish you all the best of luck.If you've got any further questions, please let me know. And if you'd like to give me any feedback, don't hesitate to drop me a comment down below. In case you'd like to support my work, you can see this page or visit my store.If you find this a helpful resource, then SHARE it with your network on social accounts! Let others know about it.Regards,Khaled Magdy

We can categorize these questions as follows. Embedded Systems Interview Questions. Embedded C: some questions about C programming, structs, typedef, pointers, the C build process, multi-file projects, memory sections, bootloader vs startup code, arrays, strings manipulations, and things like that.; Computer Architecture: some questions about memory … Signals & Systems Questions and Answers – Exponential Fourier Series and Fourier Transforms ; Signals & Systems Questions and Answers – Common Fourier Transforms ; Signals & Systems Questions and Answers – Properties of Fourier Transforms ; Signals & Systems Questions and Answers – Common Laplace Transforms – 1 7. Embedded System Multiple Choice Question on Real Time Operating systems. The section contains Embedded System multiple choice questions and answers on operating system, multitasking and commercial operating system, tasks, resource protection, linux, task swapping and characteristics of windows nt. 7/5/2019 · The widespread community in C provides vast support for Embedded Systems Programming. 2. C++. C++ is Boney Kapoor in Embedded Systems Programming. Like Boney Kapoor is not as famous as Anil Kapoor, C++ is less popular than C in regards to embedded systems but the addition of object-oriented programming makes it a great choice for veteran ... About C Programming Aptitude Questions and Answers. As we know C programming language is the first language to learn and it is also important, popular computer programming language, it is a middle level language useful for both low level (hardware level) and high level (user applications) programming. Questions and Answers; Effective Resume Writing; HR Interview Questions; Computer Glossary; Who is Who; Embedded Systems - Interrupts. Advertisements. Previous Page. Next Page . An interrupt is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention. Whenever an interrupt occurs, the controller completes the … 6/6/2022 · Top Embedded Systems Interview Questions and Answers; Example of Embedded Systems Laser Printer. Laser Printers are one of the examples of Embedded Systems which uses embedded systems to manage various aspect of the printing. Apart from performing the main task of printing, it has to take user inputs, manage communication with the computer system, … 9/7/2022 · 1) What is Android? It is an open-sourced operating system that is used primarily on mobile devices, such as cell phones and tablets. It is a Linux kernel-based system that's been equipped with rich components that allows developers to create and run apps that can perform both basic and advanced functions. 31/7/2019 · C# is being utilized as a part of numerous businesses. To have a great development in C# work, our page furnishes you with nitty-gritty data as C# prospective employee meeting questions and answers. C# Interview Questions and answers are prepared by 10+ years experienced industry experts. C# Interview Questions and answers are very useful to ... Questions and Answers; Effective Resume Writing; HR Interview Questions; Computer Glossary; Who is Who; Embedded Systems - Assembly Language. Advertisements. Previous Page. Next Page . Assembly languages were developed to provide mnemonics or symbols for the machine level code instructions. Assembly language programs consist of mnemonics, thus they should … In this post, we have put together a list of frequently asked Embedded C Interview Questions and Answers for beginner, intermediate and experienced candidates. These questions are categorized for quick browsing before the interview or to act as a helping guide on different topics in Embedded C for interviewers. 26/7/2022 · Following is a curated list of SQL questions for interview with answers, which are likely to be asked during the SQL interview. Candidates are likely to be asked SQL basic interview questions to advance level SQL interview questions for 3 years experience professionals, depending on their experience and various other factors. The below list ... Embedded Systems Interview Questions (43) Floppy Disk Interview Questions (12) Hard Disk Interview Questions (12) Hardware and Software Design Interview Questions (35) Hardware Design Interview Questions (32) Intel Interview Questions (24) Microprocessor Interview Questions (101) Motherboard Interview Questions (11) Random Access Memory Interview … Embedded Software Engineer Interview Questions. Embedded Software Engineers design, develop and install software solutions to meet company needs. They are responsible for building high-quality, fully functional embedded software systems, aligned with … FaaDoOEngineers.com Terms & Conditions. Registration to this forum is free! We do insist that you abide by the rules and policies detailed below. Keeping this in mind we have designed the most common Windows Server Interview Questions and answers to help you get success in your interview. Below is the top Windows Server Interview Questions that are asked frequently in an interview. These top interview questions are divided into two parts are as follows: Start Your Free Software Development Course. Web development, … Professional Training Institute is one of the ISO 9001:2015 certified embedded system training institute.We are the best Embedded Systems training in Bangalore. We provide 100% job placement assistance to all our student. Our top most priority is to put more and more focus on the practical aspects of the embedded systems training. We believe if the student is good with … Embedded Systems - Processors. Processor is the heart of an embedded system. It is the basic unit that takes inputs and produces an output after processing the data. For an embedded system designer, it is necessary to have the knowledge of both microprocessors and microcontrollers. Processors in a System. A processor has two essential units − The C standard doesn't care about embedded, but vendors of embedded systems usually provide standalone implementations with whatever amount of libraries they're willing to provide. C is a widely-used general-purpose high-level programming language mainly intended for … Systems engineers are also involved in creating recovery plans and redundant systems designed to survive different scenarios like natural disasters and power failures. If you're looking for a systems engineer to work on disaster planning, it's a good idea to talk about past experience on similar projects. Disaster recovery plans need to take into account physical and electronic … Embedded system security is an approach strategically to protect the software that is running on top of the embedded systems from any severe threats. Programmable hardware with the integration of the operating system and software combines to form an embedded system. They are built to work as a committed function or a group of functions. They are generally launched … 9/7/2022 · In this list of Networking interview questions, we have covered all commonly asked basic and advanced interview questions on networking with detailed answers to help you clear the job interview. The below list covers 130+ important interview questions for Networking for freshers candidates as well as Networking interview questions for ... Embedded system main helps identify a possible location in Industrial varieties machines, automobiles, medical equipment, cameras, airplanes, toys, house equipment, etc. Now, if you are looking for a job related to Embedded Systems, you need to prepare for the 2021 Embedded System Interview Questions. Every interview is indeed different as per … 6/6/2022 · 13) Explain what is the need for an infinite loop in embedded systems? Embedded systems require infinite loops for repeatedly processing or monitoring the state of the program. For instance, the case of a program state continuously being verified for any exceptional errors that might just happen during run-time such as memory outage or divide ... Here are some frequently asked Embedded Interview Questions : Q1. What is an embedded system? Embedded systems means… Embedded systems can be termed as a combination of hardware and software. Have its own CPU which contains memory, timers, peripherals, bus, reset, on-chip oscillation. Dedicated to a specific task. Q2. Why embedded systems? This set of Operating System Multiple Choice Questions & Answers (MCQs) focuses on "CPU Scheduling". 1. Which module gives control of the CPU to the process selected by the short-term scheduler? a) dispatcher b) interrupt c) scheduler d) none of the mentioned View Answer. Answer: a Explanation: None. advertisement. 2. The processes that are residing in main memory and … 2/8/2021 · This article lists 100 Embedded Systems MCQs for engineering students.All the Embedded Systems Questions & Answers given below includes solution and link wherever possible to the relevant topic.. An Embedded System (ES) is a platform in which multiple hardware and software type components are integrated together on IC technology for the … Not all operating systems and browsers have the same fonts installed. Web safe fonts are fonts that are commonly pre-installed on many computer systems, such as Arial and Times New Roman. In case the browser or operating system doesn't recognize the first font you set (e.g. Ubuntu), you should choose a web safe fallback font to display (e.g ... Linux (/ ˈ l iː n ʊ k s / LEE-nuuks or / ˈ l ɪ n ʊ k s / LIN-uuks) is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packaged in a Linux distribution.. Distributions include the Linux kernel and supporting system software and libraries, many of … 23/7/2022 · 1) Explain what is Scala? Scala is an object functional programming and scripting language for general software applications designed to express solutions in a concise manner. 2) What is a 'Scala set'