1. Sistema Embebido

1. Sistema Embebido es un Sistema electrónico de cómputo, compuesto por:
   a) Sólo hardware.
   b) Hardware u software.
   c) Sólo software.
   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

Determinista: que responde siempre de la misma manera a los mismos estímulos

2. Durante el Diseño/Desarrollo/Depuración de software, nos preocupa:
   a) Sólo el determinismo del SE.
   b) Sólo el tiempo de respuesta del SE.
   c) Tanto el determinismo como el tiempo de respuesta del SE.
   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

3. Para el Diseño/Desarrollo/Depuración de software de SE, la aplicación (escrutar, procesar, actuar):
   a) Los módulos se comunican/sincronizan mediante: Flags o Variables o Colas.
   b) Garantizar comportamiento comunitario (que ningún módulo se apropie de la CPU).
   c) El código bloqueante es inaceptable.
   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

4. Para el Diseño/Desarrollo/Depuración de software de SE, en la toma de decisiones influyen:
   a) Sólo el tipo y campo de aplicación, requerimiento del cliente y del medio (regulaciones).
   b) Sólo la idiosincrasia, conocimiento, experiencia y habilidad del profesional.
   c) Sólo el contexto: la provisión de herramientas/insumos/manufactura/...
   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

5. Los Sistemas Duros (Hard Real Time), tienen:
   a) Restricciones de tiempo rigurosos (escrutar, procesar y actuar en un plazo perentorio).
   b) Restricciones de tiempo menos rigurosas.
   c) Restricciones de tiempo nada rigurosas.
   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

6. El diseño de SE se orienta a:
   a) Aumentar: tamaño/consumo/costo.
   b) Reducir: eficiencia y confiabilidad/re-usabilidad.
   c) Asegurar: determinismo y tiempo de respuesta/seguridad.
   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

¿¿Me estas cargando?

7. Durante el Diseño/Desarrollo/Depuración del circuito eléctrico del SE, se recomienda dibujar:
   a) Un diagrama de bloques de las funciones principales y un diagrama de cada función principal.
   b) Un único diagrama conteniendo todas las funciones principales (con diagrama de bloques).

c) Un único diagrama conteniendo todas las funciones principales (sin diagrama de bloques).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

8. Las funciones de la Unidad Central de Procesamiento (CPU) son:

   a) Aceptar datos, proporciona espacio de memoria temporal de almacenamiento (guarda).

   b) Realizar operaciones aritmético/lógicas de datos.

   c) Convertir datos en información.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

9. Los componentes básicos de una computadora son:

   a) Unidad de entrada.

   b) Unidad central de procesamiento (CPU).

   c) Unidad de salida.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

10. Un lenguaje de programación es un lenguaje formal, que:

    a) Especifica un conjunto de instrucciones para que una computadora realice tareas específicas.

    b) Se usa para escribir programas y aplicaciones de software y para controlar sistemas informáticos.

    c) Cuenta con su propia sintaxis, estructura y conjunto de comandos.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

11. Respecto a la jerarquía de los Lenguajes de Programación, podemos decir que:

    a) El lenguaje ensamblador es independiente de la arquitectura de la computadora.

    b) Entre el lenguaje de alto nivel y el de máquina, existen lenguajes ensambladores.

    c) No se requiere de un programa de aplicación para convertir código ensamblador en ejecutable.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

12. En la programación Bare Metal:

    a) Es un bucle/lazo sin fin, que ejecura tareas hasta completar (RTC) su ejecución.

    b) Esta ejecución secuencial no se desvía cuando ocurre un evento de interrupción.

    c) Este enfoque se conoce como Super-Loop (polling and without Interrupts.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

13. El código fuente debe pasar por uno/varios pasos antes de convertirse en ejecutable, a saber:

    a) Preprocessor.

    b) Preprocessor, Compiler/Assembler.

    c) Preprocessor, Compiler/Assembler, Linker.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

14. Unified Modeling Language (UML) es un lenguaje de modelado de propósito general:

    a) Su objetivo es definir una forma estándar de visualizar cómo se ha diseñado un sistema.

b) Es un lenguaje de programación.

c) State Machine Diagrams, no sirven para representar el estado del sistema en instantes de tiempo.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

15. Las características de la programación modular son:

a) Código no estructurado.

b) Subdividir un programa en subprogramas separados (módulo reutilizable).

c) Dificultad de uso (y reuso), mantenimiento de leer/entender/usar/desarrollar/depurar/mantener.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

16. Para el Diseño/Desarrollo/Depuración de hardware de SE, se recurre a herramientas:

a) Con prestaciones de control y análisis, en evolución permanente.

b) Para sistematizar o mejorar la velocidad/precisión de procesos/subprocesos requeridos.

c) Herramientas informáticas: CAE/CAD/CAM.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

17. Para el Diseño/Desarrollo/Depuración de Hardware de SE, se recurre a herramientas:

a) Codificación en C aplicando el estándar de codificación.

b) Codificación: Estructurada, Modular/Bare Metal (sin RTOS)/Event-Triggered Systems.

c) Escrutar, Procesar y Actuar/Super-Loop (polling and interrupts), con modelado de tareas.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

18. Como solución a SE orientados a control, tenemos:

a) Startup (Inicio) - Inicializaciones básicas del MCU.

b) Main (Programa Principal) - Iteración perpetua de Tareas (algoritmos).

c) Drivers (Manejadores) de Entrada/Salida - Interacción con el exterior (eventos/sincronismos).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

19. Para el Diseño/Desarrollo/Depuración de hardware de SE, recurrimos a:

a) Herramientas informáticas CAE: captura esquemática, simulación, depuración, validación y ...

b) Módulos electrónicos: Placas de desarrollo: STM32-Bits ARM Cortex MCUs, NÚCLEO board.

c) Dip-switches, buttons, keyboards, leds, LCD Displays, etc. e instrumentos de laboratorio.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

20. Para el Diseño/Desarrollo/Depuración de software de SE, recurriremos a:

a) Integrated Development Environment (IDE for STM32): STM32CubeIDE.

b) Version Control System (VCS): Git Bash/GUI & Repository: GitHub.

c) Software Modeling Tool: Itemis CREATE.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

21. ¿Qué es un IDE?

    a) Es un sistema de gestión de código fuente y control de versiones distribuido.

    b) Es un servicio de alojamiento (hosting) para repositorios.

    c) Es un sistema de desarrollo, que usaremos para desarrollar/depurar software.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

22. ¿Qué es un Itemis CREATE?

    a) Es un sistema de gestión de código fuente y control de versiones distribuido.

    b) Es un servicio de alojamiento (hosting) para repositorios.

    c) Es un sistema de desarrollo, que usaremos para desarrollar/depurar Diagramas de Estado.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

23. ¿Qué es GitHub?:

    a) Es un sistema de gestión de código fuente y control de versiones distribuido.

    b) Es un servicio de alojamiento (hosting) para repositorios.

    c) Es un sistema de desarrollo, que usaremos para desarrollar/depurar Diagramas de Estado.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

24. Como solución a SE orientados a control, tenemos:

    a) Disparador (trigger) es un suceso (evento/sincronismo) que inicia acción del Sistema de Control.

    b) La acción puede ser la ejecución de una tarea encargada de leer/calcular/enviar mensaje.

    c) Un evento significativo puede resultar de una condición de programa/encuesta/interrupción".

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

25. Como solución a SE orientados a control, tenemos:

    a) Sólo Sistemas Disparados/Activados por Tiempo (Time-Triggered Systems).

    b) Sistemas Disparados/Activados por Tiempo o por Evento (Time-and Event-Triggered Systems).

    c) Sólo Sistemas Disparadores/Activados por Evento (Event-Trigger Systems).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

2. Microcontroladores

26. En una arquitectura de computadora, el bus de control sirve para:

    a) Derivar de la unidad solicitante (procesador), especificar a qué unidad/elemento acceder.

    b) Transferir información entre la unidad solicitante y otra unidad.

    c) Especificar qué hacer: dirección de las transferencias (lectura o escritura) y cuándo hacerlo.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

27. En una arquitectura de computadora, la conexión interna se realiza a través de bus/buses

    a) Sólo el bus de direcciones.

    b) Sólo el bus de direcciones y bus de datos.

    c) El bus de direcciones, el bus de datos y el bus de control.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

28. En una arquitectura de computadora, el bus de datos sirve para:

    a) Derivar de la unidad solicitante (procesador), especificar a qué unidad/elemento acceder.

    b) Transferir información entre la unidad solicitante y otra unidad.

    c) Especificar qué hacer: dirección de las transferencias (lectura o escritura) y cuándo hacerlo.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

29. ARM's Cortex-M que constituye la base de muchos microcontroladores del mercado actual, es:

    a) Un microcontrolador.

    b) Un microprocesador.

    c) Un microcomputador.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

30. Las fabricantes de silicio que usan micro...dores ARM, pueden aprovechar la disponibilidad de:

    a) Amplia gama de recursos compartidos.

    b) Herramientas de desarrollo, lenguajes de programación y sus compiladores.

    c) Sistemas operativos, Depuradores, etc.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

31. ARM ha desarrollado sus micro...dores como:

    a) Componentes del tipo COTS (Commercial Off-The-Shelf).

    b) Componentes de tipo ASIC (Application-Specific Integrated Circuit).

    c) Arquitectura IP (Intellectual Property).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

32. Los usuarios de aplicaciones quieren un producto terminado (microcontrolador):

    a) Potente en términos de lo que quieren hacer y disponible en el mercado (circuito integrado).

    b) Pueden elegir entre miles de dispositivos de cientos de fabricantes.

    c) Trabajan al nivel de la placa de circuito impreso (PCB).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

33. Cuando un ingeniero tiene que diseñar una nueva solución SoC, primero genera un prototipo:

    a) Que permite la verificación de la cantidad de memoria necesaria para el proyecto.

    b) Que no permite la verificación del tiempo de respuesta del sistema.

Hoy murió el castellano

Con el mayor de los respetos, pero este tipo es un pelotudo

Son la misma pregunta

c) El producto final dependerá de un SDK, pero un SDK ayudará en su diseño.
d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

34. Cuando un ingeniero tiene que diseñar una nueva solución SoC, primero genera un prototipo:

a) Que no permite la verificación de la cantidad de memoria necesaria para el proyecto.
b) Que permite la verificación del tiempo de respuesta del sistema.
c) El producto final dependerá de un SDK, pero un SDK ayudará en su diseño.
d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

35. Cuando un ingeniero tiene que diseñar una nueva solución SoC, primero genera un prototipo:

a) Que no permite la verificación de la cantidad de memoria necesaria para el proyecto.
b) Que no permite la verificación del tiempo de respuesta del sistema.
c) El producto final no dependerá de un SDK, pero un SDK ayudará en su diseño.
d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

36. Para guardar y ejecutar las operaciones, se utilizan registros como fuentes y destinos de datos:

a) En los procesadores modernos pueden haber 16, 32 o incluso más.
b) Para escribir o leer registros, el acceso se realiza a través de uno o más buses de datos internos.
c) Un registro particular como registro de flags o registro de estado.
d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

para mí es esta

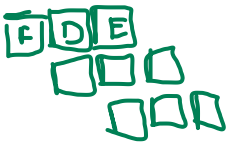
37. Para guardar y ejecutar las operaciones, se utilizan registros como fuentes y destinos de datos:

a) En los procesadores modernos pueden haber uno solo.
b) Para escribir o leer registros, el acceso se realiza a través de uno o más buses de datos internos.
c) Un registro cualquier como registro de flags o registro de estado.
d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

¿? No está en español esto

38. Un mejor enfoque de la unidad de control, es hacer lo que se llama pipelining, por cada clock:

a) Se carga una instrucción, se decodifica y se ejecuta, instrucción por instrucción.
b) Se carga una instrucción, se decodifica y ejecuta la anterior.
c) Se carga una instrucción, se decodifica la anterior y se ejecuta al anterior a la anterior.
d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).



39. El pipelining, pero trae problemas al ejecutar una instrucción de bifurcación (Branch o Jump):

¡!! ¿y el VERBO? ¿YO TAMBIÉN VOY A TERMINAR ASÍ?

a) La dirección de la siguiente instrucción se conocerá al completar la ejecución de la anterior.
b) No es necesario retrasar la siguiente recuperación ni interrumpir el pipeline.
c) Es positivo porque un programa tiene muchas más bifurcaciones que instrucciones.
d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

40. Para transferir instrucciones/datos dentro y fuera del procesador, se requiere de dos más buses:

a) Von Newman, instrucciones y datos comparten el mismo par de bus de direcciones y datos.

b) Harvard, instrucciones y datos comparten el mismo par de bus de direcciones y datos.

c) Tanto el determinismo como el tiempo de respuesta del SE.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

41. En un procesador ARM Cortex M, la palabra de datos es de:

a) 8 bits.

b) 8 bits o 16 bits.

c) 16 bits o 32 bits.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

42. Cuando el procesador ARM Cortex M comienza a funcionar (al encenderse o después de reiniciarse):

a) Necesita ejecutar código para iniciar el sistema, denominado boot (strap) code.

b) El vector de Reset apuntará a la dirección del programa de encendido o reinicio.

c) La tabla de vectores de explicación comienza en las direcciones bajas de memoria no volátil.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

43. Un área específica de memoria se utiliza para:

a) Un heap (heap), en la que el procesador puede hacer PUSH/POP (almacenar/recuperar datos).

b) Una pila (stack), en la que el procesador puede hacer PUSH/POP (almacenar/recuperar datos).

c) Un pool (pool), en la que el procesador puede hacer PUSH/POP (almacenar/recuperar datos).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

44. Los tipos de memorias disponibles según su capacidad de retener o no datos son:

A C B

A ∈ B

a) ROM/PROM/EPROM/OTP EPROM volátiles.

b) EEPROM/FLASH/FRAM/MRAM volátiles.

c) SRAM/SSRAM volátiles.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

45. Cortex-M0 y -M0+ ofrecen:

a) Un rendimiento mayor.

b) El consumo de energía más bajo de la familia Cortex-M.

c) Un conjunto de instrucciones grande.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

46. En un procesador ARM Cortex M, la capacidad de direccionamiento es de:

a) 2^{16} posiciones de memoria.

b) 2^{24} posiciones de memoria.

c) 2^{32} posiciones de memoria.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

47. Los diagramas de estado de Harel, introducen un enfoque:

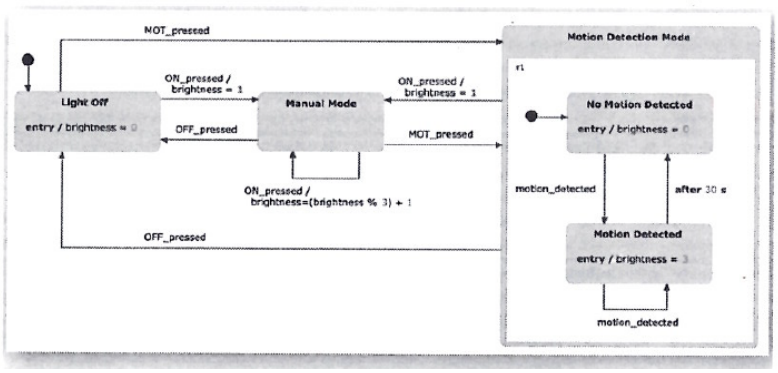
    a) Modular, jerárquico y bien estructurado.

    b) Sin estados compuestos, ni ortogonalidad (paralelismo).

    c) Sin comunicación (sin sincronización).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

48. Para el diagrama de estado de Harel de la figura, al recibir la siguiente secuencia de entradas: Indicar por qué estados evolucionará el diagrama luego de ser reiniciada.



→ ESTABA EN LAS DIAPOS

a. Reset => LIGHT OFF
b. **ON_pressed** => MANUAL MODE
c. **ON_pressed** => MANUAL MODE
d. **ON_pressed** => MANUAL MODE
e. **MOT_pressed** => NO MOTION DETECTED
f. **motion_detected** => MOTION DETECTED
g. **ON_pressed** => MANUAL MODE
h. **OF_pressed** => LIGHT OFF
i. fin

49. Del diagrama anterior, indicar por qué valores evolucionará la variable brightness (estado a estado) luego de ser reiniciada:

ACÁ iría 2 ... OJO: El %3 es la operación mod 3 (o resto de la división)

a. Reset => 0
b. **ON_pressed** => 1
c. **ON_pressed** => (1%3)+1 = 1,3
d. **ON_pressed** => (1,3%3)+1 = 1,7
e. **MOT_pressed** => 0
f. **motion_detected** => 3
g. **ON_pressed** => 1
h. **OF_pressed** => 0
i. fin

acá iría 3

50. La conexión de interfaces pasivas a pines del microcontrolador, se validan mediante cálculo aplicando:

    a) Ley de Coulomb y Faraday-Lenz.

    b) Ley de Ampere y Faraday-Lenz.

    c) Ley de Ohm y Leyes de Kirchoff.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

51. Conceptos básicos de los puertos de salida digital del microcontrolador:

    a) Los pines de un puerto de salida pueden tener uno de dos niveles lógicos de salida 0 o 1.

    b) La tensión máxima de salida 1 depende de la tecnología del microcontrolador (puede ser 3.3V o 5V).

    c) La tensión mínima de salida 0 depende de la tecnología del microcontrolador (puede ser 0V).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

52. Conceptos básicos de los puertos de entrada/salida digital del microcontrolador:

    a) Se pueden usar para interactuar con el mundo exterior y tienen algunas de las limitaciones.

    b) Los microcontroladores generalmente combinan sus pines de salida en puertos de 3 bits.

    c) Las instrucciones de un microcontrolador permiten manipular valores como 3 bits o como bits individuales.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

53. Conceptos básicos de los puertos de salida digital del microcontrolador:

    a) Los pines de un puerto de entrada/salida digital pueden cambiar de dirección.

    b) Ocasionalmente el microcontrolador envía datos a otro IC (opera como salida).

    c) Ocasionalmente el microcontrolador escribe datos a otro IC (opera como entrada).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

54. Conceptos básicos de los puertos de salida digital del microcontrolador, conectado a un LED:

    a) Usamos LEDs como indicadores en SE (niveles lógicos fijos o cambiantes).

    b) Un LED es un diodo que, polarizado en directa no conduce corriente.

    c) La corriente que circula es proporcional al brillo (cantidad de luz emitida) del LED.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

55. Conceptos básicos de los puertos de salida digital del microcontrolador, conectado a un LED:

    a) Se puede operar el LED a corrientes más bajas que la típica y obtener más brillo.

    b) Se puede operar el LED a corrientes más altas que la típica y obtener menos brillo.

    c) La hoja de datos del LED indica el brillo normal (cantidad de luz emitida) a una corriente típica.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

56. Conceptos básicos de los puertos de salida digital del microcontrolador, conectado a un LED:

    a) La diferencia de potencial en el LED (Vfwd), polarizado en directa, decrece con la corriente (leve).

    b) Una conexión válida es ánodo a alimentación (vía Rserie) y cátodo al pin de salida digital del microcontrolador.

    c) Una conexión válida es ánodo a tierra y cátodo al pin de salida digital del microcontrolador (vía Rserie).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

57. Conceptos básicos de los puertos de salida digital del microcontrolador, conectado a una carga:

    a) El microcontrolador (fuente) drena la corriente para accionar el dispositivo (carga), push-pull u open drain.

    b) La corriente fluye de la alimentación a través de la carga y del pin de salida del microcontrolador a tierra.

    c) El límite de corriente Iol dependerá del tipo de microcontrolador y del pin específico (validar con hoja de datos).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

58. Conceptos básicos de los puertos de salida digital del microcontrolador, conectado a un transistor:

    a) Para componentes que requieren más corriente o tensión que los que pueden manejar el microcontrolador.

    b) El pin de salida digital del microcontrolador se conecta a la base de un transistor NPN (vía $R_{base} + R_{PULLDOWN}$).

    c) El colector del transistor NPN se conecta a alimentación (via carga) y el emisor a tierra.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

59. Conceptos básicos de los puertos de salida digital del microcontrolador, conectado a un transistor:

    a) Para componentes que requieren más corriente o tensión que los que pueden manejar el microcontrolador.

    b) El pin de salida digital del microcontrolador se conecta a la base de un transistor PNP (vía $R_{base} + R_{PULLUP}$).

    c) El colector del transistor PNP se conecta a tierra (via carga) y el emisor a alimentación.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

60. Conceptos básicos de los puertos de entrada/salida digital del microcontrolador:

    a) No se deberá configurar la dirección deseada al inicializar el sistema (encendido o reinicio).

    b) No se podrá configurar el valor inicial o el valor de reposo.

    c) Se podrá configurar la topología de salida (push-pull, open drain, ..., c/R_{PULLUP}, con $R_{PULLDOWN}$, ...).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

61. Conceptos básicos de los puertos de salida digital del microcontrolador:

    a) Los niveles lógicos de salida 0 o 1 se logran mediante transistores internos.

    b) Los transistores internos conmutan el pin de salida entre alimentación o tierra (push-pull).

    c) La salida de los transistores va en serie (push-pull), uno a alimentación y otro a tierra.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

62. Conceptos básicos de los puertos de entrada/salida digital del microcontrolador:

    a) El microcontrolador (fuente) suministra la corriente para accionar el dispositivo (carga), sólo push-pull.

b) La corriente fluye de la alimentación del microcontrolador al pin de salida y a través de la cara a tierra.

c) El límite de la corriente (I_{OH}) dependerá del tipo de microcontrolador y del pin específico (validad con hoja de datos).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

63. Conceptos básicos de los puertos de salida digital del microcontrolador, conectado a un LED:

a) Vea que la corriente que circula por el LED no exceda el límite de corriente del pin de salida.

b) Vea que la suma de corriente que circula por cada LED no exceda el límite de corriente del microcontrolador.

c) Vea que la suma de corriente que circula por cada carga no excede el límite de corriente del microcontrolador.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

64. Conceptos básicos de los puertos de salida digital del microcontrolador, conectado a un FET canal N:

a) Para componentes que requieren más corriente o tensión que los que puede manejar el microcontrolador.

b) El pin de salida digital del microcontrolador se conecta al pin gate del FET canal N.

c) El pin drain del FET canal N se conecta a alimentación (vía carga) y el pin source a tierra.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

65. Conceptos básicos de los puertos de salida digital del microcontrolador, contacto húmedo:

a) (con tensión) es un interruptor en que la tensión la suministra una fuente interna.

b) Se conocen como contactos pasivos.

c) Proporciona aislamiento y seguridad esenciales en los sistemas eléctricos.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

66. Conceptos básicos de los puertos de salida digital del microcontrolador, contacto seco:

a) (libre de tensión) es un interruptor en que la tensión la suministra una fuente externa.

b) Se conocen como contactos activos o calientes.

c) No proporciona aislamiento y seguridad esenciales en los sistemas eléctricos.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

67. Conceptos básicos de los puertos de entrada digital del microcontrolador, conectado a un pulsador:

a) Los pulsadores de contacto no cambian de estado de forma limpia, rebotan (bounce).

b) El rebote mecánico (vibran) del pulsador de contacto puede durar algunas decenas de μs.

c) Podemos filtrar dicho rebote sólo por software, con un algoritmo anti-rebote (debounce).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

68. El diagrama de bloque de un STM32FXXXYY, usado en la placa NUCLEO-FXXXZZ (SDK), GPIO:

a) Este bus AHB1 transfiere datos digitales internamente a una velocidad de hasta 1000 Mbps.

b) El módulo HAL más complejo es `HAL GPIO`.

c) Cuenta que hay 1 banco de puertos GPIO conectados a AHB1.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

69. El diagrama de bloque de un STM32FXXXYY, HAL GPIO Module, controlador de entrada (opciones):

a) Alternate function: Schmitt trigger output interno al MCU (no válida).

b) Analog: Bypasses the Schmitt trigger (válida).

c) Read: salida de una de las líneas de registro de lectura asociadas (no válida).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

70. El diagrama de bloque de un STM32FXXXYY, HAL GPIO Module:

a) El pin de GPIO no tiene conectados circuitos robustos de protección contra sobretensiones.

b) El pin de GPIO no está protegido hasta un nivel de tensión de entrada aplicado de 5V.

c) El pin de GPIO está conectado simultáneamente a los controladores de entrada y salida.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

71. El diagrama de bloque de un STM32FXXXYY, HAL GPIO Module, controlador de entrada/salida:

a) La HAL se basa en la estructura C GPIOInitStruct() para configurar cualquier pin de GPIO.

b) El método de la HAL HALGPIOTogglePin() fuerza a bajo el nivel de un pin de GPIO.

c) El método de la HAL HALGPIOEXTICallback() conectado al IRQ EYTI de un pin de GPIO.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

72. El diagrama de bloque de un STM32FXXXYY, HAL GPIO Module, controlador de salida (opciones):

a) El pin de GPIO tiene velocidad de opreación: Low (válida).

b) El pin de GPIO tiene velocidad de opreación: Medium (no válida).

c) El pin de GPIO tiene velocidad de operación: High (no válida).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

73. El diagrama de un bloque de un STM32FXXXYY, Peripheral Memory Addresses:

a) Cada bus de la memoria (AHB1, AHB2, AHB3, ...) no tiene una velocidad máxima asignada.

b) Bloques de 50KB están asignados a los puertos GPIO A, B, C, D, F, ... respectivamente.

c) Hay bloques no utilizados, que no tienen un impacto significativo en las asignaciones de puertos.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

74. El diagrama de un bloque de un STM32FXXXYY, Memory-Mapped Peripherals:

a) El rango de direcciones de 32 bits generalmente está representando por ocho dígitos enteros.

b) Los periféricos se controlan y los datos se transfieren usando direcciones de 16 bits.

c) Desde la programación, es muy diferente si se comunica con memoria o periférico.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

75. El diagrama de un bloque de un STM32FXXXYY, HALGPIO Module, controlador de salida (opciones):

a) Alternate function: Generada desde un periférico MCU interno (no válida).

b) Tiene dos transistores MOSFET conectados al control de salida, operando: sólo en Push-pull.

c) Write: Salida de una línea de registro de escritura asociada (cada línea de control R/W)(válida).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

76. El diagrama de un bloque de un STM32FXXXYY, HALGPIO Module, controlador de salida (opciones):

a) El pin de GPIO tiene resistencias conectadas al pin: Pull-Up (no válida).

b) El pin de GPIO tiene resistencias conectadas al pin: Pull-Down, (no válida).

c) El pin de GPIO tiene resistencias conectadas al pin: Neither (válida).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

77. El diagrama de un bloque de un STM32FXXXYY, HALGPIO Module, controlador de entrada/salida:

a) La HAL no se basa en la estructura C GPIOInitStrcut para configurar cualquier pin de GPIO.

b) El método de la HALGPIOTogglePin() invierte el nivel de un pin de GPIO.

c) El método de la HALGPIOEXTICallback() conectado al IRQ EYTI de un pin de GPIO.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

3. Interrupciones

78. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), interrupciones:

a) No se utilizan los canales de comunicación de datos.

b) No las utilizan los procesos de acceso directo a la memoria (DMA).

c) Desde una perspectiva de alto nivel, es un proceso que detiene la ejecución de un programa.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

79. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), interrupciones:

a) No puede ser de hardware, como es el caso cuando se trata de un temporizador.

b) Puede ser de software (fin anormal de programa/dividir por cero/acceso a memoria inexistente).

c) En la terminología ARM, todas las interrupciones, de hardware o software, son excepciones.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

80. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), NVIC:

    a) El procesador ARM Cortex-M no contiene un dispositivo de hardware llamado NVIC.

    b) NVIC: Controlador de interrupción vectorial anidado (nested vectored interrupt controller).

    c) El propósito del NVIC no es gestionar excepciones.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

81. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), NVIC:

    a) A las interrupciones también se les puede asignar un nivel de prioridad.

    b) Una interrupción de menor prioridad puede interrumpir a una interrupción de mayor prioridad.

    c) Un registro de 12 bits tiene niveles de prioridad de interrupción (hasta 256 niveles de prioridad).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

82. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), NVIC:

    a) Entradas: muchas líneas etiquetadas como NMI (non-maskable interrupt).

    b) Entradas: muchas líneas etiquetadas como IRQs (interrupt request, maskable).

    c) No se puede responder a una IRQ si está deshabilitada.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

83. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), NVIC:

    a) Conectado dentro del núcleo Cortex-M.

    b) Compatible con CMCIS (mismo tipo de registros de control/configuración que GPIO).

    c) No integrado al software HAL y usa sus propias estructuras C para configurar sus registros.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

84. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), NVIC:

    a) Armar: permite que un dispositivo de hardware active la interrupción.

    b) Habilitar: permite interrumpir el proceso.

    c) Deshabilitar: no permite el proceso de interrupción.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

85. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), NVIC:

    a) Hay 16 líneas de interrupción de hardware que tienen conexiones directas al NVIC.

    b) Cualquier línea de GPIO de la MCU se puede configurar como una IRQ no enmascarable (canal).

    c) No pensar al canal de interrupción como una construcción de software.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

86. El diagrama de bloque de un STM32FXXXYY, NVIC, la ISR:

    a) Hay una lista de direcciones para estas pequeñas ISR almacenadas en una tabla de vectores.

    b) Esta característica da lugar a la palabra vector en el nombre NVIC.

    c) STM usa encuesta (polling) para pines GPIO del mismo puerto que comparten el mismo vector.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

87. El diagrama de bloque de un STM32FXXXYY, NVIC, la ISR:

    a) La ISR es un módulo de software compacto, para una tarea que respalda el dispositivo de IRQ.

    b) De haber una implementación con un ISR grande que da servicio a muchas fuentes de IRQ.

    c) Se emplea una técnica de encuesta (polling) para determinar la fuente exacta de la IRQ.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

88. El diagrama de un bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), NVIC:

    a) Puede gestionar hasta 266 canales de interrupción.

    b) 250 canales pueden ser internos/externos.

    c) 16 canales son internos.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

89. El diagrama de un bloque de un STM32FXXXYY, NVIC, 5 condiciones previas a generación de una IRQ:

    a) El dispositivo de interrupción debe estar armado (el bit de activación debe estar activado).

    b) La NVIC debe estar habilitada.

    c) Las interrupciones globales deben estar habilitadas.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

90. El diagrama de un bloque de un STM32FXXXYY, NVIC, 5 condiciones previas a generación de una IRQ:

    a) Debe ocurrir un evento de activación de hardware real.

    b) Estas 5 condiciones pueden ocurrir en cualquier orden.

    c) Deben ser verdaderos simultáneamente para que se reconozca y procese la IRQ.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

91. El diagrama de un bloque de un STM32FXXXYY, NVIC, 5 eventos posteriores al reconocimiento de una IRQ:

    a) Una interrupción solicitante se mantiene en pendiente con interrupciones no habilitadas.

    b) Una interrupción solicitante se mantiene en pendiente con IRQ solicitante de prioridad menor.

    c) Cumplidas las 5 condiciones previas, el estado de la IRQ solicitante pasa de pendiente a activo.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

92. El diagrama de un bloque de un STM32FXXXYY, NVIC, 5 eventos posteriores al reconocimiento de una IRQ:

    a) No se completará la instrucción que se está ejecutando actualmente.

    b) Se suspenderá el hilo actualmente en ejecución.

    c) El contenido de 18 registros se almacenará en la pila.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

93. El diagrama de un bloque de un STM32FXXXYY, NVIC, 5 eventos posteriores al reconocimiento de una IRQ:

    a) La NVIC interactúa automáticamente con la MCU usando firmware almacenado dentro de ella.

    b) Sólo se necesitan 18 ciclos de reloj, para pasar de reconocer una IRQ a ejecutar la ISR.

    c) Tiempo de lactancia de aproximadamente 150 μs para una MCU con frecuencia de 80MHz.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

94. El diagrama de un bloque de un STM32FXXXYY, NVIC, 5 eventos posteriores al reconocimiento de una IRQ:

UNA MIERDA TEXTUAL DE LAS DIAPOS

    a) Importante borrar el bit de activación del dispositivo, o se repetirá la solicitud de IRQ.

    b) Se llama reconocimiento a borrar un bit de activación (mediante una instrucción de sofware).

    c) El único dispositivo que tiene un reconocimiento automático es la interrupción de SysTick.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

4. Timers

95. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), Timers:

    a) Es un contador de funcionamiento libre que cuenta los pulsos de una fuente de reloj.

    b) El reloj maestro funciona a una velocidad rápida, por lo que debe dividirse mediante hardware.

    c) Puede contar o acumular conteos hasta que desborden (la cantidad de bits de cuenta).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

95. d)
96. d)
97. d)
98. d)
99. d)
100. d)
101. d)
102. d)

96. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), Timers:

    a) Tienen muchos usos, generar una base de tiempo precisa.

    b) Tienen muchos usos, medir la frecuencia de un tren de pulsos digitales entrante.

    c) Tienen muchos usos, medir el tiempo transcurrido en una señal de salida.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

97. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), Timers:

a) Tienen muchos usos, generar señales precisas de modulación de ancho de pulso (PWM).

b) Tienen muchos usos, generar pulso único con longitud programable y características de retardo.

c) Tienen muchos usos, generar señales periódicas de acceso directo a memoria (DMA).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

98. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), Timers:

a) El tiempo transcurrido está directamente relacionado con el número de impulsos contados.

b) El hardware para reducir la velocidad del reloj maestro varía y puede ser un prescaler o un PLL.

c) Una interrupción por desborde es a menudo el evento que ocurre en esta situación.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

99. El diagrama de bloque de un STM32FXXXYY, Timers, existen 5 categorías de timers STM:

a) Básico: Timer simple de 16 bits (sin pines de entrada ni de salida).

b) Se utilizan principalmente como maestros para otros cronómetros.

c) Se utilizan como fuente de reloj del convertidor digital a analógico (DAC).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

LA CORRECTA ES ESTA

100. El diagrama de bloque de un STM32FXXXYY, Timers, existen 5 categorías de timers STM:

a) Propósito general (GP): Timers de 16 o 32 bits (con pines de entrada y salida).

b) Pueden realizar todos las funciones descritas en la lista anterior.

c) Uno o dos canales / Uno o dos canales con salida complementaria.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

101. El diagrama de bloque de un STM32FXXXYY, Timers, existen 5 categorías de timers STM:

a) Alta resolución: Múltiples salidas de alta resolución posibles (seis subtimers).

b) Con este temporizador son posibles múltiples inserciones de tiempo muerto.

c) Este temporizador tiene el acrónimo HRTIM1 en terminología STM.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

102. El diagrama de bloque de un STM32FXXXYY, Timers, existen 5 categorías de timers STM:

a) Avanzado: Todas las características del timer GP.

b) Funciones adicionales relacionadas con el control del motor y la conversión de energía digital.

c) Hay tres salidas complementarias disponibles con una entrada de apagado de emergencia.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

5. Analógico/Digital

103. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), ADC SAR:

    a) La técnica usada en los ADC de STM, es la de Registro de Aproximación Sucesiva (SAR).

    b) Se copia una tensión analógica en un circuito de TRACK/HOLD (muestreo/retención) V-IN.

    c) La muestra instantánea de la tensión analógica variable se convertirá en un número digital.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

104. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), ADC SAR:

    a) La secuencia se repite hasta que se alcanza y procesa el bit menos significativo (LSB).

    b) Luego de lo cual, el N-Bit REGISTER contendrá el equivalente digital completo de V-IN.

    c) El módulo SAR LOGIC permitirá que el valor digital convertido se libere en paralelo o serie.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

105. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), ADC SAR:

    a) La arquitectura de un ADC SAR es bastante simple.

    b) El N-BIT REGISTER, arranca a un valor medio de escala (depende de los bits de conversión final).

    c) Por lo tanto, el N-BIT DAC genera una tensión igual a V-REF/2 (V-DAC).

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

106. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), ADC SAR:

    a) El ADC real de microcontroladores STM utiliza 12 bits.

    b) Un COMPARATOR entre V-IN y V-DAC, genera un 1 si V-IN es mayor a V-DAC; de lo contrario, genera un 0.

    c) El módulo SAR LOGIC se desplazará 1 bit hacia la derecha y repetirá el proceso de comparación.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

107. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), ADC SAR:

    a) Se puede ver que el proceso SAR converge lentamente hacia un valor final.

    b) Un ADC de 12 bits requeriría 12 comparaciones.

    c) Las opreaciones del módulo SAR LOGIC se realizan en software.

    d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

108. El diagrama de bloque de un STM32FXXXYY, ADC SAR se puede controlar por uno/más métodos:

a) Por encuesta (Polling) o por interrupción (Interrupt) o por Acceso Directo a Memoria (DMA).

b) Sólo por encuesta (Polling) o por interrupción (Interrupt).

c) Sólo por encuesta (Polling).

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

109. El diagrama de bloque de un STM32FXXXYY, ADC SAR, se puede controlar por encuesta (Polling):

a) Es el más complicado y requiere centrarse en el proceso ADC.

b) El paso inicial es inicializar e iniciar el ADC (métodos HALADCInit() y HALADCStart()).

c) Para acceder a los datos convertidos debe utilizarse el método HALADCPollForConversion().

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

110. El diagrama de bloque de un STM32FXXXYY, ADC SAR, modos de conversión:

a) Multi-channel Scan/Single Conversion, es el modo más simple.

b) Se muestrean múltiples líneas analógicas de entrada y luego se convierte en números digitales.

c) Luego, el número se lee del ADC y se utiliza en cualquier aplicación que lo requiera.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

111. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), ADC SAR modos de conversión:

según la lista es la a), pero para mi es la c

a) Single Channel/Continous Conversion, es el modo más simple.

b) Se muestrean múltiples líneas analógicas de entrada y luego se convierte en números digitales.

c) Los números digitales resultantes se almacenan continuamente en la memoria.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

112. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), ADC SAR modos de conversión:

a) Single Channel/Single Conversion, es el modo más simple.

b) Se muestrean múltiples líneas analógicas de entrada y luego se convierte en números digitales.

c) Cada canal es independiente y puede configurarse para tener diferentes parámetros propios.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

113. El diagrama de bloque de un STM32FXXXYY, usado en la placa de NÚCLEO-FXXXZZ (SKD), ADC SAR (Registro de Aproximación Sucesiva):

a) El método de la HALADCSTOP() detiene la conversión.

b) El método HALADCGetValue() lee la conversión.

c) El método de HALADCConvCpltCallback() conectado al IRQ de una ADC.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

114. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), Timer PWM:

   a) La generación de señal es simple e implica el uso de dos timers en modo de conteo ascendente.

   b) Cuando el conteo alcanza ese número, se genera un evento que depende del modo del timer.

   c) Cada timer GP tiene 2 registros de captura de comparación (CCRx), que almacenan un número.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

115. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), Timer PWM:

   a) La generación de señal es simple e implica el uso de un timer en modo de conteo ascendente.

   b) Las señales PWM se generan mediante el uso de un timer avanzado o de propósito general (GP).

   c) Cada timer GP tiene 4 registros de captura de comparación (CCRx), que almacenan un número.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

116. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), Timer PWM:

   a) La HAL se basa en la estructura C TIMOCInitTypeDef para configurar Timer PWM.

   b) El método de la HALTIMIRQHandler() maneja un IRQ de un Timer.

   c) El método de la HALTIMPWMPulseFinishedCallback() conectado al IRQ de un Timer.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

117. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), Timer PWM:

   a) Cada registro CCRx tiene una entrada del registro del contador de temporizador común (CNT).

   b) Cada registro CCRx tiene su propia entrada de frecuencia de reloj preescalada.

   c) Esta disposición permite implementar una salida de tren de pulsos PWM muy flexible.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

6. SPI

118. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), SPI:

   a) Es de las interfaces de comunicación del tipo InterSystemProtocol.

   b) Es de las interfaces que no ayuda a reducir la cantidad de GPIO digitales del micro-controlador.

   c) Es de las interfaces de comunicación del tipo sincrónica.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

119. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), SPI:

   a) Datos transmitidos entre nodo principal y el subnodo se sincronizan con el reloj principal.

   b) Los dispositivos SPI admiten frecuencias de reloj mucho más bajas que las interfaces I2C.

   c) La conexión de líenas entre el nodo principal y el subnodo es: MOSI a SDO.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

120. El diagrama de bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), SPI:

   a) El dispositivo que genera la señal de reloj se denomina nodo principal.

   b) La conexión de líneas entre el nodo principal y el subnodo es: MISO <- SDI.

   c) La señal de /CS (chip select), salida del subnodo se utiliza para seleccionar al principal.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

7. I2C

121. El diagrama de un bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), I2C:

   a) Es de las interfaces de comunicación del tipo Intra System Protocol.

   b) Es de las interfaces de comunicación para conectar sólo dos dispositivos diferentes.

   c) Es de las interfaces de comunicación del tipo asincrónica.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

122. El diagrama de un bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), I2C:

   a) Los bits se registran en los flancos ascendentes de la señal SCL. → X → va con el flanco descendente

   b) El protocolo permite conectar dispositivos de menor velocidad al microcontrolador.

   c) La señal de reloj siempre está controlada por el subnodo.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

123. El diagrama de un bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), I2C:

   a) Es de las interfaces de comunicación del tipo Inter System Protocol.

   b) Es de la interfaces de comunicación del tipo asincrónica.

   c) Es más lenta que la interfaz SPI. → Esto es correcto. Pensar que I2C tiene menos cables, así que la info tarda más. No tiene nada que ver, pero me ayuda a acordarme

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

124. El diagrama de un bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), I2C:

   a) Es de las interfaces de comunicación del tipo Inter System Protocol.

   b) Es de las interfaces de comunicación del tipo sincrónica.

   c) Es de las interfaces de comunicación del tipo full-duplex, bidireccional.

I2C usa un solo cable para datos, no puede ser full-duplex. SPI usa un cable para la ida y otro para la vuelta ==> Full-duplex

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

125. El diagrama de un bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), I2C:

a) El protocolo sólo permite conectar dispositivos de igual velocidad al microcontrolador.

b) Las líneas SDA y SCL son del tipo open-colector/open-drain.

c) La señal de reloj siempre está controlado por el subnodo.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

126. El diagrama de un bloque de un STM32FXXXYY, usado en la placa NÚCLEO-FXXXZZ (SDK), I2C:

a) Protocolo base es I2C Primer, sus variantes, SMBus (admin. System) y PMBus (admin. Power).

b) Las líneas SDA y SCL son del tipo push pull.

c) La línea SCL es para que el subnodo genera la señal de reloj.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

8. TPs

127. TP0-05-3er Proyecto para la placa NÚCLEO-F103RB (Event Driven Systems (EDS) - ...), en la práctica se pide: Se implementa un ejecutor cíclico de tareas. Las tareas tienen los tipos de estructuras de configuración (cfg) y datos (dta). Se observó la evolución del campo WCET, Worst-case execution time (μs):

a) taskA, Blocking code, se ejecuta de forma sincrónica (se ejecuta línea por línea).

b) taskB, Non-Blocking Code, se ejecuta de forma asincrónica (no se ejecuta línea por línea).

c) taskC, Update by Time Code, se ejecuta por ocurrencia de ticks.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

128. TP1-01-My First Statechart, en la práctica se pide: Al ingresar por primera vez al sitio de Itemis Create Cloud Editor, se genera automáticamente el modelo My First Statechart. El editor le propone un recorrido, que le sugiere realizar con fin de familiarizarse con el mismo:

a) Un statechart puede tener variables y modificarlas.

b) Un evento puede tener una guarda que opera como una proposición OR.

c) La sincronización entre statecharts es mediante el riase de un estado.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

129. TP1-01-My First Statechart, en la práctica se pide: Al ingresar por primera vez al sitio de Itemis Create Cloud Editor, se genera automáticamente el modelo My First Statechart. El editor le propone un recorrido, que le sugiere realizar con fin de familiarizarse con el mismo:

a) Un statechart puede tener variables pero no modificarlas.

b) Un evento puede tener una guarda que opera como una proposición OR.

c) La sincronización entre statecharts es mediante el riase de un evento.

d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

130. TP2-01-4to Proyecto para la placa NÚCLEO-F103RB (Sensor Statechart)(1 sensor), en la práctica se pide: Codificar la tarea modelo del sensor: Sensor Statechart para 1 sensor (BTN A = B1 Blue PushButton), generación periódica (tal como la tarea taskC del proyecto tsde-tp0-05-hw-sw-test) de estímulos leyendo 1 sensor, compilar, depurar y actualizar el repositorio

   a) Cada sensor produce un evento de la máquina de estado.

   b) Cada estado produce un case en el switch de la máquina de estado.

   c) Cada evento produce dos if en el case del estado al que afecta/excita.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

131. TP2-01-4to Proyecto para la placa NÚCLEO-F103RB (Sensor Statechart)(1 sensor), en la práctica se pide: Codificar la tarea modelo del sensor: Sensor Statechart para 1 sensor (BTN A = B1 Blue PushButton), generación periódica (tal como la tarea taskC del proyecto tsde-tp0-05-hw-sw-test) de estímulos leyendo 1 sensor, compilar, depurar y actualizar el repositorio

   a) Cada sensor produce dos eventos de la máquina de estado.

   b) Cada estado produce un case en el switch de la máquina de estado.

   c) Cada evento produce un if en el case del estado al que afecta/excita.

   d) Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso).

9. Respuestas

1. b)
2. c)
3. d)
4. d)
5. a)
6. c)
7. a)
8. d)
9. d)
10. d)
11. b)
12. a)
13. c)
14. a)
15. b)
16. d)
17. d)
18. d)
19. d)
20. d)
21. c)
22. c)
23. b)
24. d)
25. b)
26. c)
27. c)
28. b)
29. b)
30. d)
31. c)
32. d)

33. a)
34. b)
35. c)
36. d)
37. b)
38. d) ???
39. a)
40. a)
41. c)
42. a)
43. b)
44. c)
45. b)
46. c)
47. a)
48. Imagen
49. Imagen
50. c)
51. d)
52. a)
53. a)
54. a)
55. c)
56. b)
57. d)
58. d)
59. d)
60. c
61. d)
62. d)
63. puede ser la d
64. d)
65. d)

66. a)
67. a) decenas de ms, filtrar por software Y hardware.
68. d) Estaba la c) pero página 9 de micros dice 6 bancos.
69. b) pero solo por el (valido)?
70. d) Estaba la c) pero preg 25 de Ramiro está mal.
71. a)
72. a)
73. c)
74. a)
75. d) Consultar denise
76. c) Consultar denise
77. b) Consultar denise
78. c)
79. c)
80. b)
81. a)
82. b)
83. b)
84. d)
85. a)
86. d)
87. d)
88. c
89. d
90. d)
91. d)
92. b
93. a
94. d)
95. d)
96. d)
97. d)
98. d)

99. d)
100. d)
101. d)
102. d)
103. d)
104. d)
105. d)
106. d)
107. d)
108. a)
109. c)
110. b)
111. a)
112. a) Creo que b) y c) son en función de a), por lo que son falsas.
113. d)
114. b)
115. d)
116. d)
117. d)
118. c)
119. a)
120. a)
121. d) → Para mi está mal
122. b)
123. c)
124. b)
125. b)
126. a)
127. d)
128. a)
129. c)
130. b)
131. d)