



Extended light switch example as a Harel statechart with composite states

18. 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:

- Reset \Rightarrow LIGHT OFF
- ON_pressed \Rightarrow MANUAL MODE
- ON_pressed \Rightarrow MANUAL MODE
- ON_pressed \Rightarrow MANUAL MODE
- MOT_pressed \Rightarrow MOTION DETECTION MODE (NO MOTION DETECTED)
- motion_detected \Rightarrow MANUAL MODE
- ON_pressed \Rightarrow MANUAL MODE
- OFF_pressed \Rightarrow LIGHT OFF
- fin

19. Para el diagrama de estado de Harel de la figura, al recibir la siguiente secuencia de entradas:

Indicar por qué valores evolucionará la variable **brightness** (estado a estado) luego de ser reiniciada:

- Reset \Rightarrow 0
- ON_pressed \Rightarrow 1
- ON_pressed \Rightarrow 1 + (BRIGHTNESS * 1.5)
- ON_pressed \Rightarrow 1 + (BRIGHTNESS * 1.5)
- MOT_pressed \Rightarrow 1
- motion_detected \Rightarrow 1
- ON_pressed \Rightarrow 1
- OFF_pressed \Rightarrow 0
- fin

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

- Los pines de un puerto de entrada/salida digital pueden cambiar de dirección
- Ocasionalmente el microcontrolador envía datos a otro IC (opera como salida)
- Ocasionalmente el microcontrolador recibe datos a otro IC (opera como entrada)
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- El uC (fuente) drena la corriente para accionar el dispositivo (carga), push-pull u open drain
- La corriente fluye de la alimentación a través de la carga y del pin de salida del uC a tierra
- El límite de corriente (I_u) dependerá del tipo de uC y del pin específico (validar c/hoja de datos)
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

Ojo con los pines de entrada y salida

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

- La diferencia de potencial en el LED (V_{LED}), polarizado en directa, decrece c/la corriente (I_{LED})
- Una conexión válida es ánodo a alimentación (Vía Reseñe) y cátodo al pin de salida digital del uC
- Una conexión válida es ánodo a tierra y cátodo al pin de salida digital del uC (Vía Reseñe)
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- Para componentes que requieren más corriente o tensión que los que pueden manejar el uC
- El pin de salida digital del uC se conecta al pin gate del FET canal N
- El pin drain del FET canal N se conecta a alimentación (Vía carga) y el pin source a tierra
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

24. Conceptos básicos de los puertos de entrada/salida digital del microcontrolador, contacto húmedo:

- (con tensión) es un interruptor en que la tensión la suministra una fuente interna
- Se conocen como contactos pasivos
- Proporciona aislamiento y seguridad esenciales en los sistemas eléctricos
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

25. El diagrama de bloque de un STM32FXXXYY, HAL_GPIO Module:

- El pin de GPIO no tiene conectados circuitos robustos de protección contra sobretensiones
- El pin de GPIO no está protegido hasta un nivel de tensión de entrada aplicado de 5V
- El pin de GPIO está conectado simultáneamente a los controladores de entrada y salida
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

26. El diagrama de bloque de un STM32FXXXYY, HAL_GPIO Module, controlador de salida (opciones):

- El pin de GPIO tiene velocidad de operación: Low (válida)
- El pin de GPIO tiene velocidad de operación: Medium (no válida)
- El pin de GPIO tiene velocidad de operación: High (no válida)
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

27. El diagrama de bloque de un STM32FXXXYY, HAL_GPIO Module, controlador de entrada/salida:

- La HAL no se basa en la estructura C GPIO_Instruct para configurar cualquier pin de GPIO
- El método de la HAL HAL_GPIO_TogglePin() fuerza a bajo el nivel de un pin de GPIO
- El método de la HAL HAL_GPIO_EXTI_Callback() conectado al IRQ EXTI de un pin de GPIO
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- No puede ser de hardware, como es el caso cuando se trata de un temporizador
- Puede ser de software (fin anormal de programa/dividir por cero/acceso a memoria inexistente)
- En la terminología ARM, todas las interrupciones, de hardware o software, son excepciones
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- Conectado dentro del núcleo Cortex-M \rightarrow ESTA AFUERA
- Compatible con CMICIS (mismo tipo de registros de control/configuración que GPIO)
- No integrado al software HAL y usa sus propias estructuras C para configurar sus registros
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- Armar: permite que un dispositivo de hardware active una interrupción
- Habilitar: Permite interrumpir el proceso
- Deshabilitar: No permite el proceso de interrupción
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- La NVIC interactúa automáticamente con la MCU usando firmware almacenado dentro de ella
- Sólo se necesitan 18 ciclos de reloj, para pasar de reconocer una IRQ a ejecutar la ISR
- Tiempo de latencia de aproximadamente 150us para una MCU con frecuencia de 80MHz
- Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

IRQ = Interrupt request?

Firmware es un software hijo en un uC

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

GPDI

INT

TIM

TIM

ADC

ADC

ADC

PWM

SPi

SPi

I2C

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

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

34. 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)

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

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

- a. Single Channel/Continuous Conversion, es el modo más simple
- b. Se muestran 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)

37. El diagrama de bloque de un STM32FXXXYY, usado en la placa NUCLEO-FXXXZZ (SDK), ADC SAR:

- a. El método de la HAL HAL_ADC_Stop() detiene la conversión
- b. El método de la HAL HAL_ADC_GetValue() lee la conversión
- c. El método de la HAL HAL_ADC_ConvCpltCallback() conectado al IRQ de un ADC
- d. Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- a. La HAL se basa en la estructura C TIM_OC_InitTypeDef para configurar cualquier Timer PWM
- b. El método de la HAL HAL_TIM_IRQHandler() maneja un IRQ de un Timer
- c. El método de la HAL HAL_TIM_PWM_PulseFinishedCallback() conectado al IRQ de un Timer
- d. Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- a. Es de las interfaces de comunicación del tipo "Inter System Protocol"
- b. Es de las interfaces de comunicación del tipo "asíncronica"
- c. Es de las interfaces de comunicación del tipo "half duplex", "bidireccional"
- d. Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- a. Datos transmitidos entre el nodo principal y el subnodo se sincronizan con el reloj del subnodo
- b. Los dispositivos SPI admiten frecuencias de reloj mucho más altas que las interfaces I2C
- c. La conexión de líneas entre el nodo principal y el subnodo es: /CS => SCLK
- d. Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

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

- a. Es de las interfaces de comunicación del tipo "Inter System Protocol"
- b. Es de las interfaces de comunicación del tipo "asíncronica"
- c. Es más lenta que la interfaz SPI
- d. Ninguna de las anteriores o Todas las anteriores (lo que aplique al caso)

I2C

42. El diagrama de bloque de un STM32FXXXYY, usado en la placa NUCLEO-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-collector/open-drain"
- 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)

43. TP0-05 - Ser Proyecto p/placa NUCLEO-F103RB (Event Driven System (EDS) - ...), en la práctica se pide:

- a. Se implementa un ejecutor cíclico de tareas
- b. Cuyas tareas tienen los tipos estructuras de configuración (cfg) y datos (dta)
- c. Se observó la evolución del campo WCEI, Worst-case execution time (microseconds)
- d. task_a, Blocking Code, se ejecuta de forma asíncronica (no se ejecuta línea por línea)

44. TP1-01 - My First Statechart, en la práctica se pide:

- a. Al ingresar por primera vez al sitio de Itemis Create Cloud Editor => se genera automáticamente el modelo My First Statechart
- b. El editor le propone un recorrido, que le sugiere realizar un fin de familiarizarse con el mismo
- c. Un statechart puede tener variables, pero no modificarlas
- d. Un evento puede tener una guarda que opera como una proposición

45. TP2-01 - 4to Proyecto p/placa NUCLEO-F103RB (Sensor Statechart) (1 sensor), en la práctica se pide:

- a. Codificar la tarea modelo del sensor: Sensor Statechart para 1 sensor (BTN_A => B1 Blue PushButton), generación periódica (tal como la tarea task_c del proyecto tise-tp0_05-hw_sw_test) de estímulos leyendo 1 sensor, compilar, depurar y actualizar el repositorio
- 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)

Dudas: 10) SDK???

Palabras clave: SDK, Registro, von Neumann vs. Harvard, Pipelining, Boot strap, vector reset, tipos de memoria, MO/MO+, Harel, Pull up/down, contacto húmedo, CMCIS, NVIC, IRQ, ISR, TIMERS, ADC SAR, SPI, I2C

Que duro esto

Quiero hacer la tarea

Algunas respuestas:

1) SDK: Software development kit: conjunto de herramientas de software utilizadas en el desarrollo del MVP. Estas incluyen: bibliotecas de código Depuradores, compiladores, documentación, ejemplos y API's.
Ayudan a tener una idea de cuánta memoria requerirá el proyecto, el tiempo de respuesta y otras cosas. La idea es usarlas para evitar meternos con el hardware. El producto final NO debe depender del SDK usado en el desarrollo.

El registro es un lugar donde se almacenan o acumulan datos

Hay registros que almacenan instrucciones

Esto no apareció en este examen pero podría tomarlo:

ISA: instruction set architecture: formato en el que se escriben las instrucciones cambia de procesador a procesador

PC: program counter: registro que tiene la ubicación de la siguiente instrucción

En general, luego de ejecutar una instrucción, $PC = PC + 1$, salvo saltos o interrupciones. Creo que el PC va a volver a aparecer

El procesador funcionaría así: recibe la dirección de la instrucción en el PC.

La unidad de control busca en la memoria y decodifica la instrucción.

La unidad de control manda las ordenes (qué registros sumar, por ej.)

En general, por cada instrucción tenemos un ciclo: buscar->decodificar->ejecutar

Esperar a que termine un ciclo para empezar con el siguiente es ineficiente.

Para evitar esto se usa el Pipelining, que es hacer ciclos paralelos.

Esto acelera el procesamiento, pero también añade complejidad.

Qué pasa si necesito un resultado de la operación anterior para ejecutar la siguiente?

Si bien esto puede pasar, es poco frecuente

Código boot strap = código q se ejecuta cuando arranca o reinicia el uC

Vector de reset = dirección en memoria del código bootstrap

ISR: Interrupt service routine: código a ejecutar cuando hay una interr.

SAR: Successive approximation register: técnica que usa el CAD. Es básicamente una búsqueda binaria. Está copado como lo explica Norris.

SPI: Serial peripheral interface

I2C: inter-integrated circuit.

OJO: ISP e I2C son protocolos *INTRA* sistema, comunican distintos integrados, pero dentro de un mismo sistema.

Un protocolo inter-sistema es el USB, por ej.

Von Neumann ==> Datos y programa comparten el mismo BUS. Lento
Harvard ==> Bus de Datos por un lado, Bus de programa por otro. Rápido y complejo