

INFO MICROCONTROLADORES

Sistemas Digitales

Estos son cualquier dispositivo que tenga como función la transmisión, procesamiento, almacenamiento o generación de señales digitales

Señal Análoga

Una señal analógica quiere decir que la información, señal, para pasar de un valor a otro pasa por todos los valores intermedios esto quiere decir que es continua, puede tomar valores infinitos.

Señal Digital

Una señal digital es aquella variación que permite transmitir información discreta en amplitud y en intervalos de tiempo, dicho de otra forma, esta es discontinua. Estas son importantes pues que son fáciles de procesar por medios digitales, además de proveer varias ventajas, entre ellas:

- Reproducibilidad de resultados, no varían en proporción a la temperatura, voltaje de alimentación, antigüedad y otros factores.
- Facilidad de diseño y modelado.
- Flexibilidad, velocidad y facilidad de manipulación y procesamiento.
- Programabilidad.

Un sistema que procesa señales digitales no es capaz de comprender valores analógicos de forma directa; de hecho, un sistema digital trabaja completamente con valores codificados de manera binaria. Los circuitos internos de un sistema digital solamente son capaces de comprender la existencia de voltaje o su falta. Estos dos estados pueden ser representados por medio del sistema binario; cada combinación se hace por medio de los números binarios.

Los sistemas digitales se dividen en dos según el álgebra booleana

- Sistemas digitales Combinacionales: Sistemas cuyas salidas solo dependen del estado de sus entradas en un momento dado, no dependen de los estados previos de las entradas.
- Sistemas digitales secuenciales: Sistemas cuyas salidas dependen además del estado de sus entradas en un momento dado, de estados previos.

Los sistemas digitales se encuentran conformados por dos partes:

- **Parte lógica:** conformada por el software y el firmware.
 - o **Software:** secuencia ordenada de cada instrucción específica almacenada en memoria definiendo exactamente que tarea se realizará.

o **Firmware:** es un bloque de instrucciones de programa para propósitos específicos, grabado en una memoria de tipo no volátil, que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo. Son instrucciones básicas que indica al sistema digital como debe de funcionar, como arrancar y que operaciones iniciales debe realizar para dar paso a las funciones asignadas por el usuario.

- **Parte física:** conformada por el hardware el cual es la parte tangible de un sistema digital, conformada por partes como sus componentes eléctricos, electrónicos y mecánicos.

Sistemas embebidos

Sistema de computación diseñado para realizar una o pocas funciones dedicadas e integrada en un dispositivo. La palabra embebido implica que se encuentra adjunto o ligado dentro de algo. Un sistema se refiere a un arreglo en el que todas las unidades trabajan en conjunto de acuerdo a un plan.

La funcionalidad será la que nos definirá lo que realmente es un sistema embebido. Estos son creados para una o muy pocas funciones específicas. Un sistema embebido debe cumplir y dedicarse por completo a la función para la que fue creado.

Características de los sistemas embebidos

- Ejecutar una o muy pocas tareas: En el momento que un sistema se encuentra ejecutando muchas tareas, pierde su objetivo de ser dedicado lo que no permite la optimización de recursos y tamaño del dispositivo final.
- Procesamiento en tiempo real: Se considera que un sistema realiza procesamiento en tiempo real cuando es capaz de terminar procesos más rápido de lo que suceden eventos externos que modifiquen el sistema de modo que su reacción sea aparentemente instantánea.
- Bajo costo: El sistema embebido es concebido para ser producido en miles o millones de unidades y comercializado, por lo que debe tener un costo bajo.
- Bajo consumo de potencia: Se enfoca en la optimización del consumo energético del sistema. Tanto por el costo que generaría como el tamaño, de ser necesario de contar con una batería. Pues si requiere mayor potencia, de mayor tamaño sería la batería dejando de ser un sistema embebido.
- •Mínimo uso de memoria: El tamaño del código debe ser el mínimo posible, al igual que el uso de memoria que responde siempre a mantener dimensiones físicas pequeñas y costos bajos.
- Dimensiones físicas pequeñas: Si se desea reducir el costo y consumo energético, se debe tener la capacidad de administrar bien los recursos del sistema utilizando solo el mínimo necesario. Al momento de diseñar un sistema embebido, se debe considerar que su unidad de procesamiento sea de dimensiones físicas pequeñas para no afectar los costos de producción.
- Capaz de interactuar con el mundo físico: Un sistema embebido interactuará con su entorno por medio de sensores y actuadores, por lo que periféricos que permitan esta interacción con el mundo físico son obligatorios.

Microcontrolador

Circuito integrado programable, compuesto por una unidad central de procesamiento, memoria, periféricos de entrada/salida y buses que interconectan todas las partes, dentro de un chip. Ejecuta instrucciones grabadas en su memoria para realizar una tarea específica.

- **Unidad central de procesamiento (CPU)**

Dispositivo encargado de interpretar las instrucciones de un programa mediante la realización de operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.

Es un chip con integración a gran escala o VLSI, lo cual implica que estará integrado por miles de transistores

- Unidad lógica aritmética (ALU): circuito digital combinacional encargado de realizar operaciones aritméticas y lógicas de números binarios enteros.
 - Suma aritmética
 - Resta aritmética (complemento a 2)
 - Operaciones lógicas
 - Producto
 - Suma lógica
 - Comparación
 - Complementación
 - Enmascaramiento
 - Transferencia
 - Rotación
- Unidad de control (CU): encargada de decodificar las instrucciones y controlar cada uno de los componentes internos de la CPU para que esta funcione.
- Registros: Pequeñas memorias de rápido acceso. Su tamaño es dependiente de la arquitectura de la CPU, van desde 4 hasta 64 bits. Se dividen en:
 - Registros de propósito general
 - También conocido como registros de datos, son utilizados para guardar números enteros. Anteriormente únicamente se utilizaba el registro acumulador.
 - Registros de memoria
 - Es un pequeño espacio de memoria sumamente veloz, en el cual se almacena información que se utiliza frecuentemente, lo que agiliza la ejecución de instrucciones.
 - Registros de punto flotante
 - La FPU no utiliza los registros de propósito general, sino que sus propios registros, llamados pila de registros. Su nomenclatura es

S_n , siendo n el número de registro de punto flotante yendo de S_0 a S_{31} .

- Registros constantes
 - Los registros constantes son valores de solo lectura definidos por el hardware
- Registros de propósito específico
 - Guardan información del estado de ejecución del programa.
- Encapsulado: es lo que rodea a la oblea de silicio en sí, para darle consistencia, impedir su deterioro y permitir el enlace con los conectores externos que lo acoplaran a su zócalo de la placa base.
- Puertos: es la manera en que el procesador se comunica con el mundo externo. Un puerto es análogo a una línea de teléfono. Cualquier parte de la circuitería de la computadora con la cual el procesador necesita comunicarse, tiene asignado un «número de puerto» que el procesador utiliza como si fuera un número de teléfono para llamar circuitos o a partes especiales.
- **Memorias**

Una de las características principales de un microcontrolador es el tamaño reducido, por lo cual las memorias suelen estar dentro del mismo integrado. No se encontrarán de un gran tamaño debido al espacio reducido.

- RAM Memoria de acceso aleatorio (RAM) es el lugar donde se almacena la información volátil, puesto que al apagar o desconectar el procesador esta se perderá. Existen dos tipos de RAM: estática o SRAM, la cual mantendrá la información hasta el momento en el que se le retire la potencia, y la dinámica o DRAM, la cual se sobrescribe constantemente para que no se pierda la información contenida en ella.
- ROM La memoria de solo lectura o ROM es utilizada para almacenar datos de forma permanente; el procesador puede leer información mas no escribir información en ella. Al no necesitar una fuente de poder, esta memoria es conocida como memoria no volátil.
- EPROM ROM borrable-programable o EPROM son memorias que pueden ser reprogramadas si previamente los datos almacenados son borrados con luz ultravioleta, la cual es aplicada por medio de una ventana de cuarzo que se encuentra ubicada sobre el microcontrolador.
- EEPROM (Electrically Erasable Programmable Read Only Memory) Memoria de sólo lectura programable y borrable eléctricamente. Chip de memoria que retiene su contenido sin energía. Puede borrarse, tanto dentro del computador como

externamente. Por lo general requiere más voltaje para el borrado que el común de +5 voltios usado en circuitos lógicos.

- **Periféricos**

- Entradas y salidas

Debido a que un microcontrolador está diseñado para interactuar con su entorno físico, comunicarse con su exterior es imprescindible los microcontroladores cuentan con una serie de pines de propósito general que funcionan como entradas y salidas digitales del sistema que brindan información por medio de señales eléctricas.

- Circuitos especializados

Dentro de estos se habla de cualquier circuito electrónico integrado dentro del microcontrolador encargado de realizar una función específica. Algunos periféricos, como el reloj del sistema, son necesarios para el funcionamiento del microcontrolador. Existe una gran variedad de circuitos especializados y cada uno varía entre modelos diferentes. Es necesario analizar un modelo específico para poder detallar cada uno.

- **Buses**

Componentes de interconexión dentro del sistema, encargados de la transmisión de datos entre componentes internos. Existen tres buses dentro de un microcontrolador:

- Bus de control: encargado de la transmisión de ordenes entre módulos para el uso y acceso a las líneas de datos y direcciones.
- Bus de datos: encargado de transmitir datos entre módulos internos.
- Bus de direccionamiento: encargado de transportar la dirección de lectura o escritura entre unidades.

Arquitectura de un microprocesador

Existen dos tipos de set de instrucciones para un procesador:

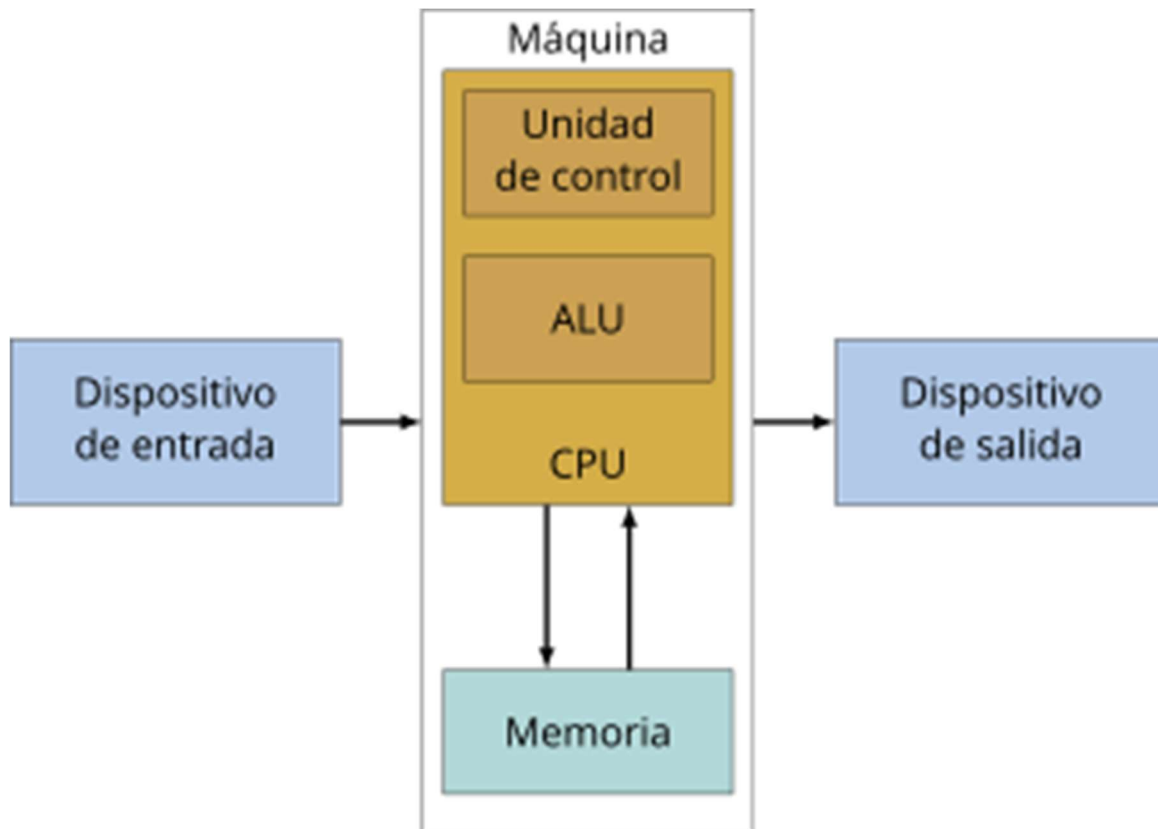
- **Arquitectura RISC o Reduce Instruction Set Computing**, el cual está basado en microcontroladores fáciles de construir y rápidos en el procesamiento
- **Arquitectura CISC o Complex Instruction Set Computing**, el cual es capaz de ejecutar instrucciones complejas, pero consumiendo mayor energía.

Arquitectura de memoria

Un microprocesador está compuesto por varios bloques interconectados entre sí pero cada uno de ellos tiene una función específica que cumplir, al diseño e interconexión de estos bloques se le denomina arquitectura.

En la actualidad existen dos arquitecturas de memoria. Estas han ido evolucionando desde su aparición, pero conservan el nombre e idea originales.

- **Arquitectura Von-Neumann** Los primeros sistemas computacionales surgieron de la arquitectura descrita por el matemático John Von-Neumann en 1945 quien expuso que para construir un sistema computacional eran necesario cuatro componentes básicos:
 - Unidad de procesamiento central
 - Unidad de memoria
 - Entradas y salidas
 - Buses de comunicación



La importancia de esta arquitectura es que opera con una sola unidad de memoria; por lo que los datos y el programa se graban en la misma memoria y comparten los mismos buses de comunicación.

- **Arquitectura Harvard** Fue desarrollada de manera casi paralela a la de Von-Neumann por estudiantes de la Universidad de Harvard, quienes trabajaban en el sistema computacional llamado Harvard Mark I. Esta arquitectura funciona con los siguientes componentes básicos:
 - Unidad de procesamiento central
 - Unidad de memoria de instrucciones
 - Unidad de memoria de datos

- Entradas y salidas
- Buses de comunicación



Este sistema planteaba una arquitectura similar a la de Von-Neumann con la diferencia de que incorporaba una memoria para instrucciones y una memoria para datos, cada una con un bus diferente, lo que permitía una mayor velocidad de procesamiento

ARM

Por sus siglas en inglés *advanced RISC machine*, es una arquitectura RISC de 32 y 64 bits. Fue desarrollada por la compañía ARM Holdings que solamente se limita a licenciar sus diseños.

La característica principal de los procesadores ARM, es su bajo consumo de potencia, lo cual los hace perfectos para el uso en dispositivos portátiles. Esta simplicidad se debe a la falta de microcódigo y su *Instruction Set Architecture (ISA)* basado en RISC que utiliza una menor cantidad de transistores en comparación con otras arquitecturas.

Córtex-A

Esta familia de procesadores está basada en la aplicación y utilización para terceros. Los fabricantes de diferentes productos compran una licencia de esta familia para poderlos adecuar al usuario final. Generalmente son utilizados en Smartphone, videojuegos y tabletas.

Córtex-R

La característica principal de esta familia es la velocidad de procesamiento, puesto que están destinados a aplicaciones en tiempo real, donde se necesita una rápida respuesta. Utilizados en el campo de equipo médico, aviación y robótica.

SecureCore

Procesadores de bajo consumo de potencia, bajo costo y buen rendimiento. Se especializan en aplicaciones de seguridad.

Córtex-M

La familia de los Cortex-M fue diseñada específicamente para el mercado de los microcontroladores. Los procesadores Cortex-M cuentan con un alto rendimiento comparado con otro procesador típico de la mayoría de microcontroladores, destacan por su bajo consumo de potencia. Han llegado a ser tan bien aceptados que dominan el mercado de los microcontroladores; es la arquitectura más empleada de 32-bits en la historia. Entre sus características principales se encuentran:

- Trabajar con un set de instrucciones Thumb-2
- Alto rendimiento y eficiencia
- Compatibilidad con lenguajes C y ASM
- Herramientas de fácil aprendizaje para desarrollo
- Soporte para sistemas operativos
- Fácil depuración
- Manejo de fallas
- El manejo de interrupciones se hace por medio del NVIC
- Son utilizados por Texas Instrument y Arduino.

Los procesadores Cortex-M tienen 6 miembros diferentes dentro de su familia; apuntan cada uno a una necesidad diferente ya sea de alto rendimiento o de ultrabajo consumo de potencia. Los procesadores Cortex-M son:

- Cortex-M0: un procesador muy pequeño (de 12 mil compuertas), de bajo costo, ultrabajo consumo de potencia y sistemas embebidos reducidos. Utiliza una versión ARMv6-M con arquitectura de memoria Von Neumann.
- Cortex-M0+: el procesador más eficiente en consumo de potencia de la familia. Diseñado para sistemas embebidos pequeños, cuenta con un tamaño similar al Cortex-M0, pero con algunas características extras. Utiliza una versión ARMv6-M con arquitectura de memoria Von Neumann.

- Cortex-M1: procesador pequeño, con un diseño optimizado para su implementación en diseños de FPGA. Instrucciones y arquitectura iguales a las del Cortex-M0.
- Cortex-M3: procesador pequeño, pero de alto rendimiento para microcontroladores de bajo consumo de potencia con un set de instrucciones alto (Thumb-2) que le permite realizar tareas complicadas de manera rápida. Cuenta con un divisor de hardware, una unidad MAC (siglas en inglés de multiply-accumulate), una unidad para depuración y diferentes herramientas para desarrollo. Utiliza una versión ARMv7-M con arquitectura de memoria Harvard.
- Cortex-M4: contiene todas las características de un Cortex-M3, con instrucciones adicionales para el procesamiento digital de señales, esto debido a que incluye una unidad DSP incorporada. También, cuenta con instrucciones SIMD (siglas en inglés para single instruction multiple data) y operaciones de un ciclo MAC. Algunas versiones incluyen un coprocesador de punto flotante con soporte para el estándar IEEE-754. Estos últimos reciben el nombre de Cortex-M4F. **Cortex-M4F es el procesador que utiliza la tiva que tenemos.**
- Cortex-M7: procesadores de alto rendimiento para microcontroladores de alto nivel con aplicaciones de procesos intensivos. Contiene todas las características de un Cortex-M4 con soporte adicional para unidad de 47 punto flotante de doble precisión y algunas características de memoria como caché y TCM (siglas en inglés para tightly coupled memory).

Microcontrolador TM4C123GH6PM

El TM4C123GH6PM de Texas Instruments es un microcontrolador de alto rendimiento de 32-bits con una arquitectura ARM Cortex-M y una amplia cantidad de herramientas para el desarrollo de aplicaciones.

Características

- Set de instrucciones: Las arquitecturas ARM implementan dos sets de instrucciones: ARM y Thumb; su diferencia es su codificación. ARM es un set de instrucciones de 32 bits de largo, mientras que las instrucciones de Thumb son de 16-bits. ARM tiene una codificación de 4 bytes, esto implica que, sus instrucciones de 32-bits se dividen en cuatro bytes mientras que Thumb cuenta con una codificación de 2 bytes. El desarrollador decide qué tipo de codificación utilizará, de modo que el procesador decodificará las instrucciones como tipo Thumb o ARM. Las instrucciones ARM son más rápidas y potentes (hay instrucciones que solo están en este modo) pero implican mayor consumo de memoria y de electricidad. Thumb por el contrario es más limitado, con instrucciones que ocupan 2 bytes, pero con menor consumo de potencia. En las versiones ARMv-7 se comienza a implementar un nuevo set de instrucciones conocido como Thumb-2. Las instrucciones Thumb-2 son una mezcla entre ARM y Thumb, de modo que las instrucciones pueden ser de 16 o 32 bits. Tiene un código 26 % menor que ARM y 25 % en mejora de rendimiento con respecto a Thumb.

- Temporizador del sistema (SysTick) Los procesadores ARM Cortex-M4F incluyen un temporizador llamado SysTick. Este se incluye en todos los diseños de la familia Cortex-M. Su importancia radica en que los periféricos incluidos dentro de un microcontrolador dependen del fabricante; sin embargo, el temporizador del sistema siempre estará presente en todos los diseños ARM, por lo que su uso no depende del mismo. Es un temporizador simple de 24-bits decreciente que se puede usar de diferentes formas.
- Nested vectored interrupt controller (NVIC) El microcontrolador TM4C123GH6PM incluye un dispositivo llamado nested vectored interrupt controller (NVIC). Este se encarga de manejar todas las interrupciones.
- System control block (SCB) El SCB, es un bloque encargado del control del sistema, configuración, control y reporte de las excepciones o fallas del mismo.
- Memory protection unit (MPU) Sistema de protección de memoria para ARM7. Utiliza el modelo PMSA (siglas en inglés para protected memory system architecture). Es el encargado de proteger regiones de la memoria, evitar traslape de regiones, los permisos de acceso a memoria y la exportación de atributos al sistema.
- Floating point unit (FPU) Coprocesador del sistema encargado de dar soporte a operaciones de precisión simple, como suma, resta, multiplicación, división, multiplicación, acumulación y raíz cuadrada. También, permite la conversión entre datos e instrucciones de enteros y de punto flotante. Cuenta con las siguientes características:
 - Instrucciones de precisión simple de 32 bits para operación de datos.
 - Combinación entre instrucciones de multiplicación y acumulación para mejora de precisión.
 - Soporte en hardware para conversión, suma, resta multiplicación con opción para acumulación, división y raíz cuadrada.
 - Registros de 32 bits dedicados
- Memoria integrada en chip.

El microcontrolador trae integrado dentro del chip las siguientes memorias:

- 32 KB de memoria SRAM de ciclo único
- 256 KB de memoria flash
- 2KB de memoria EEPROM
- ROM interna con Tivaware cargado para desarrollo en lenguaje C
- Periféricos Cuenta con diferentes periféricos encargados de las interfaces de comunicación, integración del sistema, control avanzado de movimiento y soporte analógico.

- Módulo de hibernación Provee la lógica para apagar el procesador principal y sus periféricos y volver a activarlo en eventos basados en tiempo. Entre sus características se destacan:
 - Control de la potencia de sistema con regulador discreto externo
 - Control de la potencia del sistema por medio de registros de control
 - Pin dedicado para despertar el sistema por medio de una señal externa
 - Detección de batería baja con opción a despertar cuando esto suceda
 - Retención de estados en los pines durante hibernación
- Watchdog timer: Es un control incluido en el sistema, encargado de reaccionar de una manera definida en caso de fallo del sistema o un error en el programa. El TM4C123GH6PM puede generar una interrupción enmascarable, no enmascarable o un reinicio cuando se alcanza un valor de tiempo muerto. El TM4C123GH6PM cuenta con dos módulos de watchdog timer:
 - Watchdog timer 0: usa el reloj del sistema como temporizador
 - Watchdog timer 1: usa el oscilador principal como temporizador
- JTAG

JTAG (siglas en inglés de joint test action group) es un estándar IEEE que define un puerto de acceso de pruebas y un escaneo en la arquitectura por medio de una interfaz, lo que proporciona información de los circuitos y componentes para realización de ensayos, observación, control y depuración del código.

Periféricos principales

Se entiende por periféricos a todos aquellos circuitos especializados auxiliares e independientes a la unidad central de procesamiento. El controlador TM4C123GH6PM cuenta con una gran cantidad de periféricos, como se especificó en sus características.

- **Señal de reloj:** La señal de reloj es una señal digital cuadrada encargada de la sincronización de los componentes de un sistema digital.
Esta señal oscila entre estados altos y bajos con un ciclo de trabajo del 50 %, esto implica que los primeros duran lo mismo que los otros. Se mide utilizando las unidades Hertz, que representan cantidad de ciclos por segundo. La señal de reloj es de gran importancia en cualquier sistema digital, dado que dicta la ejecución de las instrucciones dentro de la unidad de procesamiento y la sincronización de los componentes internos. La unidad de

procesamiento necesita un número establecido de ciclos de reloj para ejecutar una instrucción, de modo que, a una mayor velocidad de reloj, mayor cantidad de instrucciones se ejecutarán en menor tiempo.

La sincronización de los componentes internos por medio del reloj funciona como un metrónomo, cada flanco de subida o bajada (según se especifique), cada componente realizará una acción, de modo que los estados cambiarán todos al mismo tiempo y mantendrá el funcionamiento del sistema de manera ordenada, evitando congestión dentro del mismo y corrigiendo la diferencia de tiempos de ejecución entre componentes.

Al iniciar el diseño de un sistema embebido, debe configurarse inicialmente la velocidad de reloj y su fuente para el sistema. Es importante también recordar que cada periférico necesita estar sincronizado con la unidad de procesamiento por lo que es imprescindible la configuración y habilitación de una señal de reloj para cada periférico a utilizar.

- **Las entradas y salidas de propósito general GPIO** (siglas en inglés para general purpose inputs and outputs) son pines sin algún propósito específico definido cuyo comportamiento puede ser controlado por el usuario. Un puerto GPIO es un grupo de pines GPIO ordenados y controlados como grupo, generalmente de hasta 8 pines. Si se configuran como entradas, su propósito será introducir datos externos al sistema digital para su procesamiento. Si son configurados como salidas, su propósito será recibir la información procesada por el sistema digital y reproducirla

El controlador TM4C123GH6PM utiliza el estándar LVCMOS3.3, lo cual implica que sus circuitos utilizan tecnología CMOS y un nivel de voltaje de 3,3 V, donde un estado lógico alto de entrada varía entre 2,0 V y 3,3 V, un estado lógico bajo de entrada varía entre 0 V y 1,3 V.

El módulo GPIO provee control sobre 8 pines diferentes e independientes de cada puerto. Se cuenta con 6 puertos diferentes nombrados alfabéticamente desde A hasta F. Cada pin puede ser configurado con cada uno de los siguientes parámetros

- Configuración como entrada o salida: la configuración por defecto es de entrada.
 - En modo entrada, pueden generar interrupciones en niveles altos, bajos, flancos de subida, flancos de bajada o ambos.
 - En modo de salida pueden ser configurados para un drive strength de 2, 4 u 8 mA.
 - Opción para resistencias pull-up y pull-down, por defecto están inactivas.
 - Opción para operación open-drain.
-
- **Temporizador de propósito general (GPTM)** El temporizador de propósito general GPTM (siglas en inglés para general purpose timer module) es el módulo encargado de llevar

conteos o realizar eventos controlados por tiempo. Estos funcionan por medio del aumento o decremento en un registro de tamaño específico, de modo que, si es un contador ascendente, contará desde el valor indicado hasta que exista un desbordamiento. Si es un contador descendente este contará desde el valor hasta cero.

Modos de funcionamiento del temporizador

- Disparo único: el temporizador solamente realizará el conteo una vez, luego de esto dejará de funcionar.
- Periódico: el temporizador realizará su recorrido desde el valor especificado hasta el final, una vez este haya terminado se reiniciará el contador y comenzará el recorrido de nuevo.
- RTC (real time clock): en conjunto con un cristal de 32 768 Hz, este temporizador se utiliza para aplicaciones con un reloj de tiempo real. Esto es posible dado que 32768 es un valor de potencia 2. Por lo que se puede obtener un valor preciso de un segundo usando un contador de 15 bits.
- Contador de flancos: este modo funciona aumentando el valor del contador cada vez que ocurre uno de tres eventos: flancos de subida, flancos de bajada o ambos.

Lenguaje Informático

Un programa informático es una secuencia de instrucciones que se almacenan en algún lugar de la memoria con el objetivo de realizar una tarea específica

Si se pudiera observar un programa dentro de un sistema embebido, solamente se podría notar una larga sucesión de bits (código máquina) pues, a nivel de hardware, es lo único que comprende la unidad de procesamiento. Al escribir un programa, no se hace utilizando sucesiones de bits, puesto que esto implicaría demasiado tiempo y es muy propenso a errores. Es por esto que los primeros operadores de computadoras decidieron reemplazar estas cadenas de bits por palabras y letras de fácil entendimiento para el ser humano, lo cual dio origen al lenguaje ensamblador.

Al leer un archivo escrito en lenguaje ensamblador se sustituye cada una de las palabras y parámetros clave por código de operación correspondiente en sistema binario. Es por esto que recibe el nombre de lenguaje de bajo nivel o de nivel próximo a máquina. A finales de los años 50, surge un nuevo tipo de lenguajes de programación que tenían como objetivo que el usuario común pudiese escribir código de una manera más fácil y rápida. Estos lenguajes tienen una estructura semántica similar al lenguaje utilizado por los seres humanos por lo que reciben el nombre de lenguajes de alto nivel o de tercera generación. Entre sus ventajas se encuentran las siguientes:

- **Lenguaje C**

El lenguaje C, es un lenguaje de alto nivel que posee muchas características importantes: programación estructurada, métodos de llamadas a funciones, traspaso de parámetros, estructuras de control, entre otras.

Este lenguaje es de gran utilidad debido a su gran flexibilidad de programación; permite la habilidad de combinar comandos simples de bajo nivel en complicadas funciones de alto nivel. Provee acceso a la memoria de bajo nivel y construcciones del lenguaje que representan al código de la forma más cercana a código de máquina, a tal punto que algunos llegan a considerarlo actualmente un lenguaje de bajo nivel, motivo por el cual es el lenguaje más popular para escribir programas en microcontroladores.

La conversión de un lenguaje de alto nivel a uno de bajo y posteriormente a lenguaje de máquina, respectivamente, se hace por medio de un entorno de desarrollo integrado.

- **Entorno de desarrollo integrado (IDE)**

Un IDE es un entorno de programación que ha sido empaquetado como una aplicación. Está diseñado para maximizar la productividad del programador ofreciendo las herramientas para la creación, modificación, compilación, interpretación y depuración de código. Su objetivo es reducir el tiempo de desarrollo, pues aprender el uso de un IDE es más rápido que aprender e integrar manualmente todas las herramientas por separado. Para el controlador TM4C123GH6PM se cuenta con diferentes entornos de desarrollo integrado:

- Mentor embedded
- IAR systems
- ARM keil
- Code composer studio
- Energia

Este se encuentra compuesto por un compilador, un enlazador, un cargador de programa y en algunos casos por un módulo de depuración.

- **Compilador** Esta herramienta es la encargada de convertir el lenguaje de alto nivel a lenguaje de máquina. Sus tareas se dividen en tres fases diferentes:
 - The-front-end: encargada del análisis y generación del código para ser procesado por la fase middle-end.
 - The-middle-end: realiza la optimización del código recién generado. Esta fase consiste en mejorar este código de modo que se obtenga un código de máquina que se ejecute más rápido. Esto lo logra removiendo código inservible, descubriendo y propagando valores constantes, reubicación de procesos a lugares con menor ejecución. Genera otro código intermedio para ser procesado por la fase back-end.

- The-back-end: genera el código en lenguaje ensamblador, realiza la ubicación de los procesos en los registros y optimiza el código para el uso del hardware asociado.
- **Enlazador** Esta herramienta, también conocida como linker, es la encargada de tomar uno o más archivos de objetos generados por el compilador y combinarlos en un solo archivo ejecutable. Esto se hace debido a que un programa generalmente tendrá más de un solo archivo de objetos, como librerías, vectores de interrupciones u otros archivos externos declarados dentro del mismo código.
- **Cargador** de programa Esta herramienta, también conocida como loader, es la encargada de cargar a memoria el archivo ejecutable generado por el enlazador. También, es la encargada de inicializar los registros. Una vez el código ha pasado por todos estos pasos finalmente se tendrá un programa informático ejecutándose dentro del procesador.
- **Depurador** Este proceso, conocido también como debugging, es utilizado para probar y depurar (eliminar) errores en los programas. Funciona deteniendo la ejecución del programa a depurar cuando se cumplen ciertas condiciones que permiten examinar la situación al desarrollador. El depurador permite detener el programa en un punto determinado por una condición de ruptura. Durante esa interrupción, el usuario puede:
 - Examinar y modificar la memoria y las variables del programa.
 - Examinar el contenido de los registros del procesador.
 - Examinar la pila de llamadas que han desembocado en la situación actual.
 - Cambiar el punto de ejecución, de manera que el programa continúe su ejecución en un punto diferente a en el que fue detenido.
 - Ejecutar de modo instrucción a instrucción.
 - Ejecutar partes determinadas del código, como el interior de una función, o el resto de código antes de salir de una función.