

Periféricos Interfaces y Buses

I. Arquitectura de E/S

II. Programación de E/S

III. Interfaces de E/S de datos

IV. Dispositivos de E/S de datos

V. Buses

Buses de E/S (PCI, PC104, AGP). Sistemas de interconexión de periféricos en entornos industriales (bus I2C, bus CAN). Modelo de programación de dispositivos con los buses descritos.

VI. Controladores e interfaces de dispositivos de almacenamiento

VII. Sistemas de almacenamiento

Buses

Bloque III

- **El bus CAN**
- **El bus AGP**

Introducción al Bus CAN

El Bus CAN (Controller Area Network) es un protocolo de comunicaciones desarrollado por la firma alemana Robert Bosch GmbH para automoción:

- topología de bus
- transmisión serie de mensajes de hasta 8 bytes
- velocidad de transmisión de hasta 1 Mbit/s
- arbitrio del bus basado en prioridades (tiempo real)
- pertenece al grupo de buses de campo utilizados en control distribuido
 - control de sensores y actuadores distribuidos
 - uso de dispositivos inteligentes que disminuye el cableado
- se usa con protocolos de alto nivel

Características del bus CAN

Mensajes

- la información se envía en mensajes con un formato fijo
- puede tener tamaños de datos entre 0 y 8 bytes

Encaminamiento de la información (*routing*)

- no existe identificación de nodos, ni direcciones, lo que implica:
 - los nodos pueden entrar y salir del sistema sin que se requiera una nueva configuración
 - los mensajes se identifican por el valor de su prioridad, y aunque ésta no indica el destino, se pueden aplicar filtros en cada nodo para decidir que mensajes se aceptan
 - la información se puede enviar simultáneamente a varios nodos (*multicast*)
 - la información la reciben todos los nodos o ninguno (consistencia)

Características del bus CAN (cont.)



Velocidad de transmisión

- se puede elegir hasta el máximo de 1 Mbit/s
- debe ser uniforme y fija para todos los nodos conectados

Arbitrio

- se realiza por el identificador del mensaje que constituye su prioridad
- se utiliza durante el acceso al bus
- cualquier nudo puede empezar a transmitir en cualquier momento si el bus está libre (*multimaster*)

Características del bus CAN



Seguridad

- detección de errores
 - monitorización: el transmisor compara el valor del bit transmitido con el que encuentra en el bus
 - chequeo de redundancia cíclica (CRC)
 - *bit stuffing* (se añaden unos o ceros extra)
 - chequeo de marco de mensaje
- la probabilidad de no detectar un error es del orden de 10^{-11}

El canal físico normalmente consiste es un par de cables diferenciales, pero puede ser también una fibra óptica

Características del bus CAN

Valores del bus

- el bus puede tomar dos valores lógicos complementarios:
 - dominante o recesivo
- el caso de transmisión simultánea impera el dominante
- por ejemplo, si se implementa como una AND cableada el dominante es el 0 y el recesivo el 1
- no se dan valores físicos en la especificación

Arbitrio del bus CAN

Cuando el bus está libre cualquier nodo puede comenzar a transmitir un mensaje

Si dos o más nudos comienzan la transmisión a la vez, el conflicto se resuelve bit a bit por el identificador del mensaje (su prioridad):

- los bits del identificador se transmiten uno a uno desde el más significativo
- cada transmisor compara el bit que ha puesto con el que hay en el bus
 - si es el mismo continúa la transmisión del siguiente
 - si ha puesto un recesivo y lee un dominante se retira hasta que el bus quede libre de nuevo

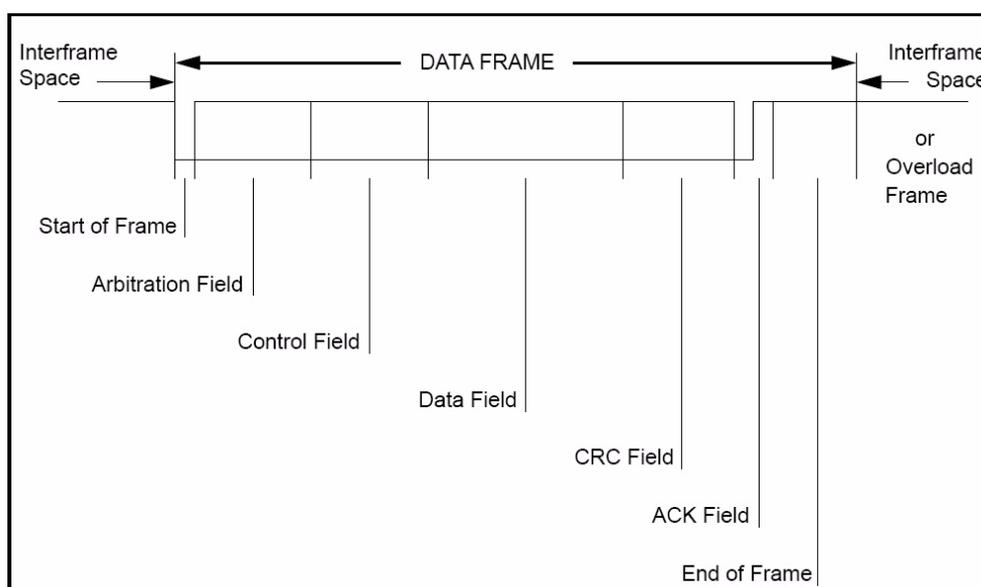
Arbitrio del bus CAN (cont.)

No debe haber dos mensajes con el mismo identificador

Hay dos especificaciones de CAN:

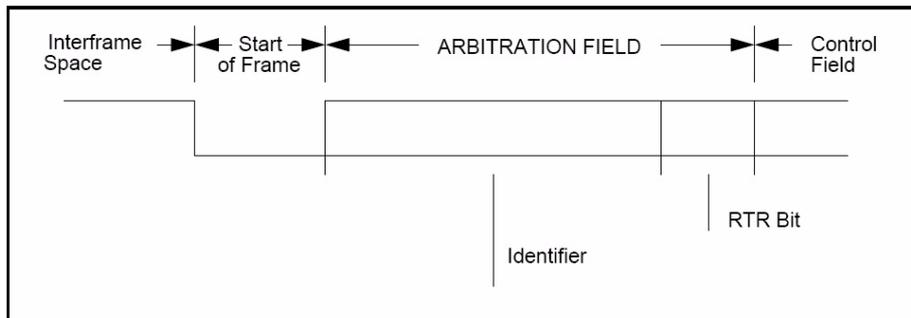
- CAN 2.0 A: identificador de 11 bits
- CAN 2.0 B: identificador de 29 bits

Formato de mensaje CAN



[3]

Campo de arbitrio



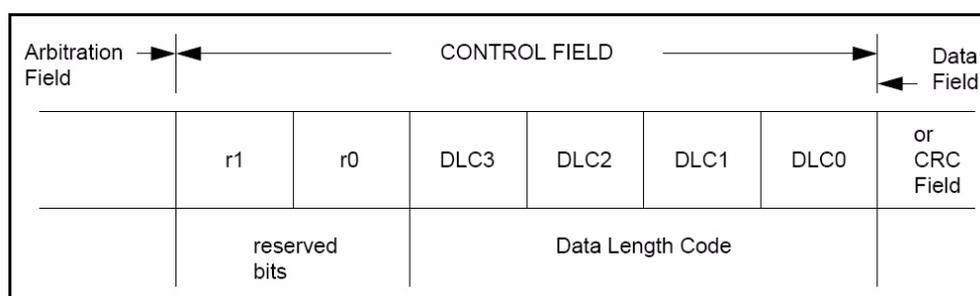
[3]

RTR: bit de requerimiento de transmisión remota

Prioridad: los bits se transmiten desde el más significativo

- para el identificador de 11 bits los 7 más significativos no pueden ser todos recesivos

Campo de control

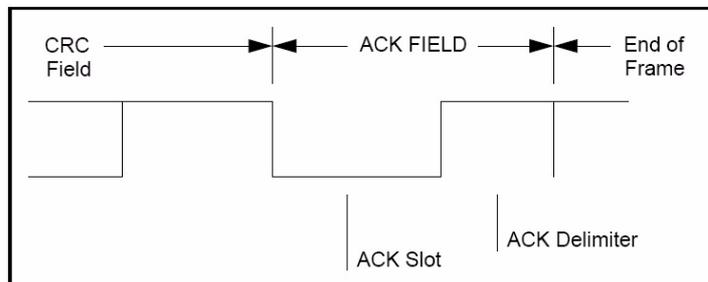


[3]

El campo de control tiene 6 bits

- los bits DLC portan la longitud del mensaje en bytes con valores de 0 a 8

Campos de CRC, ACK y fin



[3]

El CRC es una secuencia de 15 bits con el código calculado

El final del marco del mensaje consta de 7 bits recessivos consecutivos

Campos de CRC, ACK y fin

El campo de reconocimiento consta de dos bits que el transmisor los pone como recessivos:

- **ACK SLOT:** si el receptor recibe el mensaje correctamente lo ponen como dominante
- **ACK Delimiter:** se deja recessivo

Comparación con otros buses

	CAN	LIN	I2C
Compañía que lo desarrolló	Bosh	Open source	Philips
Velocidad	1Mb/s	20 Kb/s	0.1Mb/s / 0.4Mb/s
Tamaño de datos	64bits	8bits	8bits
Prioridad de mensajes	Si	No	No
Garantía de latencia	Si	***	No
Flexibilidad en la configuración	Si	***	Si
Sistema Multimaestro	Si	No	Si
Detección y señalización de errores	Si	Si	Si
Retransmisión de tramas	automática	No	programable

[5]

Protocolos de alto nivel sobre CAN

Se basan en el uso del identificador con un fin especial:

- **CANKingdom**
 - define primitivas de protocolo basadas en CAN
 - se define un master del bus en la inicialización que chequea los nodos que hay conectados a la red
 - soporta modelos gobernados por eventos y por tiempo
- **OSEK-COM**
 - principalmente usado en automoción
- **SDS (*Smart Distributed Systems*)**
 - define el nivel físico y el de aplicación
 - usado por sensores y actuadores inteligentes sobre un cable de 4 hilos
 - admite hasta 64 nodos y 126 direcciones

- **DeviceNet**
 - implementa otro protocolo (*Common Industrial Protocol*) sobre CAN
- **CANOpen**
 - implementación de objetos basados en los identificadores de mensaje
 - permite controlar dispositivos e intercambiar datos a través de los diferentes tipos de objetos que define
- **TTCAN (*Time Triggered CAN*)**
 - permite el uso determinista del bus CAN

Cada protocolo de alto nivel requiere dispositivos especiales para que los puedan controlar

El bus CAN y el tiempo real

El bus CAN es una red capaz de transmitir mensajes en tiempos acotados

- al basarse en prioridades fijas se puede aplicar el análisis de planificabilidad similar a los procesadores para obtener cotas de los tiempos de transmisión de los mensajes que genera una aplicación
- se pueden aplicar técnicas de análisis de sistemas distribuidos
- para que esto sea posible los protocolos de alto nivel que operan sobre el bus tienen que haber sido diseñados para hacer un uso adecuado de las prioridades

El bus CAN y el tiempo real (cont.)

Por ejemplo, un protocolo de nivel de aplicación podría utilizar el identificador de mensaje como sigue:

		Identifier										
Bit Number		11	10	9	8	7	6	5	4	3	2	1
Field		Message Priority				Destination Node			Communication Channel			

- **Prioridad:**
 - urgencia del mensaje
 - independiente de los nodos y la funcionalidad
 - mapea las prioridades
- **Nodo destino:**
 - al menos debe haber un nodo por cada controlador de CAN en el sistema
 - utiliza el filtrado de mensajes

El bus CAN y el tiempo real (cont.)

- **Canal de comunicaciones:**
 - identifica la tarea o thread destinatario
 - la tarea elige el canal del que quiere leer, o en el que espera la recepción de mensajes
 - papel similar al puerto de comunicaciones

Con el identificador de 29 bits las posibilidades de codificación aumentan considerablemente

Para comunicaciones se debe usar un mecanismo de particionamiento de mensajes mayores de 8 bytes en paquetes

Bloque III

- El bus CAN
- El bus AGP

Introducción al bus AGP

El bus AGP (*Accelerated Graphics Port*) se presenta como un único puerto destinado a tarjetas gráficas

- tiene 32 bits
- soporta velocidades de 66 (AGPx1), 133 (AGPx2), 266 (AGPx4), ó 533 (AGPx8), en MHz
- permite anchos de banda de hasta 2133 MBps
- se conecta al *North Bridge* o a un *Hub* controlador de la memoria al mismo nivel que el bus PCI
- las tarjetas que se conectan siempre requieren un canal de interrupción (IRQ)
- en los sistemas más modernos está en desuso en favor del PCI-Express

Bibliografía

- [1] **H.P. Messmer, "The Indispensable PC Hardware Book", 4th Ed., Addison-Wesley, 2002**
- [2] **Scott Mueller, "Upgrading and Repairing PCs", 17th Ed., QUE, 2006**
- [3] **Bosch, "CAN Specification", 1991.**
- [4] **Aplicaciones del bus CAN**
http://www.s3.kth.se/~kallej/papers/can_necs_handbook05.pdf
- [5] **<http://server-die.alc.upv.es/asignaturas/PAEEES/2005-06/A03-A04%20-%20Bus%20CAN.pdf>**
- [6] **http://en.wikipedia.org/wiki/Accelerated_Graphics_Port**