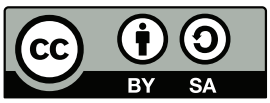
M **23-001**

C-Bus

Interface Fisica C-BUS en DccDiyTools

Daniel Vilas

(Draft v0.1)



1 Introduccion

C-Bus es un standard LCB usado y promocionado por MERG©. A bajo nivel utiliza un Bus CAN como transporte fisico de datos entre modulos (electronicos).

La idea de despligue es usar una topolgia de bus:

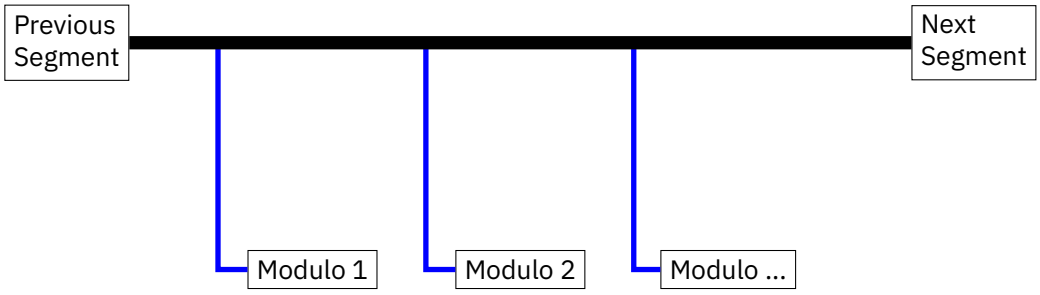


Figura 1: C-Bus Segmento

Al final de un segmento puede haber otro segmento, un repetidor, un convertidor a Ethernet,.... Desde el bus a los dispositivos es necesario tener un latiguillo.

2 Especificacion MERG

3 Topologia

CBus[1] ha sido pensado para seguir lo más fielmente posible una topologia de Bus y en los extremos una resistencia de 120 Ω.

[1] Usando CAN

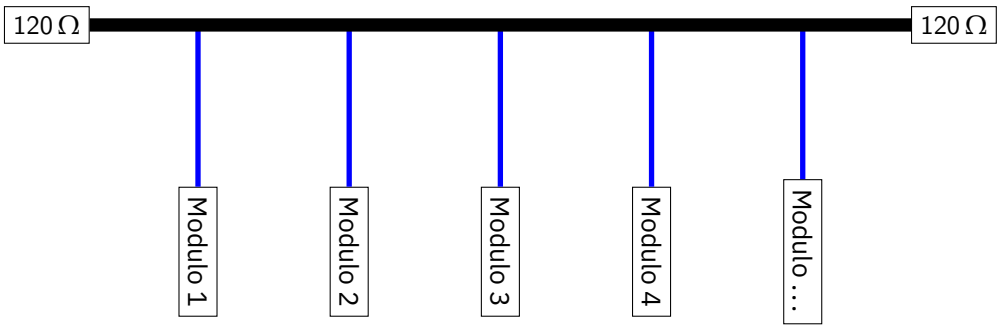


Figura 2: Topologia Bus

Pero a su vez permite algunas ramificaciones siempre y cuando la resitencia entre los conductores diferenciales sea aproximadamente 60 Ω. Ya que todos los modulos estaran conectados en paralelo a los 4 conductores.

Ademas CBUS, en general, permite utilizar otros sistemas de transporte de la informacion como puede ser TCP, UDP, MQTT,... Pudiendo haber multiples Buses y segmentos unidos formando una red mayor.

3.1 Red

Recordemos que el objetivo de CBUS es poder controlar una maqueta de tren de forma digital. Es decir desde un panel de mandos enviar señales para que suceda un cambio en la maqueta, o al reves, o ante un evento en la maqueta [2] se actualice nuestro panel de control.

[2] Cortos, ocupacion por un tren,...

El panel de control puede ser fisico, (con lucecitas y botones) o bien puede ser una pantalla de un programa como JMRI [3]

[3] JMRI o similar, usaremos JMRI por preferencias del autor

El caso minimo de uso CBus tendra un bus CAN con unos pocos modulos, partiendo el cable donde sea necesario y como mucho un CanUSB o CBusServer para conectar con JMRI. Vease 2. El caso más complejo requeria de uno o varios CBusServer cada uno conectado a uno o varios Buses CAN. El/los CBusServer/es se comunicarian mediante TCP[4] entre si y distribuyendo asi los eventos entre todos los buses CAN.

[4] Wifi, Ethernet,...

Este ultimo caso lo podemos representar con dos buses CAN donde "Modulo 2"seria el CBus-Server y JMRI se conecta a traves de Wifi o un CanUSB.

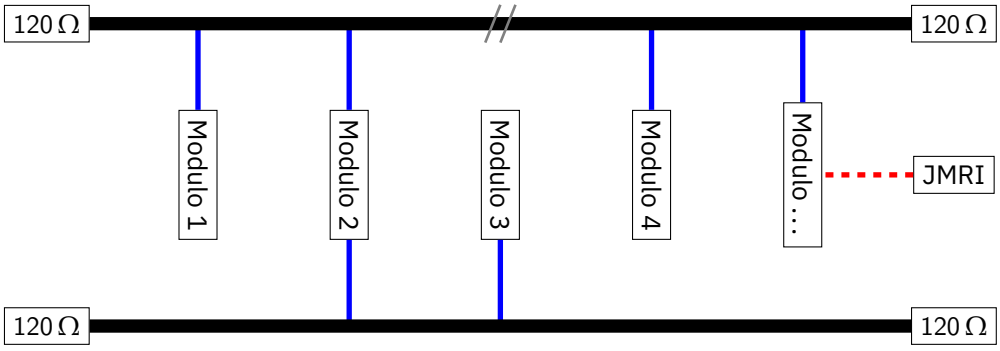


Figura 3: Topologia Bus Ejemplo Red

La red CBUS es pues el conjunto de todos los dispositivos CBUS que se pueden comunicar entre si usando el protocolo CBUS

3.2 Segmento vs Bus vs Red vs CBusServer

Sirva este apartado para aclarar y definir estos terminos. Puesto que esto son muy parecidos a una instalacion TCP/IP.

3.2.1 Bus

Como ya se tiene una idea de lo que es la Red, empezaremos con el Bus. Siendo este el canal que permite a todos los dispositivos conectados fisicamente al mismo comunicarse con una misma tecnologia. Hoy por Hoy solo hay dos disponibles [5]

[5] segun este punto de vista

- **CAN Bus**: Utilizar un par diferencial CAN como comunicacion.
- **CBusServer**: Los dispositivos se conectan mediante una conexion TCP.

De cada bus puede haber varias instancias, que no tienen que comunicarse de por si.

En resumen un Bus es un canal por lo el cual varios dispositivos se comunican entre si, sin tener que cambiar de medio.

Los buses pueden unirse o partirse, lo que hablaremos de redes o de segmentos. Si para unirlos es necesario una logica de transformacion de datos o de medio fisico, estaremos ante una red.

3.2.2 Segmento

Cuando un bus lo podamos cortar en varios trozos fisicos hablamos de segementos. En el caso de un Bus Can, al final son dispositivos conectados entre si mediante un par de cables (H y L)[6] . Por lo que estos cables se pueden cortar en segmentos. Por otra parte CBusServer puede ejecutarse en varias maquinas (cada una una instancia) y configurase para que se hablen entre si, pero al final los clientes se conectaran a una sola maquina.

[6] y como mucho Vcc y GND

En resumen un segemento es cada una de las partes en las que un bus se puede partir sin perder la caracteristica de bus.

3.2.3 Red

Cuando nos encontremos con varios buses y tengamos que comunicarse entre ambos, nos encontramos con el concepto de Red. Este concepto aparece cuando al unir dos segmentos aparece alguna de estas casuisticas:

- **Cambio de Medio**: Por ejemplo al pasar de CAN a TCP. Esta es una razon un poco debil, por que por ejemplo un repetidor CAN"segun como este echo puede cambiar de medio momentanemente, o lo mismo un firewall (que no deje pasar tramas no CBUS, como OpenLCB). Y segun lo estrictos que seamos puede ser dos buses o no.
- **Transformacion de datos**: Si los eventos deben ser transformados (cambiar ids, agrupar,...)
- **Diferentes Funciones**: Para cada bus hemos definido una funcion diferente como puede ser eventos locales a un modulo o globales a toda la maqueta. O un bus para traccion y otro para accesorios.

En la implementacion Merg, no existe este concepto, puesto que no lo han necesitado. Tener una Red, implica tener varios buses, con la consecuencia de tener que gestionar la red. Saber para que es cada uno de los buses, asignar identificadores,... Pero nos abre la puerta a diseños más complejos y optimos.

3.2.4 CBusServer

CBusServer se corresponde con un protocolo de aplicacion sobre TCP/IP para enviar tramas CBus entre aplicaciones informaticas y como esta vision forma una tecnologia de Bus.

Pero a su vez es un Sofware de Servidor donde las diferentes aplicaciones clientes se conectan con TCP y se encarga de enviar los paquetes CBus que recibe de un cliente al resto. Este sofware puede usar el Modulo CanUSB4 de Merg y/o el Modulo CanPiHat para conectarse a una Red CBus directamente.

Este ultimo caso lo podemos representar con dos buses CAN donde "Modulo 2"seria el CBusServer y JMRI se conecta a traves de Wifi o un CanUSB.

3.3 Ejemplo Practico 1 - Dos Habitaciones

Imaginemosnos una maqueta dividida en dos grandes zonas o habitaciones y con un ordenador central para mostar el estado general de la maqueta.

Como todo en la vida hay multiples soluciones, que implican desde un bus hasta una red de 4 o más buses.

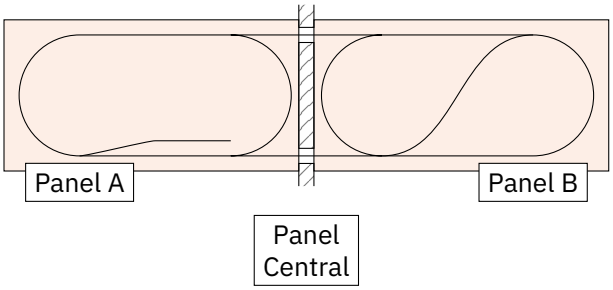


Figura 4: Ejemplo Practico

Por simplificar hemos dibujado dos ovalos, uno en cada habitacion, con dos conexiones. Simplemente representan una maqueta dividida en dos habitaciones.

Al final cada habitacion tiene sus desvios, sus detectores de ocupacion, escenografia, accesorios,... En la practica son dos maquetas que comparten unos puntos de conexion. Cada habitacion tendra su panel de control y solo estara interesada en sus propios eventos. Como mucho le interesara saber si puede enviar o recibir un tren por cada una de las conexiones.

3.3.1 Solucion de un solo bus

Una primera solucion podria ser tener todos los accesorios conectados a un solo bus, tanto de una zona como de la otra.

En esta solucion depende que en conjunto no se supere la longitud maxima de cableado y de dispostivos conectados [7] . Pero aun en el muy probale caso de que no, existe otra limitacion, que es el numero de eventos y la seguridad que los eventos de una zona son ignorados por los dispositivos de la otra.[8]

[7] Muchos metros y decenas de dispositivos

[8] Como mucho interesa algun error para eviar enviar trenes de una zona a la otra

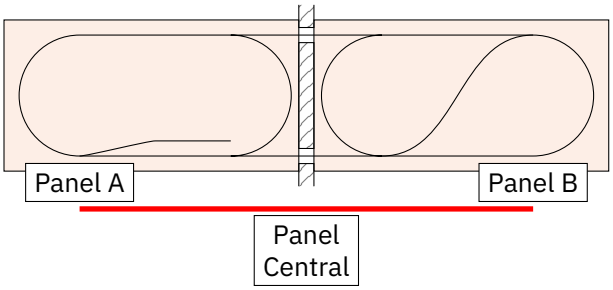


Figura 5: Ejemplo Practico - Solucion de un Bus

Para Pensar: Si usamos un CBusServer para comunicar las zonas, seguimos en un bus o son tres...

3.3.2 Dos buses

Como hemos visto que hay dos zonas podemos tener cada una con su bus propio y un ordenador conectado a ambas redes. Este ordenador procesaria los eventos para mostrar en su pantalla y como mucho pasaria eventos de error en la otra zona para evitar enviar trenes de una a otra.

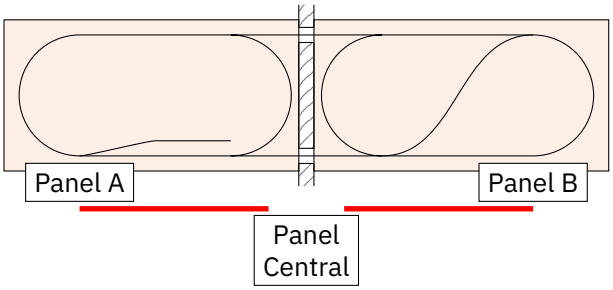


Figura 6: Ejemplo Practico - Solucion de dos Buses

Esta solucion obliga a que el PC con el Panel Central este siempre encendido y ejecutando una logica para enviar mensajes de ocupacion/disponibildad de los tramos de interconexion.

4 Cables

5 Conectores

5.1 Segmento - Segmento

5.2 Placa - Latiguillo

Para conectar las placas al bus CBUS tenemos varias opciones validas, y las usaremos segun nos sea util

5.2.1 Poca Potencia/MERG Screw Terminal Block

Para las placas de poca potencia (consumo del bus <100 mA, o con su propia fuente) podemos usar la solucion de MERG con un bloque terminal tipo MKDS - PHOENIX CONTACT.

El cable a utilizar es el de latiguillo (0.25 mm^2/22AWG) usando una ferrula adecuada (Codigo-color).

Nota: Sale mas barato comprar 2 de 1x02 que 1 de 1x04

Tabla de refencias

5.3 Media Potencia/MERG Plug Terminal Block

Cuando la placa requiera mas potencia (Consumo del bus <500 mA, ej: esp32 usando Wifi activamente) podremos usar una version plug 3,5mm o 5mm

Si es posible utilizar el cable de latiguillo (0.25 mm^2/22AWG) usando una ferrula adecuada (Codigo), pero si no se puede usar el del bus general (0.5 mm^2/20AWG) con su ferrula adecuada (codigo-color).

Tabla de refencias

5.3.1 Alta Potencia

Finalmente si la placa requiere mucha potencia, como puede ser un distribuidor de CBUS, o un motor alimentado del bus. Se podra usar la version PCB del conector Cable-Cable o soldar directamente el cable a la PCB