

# $\mu$ Cmp: MicroController multipoint protocol

¿Qué es?

# $\mu$ Cmp: MicroController multipoint protocol

- Un pequeño subsistema, que se incluye en nuestras aplicaciones
- Comunica múltiples nodos
- Todos con un *bus* en común
- En un principio pensado para RS485

# $\mu$ Cmp: MicroController multipoint protocol

- Detalles:

- Permite la interacción de  $2^{24}$  nodos

- Estructura de trama:

[ ] <- Opcional

- STX (0x2)
    - HDB2
    - HDB1
    - [DST]
    - [SRC]
    - [DATA]
    - [CRC]
    - EOT (0x4)

# $\mu$ Cmp: MicroController multipoint protocol

- Implementación de uCmp en AVR-Atmega168, el primer port.

3 partes:

Kepler – HAL – uCmp (Independiente)

# $\mu$ Cmp: MicroController multipoint protocol

## • Kepler:

Antes, sin Kepler:

- AVR-ATmega168 no podía mantener activo el generador de ticks y procesar una trama al mismo tiempo.

Después, con Kepler:

- Interrupción solo recolecta datos.
- $\mu$ Cmp crea una tarea que revisa si existe una trama en la cola por procesar.
- Kepler hace mas fácil la programación de aplicaciones

# $\mu$ Cmp: MicroController multipoint protocol

- **HAL: Hardware Abstraction Layer**
  - uCmp no tiene idea como se envía un octeto por serial.
    - Esto permite que uCmp no sea un código dependiente de una arquitectura
    - Portar uCmp a otra arquitectura de uC es mas fácil y rápido.

# $\mu$ Cmp: MicroController multipoint protocol

- uCmp:

- Recolecta una trama hasta completarla
- Valida la trama: Estructura básica, dirección de destino y Integridad (CRC)
- Si todo esta correcto, llama a la función asignada por el programador.

# $\mu$ Cmp: MicroController multipoint protocol

- **uCmpUS:** uCmp Unix Service
  - Permite a las plataformas UNIX (POSIX compatible) interactuar fácilmente con una red de nodos.
  - librería:
    - `ucmpUS_connect()`
    - `ucmpUS_create_vnode()`
    - `ucmpUS_send()`
    - `ucmpUS_next_msg()`

!!!

- Utilizando  $\mu$ Cmp:
  - Compilar una aplicación
  - Subirla
  - Arrancar uCmpUS
  - Crear un **Virtual Node** con uCmpUS

Gracias por asistir