



**ITESO, Universidad  
Jesuita de Guadalajara**

**Redes para Sistemas Embebidos (Alámbrico)**

**O2023\_ESI1180**

**Jorge Antonio Alcalá Vázquez**

**Práctica 2 – CAN**



**Nelida Paulina Hernández Moya**

**ie727824**

**Sergio Eduardo Borrayo Anzaldo**

**ie726673**

**María Luisa Ortiz Carstensen**

**ie717781**

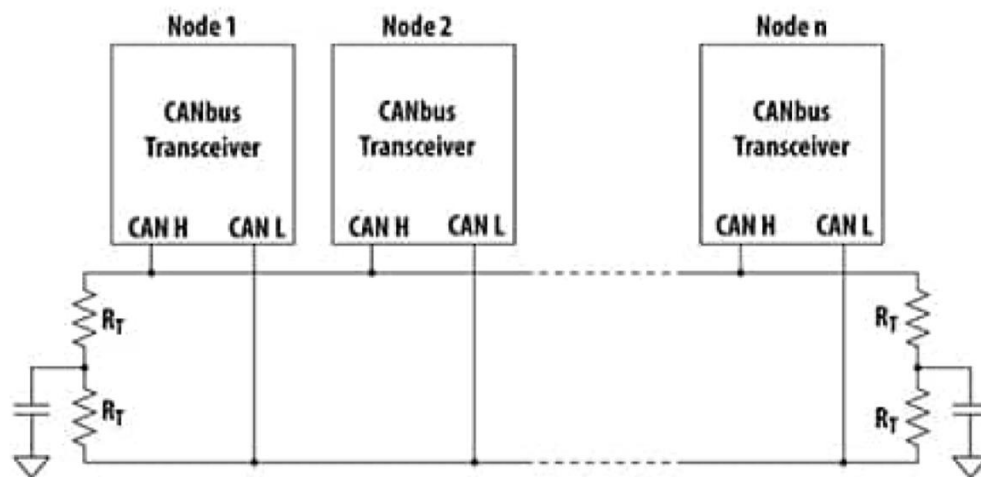
**Guillermo Alexis García Ramírez**

**ie715902**

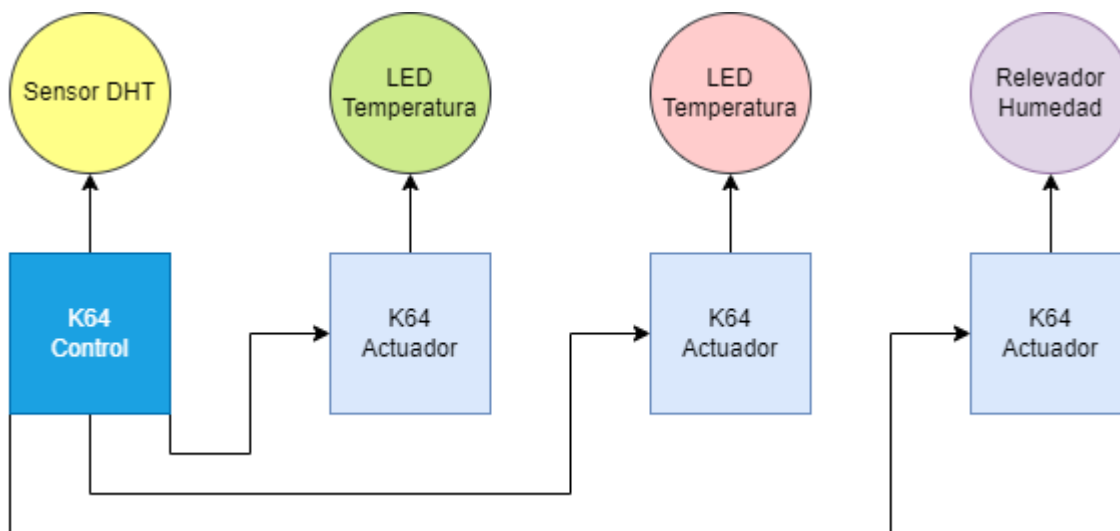
## Enunciado de la práctica

Implementar un bus de comunicación CAN donde una de las tarjetas K64F represente el “módulo de control” que mande diferentes mensajes a las otras tarjetas K64F y estas activen algún modulo que muestre que recibió el mensaje por parte de la tarjeta de control K64F.

Arquitectura: utilizando un bus de CAN, conectar al menos 4 tarjetas K64F. Una de las tarjetas funcionará como módulo de control principal y las otras 3 tarjetas realizarán diferentes tareas que recibirán por parte de la tarjeta de módulo de control.



## API de la nueva capa

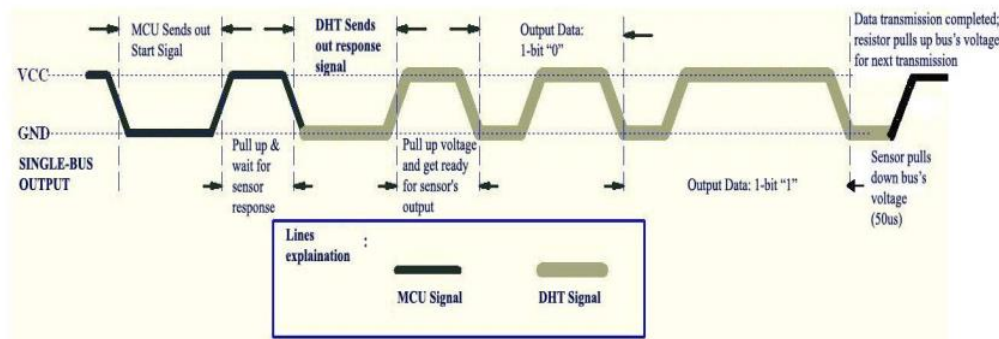


Siendo la primera de las tarjetas la que funciona como el control de las demás y siendo quien recopila datos del sensor de temperatura/humedad, desde aquí se hace todo el “arbitraje” hacia las demás tarjetas, quienes funcionan como actuadores, lo quisimos ver como el sensor de temperatura del carro, en un rango adecuado mantenemos el LED verde prendido, mientras que cuando se pasa del rango, el LED rojo se enciende. Además, el sensor de humedad a partir de cierto valor, lo consideraremos cuando empieza a llover y el relevador que enciende el funcionamiento de los parabrisas se activa.

## Cambios en los proyectos originales

Tomando como referencia el ejemplo del SDK de CAN loopback transfer, cambiamos el código fuente para la transmisión y recepción de los datos, realizando así la comunicación en las tarjetas para que pudieran funcionar como un bus.

El sensor DHT11 funciona mediante una resistencia de pull-up y una serie de señales mandadas desde el microcontrolador y esperar a que el sensor responda con otra trama de señal. Para obtener los bits esperamos a que el DHT mande una señal en bajo y luego en alto, dependiendo del tiempo que se mantenga en alto sabremos si el bit es 1 o 0.



## Problemas a los que nos enfrentamos

- El mayor reto, consistió en leer el sensor DHT, ya que se requieren saber los tiempos de inicio del sensor para poder tomar los datos adecuadamente.
- Adicionalmente, problemas básicos de configuración de los pines que no estábamos configurando adecuadamente.
- Y para la conexión de las tarjetas, batallamos un poco para que la comunicación entre las 4 tarjetas fuera exitosa, pero al final, cambiando bien los IDs logramos que fuera exitosa.

## Conclusiones

### Nelida:

- Durante el desarrollo de esta práctica aprendí bastante respecto al uso de CAN y el tema de los IDs, así como de la transmisión y la recepción de los mensajes, realmente fue divertido tener el bus y las tarjetas conectadas entre sí, para obtener datos del sensor y poderlos mostrar en las tarjetas en los diferentes actuadores; tuvimos algunos errores sobre la marcha, más enfocados hacia la incorrecta desconexión de algunos pines, o del hardware en general, pero una vez detectando esos errores, pudimos hacer funcionar el sistema de forma correcta, viendo en la terminal los mensajes enviados y como se manipularían por los actuadores.

### Sergio:

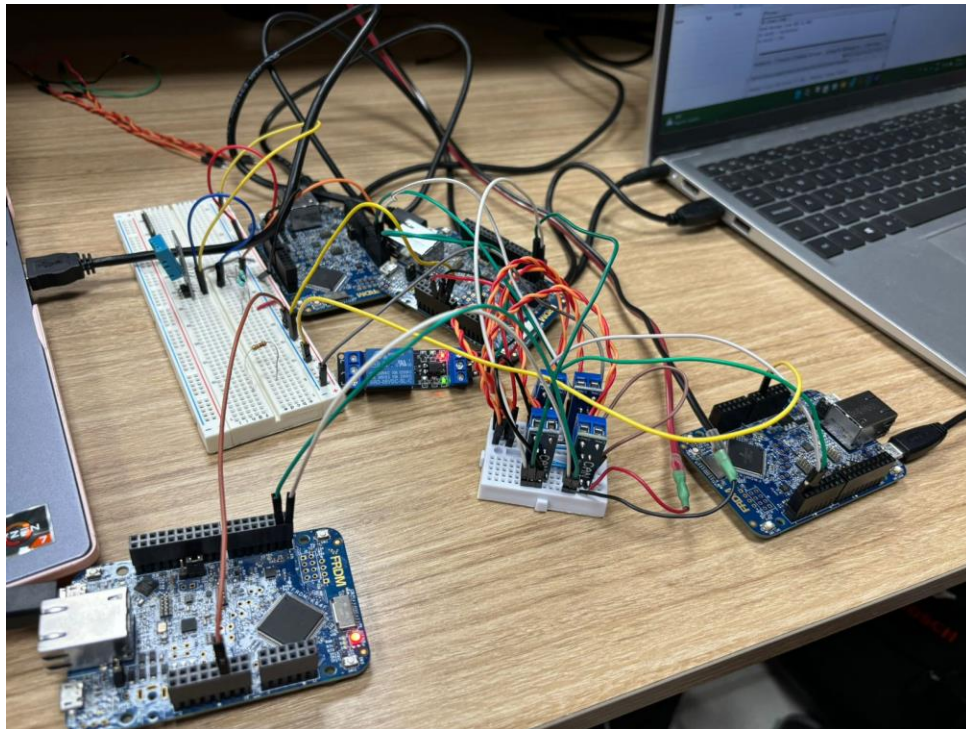
- Con esta práctica mi parte fue hacer trabajar el sensor, tuve algunas complicaciones con los cálculos de los tiempos para obtener los bits al igual que con las funciones del SDK para leer y escribir, al final descubrí que para leer se manda el número del pin y para escribir con una máscara. En cuanto a la comunicación de can hubo algunas complicaciones comunicando más de una tarjeta por algunos problemas de hardware que fuimos solucionando y los de software se fueron debuggeando, en mi caso aprendí que solamente cambiando el ID se podía mandar con la misma variable de transfer. Al final el resultado fue el esperado, los datos se mandaban perfectamente y los actuadores actuaban cuando se debía.

**María Luisa:**

- Esta práctica me ayudó a entender de mejor manera la comunicación utilizando CAN, ya que para comunicar cuatro tarjetas se necesitó un mejor control de conceptos, tanto de software, como de hardware. También, ayudó a refrescar conceptos básicos de configuraciones en embebidos para lograr utilizar las tarjetas como actuadores. La única parte complicada fue la parte física, ya que nos confundimos repetidamente por no revisar correctamente, el código en sí no fue complicado, únicamente la configuración del sensor.

**Alexis:**

- Con esta práctica logramos comunicar un master con cuatro tarjetas con actuadores, hubiera sido posible comunicar una de las k64 con el actuador de regreso con la k64 de control, ya que todas las K64 estaban seteadas tanto para recibir como para enviar, solo hay que tener cuidado de que la lógica del código sea la correcta. Me ayudó a recordar como configurar los distintos puertos y entender como funcionaría la conexión en un bus más grande de CAN.

**Imagen de la práctica****Video de la práctica funcionando**

[Mi película.MOV \(sharepoint.com\)](#)

**Repositorio con el código**

[https://github.com/hanilizalo23/RSE\\_O2023\\_CAN.git](https://github.com/hanilizalo23/RSE_O2023_CAN.git)