Programa Oficial de Postgrado en Ciencias, Tecnología y Computación
Máster en Computación
Facultad de Ciencias – Universidad de Cantabria

Manejadores de buses serie en MaRTE OS

Autor:
Mónica Puig-Pey González
Directores:
Mario Aldea Rivas y
Michael González Harbour

Grupo de Computadores y Tiempo Real
Departamento de Electrónica y Computadores
Universidad de Cantabria

Introducción

- Sistema de Tiempo Real
  - Aquél en el que lo importante no es SÓLO que el trabajo se realice, sino que éste se realice a TIEMPO

- MaRTE OS
  - Sistema de tiempo real que cumple con el estándar POSIX
  - Perfil “Sistema de Tiempo Real Mínimo”
  - Uso en aplicaciones empotradas de tiempo real y docencia
  - Permite ejecutar aplicaciones escritas en lenguajes Ada y C

- Manejadores de dispositivo
  - Aumentan posibilidades del sistema de interactuar con el exterior

Introducción

- Los buses serie
  - Información transmitida bit a bit

Comunicación
- *Simplex*
- *Duplex*
- *Full Duplex*

Ejemplos
- Puerto Serie
- USB
- I2C
- SPI
- SMBUS

Objetivos

- Motivación

Participar en el proyecto:

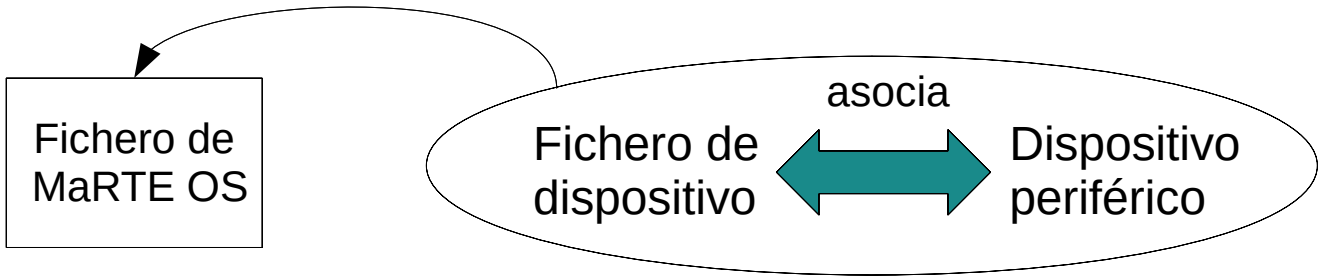
"Sistema de Cálculo de Orientación basado en GPSs no dedicados y apoyado por Sensores Inerciales MEMs de bajo Coste"

- Manejador en MaRTE OS para una tarjeta PC/104 que contiene 8 puertos serie.
- Estudio del bus serie SPI (*Serial Peripherical Interface*).
- Subsistema SPI en MaRTE OS.
- Manejador en MaRTE OS para un sensor inercial de tres ejes con comunicación SPI.

Índice

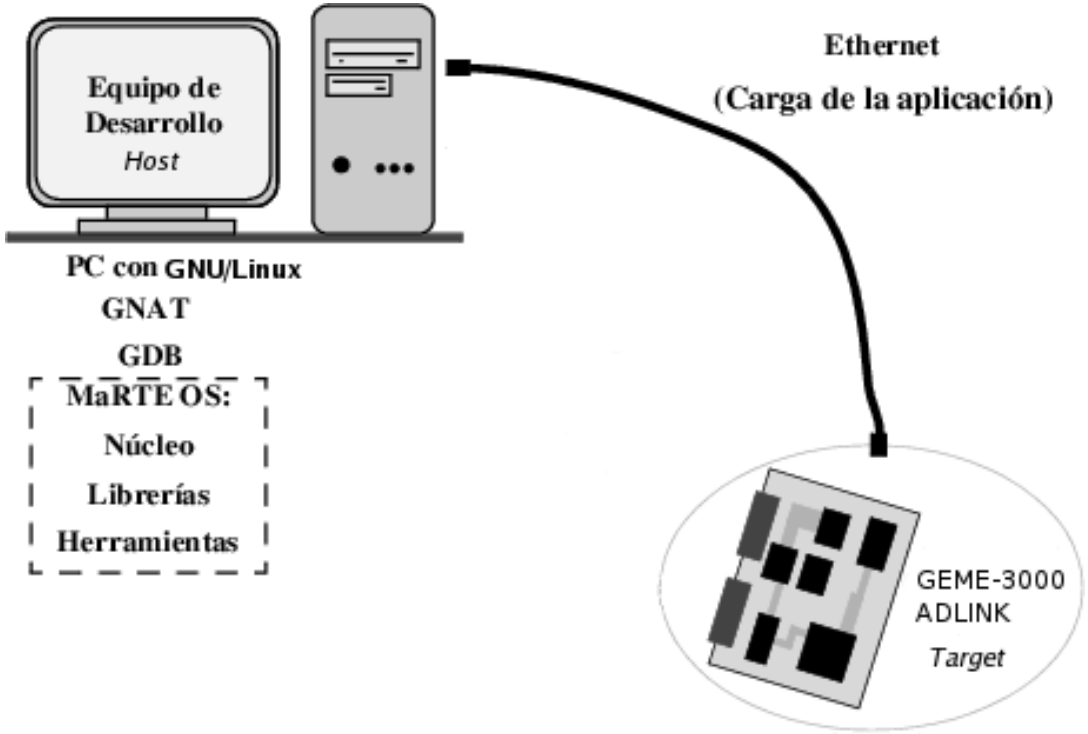
- Introducción.
- Objetivos.
- ***Drivers* en MaRTE OS.**
- Manejador para tarjeta de 8 puertos serie.
- Subsistema de gestión del bus serie SPI.
- Manejador para un dispositivo real SPI.
- Conclusiones y trabajo futuro.

Drivers en MaRTE OS

- Un manejador o *driver* es un módulo *software* encargado de controlar un dispositivo periférico facilitando al usuario su manejo.
- Oculta al usuario registros internos del dispositivo:
  - inb_p outb_p
- Interfaz POSIX de entrada/salida basada en ficheros.
  - create, open, write, read, ioctl...
  - Ejemplo de uso: fd = open("/dev/driver_c", O_RDONLY);
    read(fd, &buffer, sizeof(buffer));
- Instalación:
  - 

  - Se recompila MaRTE

Entorno de trabajo



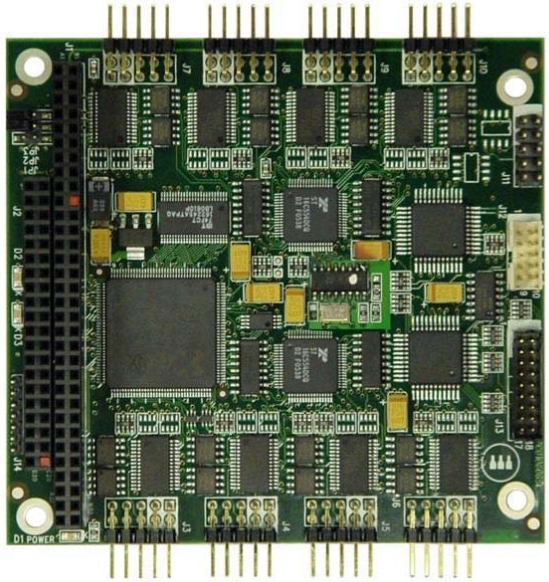
- Computador de ejecución GEME-300 ADLINK:
  - CPU: Low Voltage Pentium III, 800 Mhz
  - Puertos serie COM1 y COM2, UART 16650
  - Interfaz PC/104

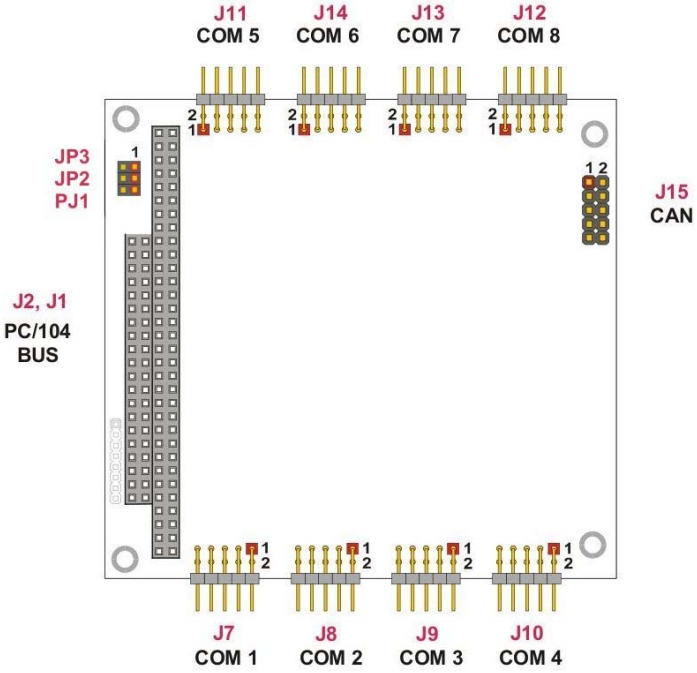
Índice

- Introducción.
- Objetivos.
- *Drivers* en MaRTE OS.
- **Manejador para tarjeta de 8 puertos serie.**
- Subsistema de gestión del bus serie SPI.
- Manejador para un dispositivo real SPI.
- Conclusiones y trabajo futuro.

Manejador para tarjeta 8 puertos serie

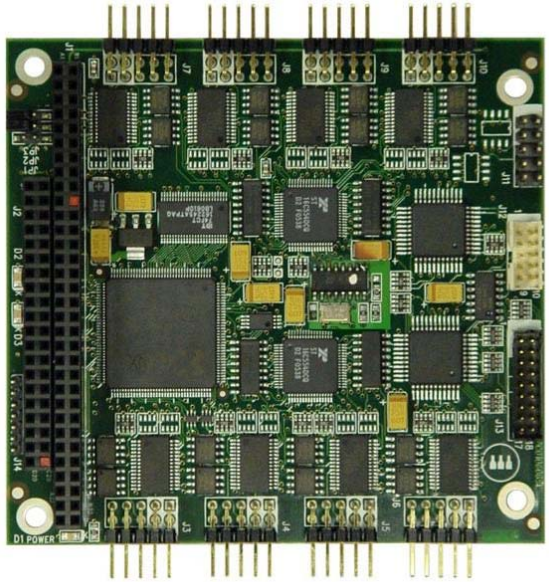
- Tarjeta COM-1274-A1 de Parvus

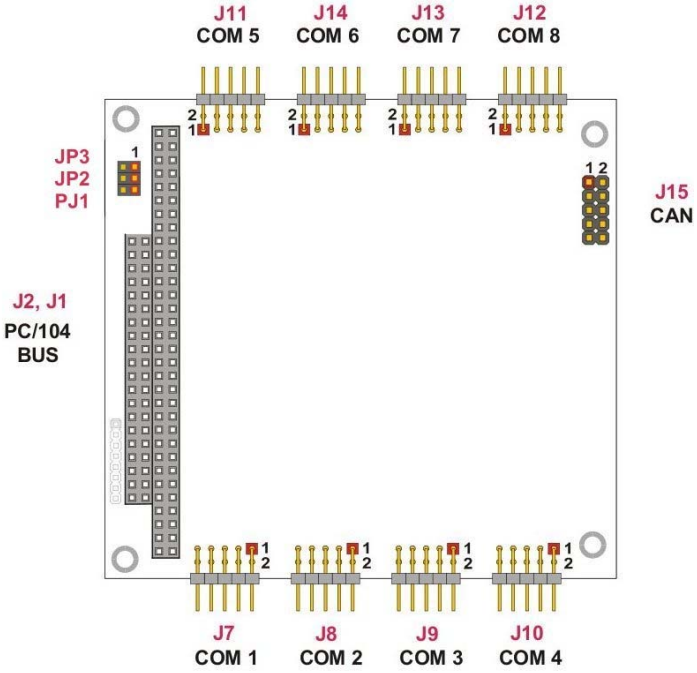




Manejador para tarjeta 8 puertos serie

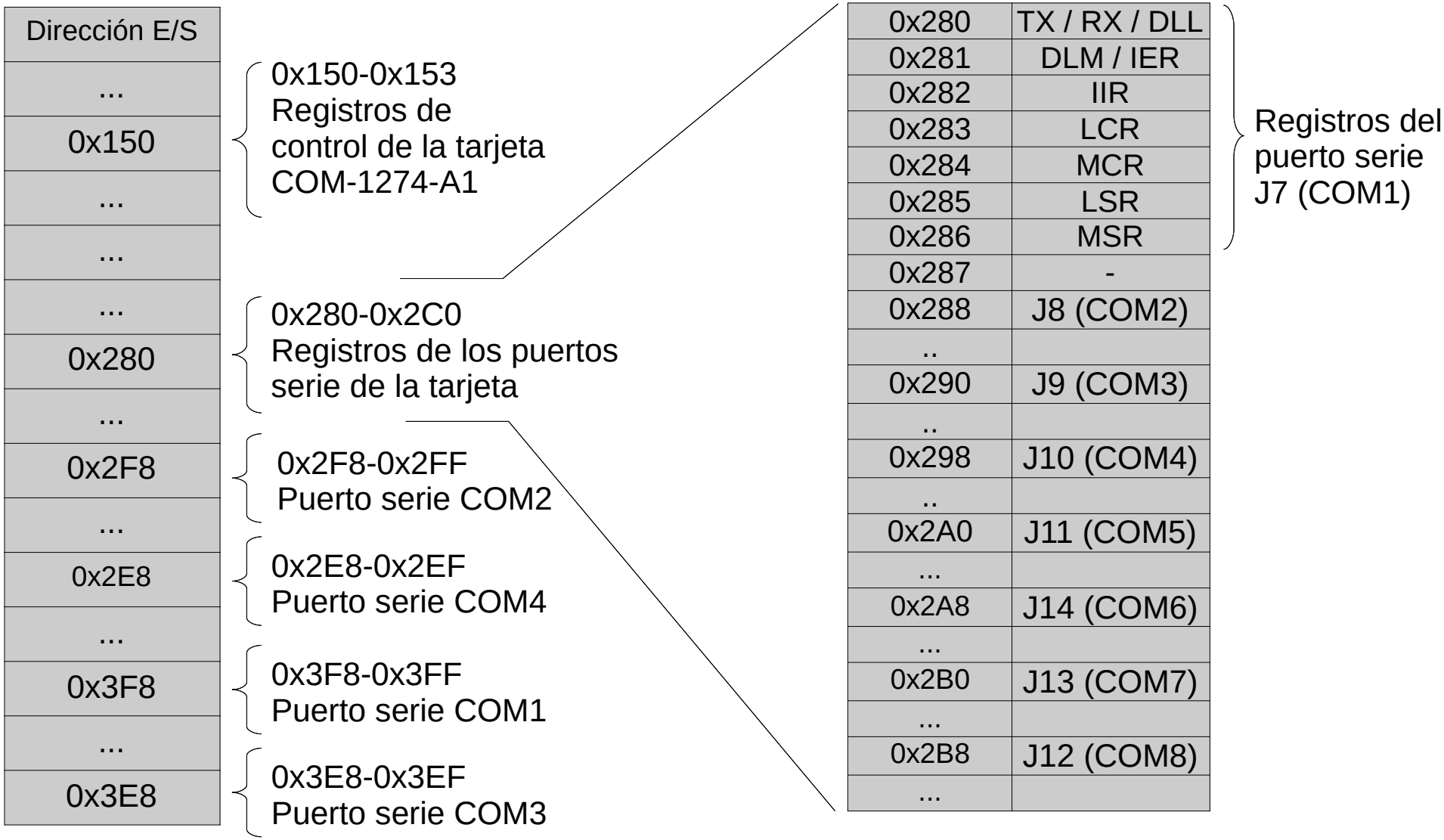
- Tarjeta COM-1274-A1 de Parvus





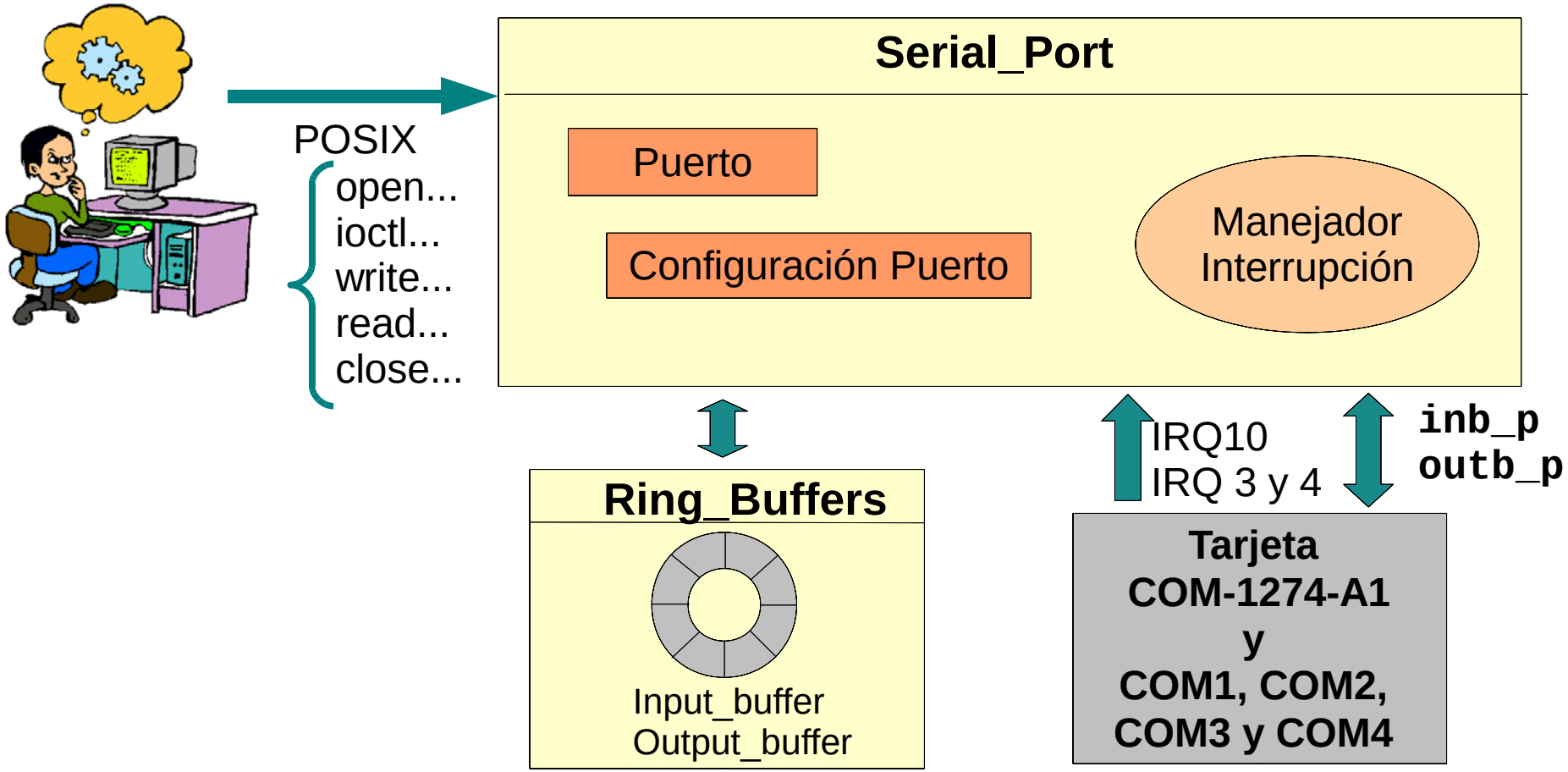
- 8 puertos serie capaces de trabajar en modo RS-232, RS-422 y RS-485
- Controlador I/O 16C554C (4 UARTS)
- Bus ISA para PC/104
- Hasta 2 *buses* CAN

Registros de la tarjeta COM-1274-A1



Manejador para tarjeta 8 puertos serie

- Arquitectura manejador COM-1274-A1



Manejador para tarjeta 8 puertos serie

- Función *ioctl*

```
int serial_port_ioctl (int file_descriptor, int request,
void* argp)
```

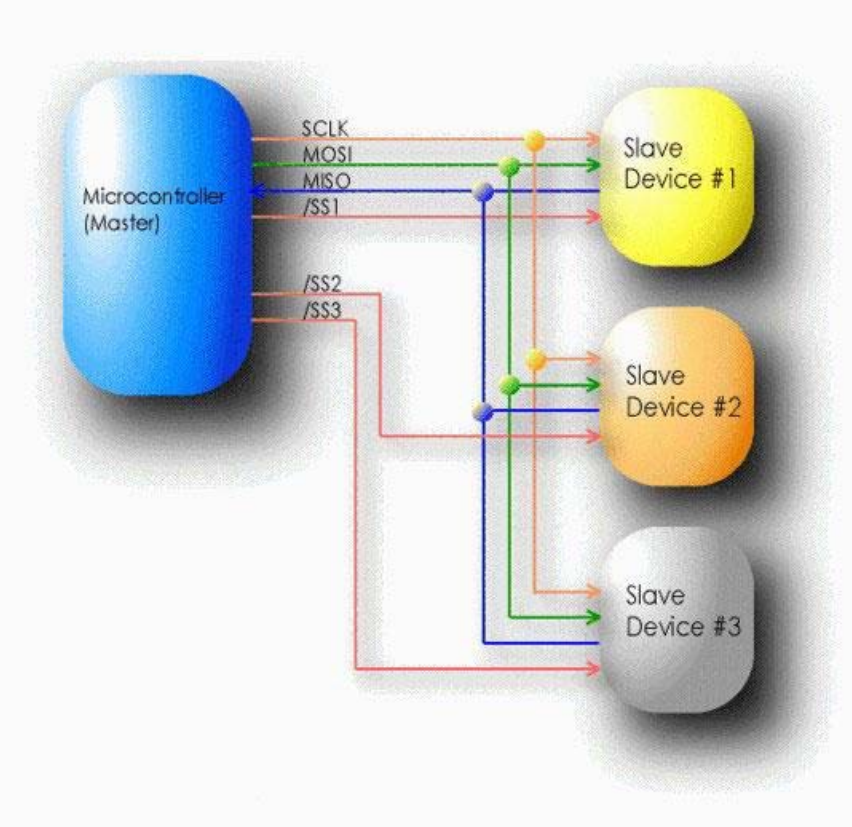
  - SERIAL_SETATTR / SERIAL_GETATTR: configuración del puerto (tamaño de la palabra, paridad, bits de stop, velocidad de transmisión)
  - SERIAL_SETSPEED / SERIAL_GETSPEED
  - SERIAL_ENABLE / DISABLE_INTERRUPTS
  - SERIAL_RESET_LAST_LINE_ERROR
  - SERIAL_FLUSH: permite vaciar el *buffer* de datos recibidos por el puerto

- Funciones *create*, *open*, *read* y *write*:
  - `int serial_port_create ()`
  - `int serial_port_open (int fd, int access_mode)`
  - `ssize_t serial_port_read (int fd, void *buffer, size_t bytes)`
  - `ssize_t serial_port_write (int fd, void *buffer, size_t bytes)`

Índice

- Introducción.
- Objetivos.
- *Drivers* en MaRTE OS.
- Manejador para tarjeta de 8 puertos serie.
- **Subsistema de gestión del bus serie SPI.**
- Manejador para un dispositivo real SPI.
- Manejador para un dispositivo real SPI.
- Conclusiones y trabajo futuro.

El bus serie SPI



El bus serie SPI tiene al menos 4 líneas:

- SCLK
- MOSI (“Master Output Slave Input”)
- MISO (“Master Input Slave Output”)
- SS (“Select Slave”) ó CS (“Chip Select”)

➔Protocolo flexible que permite escoger:
Longitud de la trama, significado y propósito

El bus serie SPI

- Posee 2 bits de configuración
  - CPOL ó “Polaridad de reloj”
  - CPHA ó “Fase de reloj”

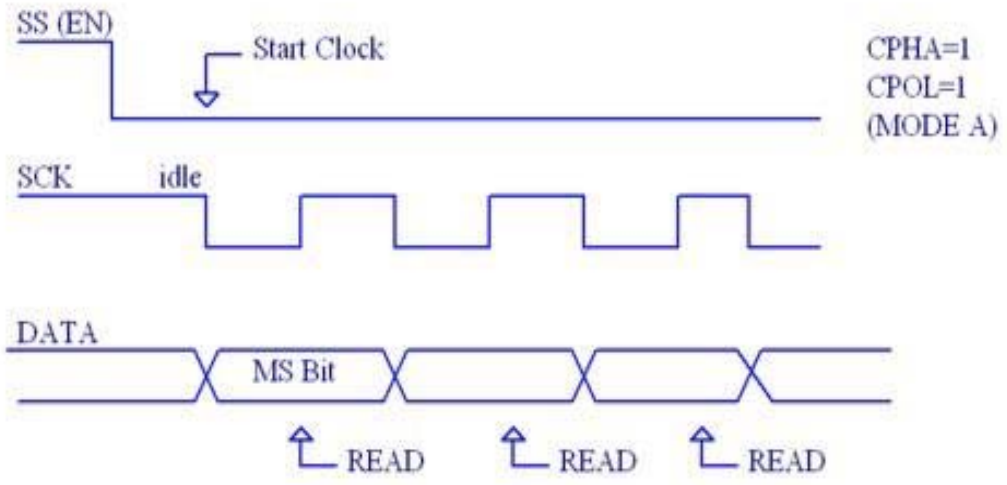
CPHA	CPOL	MODO
1	1	A ó 0
1	0	B ó 1
0	1	C ó 2
0	0	D ó 3

El bus serie SPI

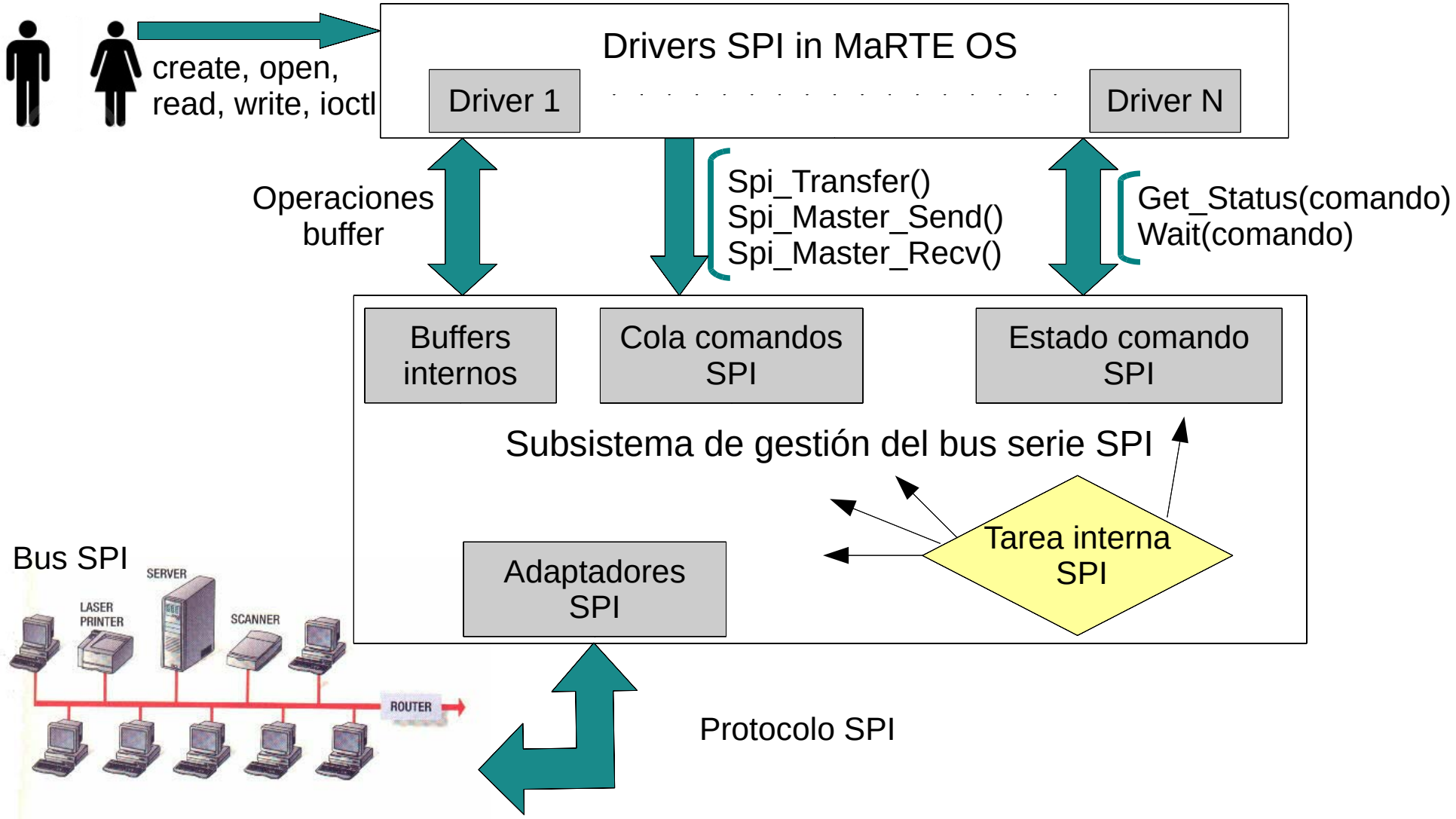
- Posee 2 bits de configuración
  - CPOL ó “Polaridad de reloj”
  - CPHA ó “Fase de reloj”

CPHA	CPOL	MODO
1	1	A ó 0
1	0	B ó 1
0	1	C ó 2
0	0	D ó 3

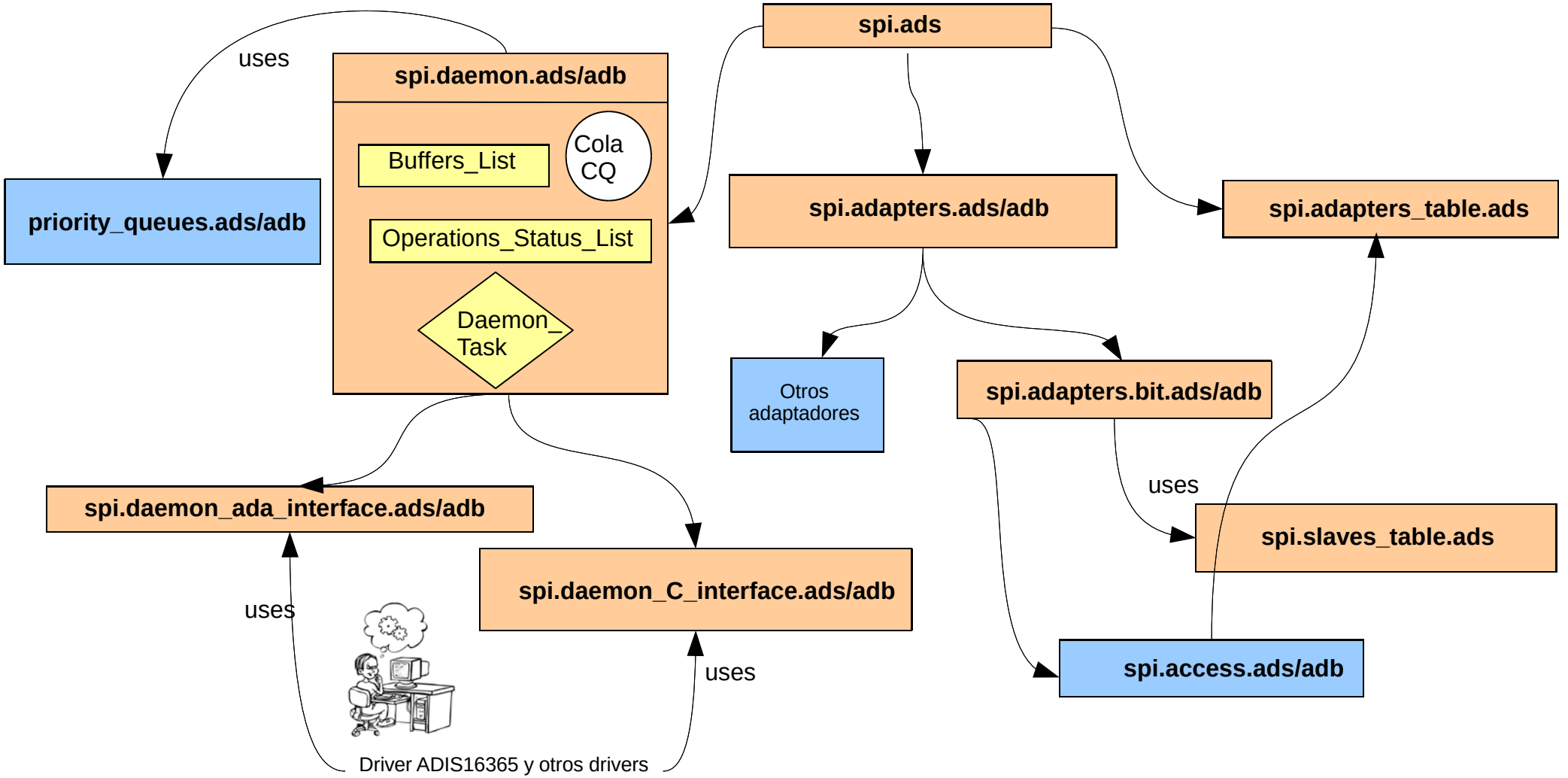
- Modo A



Subsistema de gestión del bus serie SPI



Estructura del subsistema SPI

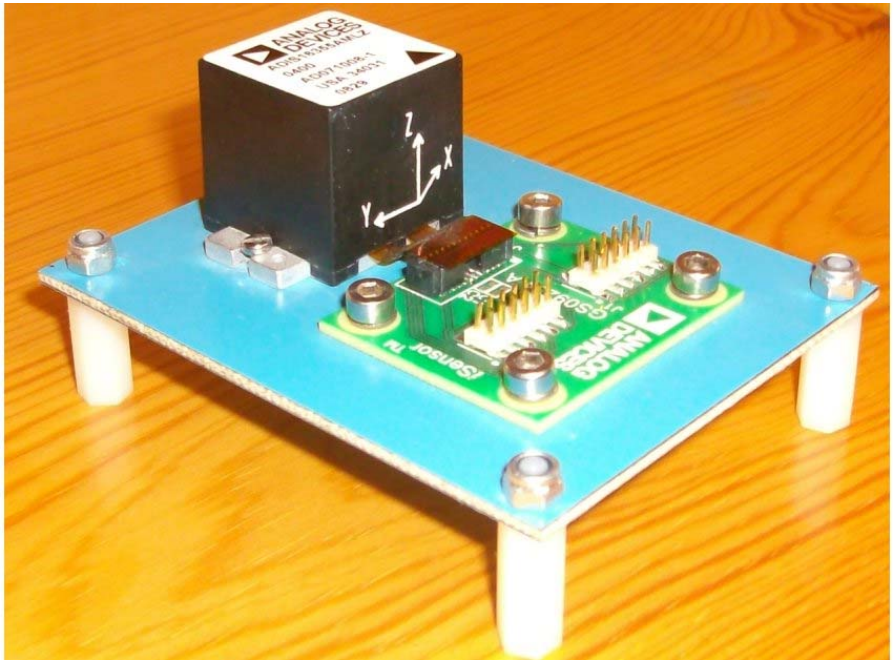


Índice

- Introducción.
- Objetivos.
- *Drivers* en MaRTE OS.
- Manejador para tarjeta de 8 puertos serie.
- Subsistema de gestión del bus serie SPI.
- Manejador para un dispositivo real SPI.
- Conclusiones y trabajo futuro.

Manejador ADIS-16355. Comunicación SPI

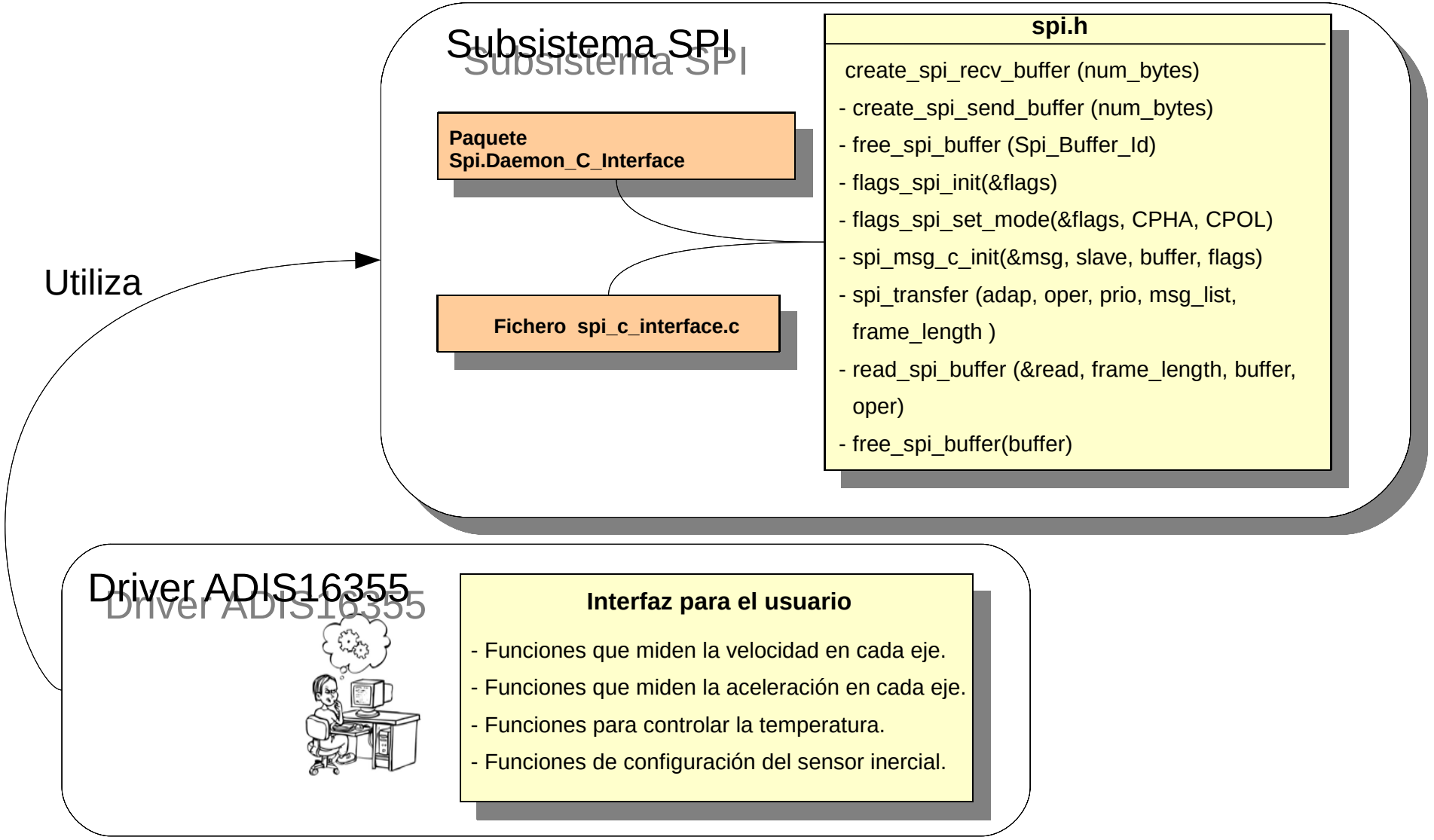
- Sensor inercial de 3 ejes y alta precisión



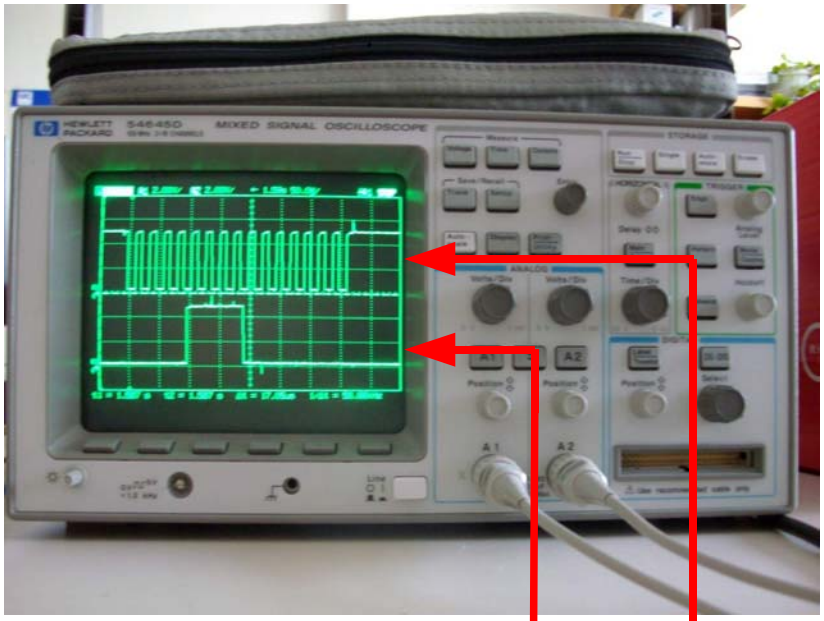
Medir

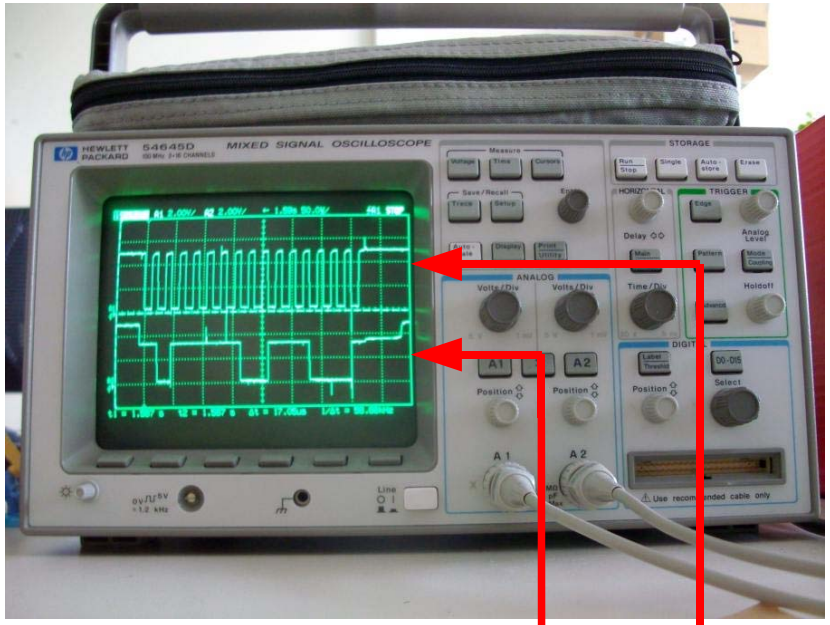
- Potencia suministrada
- Aceleración en los ejes X, Y y Z
- Velocidad de giro en los tres ejes
- Temperatura en los tres ejes

Relación con el subsistema SPI

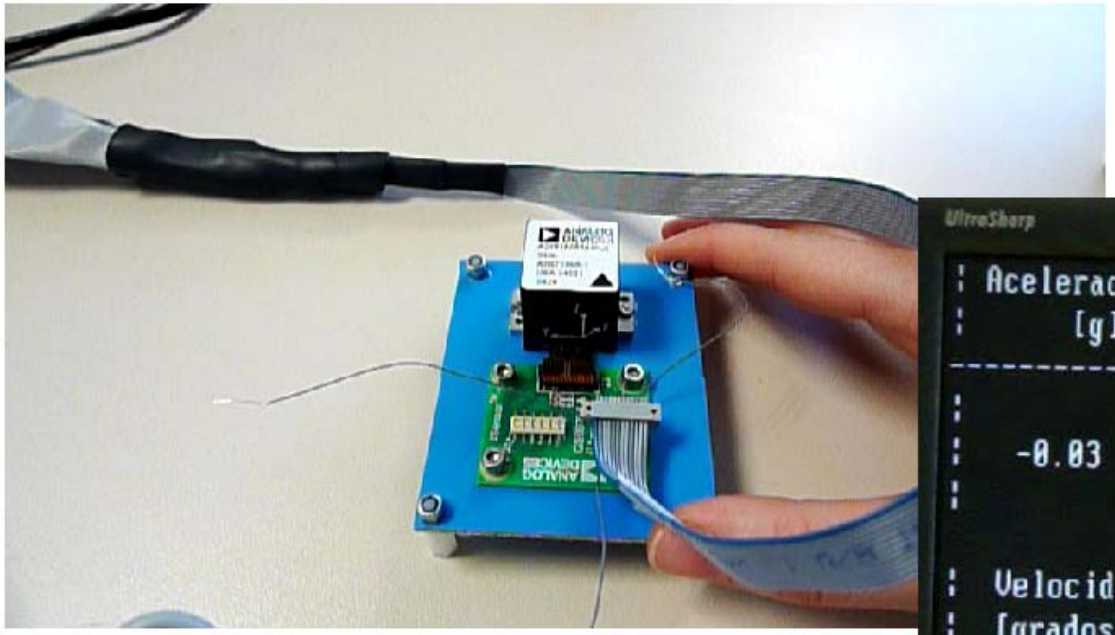


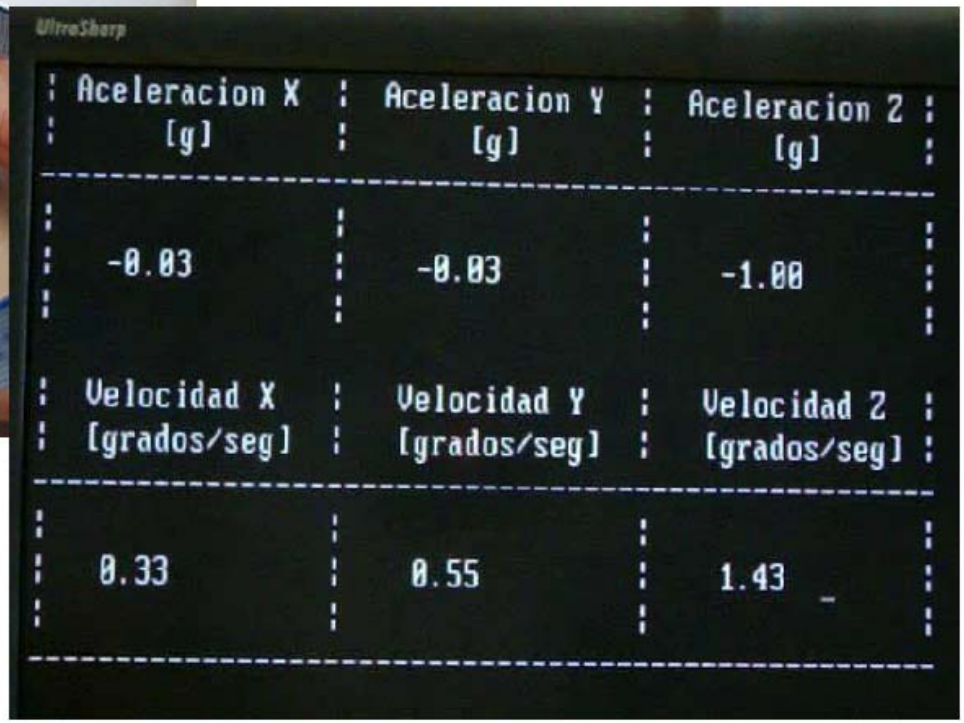
Tramas SPI del ADIS16355

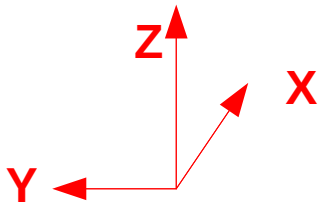




Demostración ADIS16355







Índice

- Introducción.
- Objetivos.
- *Drivers* en MaRTE OS.
- Manejador para tarjeta de 8 puertos serie.
- Subsistema de gestión del bus serie SPI.
- Manejador para un dispositivo real SPI.
- **Conclusiones y trabajo futuro.**

Conclusiones y trabajo futuro

- Objetivos cumplidos
  - Manejador tarjeta COM-1274 (hasta 6 puertos serie)
  - Subsistema SPI en MaRTE OS
  - Manejador sensor inercial ADIS16355
  - 30 archivos y 5472 líneas de código
- Trabajo futuro
  - Desarrollar software de configuración para la tarjeta COM-1274 de Parvus
  - Aumentar el número de adaptadores SPI en el subsistema
  - Creación de *drivers* para dispositivos que usen SPI

MUCHAS GRACIAS POR SU ATENCIÓN

Mónica Puig-Pey González