

Gestión de dispositivos de entrada/salida analógica, digital y por el bus serie I2C

Daniel Sangorrín López

`daniel.sangorrin@gmail.com`

Director: Michael González Harbour

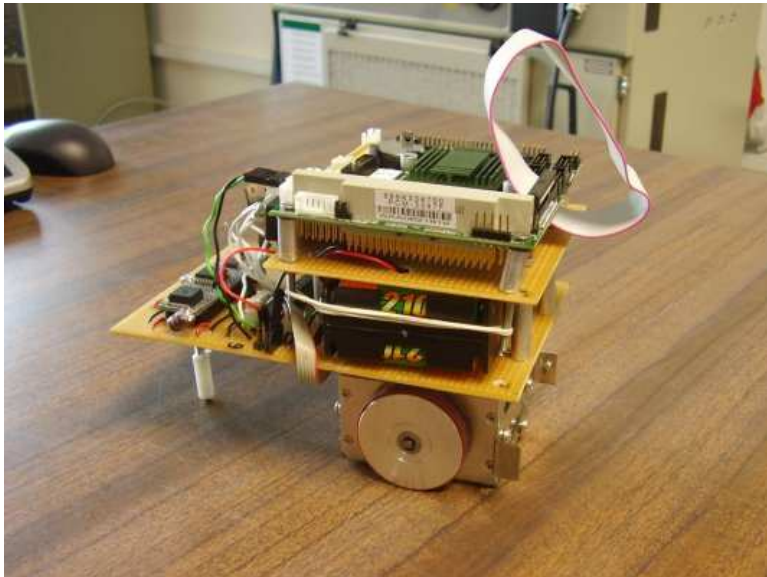
Universidad de Cantabria

Tabla de contenidos

- 1. *Motivación y objetivos del Proyecto***
- 2. Proyecto: arranque de MaRTE OS**
- 3. Proyecto: entrada/salida analógica y digital**
- 4. Proyecto: gestión bus serie I2C**
- 5. Conclusiones y trabajo futuro**

Contexto del proyecto

- **Objetivo general:** sistemas empotrados de automatización industrial y otros
- **Objetivo particular:** torneo RoboCup



Primer prototipo:

1. PC
2. Etapa de potencia
3. Motores + Encoders
4. Chasis
5. Baterías

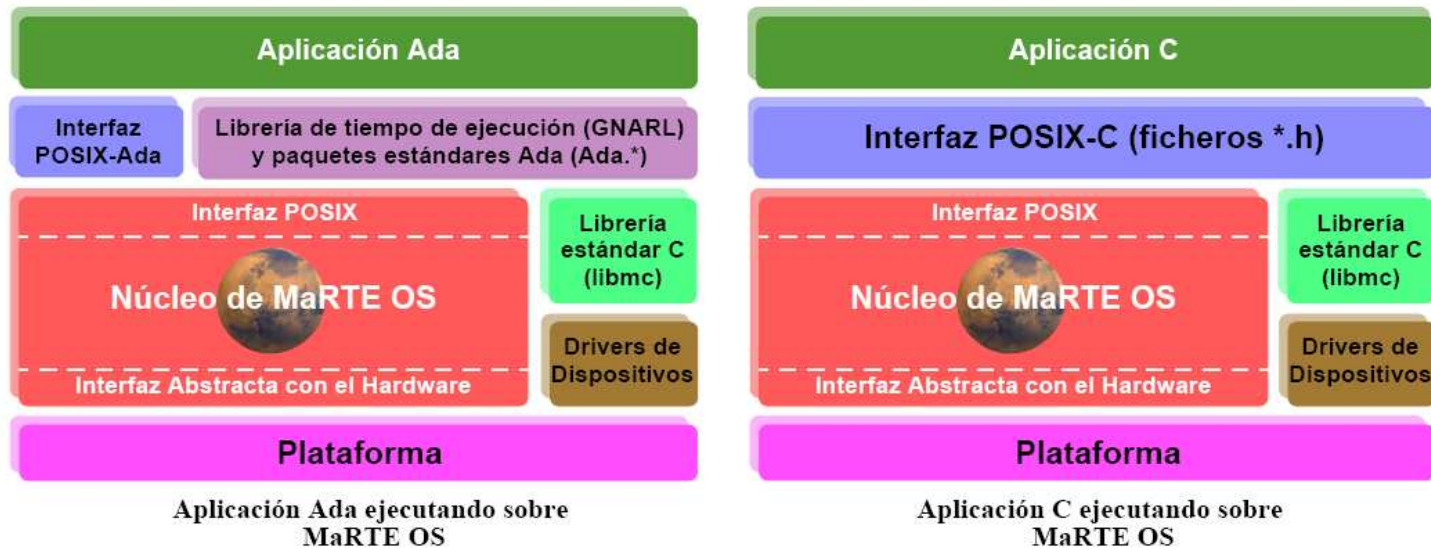
MaRTE OS

Necesitamos un **Sistema Operativo** que ...

- Nos abstraiga del uso del hardware
- Nos facilite interactuar con un entorno cambiante cumpliendo criterios de:
 - *Planificabilidad*
 - *Respuesta de peor caso*
 - *Estabilidad*
- Ligero, eficiente, sin disco duro ni terminal

MaRTE OS cumple los requisitos y es *Libre*

MaRTE OS



- `$ mgnatmake aplicacion_ada.adb`
- `$ mgcc aplicacion_c.c`
- `$ mkkernel`

Drivers en MaRTE OS

- Un *Driver* es un software intermedio entre un dispositivo y su usuario
- Ofrece una interfaz estándar basada en ficheros:
 - `create open read write close ioctl`
 - `fd = open ("/dev/demo_c", O_RDWR);`
 - `count = write(fd, "hola", 5);`
- Oculta las complejidades y *registros* internos de cada dispositivo hardware
 - `Inb_P Outb_P`
 - `E/S mapeada en memoria`

Motivación del Proyecto

- Necesidad de una nueva forma de arranque adecuada al Robot
- Problemas del primer prototipo:
 - CPU sobrecargada
 - Las ruedas resbalan
- Capacidad de expansión limitada



Objetivos del Proyecto

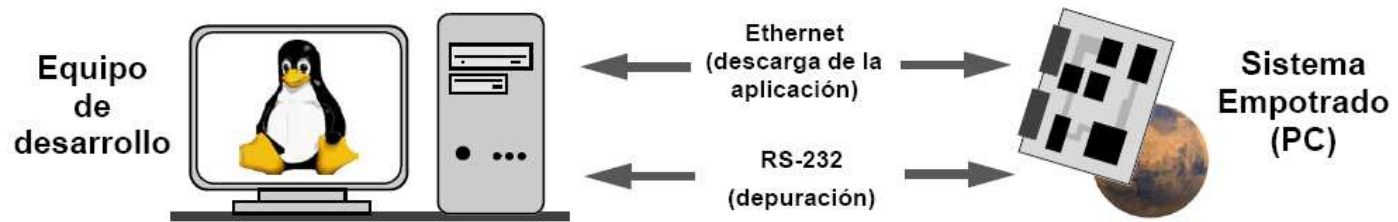
- **Arranque de MaRTE OS:**
 - **Fase desarrollo: Ethernet PXE**
 - **Fase final: CompactFlash™**
- ***Driver* para la tarjeta PCM-3718-H:**
 - **Entrada analógica**
 - **Entrada/salida digital**
- **Subsistema software para la gestión de dispositivos I2C**
 - ***Driver* brújula magnética con interfaz I2C**

Tabla de contenidos

1. Motivación y objetivos del Proyecto
2. ***Proyecto: arranque de MaRTE OS***
3. Proyecto: entrada/salida analógica y digital
4. Proyecto: gestión bus serie I2C
5. Conclusiones y trabajo futuro

Arranque en Fase Desarrollo

- **Ethernet**: usaremos el protocolo **PXE**, integrado en la placa, junto al cargador **ETHERBOOT**



Necesitamos configurar:

- Servidor **DHCP** (Ej: `/etc/dhcpd.conf`)
- Servidor **TFTP** (Ej: `/etc/xinetd.d/tftp`)
- Servidor **NFS** (Ej: `/etc/exports`)
- Emuladores: **QEMU**, **BOCHS**, **VmWare**

Arranque en Fase Final

- Utilizaremos una tarjeta **CompactFlash™**



- Muy ligera
 - Soportada por la placa
 - Apenas consume energía
-
- Necesitamos instalar y configurar el cargador **GNU/GRUB** para que arranque con nuestra aplicación

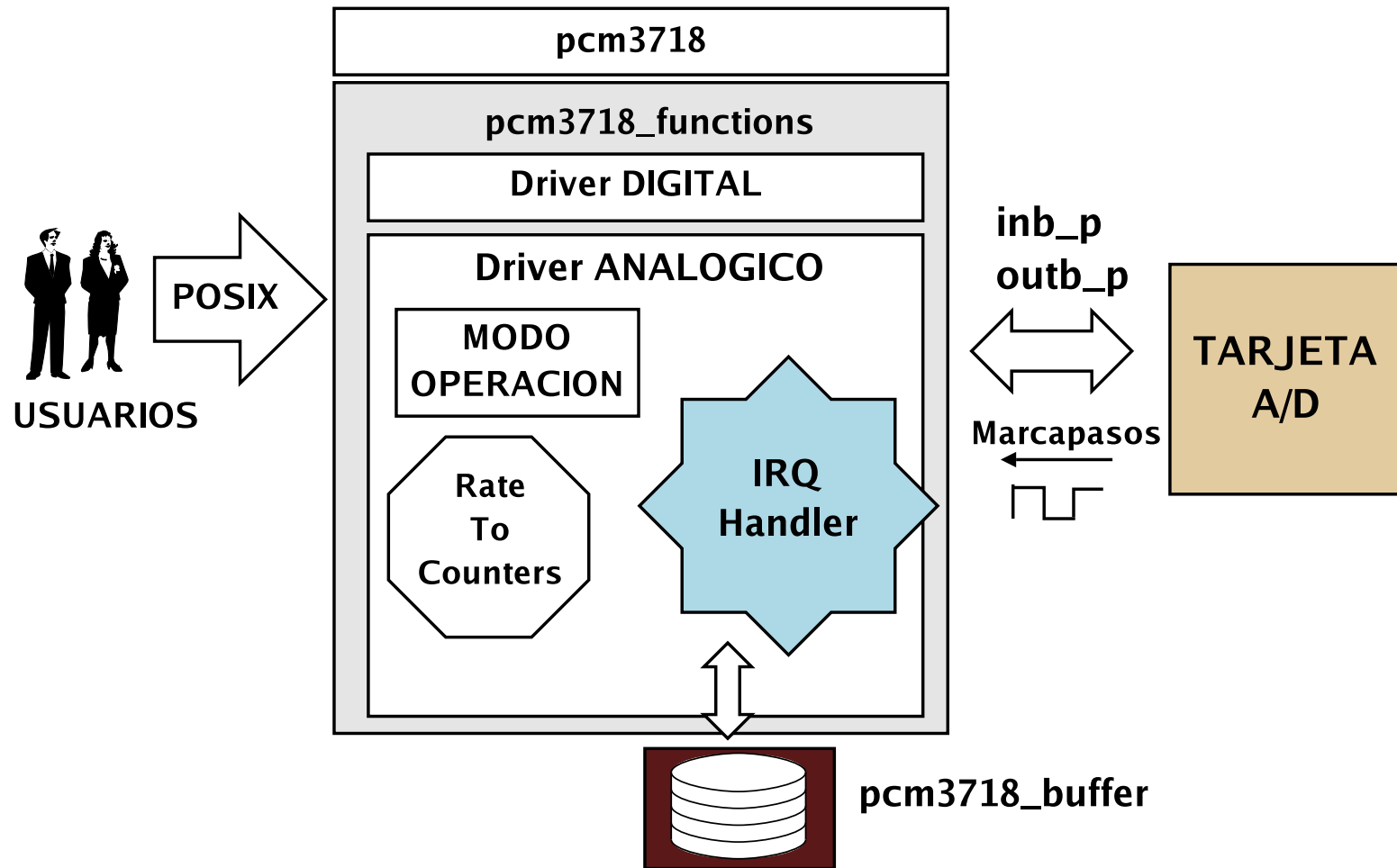
Tabla de contenidos

1. Motivación y objetivos del Proyecto
2. Proyecto: arranque de MaRTE OS
3. ***Proyecto: entrada/salida analógica y digital***
4. Proyecto: gestión bus serie I2C
5. Conclusiones y trabajo futuro

La tarjeta PCM-3718-H

- **Entrada/Salida Digital**
 - **Dos canales digitales de 8 bits.**
 - **Compatibles *TTL***
- **Entrada Analógica**
 - **16 entradas simples u 8 diferenciales**
 - **Convertor A/D 12 bits hasta 100 kHz (*DMA*)**
 - **Rangos de entrada programables**
 - ***Trigger Software, Pacer* o *Externo***
 - ***Sondeo, Interrupciones* o *DMA***
 - **Se configura mediante *jumpers***

Arquitectura tarjeta A/D



Entrada Analógica

Se distinguen cinco modos de funcionamiento

- **Software+Fixed**: captura N muestras
- **Pacer+Fixed**: captura N muestras al ritmo del generador de pulsos (*Pacer*)
- **Pacer+Scan**: captura continua de muestras al ritmo del *Pacer*
- **External+Fixed**: captura N muestras al ritmo de una señal externa
- **External+Scan**: captura continua de muestras al ritmo de una señal externa

Entrada Analógica

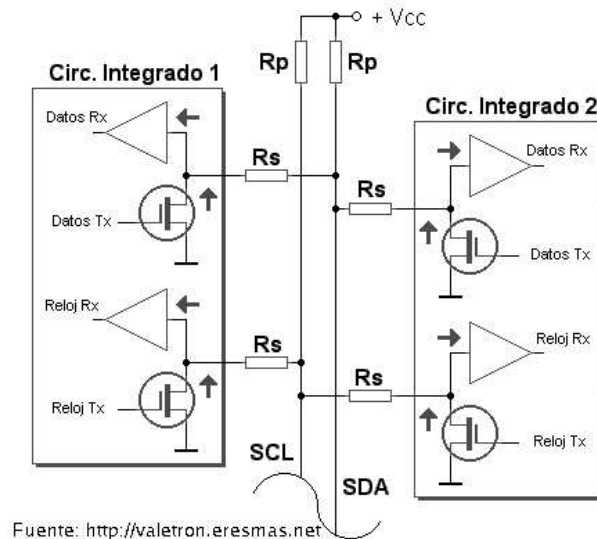
- **read**
 - **Scan**: lee muestras del Buffer
 - **Fixed**: captura y devuelve N muestras
- **ioctl**
 - **Set_Range_Of_Channel**: rangos de entrada
 - **Set_Parameters**
 - **Trigger + Mode**
 - **Start_Ch, Stop_Ch**
 - **Count**
 - **Scan_Rate**
 - **Get_Status**: muestras en el Buffer
 - **Flush**: vacía el Buffer

Tabla de contenidos

1. Motivación y objetivos del Proyecto
2. Proyecto: arranque de MaRTE OS
3. Proyecto: entrada/salida analógica y digital
4. ***Proyecto: gestión bus serie I2C***
5. Conclusiones y trabajo futuro

El bus serie I2C

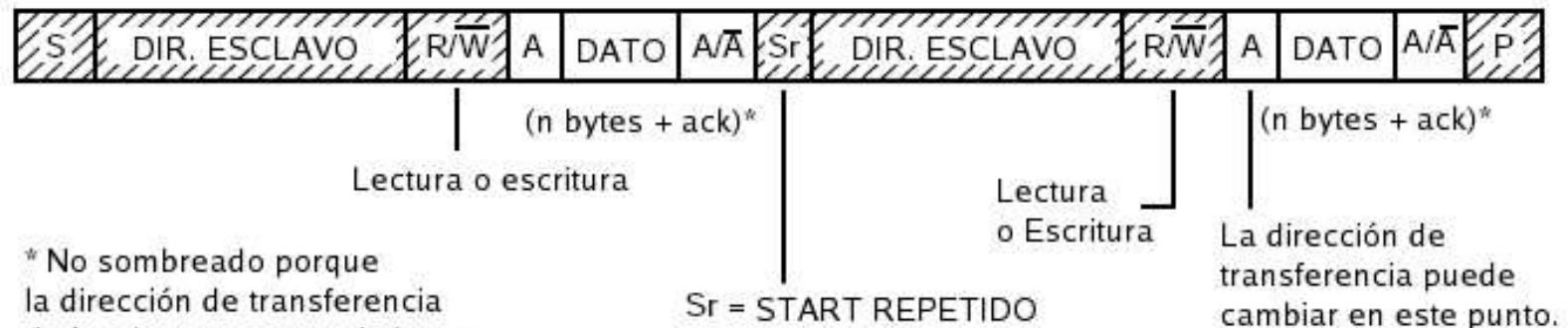
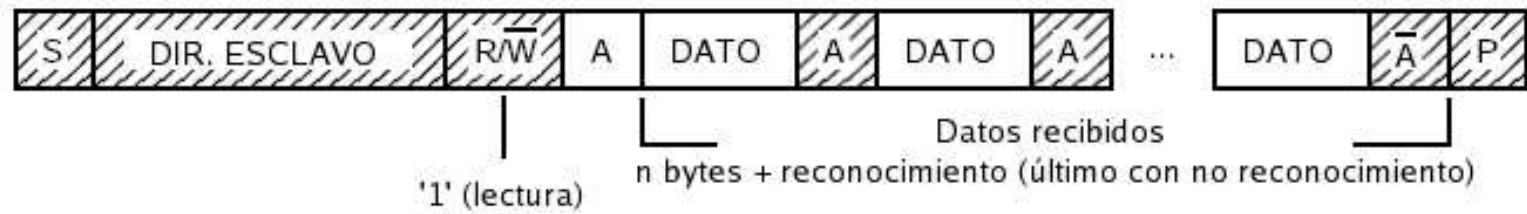
El bus serie I2C está formado por tres cables



- **SDA: datos**
- **SCL: reloj**
- **GND: masa**

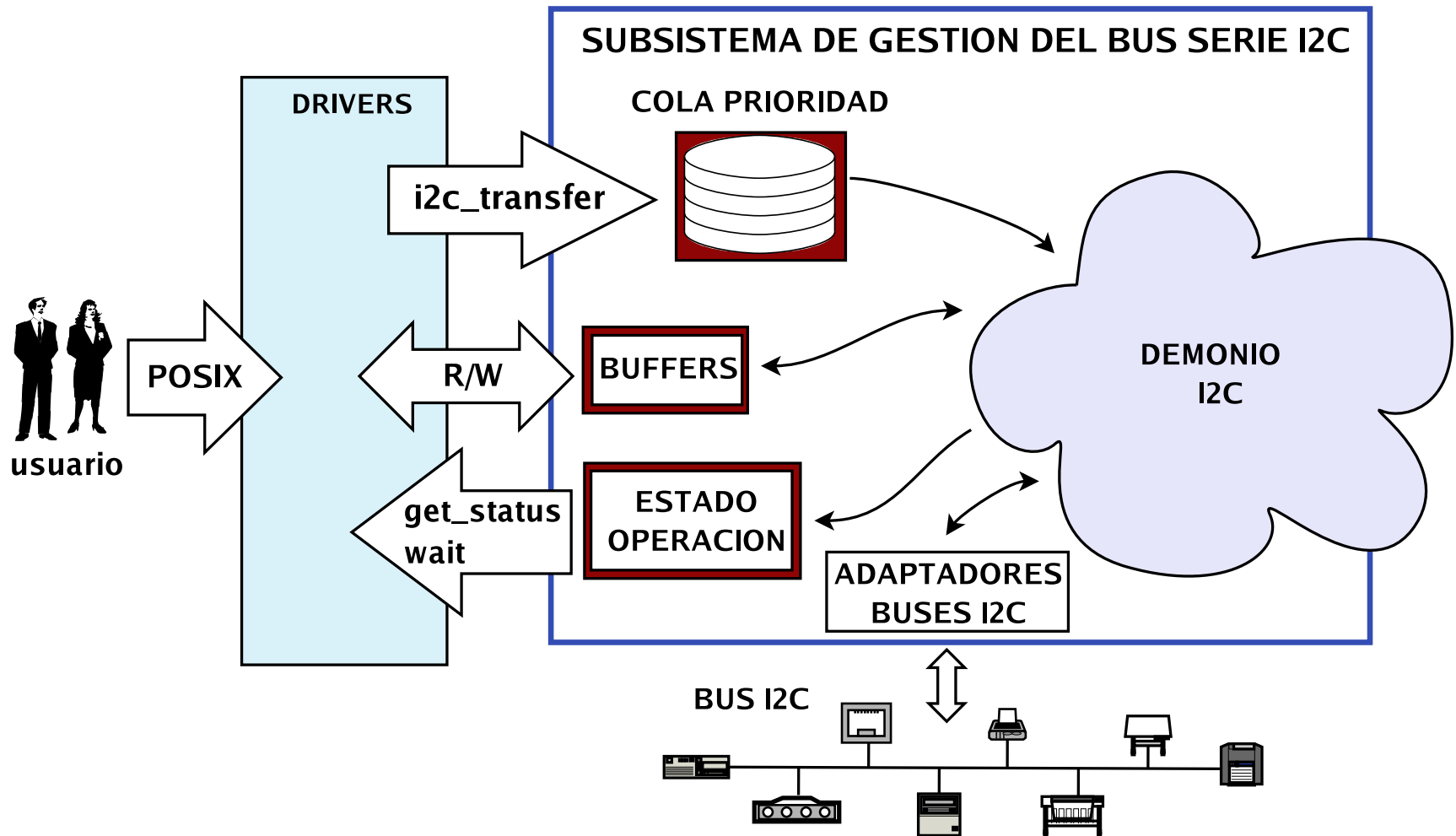
- **Son bidireccionales**
- **Drenador/Colector abierto (AND cableada)**

El protocolo I2C

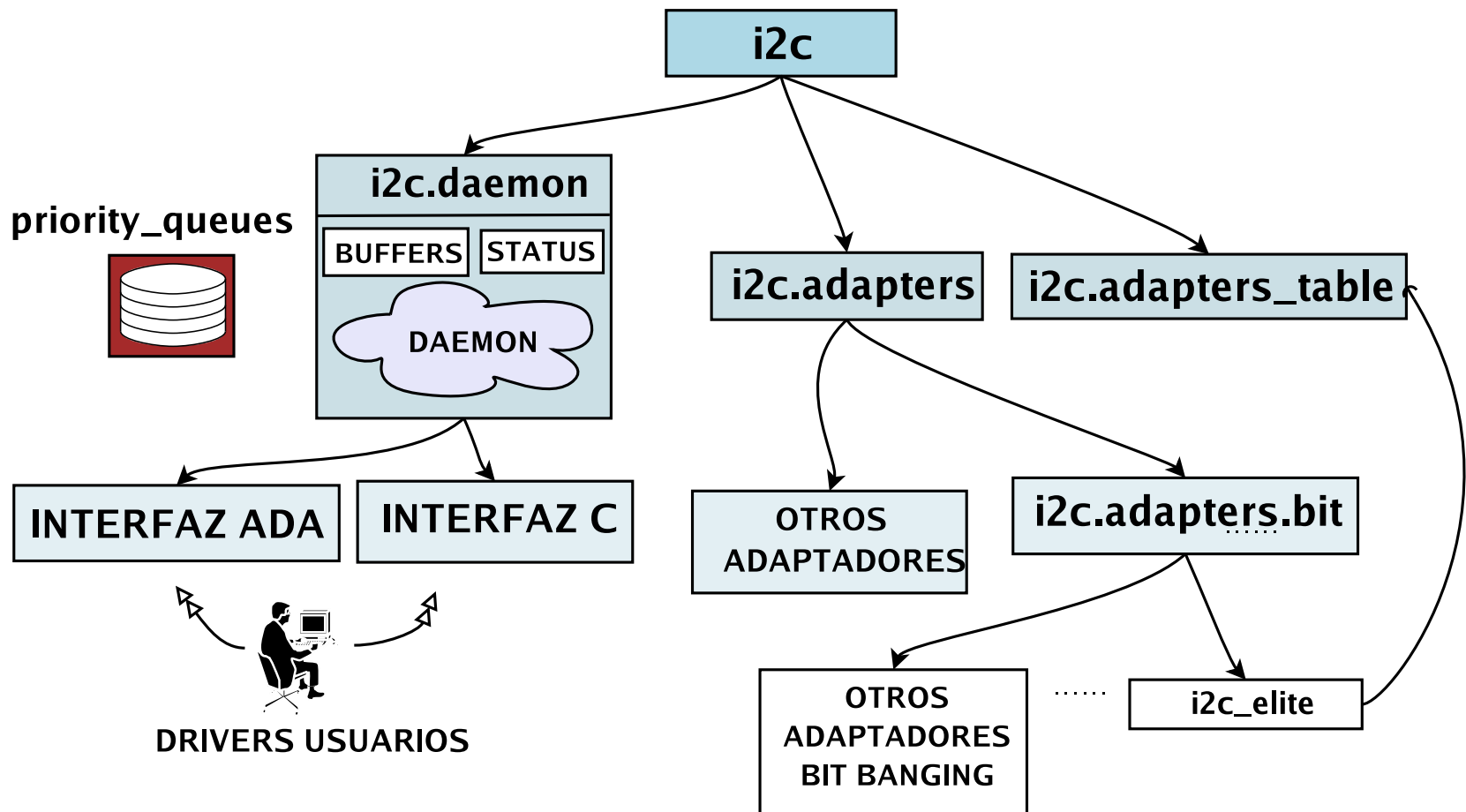


* No sombreado porque la dirección de transferencia de los datos y reconocimientos depende de los bits R/W

Subsistema gestor del bus I2C



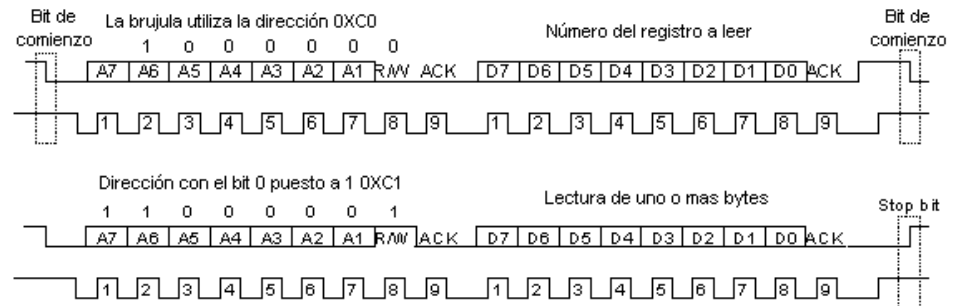
Arquitectura del subsistema I2C



Brújula Magnética



Pin 9 0V Masa
Pin 8 No conectado
Pin 7 50/60Hz
Pin 6 Calibración
Pin 5 No conectar
Pin 4 PWM
Pin 3 SDA
Pin 2 SCL
Pin 1 +5V



● `ioctl`:

- `Start_Conversion`: iniciar una captura
- `Bearing_Mode_Byte`: 8 bits (0 .. 255)
- `Bearing_Mode_Word`: 16 bits (0.0 .. 359.9)
- `Get_Status`: chequear estado de operación

● `read`: leer el valor capturado

Arquitectura Brújula

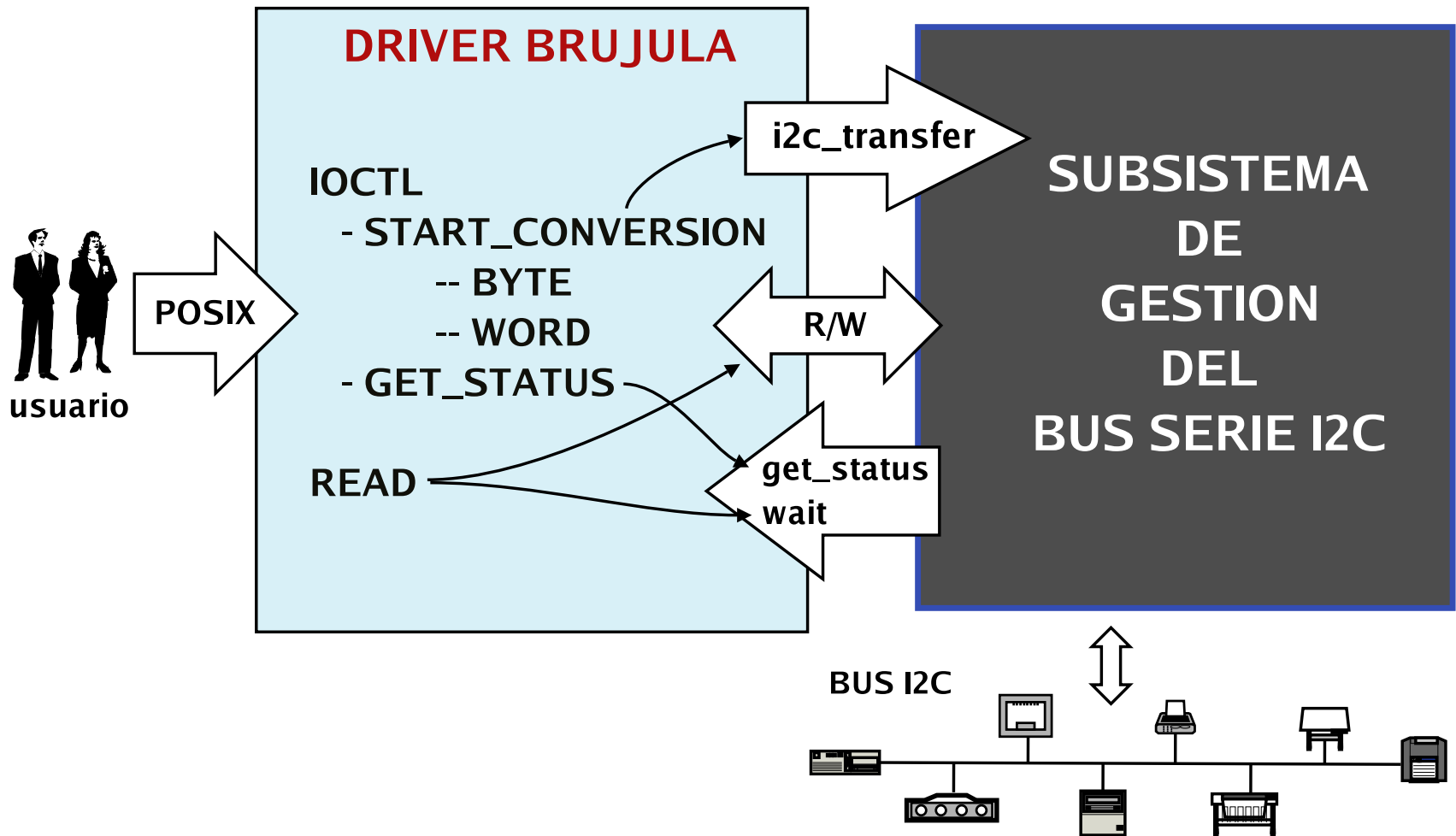


Tabla de contenidos

1. Motivación y objetivos del Proyecto
2. Proyecto: arranque de MaRTE OS
3. Proyecto: entrada/salida analógica y digital
4. Proyecto: gestión bus serie I2C
5. ***Conclusiones y trabajo futuro***

Conclusiones y futuro

- **Objetivos cumplidos**
 - **Arranque de MaRTE OS**
 - ***Driver* para la tarjeta PCM-3718-H**
 - **Subsistema gestor de dispositivos I2C**
 - ***Driver* brújula magnética con interfaz I2C**
- **43 archivos, 3085 líneas de código**
- **Trabajo futuro:**
 - 1. Periféricos con interfaz I2C**
 - 2. Adaptadores I2C**
 - 3. Utilizar la brújula**

Gracias por su atención