

Actuator Interface Card (AIC)

Versão 2.1.9

Walter Fetter Lages

fetter@ece.ufrgs.br

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Sistemas Elétricos de Automação e Energia

ENG10052 Laboratório de Robótica

Introdução

- Placa de acionamento baseada no microcontrolador dsPIC30F4012
 - 1 motor D.C. acionado por PWM
 - 1 freio eletromagnético
 - 1 Interface com *encoder* em quadratura (contador de 16 bits)
 - 1 sinal de índice
- Barramento CAN
- Interface RS-232

AIC 2.1.9

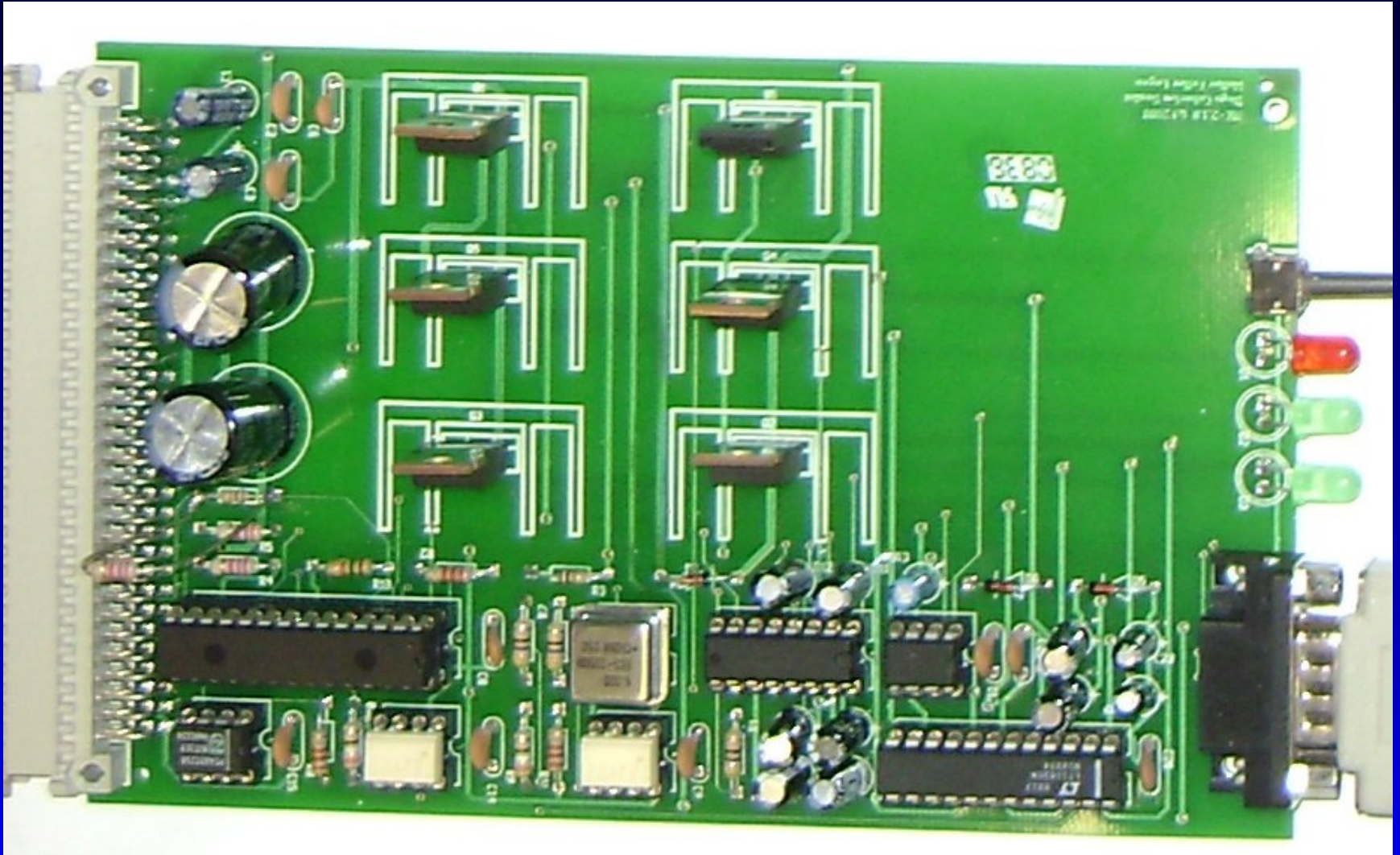
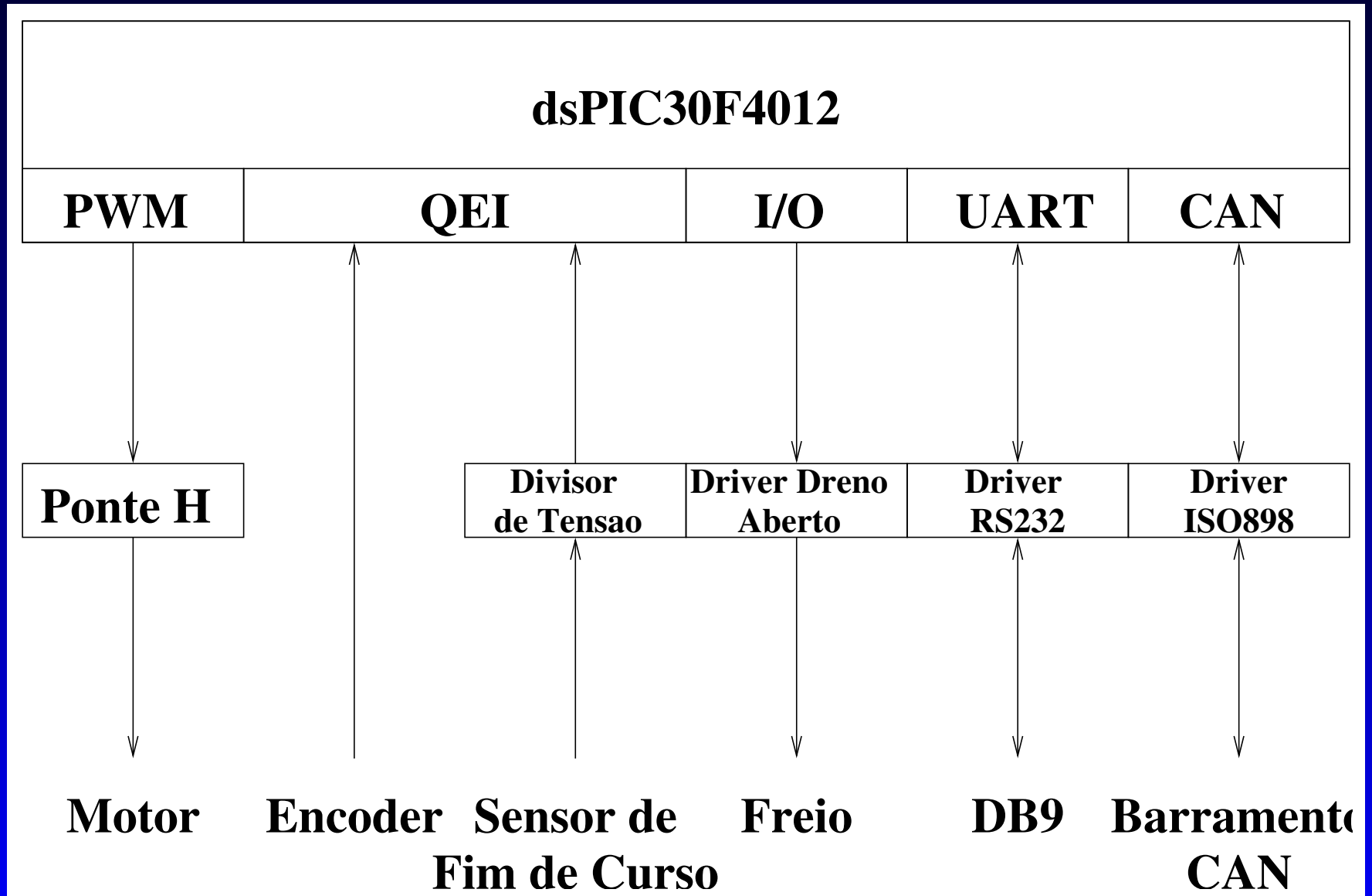
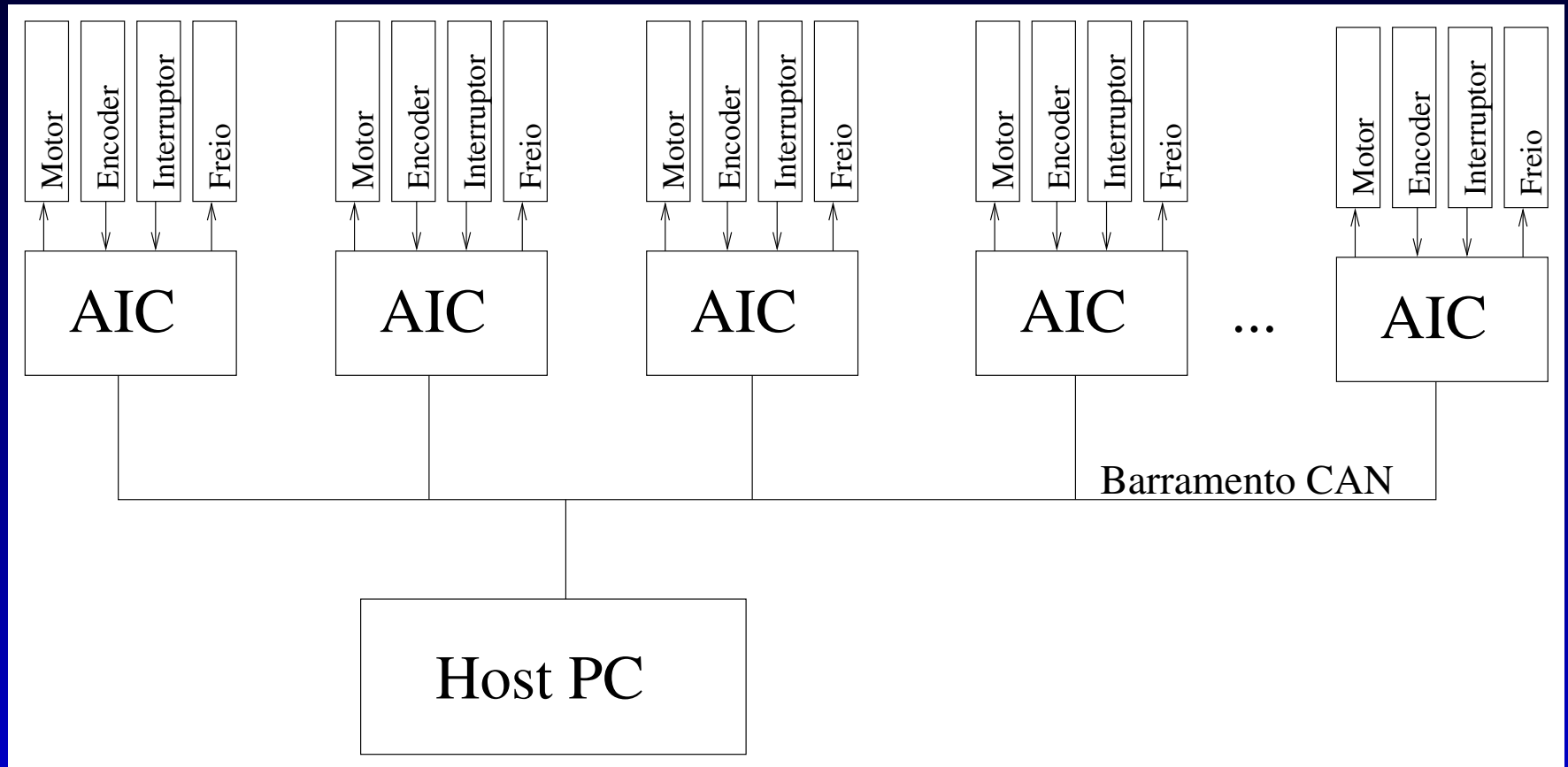


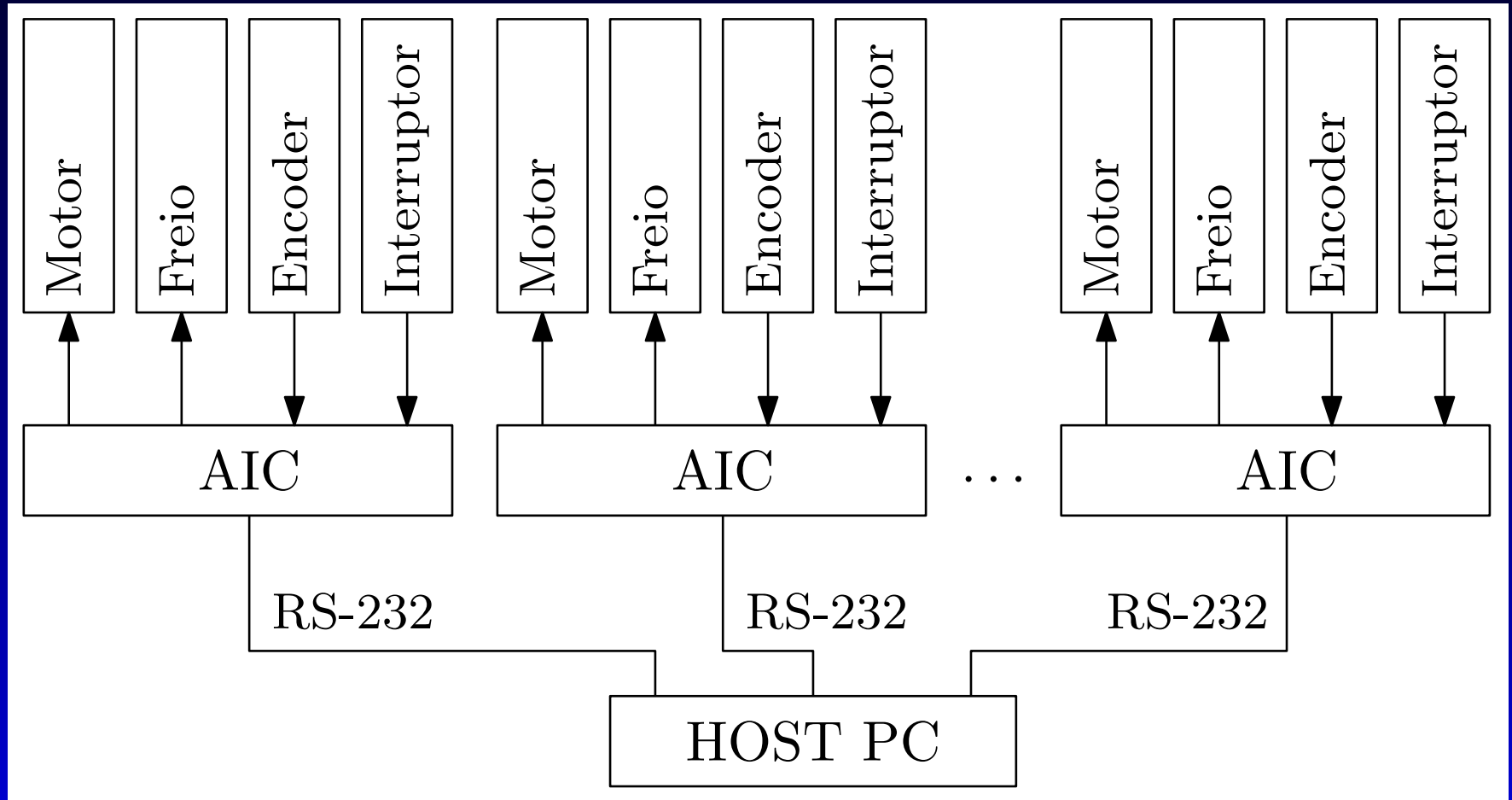
Diagrama de Blocos



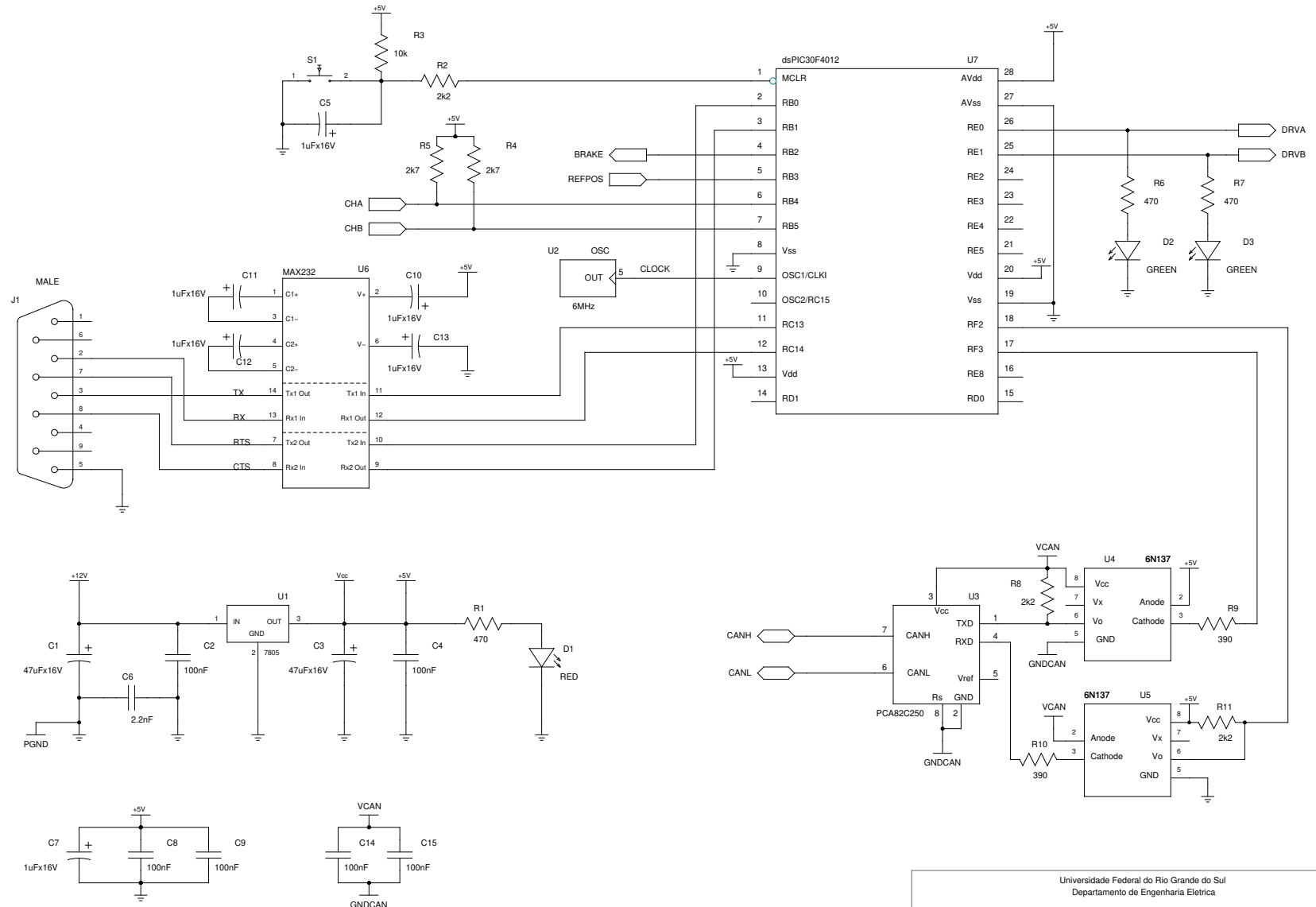
Comunicação por CAN



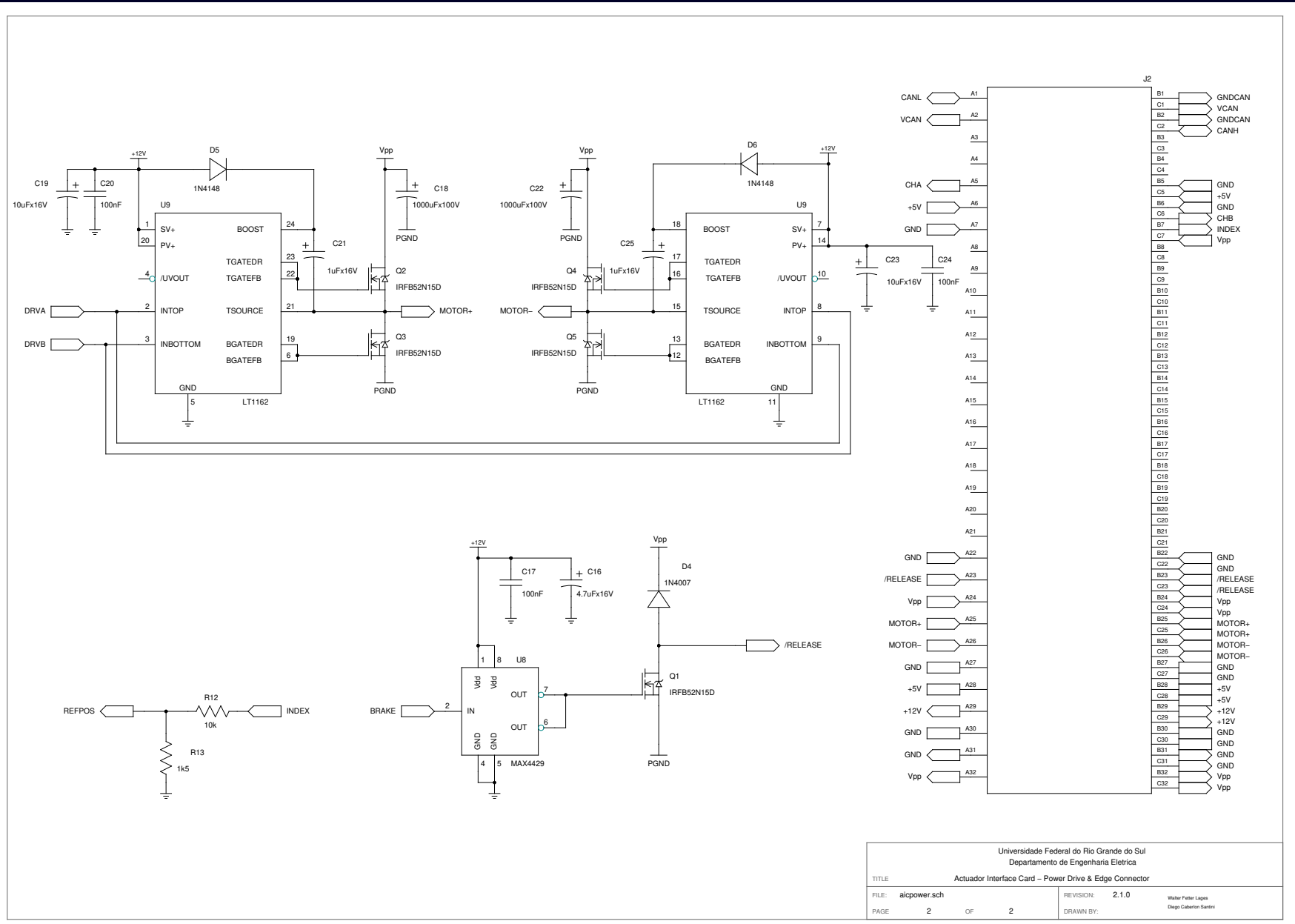
Comunicação por RS-232



Esquemático



Esquemático





Conector CAN

- DB9

Pino	Nome	Descrição
1	—	Reservado
2	CAN_L	Bit diferencial
3	CAN_GND	Terra do barramento CAN
4	—	Reservado
5	CAN_SHLD	Blindagem (Opcional)
6	GND	Terra (Opcional)
7	CAN_H	Bit diferencial
8	—	Reservado
9	CAN_V+	Alimentação externa (Opcional)

Firmware

- *Software* gravado no dsPIC
- Implementado usando o FreeRTOS
- Funções de baixo nível para suportar os comandos recebidos
 - Acionamento e liberação do freio
 - Contador de 32 bits para o encoder (funções que retornam a contagem ou o ângulo)
 - Status do índice
 - Habilitação/desabilitação do motor
 - Configuração da frequência e do ciclo de trabalho do PWM
 - Protocolo de comunicação via UART
 - Protocolo de comunicação via CAN
 - Watchdog



Protocolo

- O protocolo é baseado no pacote CAN 2.0A
- Código de comando de 11 bits
- 8 bytes de dados, se necessário



Protocolo

Bits	Descrição	
10-5	Comando	Dado
	000000 = Reservado	_____
	000001 = Reinicia AIC	_____
	000010 = Desligar motor	_____
	000011 = Aplicar freio	_____
	010000 = Aplicar tensão no motor	8 bytes
	100000 = Ler <i>status</i>	8 bytes
	110000 = Desligar freio	_____
	110001 = Ligar motor	_____
4-0	Endereço da AIC	
	00000 Reservado	
	00001 AIC 1	
	00010 AIC 2	
	:	



Biblioteca `aic_lib`

- Implementa a comunicação do *host* com a AIC
- Implementa comunicação por CAN ou RS-232
- Implementa as seguintes classes:
 - `aic_serial`**: Configuração, leitura e escrita usando um dispositivo do tipo `dev/ttySx` do Linux
 - `aic_can`**: Configuração, leitura e escrita usando o driver `SocketCan` do Linux
 - `aic_comm`**: Abstrai o canal de comunicação, através das funções:
 - `send_command()`
 - `get_status()`



Biblioteca `aic_lib`

`aic_brake`: Habilita/desabilita o freio, através das funções:

- `apply()`
- `release()`

`aic_motor`: Acionoa o motor, através das funções:

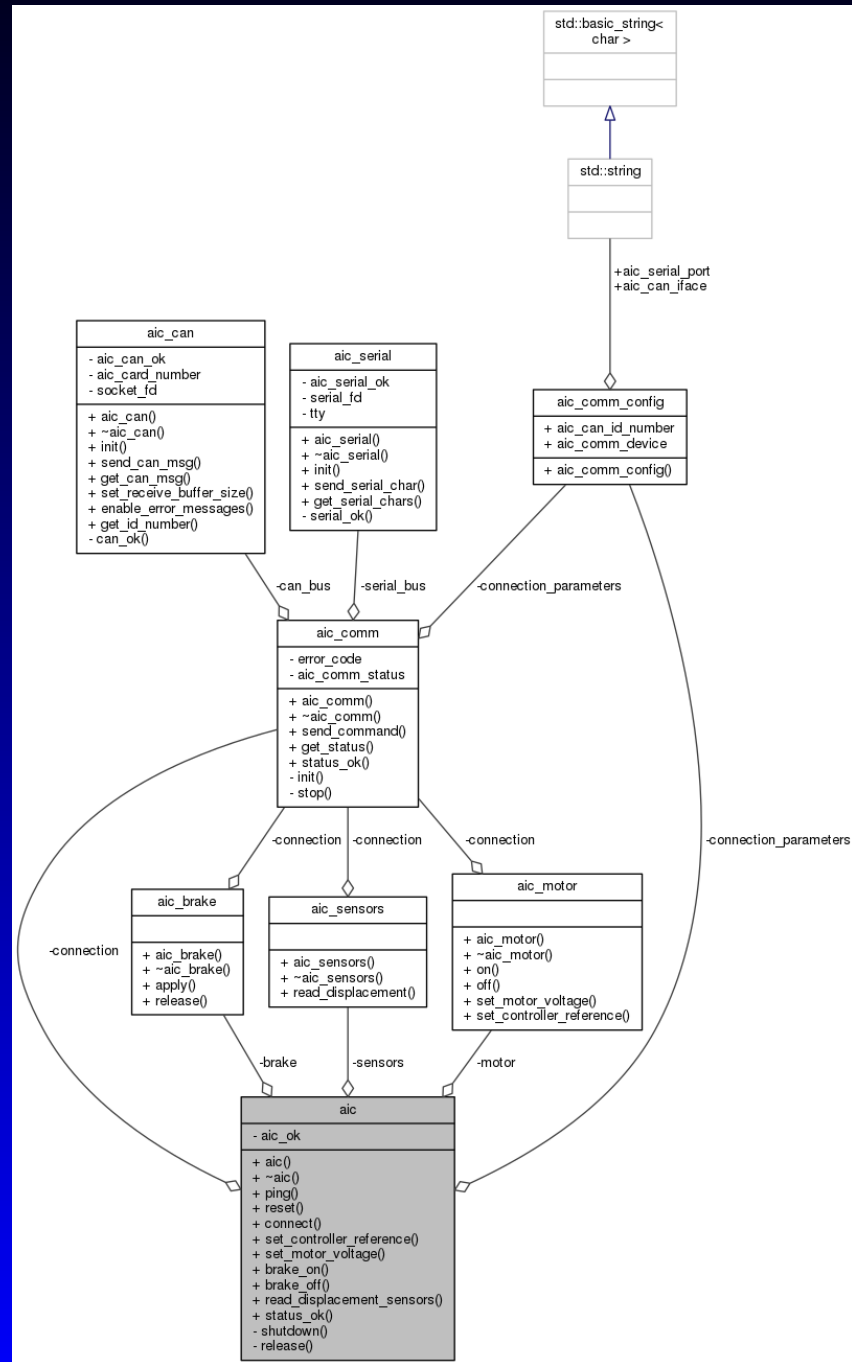
- `off()`
- `on()`
- `set_motor_voltage()`

`aic_sensors`: lê o *encoder* e so índice, através da função:

- `read_displacement()`

`aic`: Instancia objetos das classes anteriores e oferece uma visão unificada do *hardware*

Diagrama de Classes





Exemplo

```
#include <iostream>
```

```
#include <aic.h>
```

```
int main(int argc,char *argv[])
```

```
{
```

```
    aic_comm_config_t param;
```

```
    param.aic_comm_device=rs232;
```

```
    param.aic_serial_port="/dev/ttyUSB0";
```

```
    aic_board(param);
```

Exemplo

```
if (board.status_ok())  
{  
    board.brake_off();  
    board.set_motor_voltage(5);  
    usleep(1000000);  
    auto dp=board.read_displacement_sensors();  
    std::cout << "angle: " << dp.joint_displacement << std::endl;  
    board.brake_on();  
}  
board.~aic();  
return 0;  
}
```



Instalação da `aic_lib`

- A `aic_lib` não é um pacote ROS
- Não deve ser clonada em `~/colcon_ws/src`

```
mkdir ~/src
```

```
cd ~/src
```

```
git clone http://git.ece.ufrgs.br/eng10052/ufrgs\_aic
```

```
mkdir build
```

```
cd build
```

```
cmake -DCMAKE_INSTALL_PREFIX=$HOME ../ufrgs_aic
```

```
make
```

```
make install
```

```
cd ..
```

```
rm -r build
```

Uso em Pacotes ROS 2

- Não é um pacote ROS 2, portanto não é encontrada pelos mecanismos usuais
- Supondo que tenha sido instalado em `$HOME/lib` e `$HOME/include` e que binário a ser linkado seja `q2d_hardware`
- Incluir no `CMakeLists.txt` as *tags*:

```
target_include_directories(q2d_hardware PUBLIC $ENV{  
    HOME}/include)  
target_link_libraries(q2d_hardware $ENV{HOME}/lib/  
    libaic_lib.a)
```
