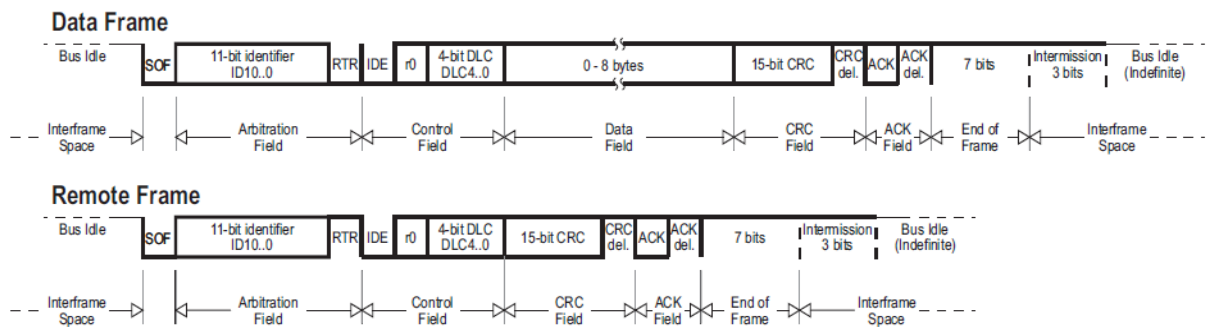


# Protocolo CAN



Será utilizada apenas a “Data frame” e será variado o tamanho da trama (DLC) e o campo de dados (DATA FIELD).

Não esquecer que os campos de dados suportam apenas 8bytes.

Será utilizado o identificador para identificar tanto os mestres como o mestre central:

- Mestre 1 – ID: 0x81;
- Mestre 2 – ID: 0x82;
- Mestre Central – ID: 0x80.

## 1. Definir função (enviada por ‘mestre’) – F

Pretende-se definir a função que os escravos irão desempenhar desde a chegada desta ordem. São enviadas 3 tramas seguidas, envia a primeira espera pelo fim do envio, envia segunda trama espera pelo fim do envio, envia a segunda e espera confirmação.

DLC pedido e resposta – 8 (para as 3 tramas)

TRAMA	Bits [7:0]							
	1	2	3	4	5	6	7	8
F.1	id_nó	idMess	tipTrama	funcao1	funcao2	ano_ini1	ano_ini2	mes_ini
F.2	dia_ini	ano_end 1	ano_end 2	mes_end	dia_end	t_ini_h	t_ini_m	t_ini_s
F.3	t_end_h	t_end_m	t_end_s	n_sensors	Sensor0	Sensor1	Sensor2	Sensor3
resp_F1/2	igual							

ID\_Nó:

- Do mestre respectivo (Mestre 1 – id: 1 ou 2 | Mestre 10 – id: 10)

Tipo\_Trama:

- F (neste caso): definir função

ID\_Message:

- ID da mensagem até entre 0 e 200 (Repete-se em todas as tramas)

Função:

- 'C'|'A' - continuousA (para todos os sensores)
- 'C'|'S' - continuousS (para sensores específicos)
- 'S'|'A' - sleep
- 'I'|'A' - intervalA
- 'I'|'S' - intervalS

data\_ini /data\_end (apenas para função 'interval' – para as outras envia 0's):

(2011/10/30)

- ano\_ini1: 20
- ano\_ini2: 11
- mes\_ini: 10
- dia\_ini: 30

T\_ini / T\_end (apenas para função 'interval' – para as outras envia 0's):

(14h10m30s)

- T\_ini\_h: 14
- T\_ini\_m: 10
- T\_ini\_s: 30

n\_sensors - nesta trama até 6 sensores (apenas para funções 'continuous' ou 'intervalS'):

- numero de sensores

sensorX (apenas para funções 'continuous' ou 'intervalS'):

- id dos sensores

**Exemplos:**

- parte 1 => 1|1|F|C|A|0|0|0  
parte 2 => 0|0|0|0|0|0|0|0  
parte 3 0|0|0|0|0|0|0|0
- parte 1 => 2|2|F|C|S|0|0|0  
parte 2 => 0|0|0|0|0|0|0|0  
parte 3 => 0|0|0|4|0|1|2|3
- intervalS – desde 2011/03/13 a 2011/03/15 das 9h15m10s às 23h305s para 4 sensores com os id's: 0, 1, 2, 3  
parte 1 => 10|3|F|I|S|20|11|3  
parte 2 => 13|20|11|3|15|9|15|10  
parte 3 => 23|30|5|4|0|1|2|3

**Resposta**

- Resposta igual à trama enviada (as 3 tramas)

## 2. Ler (enviada por 'mestre' de 1 em 1 segundo) - R

Pretende-se ler o valor dos Sensores Activos pela trama de definir função, mestre espera resposta com time out, se time out excedido ou resposta inválida pergunta novamente.

DLC pedido: 3

DLC resposta1: 8 (para a primeira)

DLC resposta2: 4 (para a segunda)

	Bits [7:0]							
<u>TRAMA</u>	1	2	3	4	5	6	7	8
<b>R</b>	id_nó	idMess	tipTrama					
<b>resp_R1</b>	id_nó	idMess	tipTrama	n_sensors	idSens0	valSens0	idSens1	valSens1
<b>resp_R2</b>	idSens2	valSens2	idSens3	valSens3				

Tipo\_Trama:

- R (neste caso): ler valor dos sensores activos - Read

**Exemplo:**

- 2|4|R

### 2.1. Resposta a Ler (enviado por 'escravo') - R

Resposta ao pedido do mestre para leitura dos seus valores.

Tipo\_Trama:

- R : responde com valor dos Sensores

N\_sensores :

- Numero de sensores activos

IdSensX :

- Id dos sensores

valSensX (até 255):

- Resultado da medição

**Exemplo:**

- parte 1 => 2|4|R|4|0|100|1|240  
parte 2 => 2|20|3|55

### 3. Ler estado de Actuador (enviada por 'mestre') - A

Pretende-se ler o estado de um actuador.

DLC pedido – 4

DLC resposta - 5

	Bits [7:0]							
<u>TRAMA</u>	1	2	3	4	5	6	7	8
<b>A</b>	id_nó	idMess	tipTrama	idAct				
<b>resp_A</b>	id_nó	idMess	tipTrama	idAct	State			

Tipo\_Trama:

- A: Ler estado de Actuador

**Exemplos:**

- Pedido: 1|6|A|0
- Resposta : 1|6|A|0|1

### 4. Activar/Desactivar/Mudar Estado Actuador - M

Pretende-se activar, desactivar ou mudar o estado de um actuador.

DLC - 5

	Bits [7:0]							
<u>TRAMA</u>	1	2	3	4	5	6	7	8
<b>M</b>	id_nó	idMess	tipTrama	idAct	State			
<b>resp_M</b>	igual							

**Exemplos:**

- Pedido: 1|7|A|0|0
- Resposta : 1|7|A|0|0

## 5. Acertar Date e Hora (enviado por 'mestre central') - C

Pretende-se acertar a hora de todos os elementos (mestres e escravos) existentes no sistema, mestre espera resposta com time out, se time out excedido ou resposta inválida envia novamente.

DLC – 8 (parte 1)

DLC – 2 (parte 2)

	Bits [7:0]							
<b>TRAMA</b>	1	2	3	4	5	6	7	8
<b>C1</b>	idGlobal	idMess	tipTrama	ano1	ano2	mês	dia	hora
<b>C2</b>	min	seg						
<b>resp_C1/2</b>	igual							

ID\_Global

- 99 – Indica ao mestre que a mensagem é para ele e não para os seus nós

Tipo\_Trama:

- C: acertar data e hora

Dados:

1. Ano1 – 20
2. Ano2 – 11
3. Dia – ex.22
4. Hora – ex.18
5. Minutos – ex. 19
6. Segundos – ex . 14

**Exemplo:**

2011/04/01 – 12h20m30s

- parte 1 => 10|9|C|20|11|4|1|12  
parte 2 => 20|30