Redes de Comunicação em Ambientes Industriais
Aula 11

Luís Almeida lda@det.ua.pt
José Alberto Fonseca jaf@det.ua.pt
Pedro Fonseca pf@det.ua.pt
Paulo Pedreiras pedreiras@det.ua.pt

Electronic Systems Lab-IEETA / DET
Universidade de Aveiro
Aveiro, Portugal

In the previous episode ... (1)

PROFIBUS

- ✓ 80's by **Siemens, aims Process control** and **factory automation**
- ✓ Standards DIN/CENELEC/IEC
- ✓ **Broadcast** serial **bus**
  - ✓ asynchronous transmission based on UARTs,
  - ✓ transmission rates up to **12 Mbit/s**
  - ✓ Max. length: **200m @ 1.5Mbit/s**, **1.2km @ 93.75kbit/s**. Extendable by repeaters
  - ✓ Max. number of nodes **127** (32 masters)

In the previous episode ... (2)

- Two main application profiles:
  - **PROFIBUS / FMS** - Fieldbus Message Specification
  - **PROFIBUS / DP** - Decentralised Peripherals
- **Hybrid** bus access control
  - **Token-passing** among masters, **Master-Slave** in each individual data transactions
- DLL services
  - SDA – **Send** Data with **Acknowledge**
  - SDN – **Send** Data with **No acknowledge**
  - SRD – **Send** and **Request** Data
  - CSRD – **Cyclic Send** and **Request** Data

In the previous episode ... (3)

TTP/C

- ✓ 1990's in the Technical University of Vienna
- ✓ **Safety-critical** applications (nodes integrated in **fault-tolerant units** (FTUs), interconnected by a **replicated bus)**
- ✓ Prompt **error detection** and **consistency checks, membership** and **clock synchronization** services
- ✓ **Multi-master**, **broadcast**, serial **bus**
- ✓ Several tx rates (e.g. **0.5Mb/s** with **MFM to 25Mb/s** on Ethernet PHY)
- ✓ Up to **64** nodes, CNI based on **Dual-port RAM**
- ✓ **TDMA** access scheme with **one slot** allocated **per node** and **per round**

In the previous episode ... (4)

- In **each slot** nodes may send **one frame** (up to 240 bytes). Each frame may contain **several messages.**
- All message transmission instants are stored in a **distributed static table (MEDL).** The messages cycle may span over several TDMA rounds **(Cluster Cycle).**
- Frames:
  - **I-Frame**: Protocol information frame (carries the C-state / Controller state information: clock, MEDL position, membership view)
  - **N-Frame**: Normal data frame
- TTP/C is **table based** (schedulability is implicitly verified when **building the schedule table**).
- An important aspect is that different nodes access the bus in **exclusive slots** and thus, **do not interfere**.

CAN
Controller Area Network

www.can-cia.de

CAN

- ✓ Created by **Bosch**, GmbH, for use in the **automotive industry**.
- ✓ Version 2.0 released in **1991**.
- ✓ ISO Standards 11519 (94) e 11898 (95)
- ✓ Expanded to process control, manufacturing automation and embedded application domains
- ✓ Defines **physical** and **data link** layers
- ✓ Many application layers (CANOpen, DeviceNet, …)

CAN

- **Multi-master**, **broadcast**, serial **bus**
- Transmission rate from **5 Kbit/s** to **1 Mbit/s**
- On transmission, nodes synchronize on bit level (requires bus spatial coherency on bit basis)
- Length depends on tx rate (aprox. 40m @ 1Mbit/s, 1000 @ 50Kbit/s)
- Max. number of nodes depends on bus transceivers (32, 64, 128…)

CAN

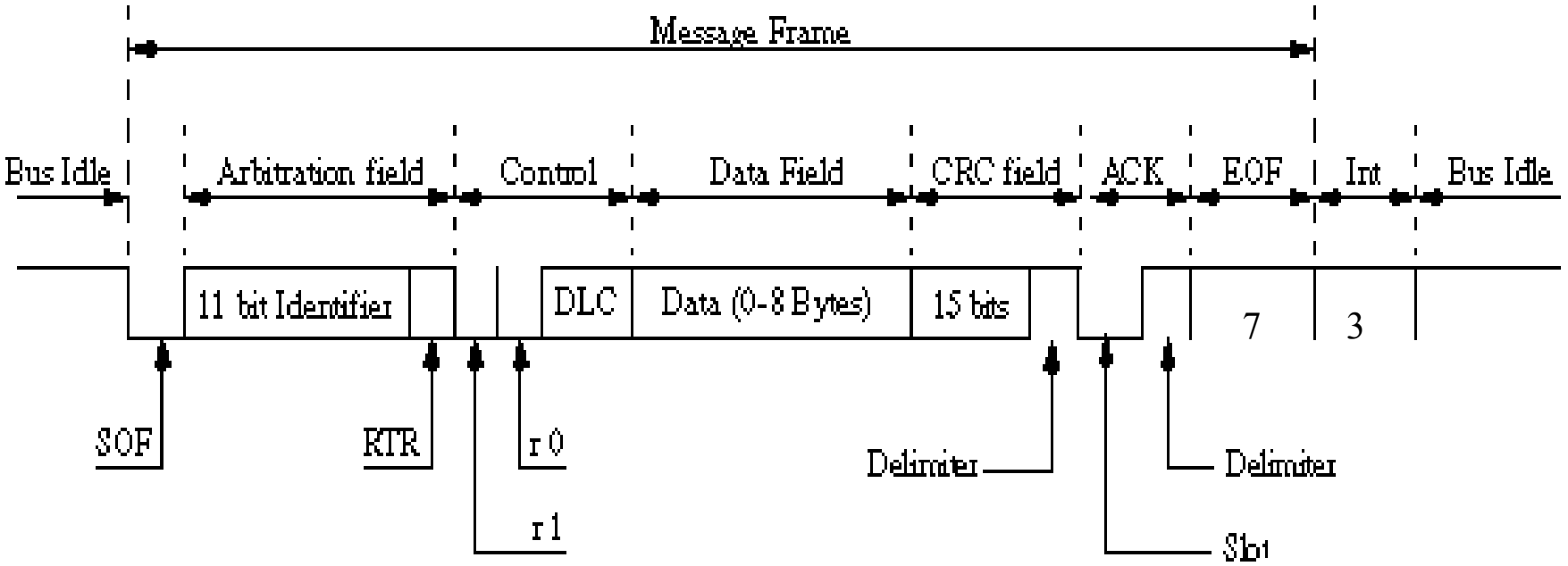
- ✓ Bit encoding using **NRZ**
- ✓ Tx/Rx synchronization using **bit stuffing**
- ✓ Frame error detection using **15 bit CRC**
- ✓ Error confinement (bus off after N errors)
- ✓ Error recovery based on automatic retransmission
- ✓ Transmission in differential voltage mode

CAN

- ✓ **Asynchronous** bus **access**
- ✓ Data **payload** between **0 and 8 bytes**
- ✓ **Source-addressing** (message identifiers) with 11 bits in version A and 29 bits in version B
- ✓ **Non-destructive arbitration** based on message identifiers (establish priority)
- ✓ Bit-wise deterministic collision resolution **CSMA/BA** (NBA?,CA?,DCR?)

CAN

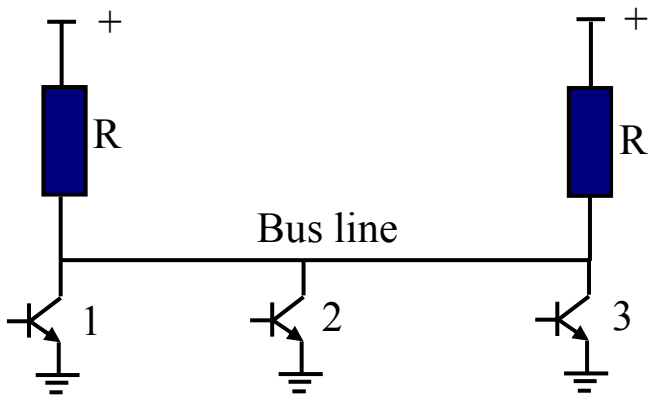
- ✓ **CAN 2.0A message frame**



SOF: start of frame
RTR: Remote transmission request
r0,r1: reserved bits
ACK: acknowledgement field
EOF: end of frame
Int: intermission gap

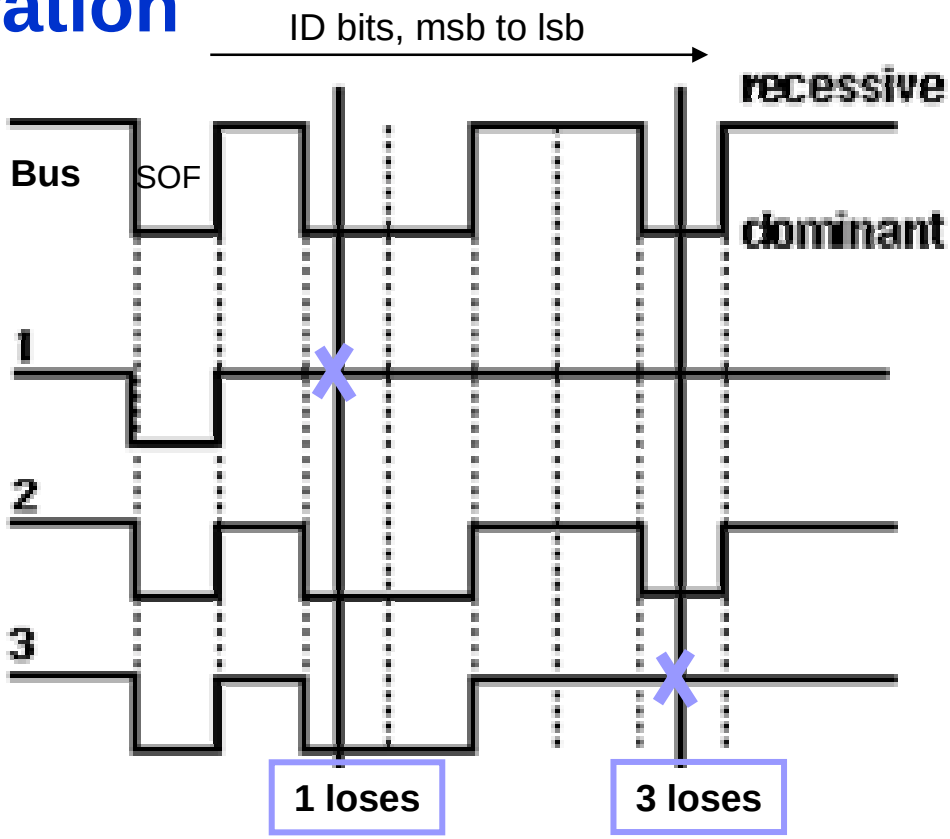
CAN

- **Bit-wise deterministic collision resolution**
  - **Wired-AND** implemented with bus drivers
    - '**0**' – Dominant level
    - '**1**' – Recessive level



CAN

- **Prioritized arbitration**
  - All nodes transmit and listen every bit
  - If different, then lost arbitration and backoff



CAN

- **Schedulability analysis**
  - Nodes ready to transmit, **wait** for any **on-going** transmission (**blocking**) as well as for the transmission of all **pending higher priority** messages (**interference**)
  - From a traffic scheduling point of view, CAN implements an **on-line static priorities-based policy**.

CAN

- **Schedulability analysis**
  - The messages' **transmission times** vary with the message contents because of **bit stuffing**
  - However, it can be **upper bounded** by

$$C = 13 + 34 + databits + \left\lfloor \frac{34 + databits - 1}{4} \right\rfloor$$

Data bits	0	8	16	24	32	40	48	56	64
Ci (bits)	55	65	75	85	95	105	115	125	135
Eff (%)	0	12	21	28	34	38	42	45	47

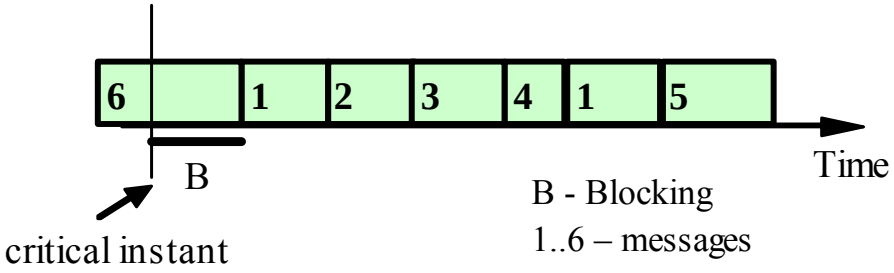
CAN

- **Schedulability analysis**
  - Use common analysis but account for blocking due to non preemption (B) and release jitter (J)

(τ = bit time)

$$\sum_{1}^{N} \frac{C_i}{T_i} + \max_{1..N}\left(\frac{B_i + J_i + \tau}{T_i}\right) < N(2^{1/N} - 1)$$

$$B_i = \max_{j \text{ in } lp(i)} (C_j)$$



CAN

- **Schedulability analysis**
  - For response time analysis with fixed priorities
    - Considering precise periodic triggering

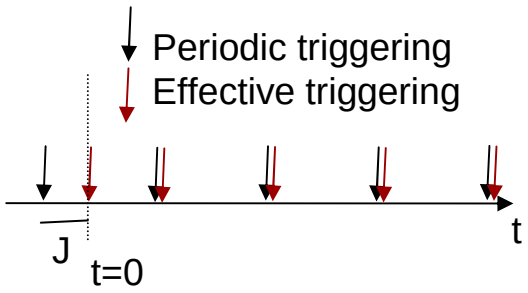
$$Rwc_i = I_i + C_i$$

$$I_i = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{I_i + \tau}{T_j} \right\rceil * C_j$$

    - Considering release jitter of each message due to internal queuing (J_i)

$$Rwc_i = J_i + I_i + C_i$$

$$I_i = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{J_j + I_i + \tau}{T_j} \right\rceil * C_j$$



CAN

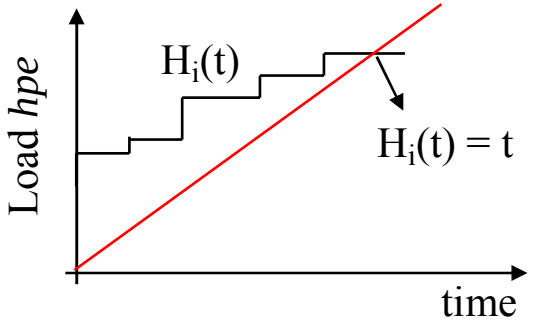
- **Schedulability analysis**
  - The previous equations for I_i can be solved using the usual iterative process (case without release jitter)

$$I_i(n+1) = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{I_i(n) + \tau}{T_j} \right\rceil * C_j$$

$$\text{until} \quad I_i(n+1) = I_i(n) \quad \text{or} \quad I_i(n+1) > D_i$$

$$\text{with} \quad I_i(0) = B_i + \sum_{j \text{ in } hp(i)} C_j$$

$$H_i(t) = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{t + \tau}{T_j} \right\rceil * C_j + C_i$$



CAN: Basic vs. Full CAN

- ✓ Commercial designations, relate with operational features
- ✓ Mainly associated with message filtering and
- ✓ Existance of multiple transmission buffers

BasicCAN

- ✓ Partial ID filtering (e.g. 8 bits mask)
- ✓ Complemented with application-level decision of acceptance
- ✓ Single transmission buffer

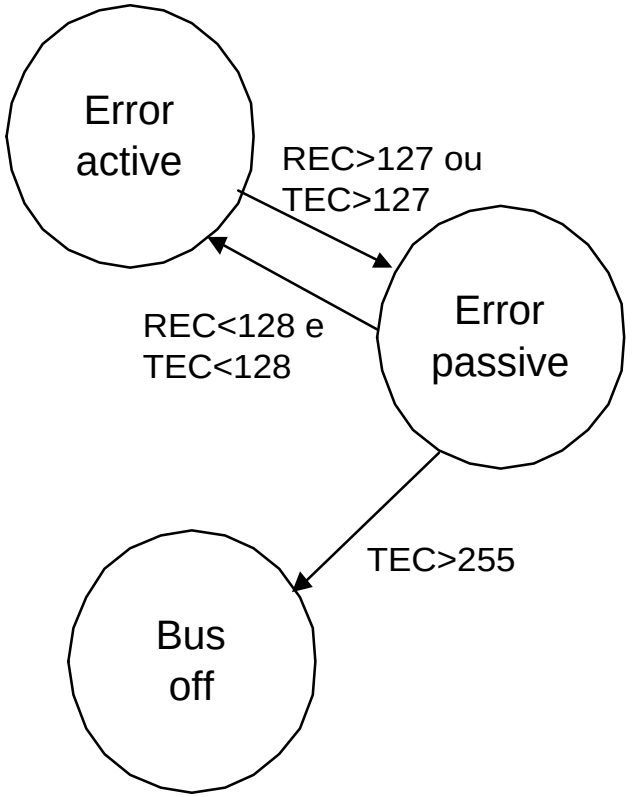
FullCAN

- ✓ Full ID filtering
- ✓ Association of data buffers to IDs
- ✓ Automated processing of RTR
  - ✓ The controller automatically sends replies to RTR frames if it has an associated buffer
- ✓ Multiple transmission buffers

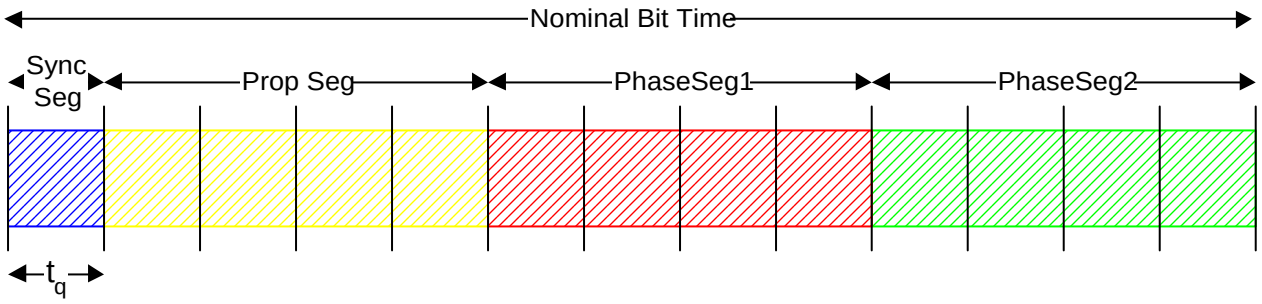
CAN: Error confinement

- CAN controllers keep 2 counters:
  - REC: Receive Error Count
  - TEC: Transmit Error Count
- Counters are:
  - Incremented upon errors
  - Decremented upon successful transactions

    (but with **different** factors)

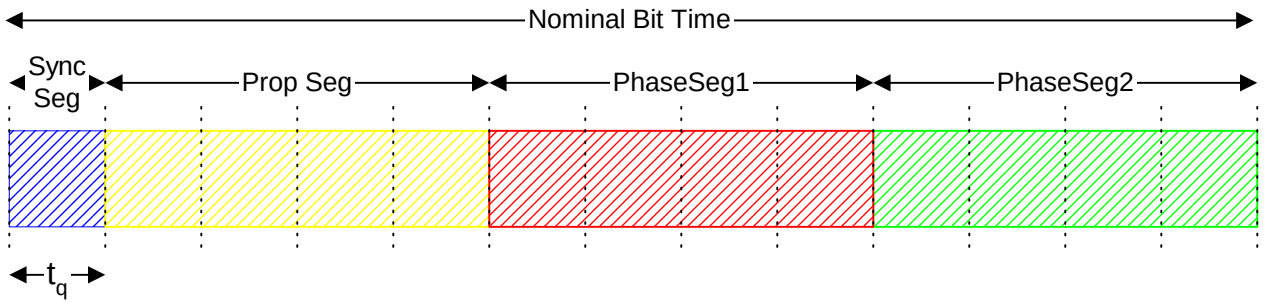


CAN: Bit timing



- CAN divides the bit in 4 segments:
  - SyncSeg
  - ProgSeg
  - PhaseSeg1
  - PhaseSeg2
- All segments are integer multiples of the *time quantum* (t_q)
- Consequently the bit time is an integer multiple of t_q

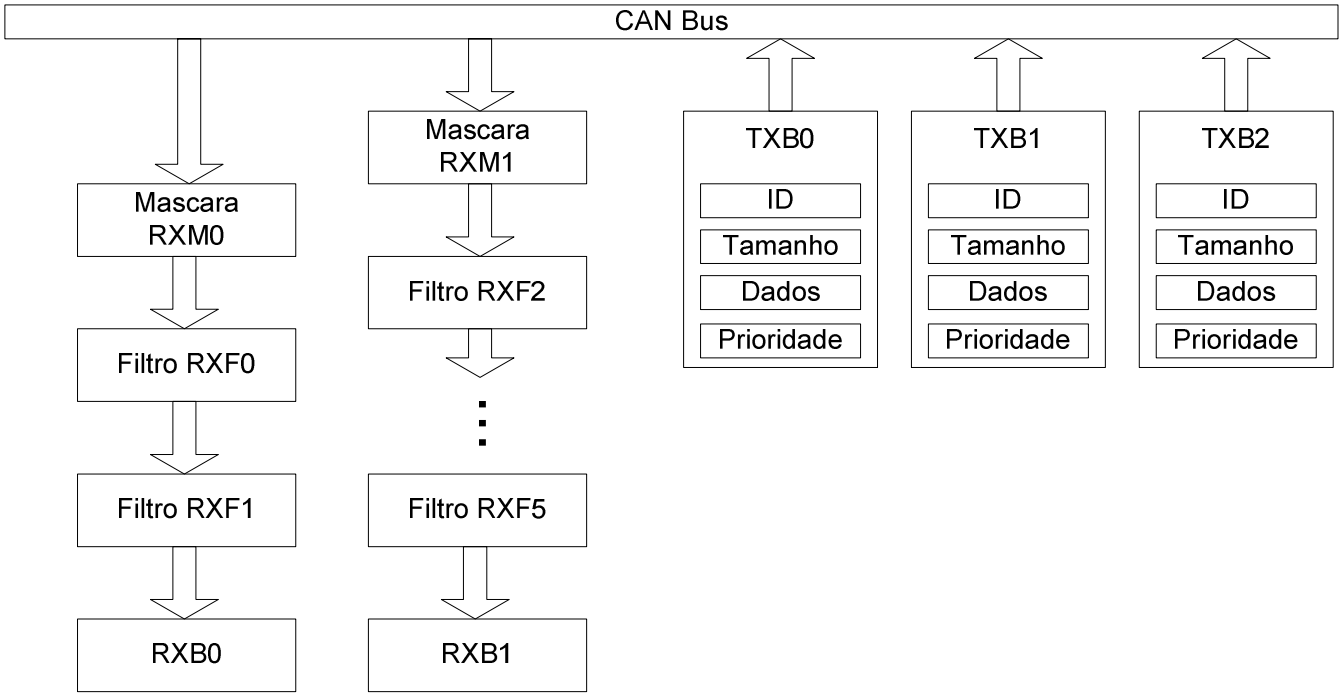
CAN: Bit timing



- ✓ SyncSeg
  - ✓ Accounts for clock error phase
- ✓ PropSeg
  - ✓ Accounts for the propagation delay (bus and electronic circuits)
- ✓ PhaseSeg1, PhaseSeg2
  - ✓ Accounts for clock skew
- ✓ Bit value is sampled between PhaseSeg1 and PhaseSeg2
- ✓ There may be constraints between the values of the diverse segments

O Controlador PIC18Fxx8

Estrutura geral (PIC18FXX8)



Inicialização do controlador

- Entrar em modo de configuração
  - Registo **CANCON REQOP2=1**
  - Esperar que o controlador no modo de config. (registo **CANSTAT**)
- Configurar Bit Rate
  - registo **BRGCON1, BRGCON2, BRGCON3**
- Configurar mascaras:
  - Registos **RXMx** (x=0,1 nº da mascara)
- Configurar filtros:
  - Registos **RXFx** (x = 0,1,2,3,4,5 nº do filtro)
- Entrar no modo pretendido
  - **Normal, loopback, listen only, disable**
- Configurar entrada e saída
  - (**RB3, RB2** – Tx_CAN, Rx_CAN)

Envio de uma mensagem

- Preencher os registos da mensagem
  - Identifação (**TXBnSIDH**, **TXBnSIDL** n=0,1,2 Buffer de transmissão)
  - Registo TXBnSIDL (n=0,1,2) colocar bit **EXIE=0** para Mensagem com identifacação standard
  - Tamanho da mensagem (**TXBnDLC**, n=0,1,2 Buffer de transmissão)
  - Mensagem **TXBnDx**, n=0,1,2 Buffer de transmissão, x=0…7 bytes)
  - Prioridade relativa das mensagens a enviar (Registo **TXBnCON**)
- Pedir o envio da mensagem
  - Bit **TXREQ = 1** do registo **TXBnCON**, n=0,1,2 Buffer de transmissão

Recepção de uma mensagem

- Verificação de qual o buffer que contém a mensagem
  - Bit **RXFUL** do registo **RXBnCON** (n=0,1 buffer de recepção)
  - Ou, por interrupção (implícito)
- Aceder aos campos da mensagem
  - Identificador (**RXBnSIDL** e **RXBnSIDH**)
  - Tamanho (**RXBnDLC**, n=0,1 buffer de recepção)
  - Conteúdo (**RXBnDx**, n=0,1 buffer de recepção, x=0…7 bytes)
- Libertar buffer de recepção
  - bit **RXFUL = 0**

Summary:

CAN

- ✓ Created by **Bosch**, Version 2.0 released in **1991**.
- ✓ Expanded to process control, manufacturing automation and embedded application domains
- ✓ Defines **physical** and **data link** layers only
- ✓ **Multi-master**, **broadcast**, serial **bus**
- ✓ Transmission rate from **5 Kbit/s** to **1 Mbit/s**
- ✓ Length depends on tx rate
  (aprox. 40m @ 1Mbit/s, 1000 @ 50Kbit/s)
- ✓ Maximum number of nodes depends on bus transceivers (32, 64, 128…)

Summary:

- ✓ Bit encoding using **NRZ**
- ✓ Tx/Rx synchronization using **bit stuffing**
- ✓ Data **payload** between **0 and 8 bytes**
- ✓ **Source-addressing** **(**11/29 bits in Vers. A/B resp)
- ✓ **CSMA with Non-destructive arbitration** based on msg. IDs
- ✓ Analysis:
  - ✓ **Common analysis, acounting for the non-preempmtion**
    - ✓ **Utilization**

$$\sum_{1}^{N} \frac{C_i}{T_i} + \max_{1..N}\left(\frac{B_i + J_i + \tau}{T_i}\right) < N(2^{1/N} - 1)$$

    - ✓ **Response time**

$$Rwc_i = I_i + C_i$$

$$I_i = B_i + \sum_{j \text{ in } hp(i)} \left\lceil \frac{J_i + I_i + \tau}{T_j} \right\rceil * C_j$$