Intro to bxCAN

1) We have a dual CAN in STM32 the first one is the Master CAN and the second is the Slave CAN.
2) 3 Transmit mailboxes with configurable transmit priority
3) Both master and slave are the same except for the master who is responsible for “Filter Configuration”. For the slave, we also set up this filter configuration as it will receive according to it.
4) Each has a scheduler that arranges the 3 mailboxes with respect to priority and sends or receives them in this order.

bxCAN operating modes:

We select the operating mode using 2 bits : SLEEP & INRQ (initialization request) and read the value of 2 status bits : SLAK (sleep acknowledge if it’s 1, we are in the sleep mode) and INAK (initialization acknowledge if it’s 1, we are in the initialization mode).

1) **Sleep mode:** default mode after RESET. Low Power mode when bxCAN is not used. SLAK = 1, INAK = 0. To be in this mode, we Reset the INRQ bit (INRQ = 0) and SET the SLEEP bit (SLEEP = 1) and we wait for the ack SLAK to be 1.
2) **Initialization mode:** Configuration mode. SLAK = 0, INAK = 1. To be in this mode, we SET the INRQ bit (INRQ = 1) and Reset the SLEEP bit (SLEEP = 0) and we wait for the ack INAK to be 1.
3) **Normal mode:** transmitting & receiving mode. SLAK = 0, INAK = 0. To be in this mode, we Reset the INRQ bit (INRQ = 0) and Reset the SLEEP bit (SLEEP = 0) and we wait for synchronization. Before entering Normal mode, the bxCAN must synchronize with the CAN bus, so it waits until the bus is idle (this means 11 consecutive recessive bits have been read on pin CANRX).
   In Normal mode, we can select to be in Test mode or operation mode.

bxCAN Test Modes:

Selected using SILM bit & LBKM bit in CAN_BTR register if we are already in the Normal mode. if SILM = 0, LBKM = 0.

1) **Silent mode:** Receive only mode. This node only sends 1’s (only recessive bits) on the CAN bus as if this node isn’t seen on the bus. SILM = 1, LBKM = 0.
2) **Loopback mode:** the bxCAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) in a Receive mailbox. This mode is used when we don’t have a CAN bus and want to test the bxCAN. SILM = 0, LBKM = 1.
3) **Loopback combined with silent mode:** This mode can be used for a “Hot Self Test”, meaning the bxCAN can be tested like in LoopBack mode but without affecting a running CAN system connected to the CANTX and CANRX pins. In this mode, the CANRX pin is disconnected from the bxCAN and the CANTX pin is held recessive. SILM = 1, LBKM = 1.

bxCAN Transmission Handling:

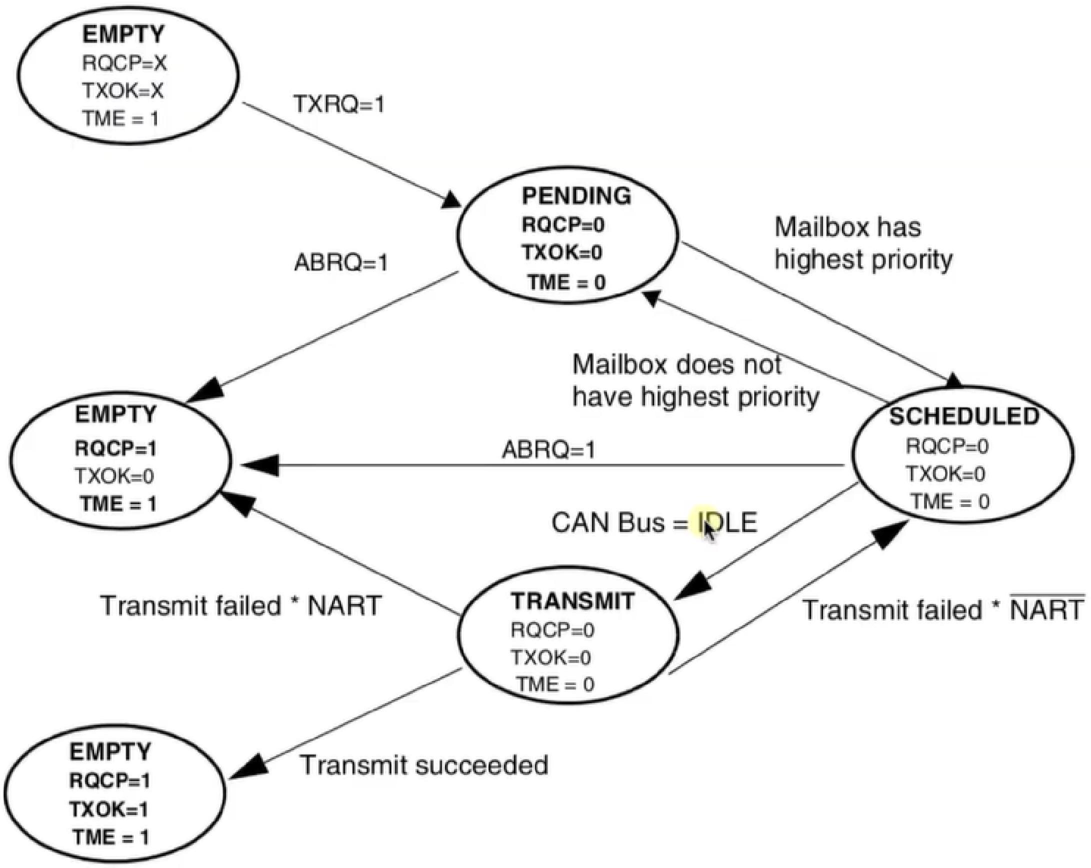
1) Check for an empty mailbox by reading the values of TME2, TME1 and TME0 in the CAN_TSR and choose one empty mailbox.
2) Then set up the identifier in the CAN_TIxR, the Data length code (No. of bytes in the message from 0 to 8 bytes) in the CAN_TDTxR and the Data in the CAN_TDLxR & CAN_TDHxR.
3) After setting up the identifier, DLC and data, Request the transmission by setting TXRQ in CAN_TIxR to 1.

4) The Mailbox enters the pending state and waits to become the highest priority then it's scheduled for transmission.
5) The 2 bits : RQCP & TXOK in CAN_TSR are indicators (the hardware sets them to 1) of successful transmission.
6) If the transmission fails, the ALST in CAN_TSR is set to 1 in case of Arbitration Lost (Bus is lost to another Master with dominant bit) and the TERR is set to 1 in case of transmission error detection.
7) For both cases, the Automatic Retransmission Mode is by default ON and the frame is sent again. We can turn this mode off by setting the NART bit in the CAN_MCR.

bxCAN Transmission Priority:

- The Normal Priority Mode:
    1) Identifier Priority: The Message with the lowest identifier value has the highest priority because of the Arbitration bus (0 is the dominant bit). If the identifier values are equal, the mailbox priority is taken under consideration.
    2) Mailbox Priority: The lower mailbox number is scheduled first.
- FIFO Priority (First In First Out): The first to request transmission is the first to be scheduled. This priority mode can be activated by the bit TXFP in the CAN_MCR.

bxCAN Summary:



bxCAN Registers:

1) CAN_MCR (Master Control Register):
    - Bit 0 - INRQ (Initialization Request): clear to 0 to switch to normal mode and wait 11 consecutive recessive bits to be able to transmit and receive and the

(INAK bit in CAN_MSR register is cleared to 0 by hardware as an indication that we're ready).
To switch to Initialization mode, we set this to 1 by software. The CAN hardware waits until the current CAN activity (Transmit or receive) is done and then enters the initialization mode and hardware sets the INAK bit to 1 as an indicator.

- Bit 1 - SLEEP: Set to 1 to enter the sleep mode. It will enter this mode as soon as the current CAN activity (Transmit or receive) is done. It's cleared by software to exit the sleep mode.
  When the AUWU is set to 1 (Automatic Wake Up mode is ON), the SLEEP bit is cleared by hardware when a SOF bit (Start Of Frame bit) is detected on pin CANRX.
- Bit 5 - AWUM : When we set it to 1, the Sleep mode is left automatically by hardware on CAN message detection. The SLEEP bit of the CAN_MCR register and the SLAK bit of the CAN_MSR register are cleared by hardware.
- Bit 2 - TXFP (Transmit FIFO priority): It is controlled by SW. If it's
  0 -> Identifier & mailbox number priority.
  1 -> FIFO priority (The request order).
- Bit 4 - NART (No Automatic Retransmit) : If it's 0, the CAN HW will automatically retransmit the Message until it's transmitted successfully.
  If it's 1, the message will be transmitted once independently of the transmission result.

2) CAN_TIXR (Tx Mailbox Identifier Register):
   The Standard identifier is 11 bits (from 31 to 21) & the total extended identifier is 29 bits (from 31 to 3).
   - Bits 31:21 - STID: Represents the standard identifier or the MSBs of the extended identifier depending on the IDE bit value (If we use the Standard or the Extended identifier CAN).
   - Bits 20:3 - EXID: The LSBs of the extended identifier.
   - Bit 2 - IDE (Identifier Extension) : 0 -> Standard identifier, 1 -> Extended identifier.
   - Bit 0 - TXRQ (Transmission Mailbox Request): Set by SW to request the transmission (then mailbox enters the pending state) & cleared by HW to indicate that the mailbox becomes empty.

3) CAN_TDTXR (Mailbox Data Length Control and Time Stamp Register):
   - Bit 3:0 - DLC (Data Length Code): define the No. of bytes of a data frame. A message can contain from 0 to 8 bytes depending on the value of DLC.

4) CAN_TDLXR (Mailbox Data Low Register): A message can contain from 0 to 8 bytes starting from byte0, so the 1st four bytes are stored in this register (from DATA0 to DATA3 bytes).

5) CAN_TDHXR (Mailbox Data High Register): The last four bytes of data are stored in this register (from DATA4 to DATA7).

6) CAN_TSR (Transmit Status Register):

- Bit 28:26 - TME2:0 : Each bit is set by HW to indicate that no transmit request is pending for its corresponding mailbox.
- Bit 0,8,16 - RQCP0:2: Request completed for mailbox 0:2 is set by HW to indicate that the last request has been performed (transmit or abort). It’s cleared by SW by writing 1 or by HW on transmit request (for example when TXRQ0 set in TI0R register). And by clearing it all other flags are also cleared (as we are in another transmit request).
- Bit 1,9,17 - TXOK0:2: The HW updates each bit after each transmission attempt. If 0 -> the previous transmission failed, 1 -> the previous transmission was successful.
- Bit 2,10,18 - ALST0:2: is set to 1 if the previous Tx request failed due to Arbitration Lost.
- Bit 3,11,19 - TERR0:2: is set to 1 if the previous Tx request failed due to an error.

7) CAN_MSR (Master Status Register):
    - Bit 0 - INAK (Initialization acknowledge): This bit is set by hardware and indicates to the software that the CAN hardware is now in initialization mode. It is cleared by hardware when the CAN hardware has left the initialization mode when it’s synchronized on the CAN bus (to monitor a sequence of 11 consecutive recessive bits on the CAN RX signal).
    - Bit 1 - SLAK (Sleep acknowledge): This bit is set by hardware and indicates to the software that the CAN hardware is now in Sleep mode.
It is cleared by hardware when the CAN hardware has left Sleep mode.

8) CAN_BTR (Bit Timing Register):
    - Bit 31 - SILM (Silent Mode Debug): If it’s set to 1, the Silent Test Mode is enabled (we need to be in Normal Mode before).
    - Bit 30 - LBKM (Loopback Mode Debug): If it’s set to 1, the Loopback Test Mode is enabled (we need to be in Normal Mode before).

    We can enable the combined Mode by setting both SILM & LBKM to 1.