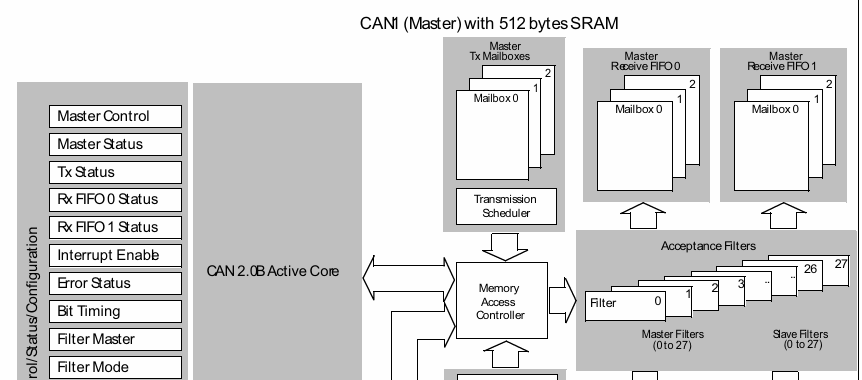
By: Abdelrahman Matarawy

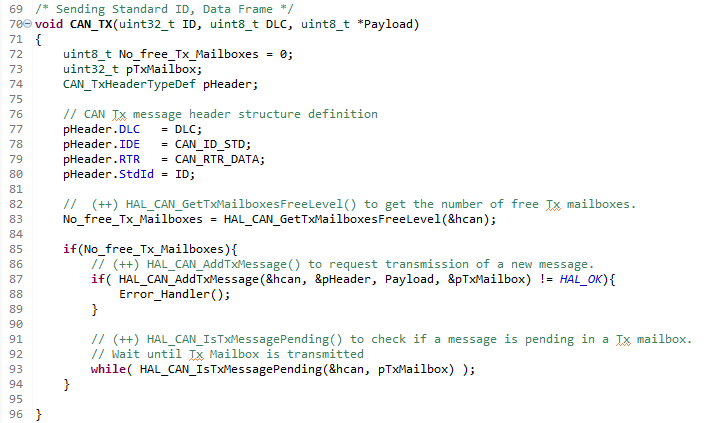
CAN Controller

CASE STUDY 1

* **Overview of CAN Controller:**
  + **Tx mailboxes:** 
    - Three transmit mailboxes are provided to the software for setting up messages. The transmission Scheduler decides which mailbox has to be transmitted first.
  + **Acceptance filters:** 
    - The bxCAN provides 14 scalable/configurable identifier filter banks for selecting the incoming messages the software needs and discarding the others.
  + **Receive FIFO:** 
    - Two receive FIFOs are used by hardware to store the incoming messages. Three complete messages can be stored in each FIFO. The FIFOs are managed completely by hardware.

****

* **CAN Transmitter Code:**

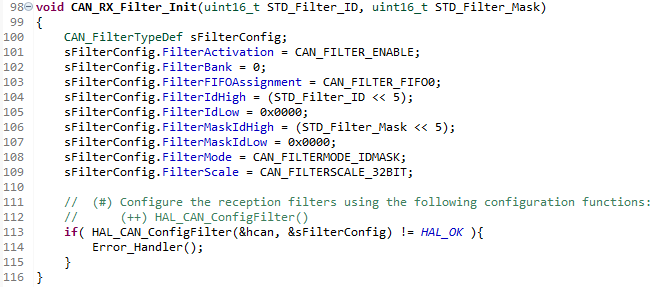


* **Receiver Filter:**

**A screenshot of a document

Description automatically generated**

* + I work For ID Mask as 1st Register Work for ID, while 2nd Work for Mask.
    - On Mask Register if I define bit as 1 so CAN Controller will Compare ID Register with ID Received from Tx.
  + Also work on Register as 32 Bits.
  + **Code**:

****

* **CAN Receiver Code:**

**A screen shot of a computer

Description automatically generated**

* **Main Code:**

**A diagram of a sleep system

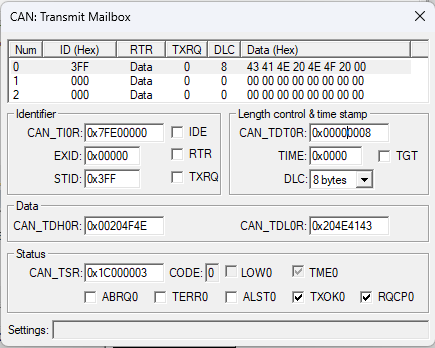
Description automatically generated**

* + 1st We Switch to Init Mode to define how CAN Controller Work.
  + 2nd We Switch to Normal mode to start Running Tx and Rx Transmission Process.
  + **Code:**

**A screenshot of a computer program

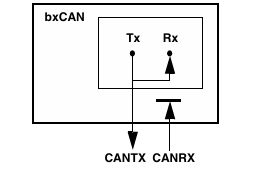
Description automatically generated**

* **Frame In Transmitter mailbox:**

****

* **Transmitting Data and Listen what I send using Loopback Mode: “**In this mode, the bxCAN performs internal feedback from its Tx output to its Rx input.”

A screenshot of a computer

Description automatically generated****