

**CAN Configuration for Jetson AGX Orin**

*Structured for RAG Readability*

**Hardware Specifications**

* **Number of CAN Controllers**: 2
* **Controller Base Addresses**:
  + mttcan@c310000 (CAN0)
  + mttcan@c320000 (CAN1)
* **Default Pin Configuration**: GPIO (requires manual configuration for CAN functionality)

**Pin Configuration (40-pin Header J30)**

* **CAN0**:
  + **DIN (Data In)**:
    - Pinmux Address: 0x0c303018
    - Value: 0x458
  + **DOUT (Data Out)**:
    - Pinmux Address: 0x0c303010
    - Value: 0x400
* **CAN1**:
  + **DIN (Data In)**:
    - Pinmux Address: 0x0c303008
    - Value: 0x458
  + **DOUT (Data Out)**:
    - Pinmux Address: 0x0c303000
    - Value: 0x400

**Kernel Configuration**

* **Device Tree Nodes** (Enabled by Default):

mttcan@c310000 { status = "okay"; };   
mttcan@c320000 { status = "okay"; };

* **Verify Status**:

cat /proc/device-tree/mttcan@c310000/status # Output: "okay"

**Key Features**

* **Bit Rate**: Configurable from **10 kbps to 1 Mbps** (base rate)
* **CAN FD Support**:
  + Max data bit rate: **15 Mbps** (hardware-dependent)
  + Use ip link set canX up type can dbitrate <value> fd on to enable
* **Transmission Delay Compensation (TDCR)**:
  + Adjust via sysfs for higher bit rates:

echo 0x600 > /sys/devices/c320000.mttcan/net/can1/tdcr

**Configuration Steps**

1. **Enable Pinmux**:

busybox devmem 0x0c303018 w 0x458 # CAN0\_DIN   
busybox devmem 0x0c303010 w 0x400 # CAN0\_DOUT

1. **Load Kernel Drivers**:

modprobe can   
modprobe can\_raw   
modprobe mttcan

1. **Bring Up Interfaces**:

ip link set can0 up type can bitrate 500000   
ip link set can1 up type can bitrate 500000

**Debugging**

* **Loopback Test**:

ip link set can0 type can bitrate 500000 loopback on   
candump can0 &   
cansend can0 123#abcdabcd

* **Check Clock Rates**:

cat /sys/kernel/debug/bpmp/debug/clk/can0/rate   
cat /sys/kernel/debug/bpmp/debug/clk/can0/pto\_counter

**Notes**:

* Transceiver requirements: **3.3V** compatible, supports ≥5 Mbps for CAN FD.
* For production, use config-by-hardware.py to persist pinmux changes.

This structure ensures compatibility with RAG systems while maintaining technical accuracy.

⁂