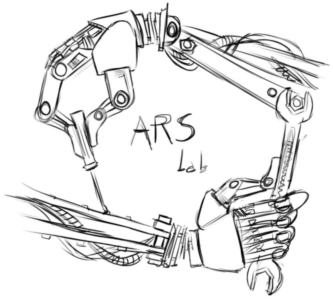
The Controller Area Network (CAN) Interface

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
santoro@dmi.unict.it

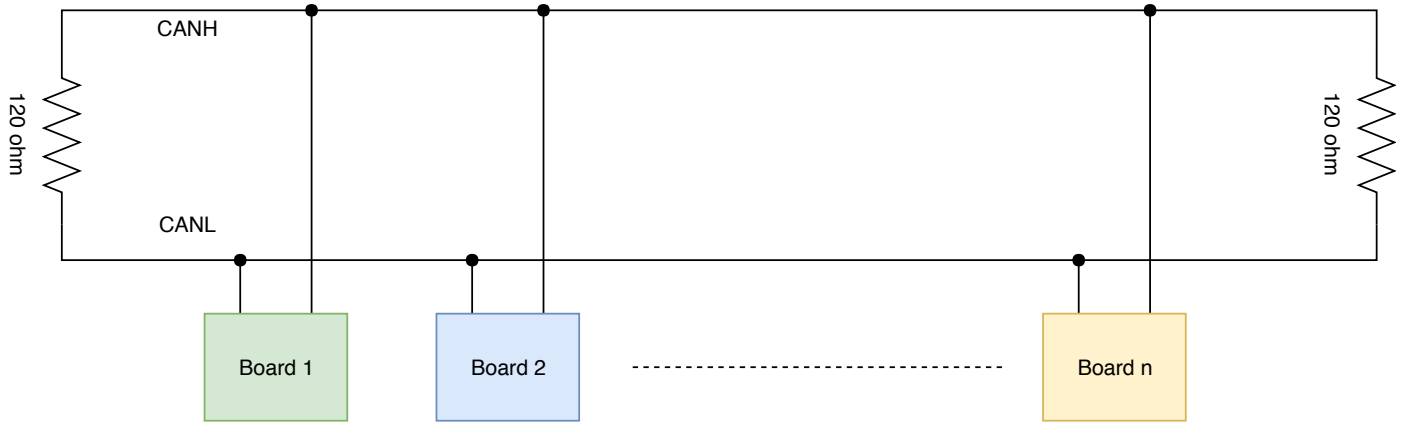


L.S.M. 1 Course

What is CAN?

- The **CAN—Controller Area Network** is a **communication network** designed to interconnect MCU-based boards using the *"computer network paradigm"*
- It has been introduced by Bosch to support communication in industrial automation environments
- It is widely used in robotics, industrial automation and (above all) transportation environments: cars and airplanes

CAN: Physical layer and Arbitration



- From the physical point of view, CAN uses a **twisted-pair bus** terminated, on both sides, by 120 Ω resistors
- Devices are attached to the bus by means of two signals called **`CANH`** (CAN high) and **`CANL`** (CAN low)
- All devices are “peers” and roles (e.g. master or slaves) do not exist
- Any device may decide to start to transmit in any moment, so an **arbitration policy** must be employed

CAN: Bus Arbitration

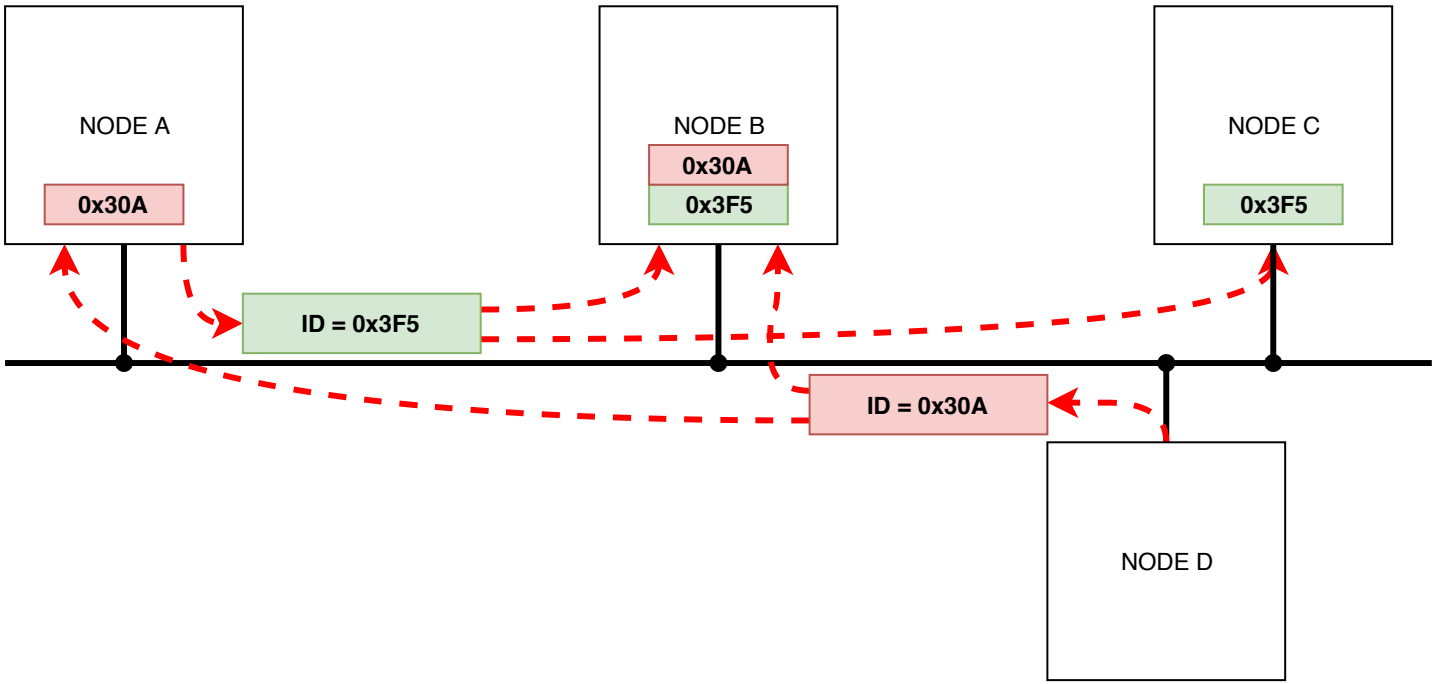
- Data is transmitted serially, one bit at time
- The maximum speed is defined by the standard as 1 *Mbps*
- Transmitted bits are:
  - **"1"**, called **recessive bit**
  - **"0"**, called **dominant bit**
- **Arbitration:**
  - When two devices start trasmitting simultaneously, the device that is sending a **dominant bit** wins!
  - The output circuit of the CAN interface has an electronics able to "promote" dominant bits
  - Each device is able to "listen to" what it is currently transmitting, so it can stop transmission if a recessive bit is "cancelled" by a dominant bit, thus avoiding collisions

CAN: Data Packet

- A CAN Packet is made of three main parts:
  - **Payload**, 8 bytes, application-dependent
  - **CAN-ID**, 11 bits, used to identify the data and its priority
  - **Control bits**, various bits for signalling
- The **CAN-ID**
  - It is a **tag** that identifies the data that is being transmitted
  - Addressing and transmission exploits the CAN-ID according to a **publisher-subscriber** paradigm

1. A node (*publisher*) sends its data using a certain CAN-ID; the packet is initially broadcasted to all other nodes
2. Each node interested in that CAN-ID (*subscriber*) "captures" the data packet and forwards it to the upper layers (software)

The Publisher-Subscriber Mechanism



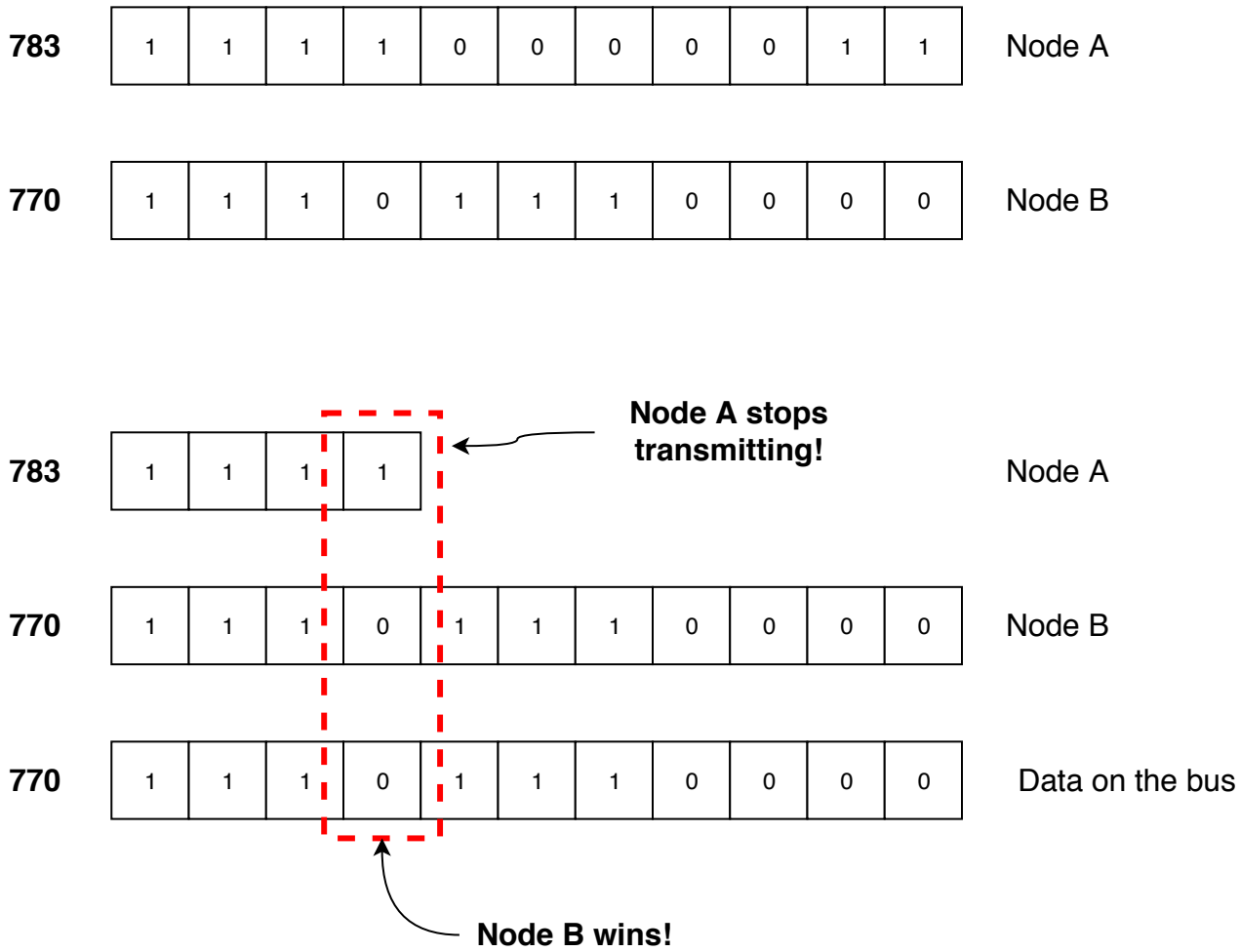
- A CAN node interested in a certain packet ID, “subscribes to it”
- The subscription means to program its local interface to catch that ID
- When a node sends a packet with that ID, the interface of all nodes receive it, but those not programmed for catching ignores the packet
- The interfaces programmed for catching process the packet and forward it to the software

CAN Packet

- **SOF:** Start-of-frame, (1 dominant bit)
- **ID:** CAN-ID of the packet (11 bits)
- **RTR:** Remote Transmit Request (1 bit)
- **IDE:** Extended Identifier (1 bit)
- **DLC:** Payload Data Length (4 bits)
- **Data:** Payload Data (0 to 64 bits)
- **CRC:** Cyclic-Redundancy-Check Code (15 bits)
- **ACK:** Acknowledge (1 bit)
- **EOF:** End-of-frame (7 recessive bits)

CAN Priority Mechanism

- The **CAN-ID** is exploited also to estabilish the **priority** of a message
- The priority mechanism exploits dominant bits that have priority over recessive bits



The Acknowledge Mechanism

- When a packet is transmitted, the sender puts a **recessive bit** in the ACK slot
- If during transmission, a node interface is catching the packet (since the software made a subscription to that ID), that interface puts a **dominant bit** in the **ACK slot**
- The sender can thus identify if at least one node has received the packet
- If no node acknowledges the packet, the sender tries to re-transmit it at most 127 times
- These operations are performed “on-the-fly” by the hardware, the software is never involved

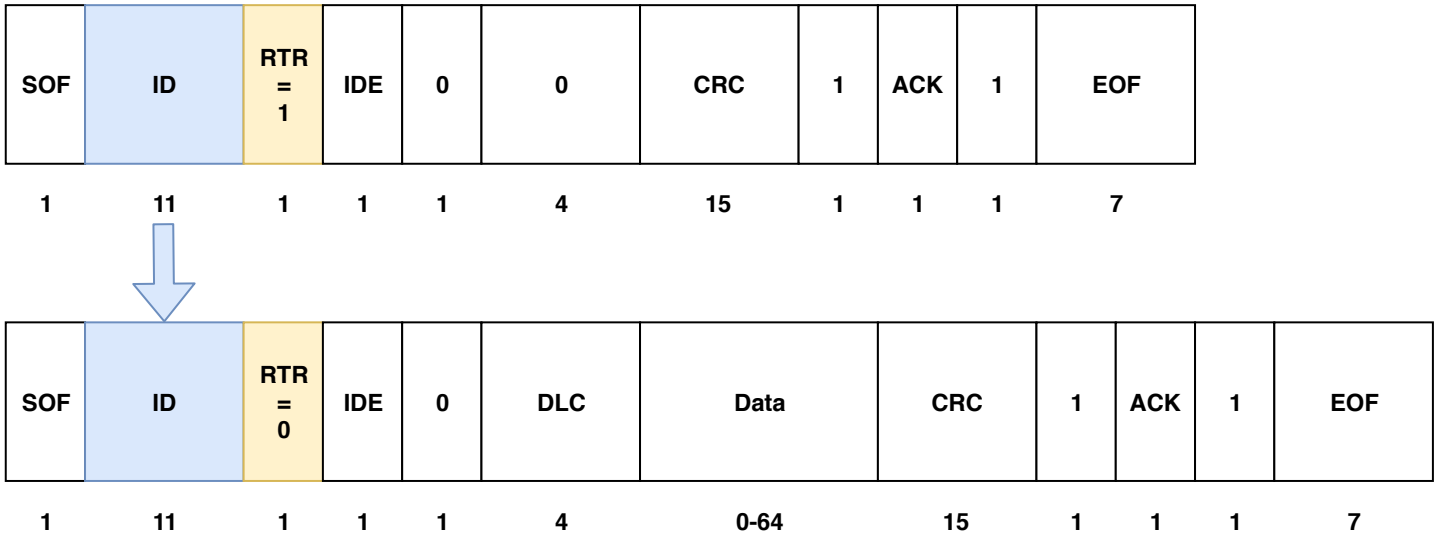
CAN Transaction Types

- **Simple (Push) Transactions:** data is transmitted (periodically) by a node (RTR = 0)

SOF	ID	RTR = 0	IDE	0	DLC	Data	CRC	1	ACK	1	EOF
1	11	1	1	1	4	0-64	15	1	1	1	7

CAN Transaction Types

- **Data Request Transactions:**
  - A node requests a data ID by sending a frame with that ID and **RTR = 1**
  - The node that can send that ID replies with a data packet with **RTR = 0**



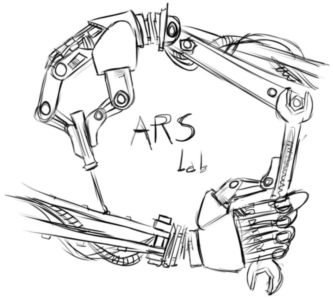
Format of Data Payload

- The structure of **Payload Data (8 bytes)** is application-dependent and free to be defined by the developer
- However, there are some specifications that define formats for certain kind of applications:
  - **CANopen:** industrial automation
  - **DeviceNet:** industrial automation
  - **CANaerospace:** avionics
  - **UAVCAN:** avionics and robotics
  - **...**

The Controller Area Network (CAN) Interface

Corrado Santoro

ARSLAB - Autonomous and Robotic Systems Laboratory
Dipartimento di Matematica e Informatica - Università di Catania, Italy
santoro@dmi.unict.it



L.S.M. 1 Course