Introduction

The Devices Communication Protocol (DCP) is a bidirectional, single-wire communication protocol developed at LMI-UFPB. It's designed to support a multi-master architecture for device communication. The primary goal of DCP is to enable reliable data exchange between devices without requiring complex arbitration mechanisms typically found in other protocols. It is engineered to reduce wiring complexity while ensuring robust, low-power, and scalable communication. DCP is particularly suited for applications such as automotive telematics, industrial sensor networks, and modular control systems, where both reliability and minimal physical wiring are crucial.

DCP Purpose

DCP provides an efficient and cost-effective solution for inter-device communication by combining clock and data in a single time-modulated signal. It incorporates robust collision detection and avoidance methods based on the CSMA/CD paradigm with binary countdown arbitration, which effectively manages simultaneous transmissions. Additionally, DCP enables hot-join capabilities so that devices can dynamically enter or exit the bus without interrupting ongoing communications. The protocol also supports multiple speed modes (Slow, Fast I, Fast II, and High Speed) to accommodate diverse system requirements and electrical conditions.

Protocol Overview and the Role of Δ

At the heart of DCP is the concept of a base time unit, Δ, which serves as the fundamental timing reference for defining the duration of all pulses and intervals on the bus. The value of Δ is determined by system constraints—such as bus capacitance, pull-up resistor values, and the desired bit rate—and is used to scale the pulse durations for both data and control signals. By adjusting Δ, the protocol can be tuned for either slower transmissions (using a larger Δ) to tolerate noise and capacitance or for high-speed communications (using a smaller Δ) that require tighter timing control.

Every time-related parameter in DCP—including bit pulse lengths, synchronization intervals, rise and fall times, and overall cycle duration—is expressed as a multiple of Δ, providing a scalable and adaptable communication framework.

Protocol Features

Bus Speeds

DCP defines four operational classes based on Δ, each suited for different application scenarios. The table below summarizes each mode, its recommended Δ time unit, and typical device clock frequencies:

Mode	Δ (Time Unit)	Description
Slow	$\Delta = 20\,\mu s$	Devices operating at 4 MHz
Fast I	$\Delta = 4\,\mu s$	Devices operating at 20 MHz
Fast II	$\Delta = 2.5\,\mu s$	Devices operating at 32 MHz
High Speed	$\Delta = 1.25\,\mu s$	Devices operating at 64 MHz

Table 1: DCP Speed Classes and Corresponding Δ Values

Bus Communication

DCP transmits data using single-wire communication that embeds clock and data signals via time modulation. At the physical layer, the communication occurs on a single conductor that is pulled high by a resistor connected to the supply voltage (V_{DD}). Devices actively drive the bus low to assert a logical "0," while a passive pull-up network maintains a logical "1." The voltage levels are defined such that the high-level voltage, V_h, is at least 70% of V_{DD}, and the low-level voltage, V_l, remains below 30% of V_{DD}.

Signal encoding in DCP relies entirely on the timing reference Δ. A logical "0" is transmitted as a high-level pulse lasting Δ followed by a low-level pulse also lasting Δ, whereas a logical "1" uses a high-level pulse of duration 2Δ followed by a low-level pulse of duration Δ.

In addition, synchronization signals are defined to mark the start of a transmission: the Start Sync signal, which lasts 25Δ for transmitting controllers and 50Δ for target devices, initiates the transmission and synchronizes the receiver's clock; and the Start Bit, composed of a high-level pulse of 7.5Δ immediately followed by a low-level pulse of 7.5Δ, signals the transition to the data payload.

- **Start Sync:** Initiates a transmission and synchronizes the receiver's clock. For the transmitting controller, it is a low-level pulse lasting 25Δ, while target devices use 50Δ to ensure proper clock recovery.
- **Start Bit:** Marks the end of synchronization and the beginning of data payload transmission. It consists of a high-level pulse of 7.5Δ immediately followed by a low-level pulse of 7.5Δ.

The bit stream is framed within a clearly defined cycle time that includes a rise time, a brief settling period, and a fall time, thereby eliminating the need for a separate clock line.

Hot-Join Mechanism

The hot-join mechanism in DCP allows devices to connect to an operational bus without disrupting active communications. When a new device is powered up, it initially enters a passive listening state, monitoring the bus for the Start Sync signal to acquire the timing information it needs. Upon detecting the Start Sync, the device synchronizes its internal clock and configures its transmission parameters according to the appropriate Δ value. Only after confirming an idle state on the bus does the device transition to active communication mode. This careful integration ensures that hot-joining does not disturb ongoing data exchanges.

Multi-Controller Setup

DCP supports multiple controllers on the same bus without causing interference. Each controller first performs carrier sensing by listening to the bus, which is maintained high by the pull-up resistor when idle. When multiple controllers attempt to transmit simultaneously, DCP employs a binary countdown arbitration process. During this process, each controller sends its unique binary address along with the data. While transmitting, a controller monitors the bus; if it sends a logical "1" but detects a logical "0," it infers that a higher-priority controller is active and withdraws from the arbitration process. Controllers that lose arbitration then wait for a random backoff period before retrying. This method ensures orderly access and minimizes the probability of repeated collisions.

Collision Detection and Avoidance Strategies (CSMA/CD)

The collision detection and avoidance strategy in DCP is a multi-layered approach based on CSMA/CD and a priority-based binary backoff mechanism.

Before transmitting any data, each device must monitor the bus to determine if it's currently occupied. If the bus is busy (logical Low), the device must defer transmission and wait for the bus to become idle. When a device detects an idle bus, it doesn't immediately transmit. Instead, it waits for a specific period before attempting transmission. This waiting time depends on the device's address:

- Generic Packets: Devices intending to send a generic packet wait $\frac{(Id+6)\Delta}{4}$ seconds.
- L3 Packets (Addresses 250-255): Devices sending L3 packets wait $\frac{(MasterId-249)\Delta}{4}$ seconds, where 'MasterId' is the device's address. This shorter waiting time gives priority to devices in this address range.

After the deferral period, the device begins transmitting its message bit-by-bit. Crucially, during transmission, each device continuously monitors the bus. It compares the level it is writing to the bus with the actual level present on the bus.

If a device detects a discrepancy between the level it's sending and the level on the bus, it indicates a collision has occurred. Specifically, if a device writes a '0' (Low) to the bus but reads a '1' (High), it means another device is simultaneously transmitting a '1'. This signals a collision. The device emitting the '0' does not detect the collision because the dominant low level overwrites the high level. The device detecting the discrepancy immediately stops transmission and enters an error state, signaling loss of arbitration.

After detecting a collision, the colliding devices enter a binary backoff procedure. This involves waiting for a random period determined by their address before attempting to retransmit. The lower the address, the higher the priority in regaining access to the bus. This mechanism helps resolve contention and prevents continuous collisions.

Devices must be able to detect communication failures during any phase of transmission – Start Sync, Start Bit, or Payload. Signals outside a defined tolerance margin (2% of error) are considered invalid, and the communication is terminated to prevent corrupted messages from being received.

Electrical specifications and timing for I/O stages and bus lines

Electrical Specification

Table 2: Electrical characteristics of the DATA line

Symbol	**Parameter**	**Conditions**	**Min**	**Max**
V_{IL}	LOW-level input voltage		V_{GND}V	V_{GND} + 0.3 * V_{DD} V
V_{IH}	HIGH-level input voltage		0.7 * V_{DD} V	-
t_{fall}	Output fall time	V_{IHmin} to V_{ILmax}	-	400ns [(1)]
t_{rise}	Output rise time	V_{ILmax} to V_{IHmin}	-	400ns [(1)]
R_p	Pullup Resistor	-	$\frac{V_{DD}-V_{OL}}{I_{OL}}\Omega$	$\frac{4\mu}{0.8473 \cdot C_{bus}}\Omega$ [(2)]

[(1)] Calculated by 2% (moe) of 20μs, which is Δ_{SLOW}.

[(2)] 4μ is (moe) Δ_{SLOW}, which is our max rise time, for other speeds the function is given as $\frac{(moe_{speed})}{0.8473 \cdot C_{bus}}\Omega$.

Timing Specification

Table 3: Characteristics of the DATA line for different speed configurations

Symbol	SLOW	FAST I	FAST II	ULTRA
t_Δ	20 μs	4 μs	2.5 μs	1.25 μs
t_0	20 μs	4 μs	2.5 μs	1.25 μs
t_1	40 μs	8 μs	5 μs	2.5 μs
$tMin_{bit0}$	32 μs	6.4 μs	4 μs	2 μs
$tMax_{bit0}$	48 μs	9.6 μs	6 μs	3 μs
Tr_{MAX} [(1)]	49.952 kbps	249.952 kbps	399.952 kbps	799.952 kbps
Tr_{MIN} [(2)]	24.976 kbps	124.976 kbps	199.976 kbps	399.976 kbps

[(1)] Max theoretical throughput for a 1 second transmission, consisting only of bit 0. [(2)] Min theoretical throughput for a 1 second transmission, consisting only of bit 1.

Timing Diagram

The timing diagram depicted in Figure 1 illustrates the fundamental signal structure of a typical DCP transmission sequence. The communication begins with the *Start Sync* pulse, a prolonged low-level signal whose duration is determined by the transmitter's role on the bus. This signal allows all listening devices to synchronize their internal clocks and determine the class of the transmitter—whether it is a controller or a peripheral node. Once synchronization is achieved, a distinct high-low sequence of equal duration follows, known as the *Start Bit*. This bit serves as a delimiter, clearly indicating the conclusion of the synchronization phase and the imminent start of data transmission. Subsequent to this preamble, the payload is transmitted as a time-modulated sequence of logical "0"s and "1"s, each defined by specific high and low pulse durations that are multiples of the base unit Δ.

An oscilloscope capture is shown in Figure 2 to illustrate real conditions. This image provides a physical representation of the DCP waveform during transmission, confirming the accuracy of the protocol's time-domain characteristics. Both the synchronization and data segments adhere to the specified

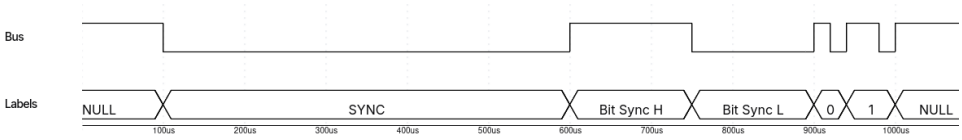

Figure 1: DCP protocol fundamental signals diagram

pulse widths, demonstrating stable transitions and clear signal definition. These characteristics are essential for ensuring reliable communication, particularly in systems with variable electrical loading or bus capacitance.

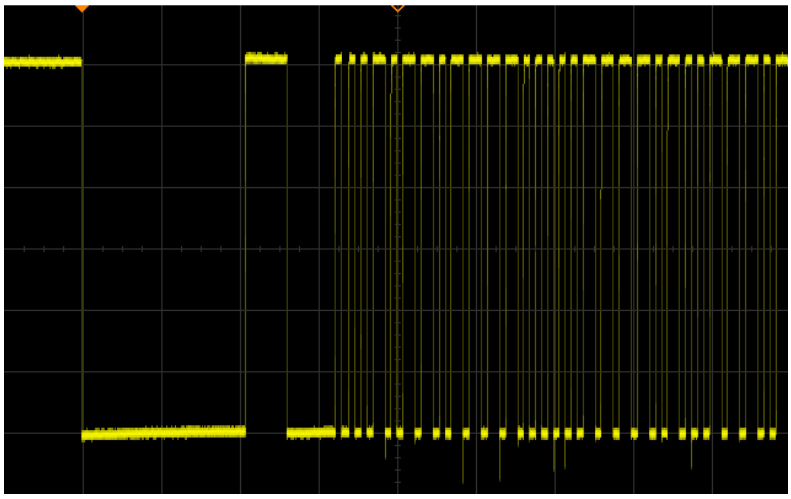

Figure 2: Oscilloscope capture of a DCP transmission, showing Start Sync, Start Bit, and payload.

Wiring Example

The wiring configuration for a typical DCP setup is shown in Figure 3. In this example, three devices are interconnected via a shared single-wire bus. A pull-up resistor is placed between the signal line and the supply voltage VDD for each device. These resistors ensure that the bus remains at a high logic level accounting for the extra capacitance added on each node.

This configuration allows any device to monitor the line state and participate in communication when appropriate. The use of multiple pull-up resistors in parallel does not disrupt the protocol as long as the total effective resistance complies with the electrical constraints defined earlier. It is important to size the pull-up resistors with the expected bus capacitance in mind, maintaining rise times within the allowed specifications for the selected speed class. This simple yet robust topology supports the protocol's hot-join and multi-master features by facilitating seamless electrical integration and reliable signal propagation.

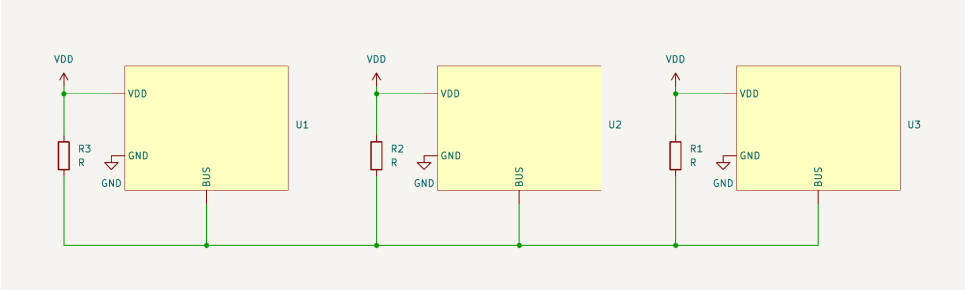

Figure 3: Example wiring configuration with three DCP-compatible devices sharing a single communication bus.