Hello and welcome to this presentation of the Flexible Data rate Controller Area Network interface. It will cover the main features of this interface, which is widely used to connect the microcontroller to a CAN network.
The Flexible Data-rate Controller Area Network is a standard serial differential bus broadcast interface that enables the microcontroller to communicate with external devices connected to the same network bus. The FD-CAN interface is highly configurable, enabling nodes to easily connect using just two wires. Applications benefit from a Multi-master concept with message priority, object-oriented communication (no node addressing, but content identification), real-time capability with low message transfer latency and system wide message consistency (error detection & management mechanism).
The CAN sub-system supports 2 FD-CAN controllers named FD-CAN1 and FD-CAN2. These 2 controllers are independent (except for the Clock Calibration Unit and RAM which are shared) and have the same functionalities (except for the TTCAN which is supported only by FD-CAN1). These controllers support both the Basic Extended CAN protocol versions 2.0 A and B with a maximum bit rate of 1 Mbit/s, as well as CAN FD protocol version 1.0 with up to 64 data bytes and a data bit rate of up to 8 Mbit/s.

A shared 10-Kbyte Message RAM memory is available. Fully programmable, this RAM is used to contain the filters, buffers, FIFOs and the triggers for the Time-Triggered CAN (TTCAN).

The total amount of memory that would be required to support the full configuration for a single controller is 17.4 Kbytes (max). As only 10-Kbyte RAM memory is available, some trade-offs have to be made in function of the application.
Each controller also supports 2 independent maskable interrupts, each one having 30 fully configurable interrupt flags.

The controllers have a power-down mode. They support error logging, AUTOSAR, J1939 and separate signaling on reception of high-priority messages.

Controller FD-CAN1 supports the Time Triggered CAN (TTCAN), including event synchronized time-triggered communication, global system time, and clock drift compensation.

The Clock Calibration Unit can be optionally used to generate a calibrated clock for both FD-CAN1 and FD-CAN2 controllers. It uses the HSI internal RC oscillator and the PLL, by evaluating CAN messages received by the FD-CAN1.
The FD-CAN has three main operating modes: Initialization, Normal and Sleep.

After a hardware reset, the FD-CAN enters Initialization mode via software. In this mode:

- The peripheral must be configured (bit timings and RAM allocation). In the ‘Bit timing’ configuration, the rate is set then the sampling point is adjusted according to the actual serial bus-line.
- The CAN controller then synchronizes itself with the CAN bus by waiting for 11 consecutive recessive bits.

When the CAN is in Normal mode, the user can select different specific modes:

- ‘Classic’ CAN mode
- FD CAN mode: It can be Long Frame and/or Fast Frame mode
• TTCAN mode: Time-Triggered Communication. In this mode, the Automatic Retransmission Feature must be disabled (DAR mode)
• Restricted mode: the controller is able to receive data frames and acknowledge them, but does not send frames. It can be used in applications that adapt themselves to different CAN bit rates.
• Bus Monitoring mode: the controller is able to receive data frames (but cannot acknowledge them). It can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits.
• Test modes
  o External Loop Back mode: the controller treats its own transmitted messages as received messages
  o Internal Loop Back mode: the controller can be tested without affecting a running CAN system

Upon a CPU request, the FD-CAN is in Sleep mode which operates at a lower power (Note: In Sleep mode, the internal pull-up is active on pin CANTX).
This simplified block diagram of the CAN sub-system in a single FD-CAN configuration shows its basic functional and control features.

- Three types of registers: Control configuration registers, filter configuration registers and status registers. Time-Triggered registers are showed independently as they are only present in the FD-CAN1 controller.
- A fully configurable RAM memory stores all the filters and messages needed by the TX and RX handlers. The allocation of these different sections is done by programming the RAM configuration registers.
- The TX handler manages prioritization and frame synchronization before sending the messages to the CAN core protocol handler.
- The RX handler receives messages from the CAN core and accepts or not the message function of the filters.
loaded from the RAM.

- Finally, a Clock Calibration Unit can be used to generate a calibrated clock from the HSI internal RC oscillator and the PLL, by evaluating CAN messages received by FD-CAN1.
- Filters and triggers are directly sent by the CPU through the APB bus.
This simplified block diagram of the FD-CAN in a dual-CAN configuration shows its basic functional and control features.

Compared to single configuration, we can see that:

- The register map, TX Handler, RX Handler and CAN Core are duplicated. Note nevertheless that the Time-Triggered registers are only present for FD-CAN1.
- The RAM memory also stores 2 sets of data sections, one for FD-CAN1 and the other for FD-CAN2.
- Finally, the Clock Calibration Unit, when used, generates a clock for both FD-CAN1 and FD-CAN2.
The controller area network (CAN bus) was originally designed for automotive applications, but is now also used in many other contexts.
An FD-CAN controller peripheral provides two independent interrupt lines. The CAN sub-system provides then:

- 4 independent interrupt lines for the controllers
- 1 interrupt line for the Clock Calibration Unit.

You can see in this slide the complete list of possible interrupt events.
Here is an overview of the CAN sub-system low-power configuration modes. The device is not able to perform any communications in Stop or Standby modes. It is important to ensure that all CAN traffic is completed before the peripheral enters Stop or Standby modes.
While the CPU Core is in Debug mode (i.e. stopped at a breakpoint), then

- FD-CAN remains in its normal functioning mode. In particular, reception continues as normal and this may lead to reception overrun errors when FIFOs or buffers are full.
- Registers of the type “reset on read” or “set on read” are disabled; reading them will not affect their value.
For additional information, refer to the training for these peripherals which may affect FD-CAN behavior:

- Reset and clock controller (RCC) for more information about the CAN clock control and enable/reset.
- Nested vectored interrupt controller (NVIC) for more information about the mapping of the FD-CAN's interrupts.
- General-purpose I/Os (GPIO) for more information about the FD-CAN's input and output pins.
- Debug Support (DBG) for more information about the FD-CAN's behavior when the CPU is halted.
Application notes covering the CAN topic are available on www.st.com. To learn more about the CAN interface, you can also visit a wide range of web pages discussing the CAN communication protocol and bus monitoring tools. Many digital oscilloscopes support direct reading and analysis of data transmitted over the CAN bus.