

CAN Newsletter Online

OPEN SOURCE

CAN FD interface for Arduino

CANFDuino is an open-source project for Arduino. It is available on Crowdsupply as part of the Microchip-Get-Launched design program, using the ATSAMC21G18A micro-controller.



The Arduino-compatible prototyping platform provides two CAN FD interfaces (Source: Togglebit)

Recently, Togglebit has launched the CANFDuino dual CAN Arduino-compatible platform on Crowdsupply. The board provides two CAN FD ports and is qualified for bit rates up to 5 Mbit/s. It uses 9-pin D-sub connectors with the CiA-recommended pin-assignment. Traditional open-source CAN solutions require the purchase of multiple shields, wiring, coding, and provide no enclosure or connectors. The finished product is “ready to modify”, providing prototyping space to add own hardware, and open-source Arduino IDE (integrated development environment) supported software libraries with example code.

The CANFDuino is based on the ATSAMC21G18A micro-controller with two CAN FD controllers by Microchip. It features an SD card slot, 10 analog inputs, up to 24 digital I/O ports, one I²C, and one SPI interface. The product coming in a rugged

aluminum case can be powered by 3,3 V or 5 V. The used MCU is a 64-MHz Cortex M0+ Arm processor, providing 32 KiB SRAM and 256 KiB flash memory. The board has prototyping area for soldering SMT or hole-through components.

The board can be used, for example, for CAN FD network sniffing from a web browser or terminal program, OBD-II data-logging to SD card, and transmitting or outputting analog or digital data. It can act also as bridge connecting two CAN FD segments. The resources to add support for the board to the Arduino IDE can be found on Github, and once installed the user has access to code samples. This includes example codes to display CAN FD frames on a terminal, to receive CAN FD frames, to modify them, to transmit them on the other CAN FD port. There is also an example code to store OBD-II message on the SD card. Additional software examples to transmit I/O data is available, too.

[hz](#)