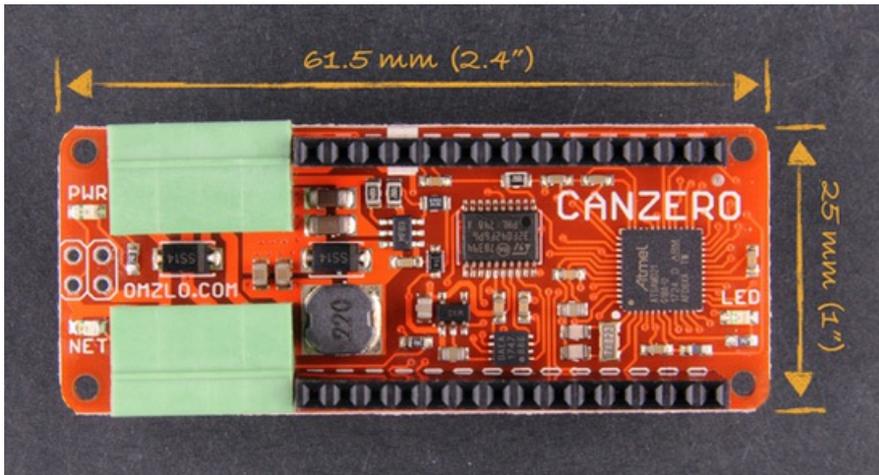


Equipped with two MCUs

The Canzero board by Omzlo can be connected to CAN-based Nocan-wired networks. The product features a Microchip and an ST-Microelectronics Cortex-M0+ processor.



The STM32F042 micro-controller provides the on-chip CAN module (Photo: Omzlo)

The Arduino-compatible board is equipped with two MCUs: the 48-MHz SAMD21G18A by Atmel coming with a 256-KiB flash memory and a 32-KiB RAM as well as the 48-MHz STM32F042 with an on-chip CAN module by ST-Microelectronics providing a 32-KiB flash memory and a 6-KiB RAM. Both MCUs communicate with each other through SPI as well as through two additional general-purpose I/O lines.

The board measures 61,5 mm by 25 mm and shares the same mechanical footprint as the Arduino MKR Zero as well as the same double row of 14-pin 2,54-mm headers. Most shields designed for the Arduino MKR Zero should work on this board, said the supplier on its website.

The main MCU (SAMD21) is used by many Arduino-compatible boards. It runs compiled Arduino sketches created by the user. Canzero "sketches" can be developed in the familiar Arduino IDE and the resulting applications can then be uploaded over the network to any selected node, due to the boot-loader of the MCU. The board is intended for [Nocan networks](#), which is a wired network composed of Arduino-compatible nodes connected through a Classical CAN network.

The second MCU (STM32F0) acts as a smart CAN interface. It unburdens the main MCU from tasks such as network packet buffering and re-assembly. It forces the main MCU to reset and jump into the boot-loader when it receives a specially formed packet. This allows recovering from blocked applications or program errors, without physical access to the node or without initiating a power-down of the whole network. The CAN transceiver is the MCP2562 from Microchip. The CAN interface runs by default at 125 kbit/s. The connection to the bus-lines is done by means of two 4-pin pluggable 3,5-mm headers (CAN_H and CAN_L as well as V_{CC} and GND).

[hz](#)