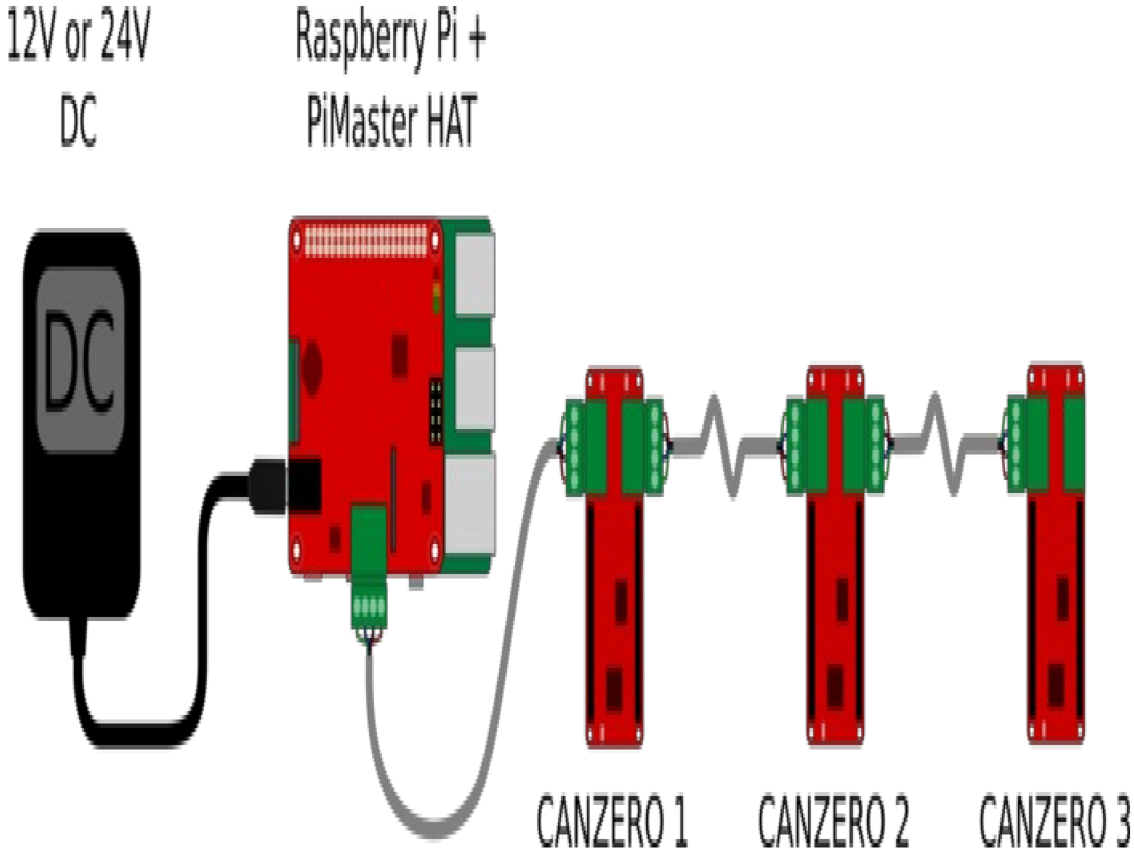
CAN-based IoT platform

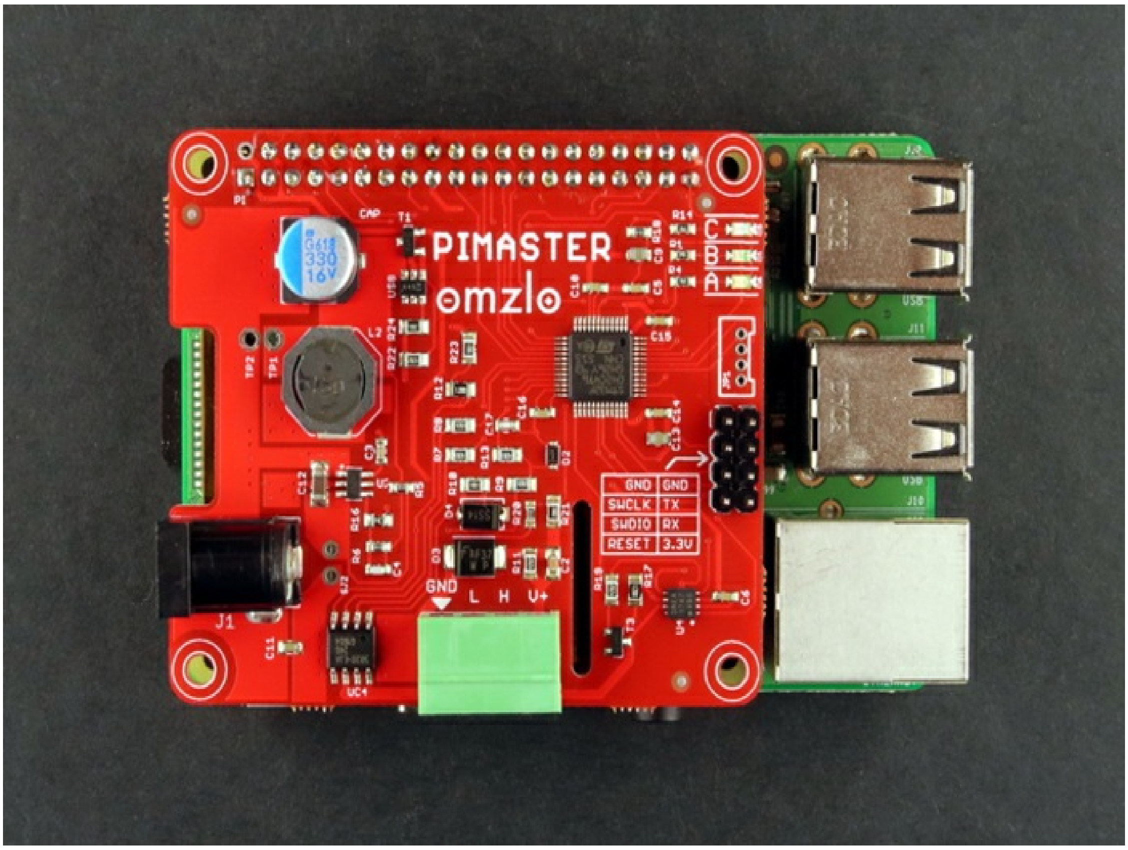
Omzlo has launched the Nocan Arduino-compatible wired IoT platform for makers. It comprises one Pimaster board and several Canzero modules.



The master board and the slave modules are daisy-chained (Photo: Omzlo)

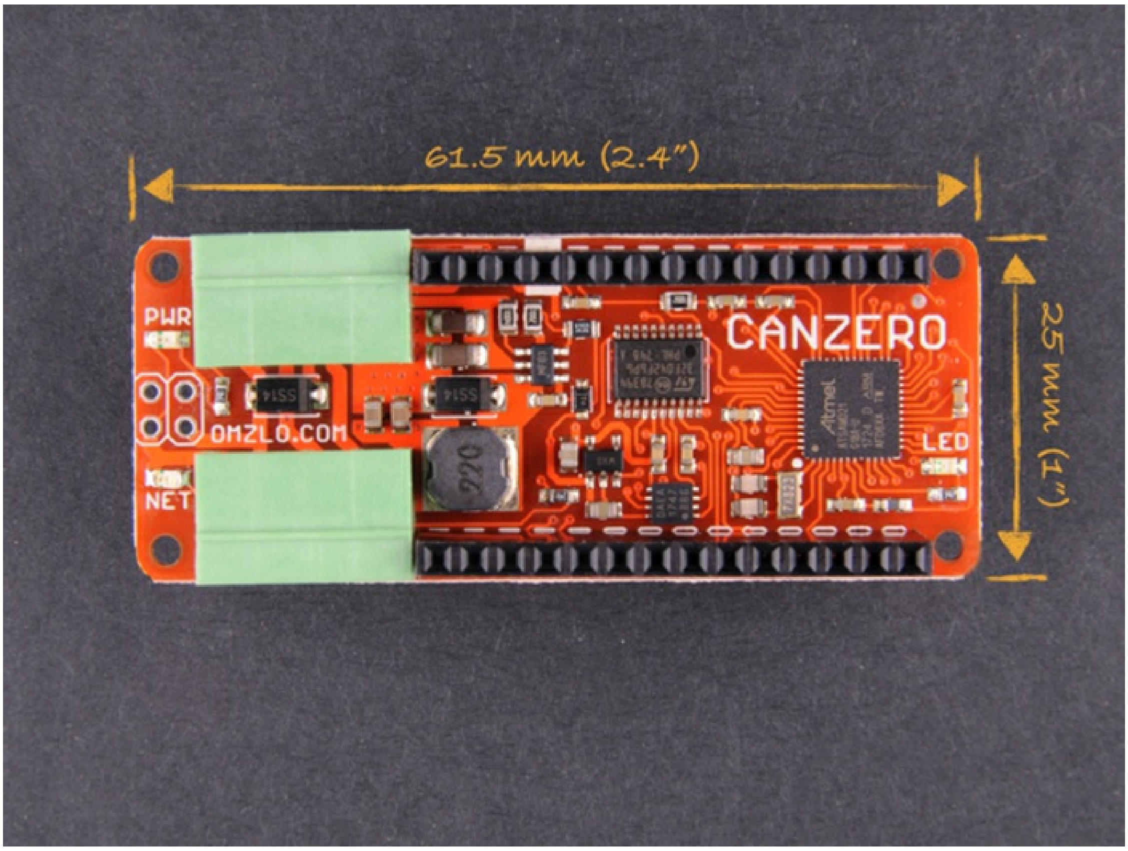
The Nocan IoT platform is a wired network composed of Arduino-compatible nodes connected through a Classical CAN network. It targets applications, in which a wireless communication is not suitable due to reliability reasons, for example. The nodes are connected together forming a "daisy chain" with a single four-wire cable that brings both power and networking over distances of up to 300 m. The CAN transmission speed is 125 kbit/s. Power is provided as 12 V_{DC} or 24 V_{DC}. Each node in the network steps it down to 5 V_{DC} or 3,3 V_{DC}.

The CAN network is composed of a set of nodes, managed by a single master. Each node is an Arduino-compatible board called Canzero, which is based on the Microchip’s SAMD21 micro-controller and offers the same pinout and size as the Arduino MKR Zero. “You can write sketches in the Arduino IDE and then upload the resulting compiled binaries directly to any connected node over the network. The master is based on a Raspberry Pi combined with a specifically designed HAT: the Pimaster. The master takes care of managing the network and acts as a gateway between the CAN network and the ‘outside world’, explained Alwyn Nixon-Lloyd on Omzlo’s website. The network can optionally be controlled over the Internet with a smartphone, thanks to the Blynk platform. All hardware, software, and specifications of the Nocan platform is provided as open-source/open-hardware.



Raspberry Pi board with the shown master module provides connectivity to Ethernet or WLAN (Photo: Omzlo)

The slave modules feature two MCUs communicating via SPI as well as through two additional GPIOs. The module shares the same mechanical footprint as the Arduino MKR Zero, measuring 61,5 mm by 25 mm. It also shares the same 2 x 14 GPIO pin 2,54-mm headers. The product provides two four-pin 3,5-mm pluggable headers designed to connect the board to the CAN network. “However, contrary to the MKR Zero, the Canzero does not offer an SD card slot, an I^2S sound connector or a USB connector,” explained Alwyn Nixon-Lloyd. “Once compiled, Arduino sketches can be uploaded to the Canzero over the network, thanks to a dedicated bootloader burned in the SAMD21 MCU.” The second MCU, an STM32F0, provides the CAN connectivity, offloading the main MCU from tasks such as network packet buffering and re-assembly. It also resets the main MCU if requested.



The Canzero slave modules are based on Microchip’s SAMD21G18A micro-controller with a 48-MHz Cortex M0+ core and an STMF042 Cortex M0+ with on-chip CAN controllers (Photo: Omzlo)

The concept of "channels" used in Nocan network simplifies the construction of distributed applications. You don't need to worry about the address of the node(s) you want to communicate with: simply send a message on a channel and all nodes that have subscribed to that channel will receive it. This approach is very similar to MQTT. For example, if a Canzero node in the CAN network is connected to a relay and has created a channel called "relay", then closing the relay remotely could be as simple as typing the following command: nocanc publish "relay" "close". Similarly, if a slave node has a temperature sensor and publishes temperature data on a channel called "temperature", then reading the temperature remotely could be as simple as typing the following command: nocanc read-channel "temperature".

The Nocan platform uses dynamic node addresses. To overcome the 8-byte limit of the Classical CAN frames, the Nocan platform uses a transport protocol allowing segmenting and re-assembling 64-byte data. CAN FD has not been adopted yet due to availability of MCUs support it.

hz