CANopen and the Raspberry Pi

Even in the industrial environment the Raspberry Pi is used as an entry-level computer. A variety of CAN connection options are available for the Raspberry Pi and there are different ways to support CANopen on it.

The Raspberry Pi is a series of small single-board computers developed by the Raspberry Pi Foundation. The basic goal of the foundation is to promote the teaching of computer science in schools and developing countries. After several previous generations, the Raspberry Pi 3 was released in February 2016. All models feature a Broadcom System on Chip (SoC), which includes an ARM-compatible CPU and an on-chip graphics processing unit. The onboard memory ranges from 256 MiB to 1 GiB RAM. Secure Digital (SD) cards are widely used to store the operating system and program memory in either the SDHC or Micro SDHC sizes. Most boards have various USB slots, HDMI, and composite video output. Lower level output is provided by a number of GPIO and the Pi 3 offers onboard Wi-Fi and Bluetooth support. The Raspberry Pi Foundation provides a Debian-based Linux for the board. Third-party providers offer various other Linux distributions or other operation systems. Unfortunately, none of the Raspberry Pi boards provide a CAN interface onboard, thus external CAN extension boards are required.

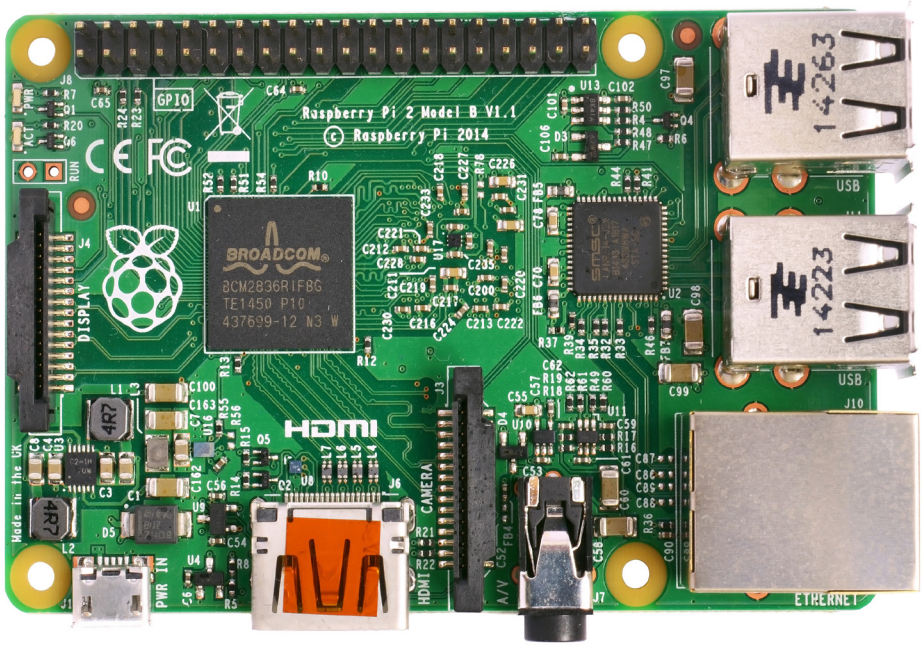

Figure 1: The Raspberry Pi 2 was used in a stamp machine (Photo: Multicherry, CC-BY-SA 4.0)

Industrial devices and clones

Customary Raspberry Pis are delivered as single-board computers without housing and without an industrial power supply. Because of their specific construction, derivatives are more appropriate for industrial use. One example for a derivative is the emPC-A/RPI by Janz Tec: the Raspberry Pi is boxed in an IP20 case and provides an industrial voltage supply of 24 V as well as improved EMC protection. Furthermore, the emPC-A/RPI provides additional interfaces. Especially the CAN interface including a transceiver and a DSUB-9-plug connector is highly attractive for CAN and CANopen applications. A back-end for CAN analysis software and a CANopen gateway are pre-installed by default. The back-end is the so-called Horch server, which is able to send and receive CAN messages and to transmit CAN telegrams via TCP/IP. Both the Horch server as well as the protocol are open source and hosted on Git Lab. By using the Horch server, CAN applications can be created because of the access via the emPC-A/RPI and TCP/IP to the CAN network. Furthermore, different providers offer commercial CAN analysis tools using the Horch protocol. For example, the German software company Emtas offers their versatile CAN Interpreter and CANopen Device Explorer.

The Banana Pi is a Chinese clone of the original Raspberry Pi. Its developers have no connection to the Raspberry Pi Foundation. Nevertheless, the M1 and M1+ board devices provide an integrated CAN controller and CAN-RX and CAN-TX pins are available at the GPIO connectors. The integrated CAN controller provides a higher performance compared to a SPI-connected one. The CAN4Linux, another open-source Linux kernel device driver, has been ported to this target.

CAN connectivity

To achieve a CAN connection, a Raspberry Pi always needs an additional module often connected via SPI. One may develop these modules in-house or buy them as a ready-to-use component. These additional modules include a CAN controller and a CAN transceiver with an appropriate D-Sub 9 plug connector or screw terminal.

Current versions of the Raspian Kernel provide a Socket CAN support for the MCP2515 CAN controller inclusively. The following steps are required to activate the Socket CAN modules:

- enable automatic loading of SPI in raspi-config
- modify /boot/config.txt to configure CAN controller on SPI
  - dtparam=spi=on
  - dtoverlay=mcp2515-can0-overlay, oscillator=16000000,interrupt=25
  - dtoverlay=spi-bcm2835-overlay
- activate CAN interface:
  - sudo /sbin/ip link set can0 up type can bitrate 250000

Provided that there is a support by a Linux device driver, one may alternatively use one of the USB ports to connect a CAN-to-USB interface. Some of the Raspberry Pis are equipped with up to four USB ports and enable users to connect more than one CAN interface. But for industrial demands, integrated solutions like Janz Tec's emPC-A/RPI are more suitable.

CANopen on Raspberry Pi

A CAN API – both Socket CAN or CAN4Linux – is the basis for CANopen protocol stacks or CANopen gateways. The software company Emtas provides a CANopen Master/Slave protocol stack running on the above-mentioned Raspberry Pi clones and variants. This protocol stack is available for Socket CAN and CAN4Linux and provides all CANopen features specified in CiA 301 and CiA 302. With this CANopen stack, various customer specific CANopen applications can be developed.

Another option is the use of the device as a CANopen-TCP/IP gateway according to CiA 309-3. Emtas provides source code variants of the 309-3 gateway that are adaptable to the customer's requirements as well as binary releases of the gateway which are pre-compiled for the emPC-A/RPI. Furthermore, a usable demo version is already pre-installed on these embedded systems. Even Modbus/TCP-CANopen gateways like CiA 309-2 run on this hardware. The CiA 309-3 specification defines an ASCII-based protocol where CANopen services are mapped into ASCII strings that can be exchanged through a TCP/IP socket.

The ASCII protocol for CANopen defines commands that are composed of tokens that are separated by white-spaces and finalized by CRLF characters. All commands sent to the gateway are confirmed and preceded with a sequence number that is enclosed in square brackets. The sequence number is an Unsigned32 number, which is sent back from the gateway with the answer. But these numbers are not used with event-triggered messages like PDO indications delivered to the TCP/IP client from the gateway. After the sequence number is sent, the command starts with an optional network-ID and the node-ID, which is addressed and followed by the specific command. All commands are defined in CiA 303-3 in Backus-Naur Form (BNF).

E.g. the definition for a SDO request:

"[" "]" [[net] node] r[ead]

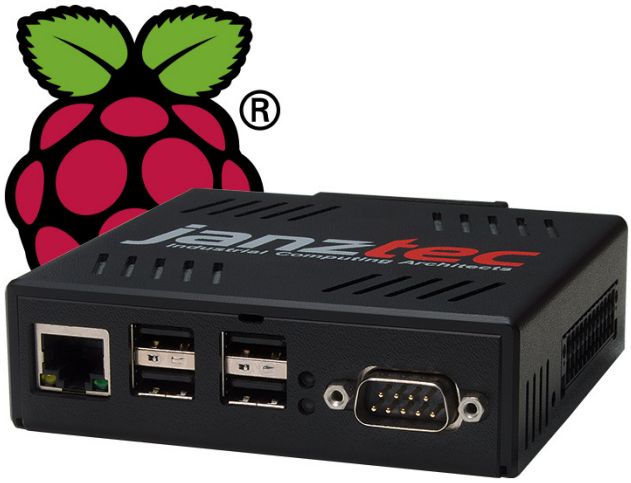

Figure 2: The emPC-A/RPI (Photo: Janz Tec)

An example for such a request:

[2232] 1 43 r 0x1000 0 u32

This means that the value of the object 1000_h sub-index 0_h shall be read from node 43 in net 1. The answer from the gateway might be the following:

[2232] 0x00010191

If a CANopen gateway – such as the one on the emPC-A/RPI – only supports a single CANopen network, the network number can be omitted. In addition to the exemplary SDO service, the CiA 309-3 protocol and the gateway also support other CANopen services like NMT, Heartbeat, Node Guarding, PDO, and LSS. Clients can connect these gateways via an RJ45 Ethernet jack. To connect the Pi3, even WLAN may be used in terms of a WLAN-to-CAN or -CANopen gateway. Once the task of the CiA SIG CANopen IoT, which was described in the previous edition of the CAN Newsletter, is completed, the CANopen gateway application by Emtas will also support URI queries and XML queries in addition to the ASCII protocol.

CANopen inside – the stamp machine

The control unit of a unique stamp machine is one notable example for the use of CANopen on Raspberry Pi. Developed by the engineers of Emtas, it underlines the wide range of CANopen applications. The control unit uses a Raspberry Pi 2 because the development was started before the availability of the emPC-A/RPI. Because of this, an external CAN board had to be used and SocketCAN was used as CAN API.

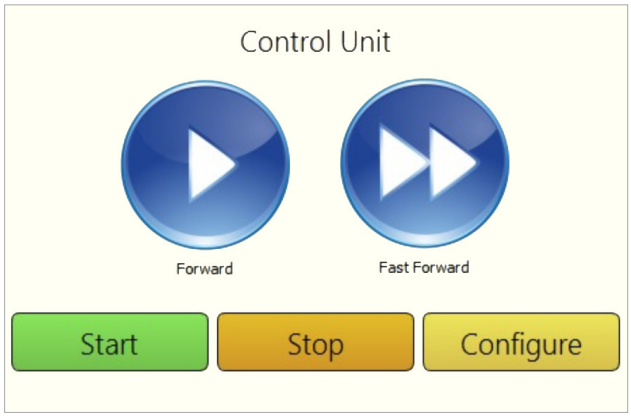

Figure 3 UI on touch display (Photo: Emtas)

In addition to the master – the Raspberry Pi – the network consists of five stepper drives with a CANopen interface that unfortunately does not support CiA 402. The task of the control unit is to control the movements of five motors moving single sheets of paper from a stack of papers to the stamp and to stamp it. The project required precise movements, but not necessarily fast reactions.

The 309-3 CANopen-TCP/IP gateway running in this Raspberry Pi had been used to configure and control CANopen devices. Because of the use of the CiA 309-3 protocol, it had been possible to run the control software on a desktop PC during the development. For future modifications of the machine, it will be possible to omit the touch-display at the machine and to control it from a PC via Ethernet. The control software had been developed using the Qt framework running on various platforms. The GUI and the control application were written with Qt. The UI of the application is displayed on a touch-display connected to the Raspberry Pi. The application communicates with the CANopen-TCP/IP gateway locally via TCP/IP sockets.

The Raspberry Pi and its clones or enhancements are suitable for various CAN and CANopen applications that range from CAN monitoring to the controlling of complete CANopen networks. Besides the aforementioned approaches that use protocol stacks or gateways, various IEC 61131 run time environments from various vendors are available for the Raspberry Pi too. Some of them also include CAN and CANopen support. The engineers at Emtas have gained a lot of experience with various embedded Linux controllers and are able to assist customers with CAN and CANopen projects based on the mentioned boards and devices. ◀

Author

Torsten Gedenk
Emtas
ged@emtas.de
www.emtas.de

TTControl
HYDAC INTERNATIONAL

HY-TTC 32 - Compact Control Unit for Cost-Sensitive Applications, Smaller Machines and Implements

Flexibility and User Friendliness

- Extensive I/O set (30 I/Os with various options for configurability by software)
- Automatic baud-rate detection
- Programmable in C/C++ or CODESYS including support for CANopen® master
- 1 x CAN bus termination configurable via connector pin outputs

Safety

- Certified according to EN ISO 13849 PL c
- CANopen safety

Robustness and Availability

- Aluminum die-cast housing for extremely rough work conditions
- Maximum current up to 24 A

Connectivity

- 2 x CAN 125 kbit/s up to 1 MB/s
- CANopen conformity

Performance

- 32 bit / 180 MHz TI TMS570 dual core lockstep processor (ARM architecture)
- Up to 2.3 MB RAM / 11 MB Flash
- Floating-point-unit

www.ttcontrol.com


Safety Certified ECUs


General Purpose ECUs


I/O Modules


Safe I/O Modules


Operator Interfaces