GPIO protocol

GPIO protocol

Supported device types and versions

The protocol supports communication via inputs/outputs of Raspberry PI and computers built on an RPI Compute Module. Currently supported are:

- Raspberry PI (version 2, 3, and 4) based on a pigpio library - communication via GPIO (General Purpose I/O) pins
- Techbase NPE X500 M3 (DIN-mounted industrial computer built on RPI Compute Module 3 with industrial inputs and outputs)

The communication was tested with Raspberry PI (version 3) and NPE X500-M3-MAX-3G.

For Raspberry PI, the protocol supports reading of digital inputs (instantaneous values as well as counting signal changes with an optional time filter) and setting digital outputs to a constant value or pulse-width modulation (PWM).

For NPE X500 the protocol supports reading of digital and analog inputs and status of user button, writing to digital outputs, setting relay outputs, working with user LEDs, and a buzzer.

Note: On Raspberry PI, the KOM process must be run under the root user to gain access to the GPIO. It is possible to achieve by setting the setuid bit for the kom binary. Under the user *pi*, you can do the following:

cd /opt/d2000/bin
sudo chown root kom
sudo chmod 4755 kom

Note: On NPE X500, the KOM process must be run under the root user to gain access to the GPIO (see the previous note for Raspberry PI). An alternative is adding user d2000 to groups and setting appropriate access rights. Under the *root* user, you can do the following:

sudo usermod -a -G gpio d2000

For access to serial ports also:

sudo usermod -a -G dialout d2000

In order for the d2000 user to access the LEDs and the buzzer, it is necessary to set the rights for the respective files each time the computer starts up. Cre ate a file *d2000init* in the */etc/init.d* directory containing:

```
#!/bin/bash

### BEGIN INIT INFO
# Provides:          script
# Required-Start:    $all
# Required-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:
# Short-Description: D2000Kom
# Description:       Support for D2000 KOM (work with leds/buzzer)
### END INIT INFO

DESC="Set permissions for LED/BUZZER for D2000 KOM"
chmod -R o+w /sys/class/leds/LED1
chmod -R o+w /sys/class/leds/LED2
chmod -R o+w /sys/class/leds/BUZZER
```

Then run:

chmod 755 /etc/init.d/d2000init
update-rc.d d2000init defaults

Communication line configuration

- communication line category: **API**.
- By default, a single line with a single station is configured for one device. If there is a need to stop communication with selected inputs/outputs, there may be several stations on one line. Due to the load distribution or for the rapid reading of the digital inputs, several lines with one or more stations per-line can be configured.

Communication line parameters

Communication line - configuration dialog box - the **Protocol parameter** tab.
Parameters defined in the field have an effect on some optional protocol parameters. The following line protocol parameters can be defined:

Table 1

Full name	Description	Units / size	Default value
Device Type	Type of device. Currently supported are:: • Raspberry PI • NPE X500	-	Raspberry PI
Library Name	Name of the library with communication functions for a particular device. The used values are: • libpigpio.so for Raspberry PI • libx1000gpio.so for NPE X500	-	-
Read Delay Ms	Delay after one reading of the values of all I/O tags. Using this parameter, it is possible to control the reading frequency more finely than using the polling parameters in the configuration of the time parameters of the station.	ms	1

Communication station configuration

- Communication protocol: **GPIO Protocol**.
- Station address: n/a.

I/O tag configuration

Possible I/O tag types: **Ai, Ao, Ci, Co, Di, Do.**

The address format of the I/O tag depends on the device type.

I/O tag addresses for Raspberry PI

In the following table, *id* defines the number of ping in BCM (Broadcom) numbering.

Following image copied from http://pinout.xyz shows the BCM pin numbering on the GPIO connector:

blocked URL

Address	Description	I/O tag type	Examples

DI,*id* DI_UP,*id* DI_DOWN,*id*	The GPIO pin will be configured as a digital input. If the voltage of 3.3V is applied to it, the value of input will be 1. If 0V (ground) voltage is applied to it, the value of input will be 0. Variants DI_UP and DI_DOWN configure the internal pull-up resp. pull-down resistors, so the voltage 3.3V (DI_UP) or ground (DI_DOWN) is connected to the input via the resistor, so even without external voltage applied the input is in a defined state. The DI variant configures the pin so that the pull-up/pull-down resistors are disconnected. Note: pigpio library supports reading from GPIO pins 0-53.	Di, Ci, Ai	DI,25 DI_UP,24
DO,*id*	The GPIO pin will be configured as a digital output. If a value of 1 is written to it, the output will be 3.3V. If 0 is written, the output voltage will be 0V (ground). Note: pigpio library supports writing to GPIO pins 0-53.	Dout, Co, Ao	DO,24
TRIGGER_TOON,*id[,filter]* TRIGGER_TOON_UP,*id[,filter]* TRIGGER_TOON_DOWN,*id[,filter]* TRIGGER_TOOFF,*id[,filter]* TRIGGER_TOOFF_UP,*id[,filter]* TRIGGER_TOOFF_DOWN,*id[,filter]* TRIGGER,*id[,filter]* TRIGGER_UP,*id[,filter]* TRIGGER_DOWN,*id[,filter]*	The GPIO pin will be configured as a counter of input events with an optional filter. The counter can register rising signal edges 0V 3.3V (TOON), falling edges 3.3V 0V (TOOFF), and any signal changes (without TOON/TOOFF). Variants with UP/DOWN configure the internal pull-up resp. pull-down resistors, the same way as for DI_UP/DI_DOWN addresses. Parameter *filter* is used to set a time filter (in microseconds) to filter the noise (if there is a shorter time interval between two events than the *filter*, the second event is ignored). If not specified, all events are considered. The value of the I/O tag is equal to the number of registered events, the maximum value being 2^31-1 i.e. 2147483647 and then the counter goes again from 0. Note: The pigpio library uses a standard input sampling with a period of 5 microseconds, i.e. with a frequency of 200 kHz.	Ci, Ai	TRIGGER,24 TRIGGER,25,1000 TRIGGER_UP,12 TRIGGER_TOON,12
PWM,*id*	The GPIO pin will be configured as PWM (pulse width modulation) output. It is then possible to write values 0-255 controlling the width of the pulse from fully off to fully on. Note: pigpio library supports PWM output for GPIO pins 0-31.	Dout, Co, Ao	PWM,12
REVISON	The revision of the hardware (the number from the "Revision" line from the /proc/cpuinfo file. For example, for RPI 3 this file contains a row "Revision : a02082" and the revision value is 10494082 (after conversion from the hexadecimal system).	Ci	REVISON

I/O tag addresses for NPE X500

In the following table, *id* defines the number of input/output (e.g. DI, DO, AO). The number of inputs and outputs depends on the particular model. The notes are referring to the NPE X500-M3-MAX-3G that was tested.
Note 1: Output points whose addresses contain _BUF use buffered writing. This allows the values of such objects not only to be written but also to be read, which can be useful, for example, after the start of the KOM process.
Note 2: With the tested model, the reading of the digital input took less than 1 ms, while the reading of the analog input took approximately 20 ms.

Address	Description	I/O tag type	Examples
DI,*id*	Digital input (DI) Note: The tested device under had inputs DI1-DI4. The value 0 means that the input is connected to the ground, the value 1 means that the input is disconnected from the ground and has a 3.3 V voltage supplied by the device.	Di, Ci, Ai	DI,1 DI,2
DO,*id* DO_BUF,*id*	Digital output (DO). If buffered (DO_BUFF), the value is also read (during startup and periodically). Note: The tested device had open collector outputs DO1-DO4, i.e. by writing the value 1 the output is connected to the ground.	Dout, Co, Ao	DO,1 DO_BUF,4
DIO,*id* DIO_BUF,*id*	Digital input/output. Based on the I/O tag type, the GPIO port is configured as input (Di, Ci, Ai) or output (Dout, Co, Ao). If the I/O tag configured as an output is also buffered (DIO_BUFF), the value is also read (during startup and periodically). Note: The device tested had digital inputs/outputs DIO1-DIO4 configurable in pairs (i.e., 1 and 2 resp. 3 and 4 must be configured the same way - as input or output).	Input: Di, Ci, Ai Output: Dout, Co, Ao	DIO,2 DIO_BUF,3
RELAY,*id* RELAY_BUF,*id*	Relay output. If buffered (RELAY_BUFF), the value is also read (during startup and periodically). Note: The tested device did not have any relay outputs, so this functionality is not tested.	Dout, Co, Ao	RELAY,1 RELAY_BUF,2
AI,*id*	Analog input (AI). Note: The tested device had analog inputs AI1-AI4 with 12-bit A/D converters that converted the input voltage 0-10 V to a number 0-4095.	Ci, Ai	AI,1 AI,3
LED,*id* LED_BUF,*id*	LED output. If buffered (LED_BUFF), the value is also read (during startup and periodically). Note: the tested device had two LEDs (red LED,1 and green LED,2) that lighted up after writing 1 and went out after writing 0.	Dout, Co, Ao	LED,1 LED_BUF,2
BUZZER BUZZER_BUF	Buzzer. If buffered (BUZZER_BUFF), buzzer status is also read (during startup and periodically). Note: The tested device had a buzzer that started to output a tone after writing 1 and became silent after writing 0.	Dout, Co, Ao	BUZZER BUZZER_BUF
BUTTON BUTTON_BUF	User button status. Note: The tested device had a button that behaved like a DI - it had normally a value of 1 and when pressed, had a value of 0.	Di, Ci, Ai	BUTTON

Literature

- http://www.a2s.pl/en/npe-x500-p-7743.html
- http://www.a2s.pl/products/NPE_X500/NPE_X500M3_EN.pdf

Blog

You can read a blog about GPIO protocol: GPIO protocol is here to help

Changes and modifications

Document revisions

- Ver. 1.0 - August 30th, 2018 – Document created

Related pages:

Communication protocols