

Fork

Programming Guide

Revised December 22, 2020

Table of Contents

Abbreviations	3
Introduction	4
Digital Input (DI) commands	5
Digital Output (DO) commands	6
Analog Input (AI) commands	7
Analog Output (AO) commands	9
Digital Interface Lines (I) commands	10
UART commands	10
SPI Master commands	11
I2C commands	13
Ethernet (ETH) commands	14

Abbreviations

DI – digital input

DO – digital output

AI – analog input

AO – analog output

I - digital interface

– channel number of input/output pin

PWM – pulse width modulation

FREQ – frequency

DUTY – duty cycle

CNT – count

Introduction

Fork API uses standard GET queries to ease human readability and support standard internet browsers like Firefox or Chrome.

API commands are case insensitive.

You can access periphery one by one

```
"http://192.168.0.205/control?DI2"
```

```
"http://192.168.0.205/control?AI5"
```

```
"http://192.168.0.205/control?DO1=1"
```

or make multiple request at a time

```
"http://192.168.0.205/control?DI2&AI5&DO1=1"
```

in case of multiple request by default you`ll get plain text comma separated results, like "0,3.300,1"

Digital Input (DI) commands

Command	Description	Input value	Return value
DI#	Reads DI channel # value	-	0/1
DI_ALL	Reads all DI values	-	array 0/1
DI#_PULLUP	Turns on/off DI channel # pullup	0/1	0/1
DI#_PULLUP	Reads DI channel # pullup status	-	0/1
DI_PULLUP_ALL	Reads all DI pullup status	-	array 0/1
DI#_CNT_START	Start counter on DI channel, return value of counter on DI channel	DI0..DI3	0..9999999
DI#_CNT	Return value of counter on DI channel	DI0..DI3	0..9999999
DI#_STOP	Stop counter on DI channel, return value of counter on DI channel	DI0..DI3	0..9999999
DI#_CNT_RESET	Reset counter on DI channel, return value of counter on DI channel	DI0..DI3	0

Examples:

"<http://192.168.0.205/control?DI1>" – "0"

"http://192.168.0.205/control?DI_ALL" – "0,1,0,0,1,1,0,0"

"http://192.168.0.205/control?DI2_PULLUP" – "1"

"http://192.168.0.205/control?DI4_PULLUP=1" – "1"

"http://192.168.0.205/control?DI1_CNT_START" – "0"

"http://192.168.0.205/control?DI1_CNT" – "345"

"http://192.168.0.205/control?DI1_STOP" – "40"

"http://192.168.0.205/control?DI1_CNT_RESET" – "0"

Digital Output (DO) commands

Command	Description	Input value	Return value
DO#	Writes DO channel # value	0/1	0/1
DO#	Reads DO channel # value	-	0/1
DO_ALL	Reads all DO values	-	array 0/1
DO#_PWM_EN	Start PWM on DO channel #	0/1	0/1
DO#_PWM DO#_PWM_DUTY	Set PWM duty cycle of DO channel #	0 .. 100	0 .. 100
DO#_PWM_FREQ	Set PWM frequency of DO channel #	0 .. 100000 1k .. 100k	0 .. 100000
DO_PWM_ALL	Reads all DO_PWM values. Response consists of three arrays. First array is flags, enabled/disabled PWM. Second array is frequencies. Third array is duty cycles	-	array 0/1, array 0 .. 100000, array 0 .. 100
DO#_PWM_CNT	Set count of impulses and start PWM	0 .. 1000	0 .. 1000
Available in Fork hardware ver. 3.0			
DO_OPENDRAIN	Set open drain for DO port	0/1	0/1
DO_OPENDRAIN	Read state open drain setting of DO port	-	0/1

Examples:

"<http://192.168.0.205/control?DO1>" – "0"

"<http://192.168.0.205/control?DO2=1>" – "1"

"http://192.168.0.205/control?DO_ALL" – "0,1,0,0,1,1,0,0"

"http://192.168.0.205/control?DO2_PULLUP" – "1"

"http://192.168.0.205/control?DO4_PULLUP=1" – "1"

"http://192.168.0.205/control?DO1_PWM=23" – "23"

"http://192.168.0.205/control?DO1_PWM_FREQ=100" – "100"

"http://192.168.0.205/control?DO1_PWM_FREQ=1k" – "1000"

"http://192.168.0.205/control?DO1_PWM_ALL" –
"1,0,0,0,0,0,0,0,1000,0,0,0,0,0,0,0,23,0,0,0,0,0,0"

"http://192.168.0.205/control?DO1_PWM_CNT=15" – "15"

"http://192.168.0.205/control?DO_OPENDRAIN=1" – "1"

Analog Input (AI) commands

Command	Description	Input value	Return value
AI#	Reads AI channel # value	-	-10.000 .. 10.000
AI_ALL	Reads all AI values	-	array -10.000 .. 10.000
AI#_TRIG	Set the source of trigger signal. DI channels can be used or "0" to turn trigger off	0/DI0..DI7	0/DI0..DI7
AI#_TRIG	Reads the value of AI channel captured on last trigger signal. Every triggering overwrites previous value. Getting "AI# wait" means no data was still captured.	-	-10.000 .. 10.000 or AI# wait
AI#_TRIG_POLARITY	Select polarity of trigger on rising or falling edge	0_1/1_0	0_1/1_0
AI#_SINGLE_TRIG	Set the source of single trigger signal. For trigger uses DI channels	0/DI0..DI7	0/DI0..DI7
AI#_SINGLE_TRIG	Reads the value of AI channel captured on first trigger signal. Getting "AI# wait" means no data was still captured.	-	-10.000 .. 10.000 or AI# wait
AI#_SINGLE_TRIG_W	Set the source of trigger signal. For trigger uses DI channels. This type of trigger uses property of HTTP, that can send reply after some time. You don't need to poll AI value like AI#_trig or AI#_single_trig, but you cannot send another requests until trigger or timeout occurs. Timeout is 30s.	DI0..DI7	-10.000 .. 10.000
AI#_FILTER	Turn on/off filtering values of AI channel.	0/1	0/1
Available in Fork hardware ver. 3.0			
AI_DIFF	Read differential input	-	-1.500 .. 1.500

Examples:

"<http://192.168.0.205/control?AI3>" – "2.8"

"http://192.168.0.205/control?AI_ALL" – "2.34,1.34,6.01,2.04,2.34,1.34,6.01,2.04"

"http://192.168.0.205/control?AI3_trig=DI4" – "DI4"

"http://192.168.0.205/control?AI3_trig" – "2.34"

"http://192.168.0.205/control?AI3_trig_polarity=0_1" – "0_1"

"http://192.168.0.205/control?AI3_single_trig=DI4" – "DI4"

"http://192.168.0.205/control?AI3_single_trig" – "2.34"

"http://192.168.0.205/control?AI3_single_trig_w=DI4" – "2.34"

"http://192.168.0.205/control?AI3_filter=1" – "1"

Analog Output (AO) commands

Command	Description	Input value	Return value
AO#	Set value of AO channel	0.000 .. 10.000	0.000 .. 10.000
AO#	Reads value of AO channel	-	0.000 .. 10.000
AO_ALL	Read all values of AO channels	-	array 0.000 .. 10.000
AO#_GEN_EN	Enable/disable generator on AO channel	0/1	0/1
AO#_GEN_FREQ	Set frequency of signal on AO channel	0..100000	0..100000
AO#_GEN_DC	Set DC level of signal on AO channel	0.000 .. 10.000	0.000 .. 10.000
AO#_GEN_AMPLITUDE	Set amplitude of signal on AO channel	0.000 .. 10.000	0.000 .. 10.000
AO#_GEN_MIN	Set min value of signal, changes the parameters DC and amp, respectively	0.000 .. 10.000	0.000 .. 10.000
AO#_GEN_MAX	Set max value of signal, changes the parameters DC and amp, respectively	0.000 .. 10.000	0.000 .. 10.000
AO#_GEN_PHASE	Set phase of signal, when starting generator	0 .. 359	0 .. 359
AO#_GEN_PHASE_2	Set phase shift of AO1 relative AO0 and start generator AO0 and AO1	0 .. 359	0 .. 359
AO#_GEN_CNT	Set count of periods, which generates on AO channel	1 .. 1000	1 .. 1000
AO_GEN_ALL	Get settings about each generator in next order: enable/disable, mode, frequency, dc level, amplitude	-	
AO#_MODE_SIN	Set sine wave form on AO channel	-	SIN
AO#_MODE_TRI	Set triangle wave form on AO channel	-	TRI
AO#_MODE_SAW	Set saw wave form on AO channel	-	SAW
AO#_MODE_FFG	Set free-form generator on AO channel	-	FFG

Examples:

"<http://192.168.0.205/control?AO0=2.8>" – "2.8"

"<http://192.168.0.205/control?AO0>" – "2.8"

"http://192.168.0.205/control?AO_ALL" – "2.8,3.1"

"http://192.168.0.205/control?AO0_GEN_EN=1" – "1"

"http://192.168.0.205/control?AO0_GEN_FREQ=1000" – "1000"

"http://192.168.0.205/control?AO0_GEN_DC=3.3" – "3.3"

"http://192.168.0.205/control?AO0_GEN_AMPLITUDE=0.1" – "0.1"

"http://192.168.0.205/control?AO0_GEN_MIN=2" – "2"

"http://192.168.0.205/control?AO1_GEN_MAX=7" – "7"

"http://192.168.0.205/control?AO1_GEN_PHASE=60" – "60"

"http://192.168.0.205/control?AO0_GEN_PHASE_2=90" – "90"

"http://192.168.0.205/control?AO0_GEN_CNT=50" – "50"

"http://192.168.0.205/control?AO_GEN_ALL" –
"0,SIN,1000,5.000,5.000,0,SIN,1000,5.000,5.000"

"http://192.168.0.205/control?AO1_MODE_SIN" – "SIN"

Digital Interface Lines (I) commands

Command	Description	Input value	Return value
I_MODE	Set interface mode. Be careful when setting interface mode and look at the pinout.	UART# UART#_RX UART#_TX SPI I2C	UART1 UART2 UART3 UART4 SPI I2C
I_UART#_EN	Enable or disable RX and TX lines of UART	0/1	0,0/1,1
I_UART#_EN	Read state of UART RX and TX	-	0/1,0/1
I_UART#TX_EN	Enable or disable TX lines of UART	0/1	0/1
I_UART#TX_EN	Read state of UART TX	-	0/1
I_UART#RX_EN	Enable or disable RX lines of UART	0/1	0/1
I_UART#RX_EN	Read state of UART RX	-	0/1
I_SPI#_EN	Enable or disable SPI	0/1	0/1
I_SPI#_EN	Read state of SPI	-	0/1
I_I2C#_EN	Enable or disable I2C	0/1	0/1
I_I2C#_EN	Read state of I2C	-	0/1

UART commands

Command	Description	Input value	Return value
I_UART#	Send packet of bytes. For send non-printable bytes use percent-encoding format.	string and url-encoded bytes	string and url-encoded bytes
I_UART#	Read uart RX buffer. Non-printable bytes are shown as url-encoded bytes.	-	string and url-encoded bytes
I_UART#_RECVSIZE	Get size of received bytes in rx buffer	-	0..512
I_UART#_BAUDRATE	Set baudrate of uart# (bits/s)	1200 .. 1000000	1200 .. 1000000
I_UART#_BAUDRATE	Read baudrate of uart# (bits/s)	-	1200 .. 1000000
I_UART#_STOPBITS	Set stopbits of uart#	1/2	1/2
I_UART#_STOPBITS	Read stopbits of uart#	-	1/2

Examples:

"http://192.168.0.205/control?I_MODE=UART1" – "UART1"

"http://192.168.0.205/control?I_UART1=rate28" – "rate28"

"http://192.168.0.205/control?I_UART1=%04%24%03%06" – "%04%24%03%06"

"http://192.168.0.205/control?I_UART1" – "Door closed."

"http://192.168.0.205/control?I_UART1" – "axf%24%03%06"

"http://192.168.0.205/control?I_UART1_BAUDRATE=115200" – "115200"

"http://192.168.0.205/control?I_UART1_STOPBITS=1" – "1"

SPI Master commands

Command	Description	Input value	Return value
I_SPI_WRITE	Send packet of bytes. For send non-printable bytes use percent-encoding format.	string and url-encoded bytes	string and url-encoded bytes
I_SPI_READ	Read packet of bytes. Non-printable bytes are shown as url-encoded bytes.	0..512	string and url-encoded bytes
I_SPI_WRITE_READ	Read while write. Count of reading bytes equal count of writing bytes.	string and url-encoded bytes	string and url-encoded bytes
I_SPI_BAUDRATE	Set baudrate for SPI	175781, 351563, 703125, 1406250, 2812500, 5625000, 11250000, 22500000	175781, 351563, 703125, 1406250, 2812500, 5625000, 11250000, 22500000
I_SPI_BAUDRATE	Read baudrate of SPI	-	175781, 351563, 703125, 1406250, 2812500, 5625000, 11250000, 22500000
I_SPI_DATASIZE	Set size of write/read data	8/16	8/16
I_SPI_DATASIZE	Read size of write/read data	-	8/16
I_SPI_POLARITY	Set clock polarity of SPI	low/high	low/high
I_SPI_POLARITY	Read clock polarity of SPI	-	low/high
I_SPI_PHASE	Set clock phase of SPI	1/2	1/2
I_SPI_PHASE	Read clock phase of SPI	-	1/2
I_SPI_FIRSTBIT	Set MSB or LSB first bit of SPI	msb/lsb	msb/lsb
I_SPI_FIRSTBIT	Read type of first bit of SPI	-	msb/lsb

Examples:

"http://192.168.0.205/control?I_MODE=SPI" – "SPI"

"http://192.168.0.205/control?I_SPI_WRITE_READ=%9F%00%00%00%00" –
"%FF%1FG%01%00"

"http://192.168.0.205/control?I_SPI_READ=5" – "%01%02%03%04%05"

"http://192.168.0.205/control?I_SPI_BAUDRATE=22500000" – "22500000"

"http://192.168.0.205/control?I_SPI_POLARITY=low" – "LOW"

I2C Master commands

Command	Description	Input value	Return value
I_I2C_WRITE	Writes bytes to device Format 7bit address: [1byte address] [data bytes] Format 10bit address: [2bytes address] [data bytes]	string and url-encoded bytes	string and url-encoded bytes
I_I2C_READ	Reads bytes from device Format 7bit address: [1byte address] [2bytes size to read] Format 10bit address: [2bytes address] [2bytes size to read]	string and url-encoded bytes	string and url-encoded bytes
I_I2C_WRITE_READ	Writes then reads bytes with repeated start to/from device Format 7bit address: [1byte address] [1byte size to write] [2bytes size to read] [data bytes to write] Format 10bit address: [2bytes address] [1byte size to write] [2bytes size to read] [data bytes to write]	string and url-encoded bytes	string and url-encoded bytes
I_I2C_ADDR_SIZE	Set address size for I2C	7/10	7/10
I_I2C_ADDR_SIZE	Read address size for I2C	-	7/10
I_I2C_BAUDRATE	Set I2C baudrate	0..400000	0..400000
I_I2C_BAUDRATE	Read I2C baudrate	-	0..400000
I_I2C_TIMEOUT	Set timeout in ms for I2C bus errors	disable/0..30000	disable/0..30000
I_I2C_TIMEOUT	Read timeout in ms for I2C bus errors	-	disable/0..30000

Examples:

"http://192.168.0.205/control?I_MODE=I2C" – "I2C"

"http://192.168.0.205/control?I_I2C_BAUDRATE=400000" – "400000"

"http://192.168.0.205/control?I_I2C_ADDR_SIZE=7" – "7"

"http://192.168.0.205/control?I_I2C_TIMEOUT=1000" – "1000"

"http://192.168.0.205/control?I_I2C_BAUDRATE=400000" – "400000"

"http://192.168.0.205/control?I_I2C_WRITE_READ=@%01%00%03%E3" – "%05a%01"

Ethernet (ETH) commands

Command	Description	Input value	Return value
ETH_MODE	Enable or disable DHCP on device	STATIC/ DYNAMIC	STATIC/ DYNAMIC
ETH_MODE	Read status of DHCP	-	STATIC/ DYNAMIC
ETH_IP	Set IP address of device	[0-255].[0-255].[0-255].[0-255]	[0-255].[0-255].[0-255].[0-255]
ETH_IP	Read settings of device IP address	-	[0-255].[0-255].[0-255].[0-255]
ETH_IP_MASK	Set subnet mask of device	[0-255].[0-255].[0-255].[0-255]	[0-255].[0-255].[0-255].[0-255]
ETH_IP_MASK	Read subnet mask of device	-	[0-255].[0-255].[0-255].[0-255]
ETH_GATEWA Y	Set gateway IP address	[0-255].[0-255].[0-255].[0-255]	[0-255].[0-255].[0-255].[0-255]
ETH_GATEWA Y	Read gateway IP address	-	[0-255].[0-255].[0-255].[0-255]
ETH_NAME	Set name of device. Name consist of 19 characters.	string	string
ETH_NAME	Read name of device	-	string
ETH_SAVE	Save settings in memory, settings are applied after reset by power	-	SAVE
ETH_LOAD	Load settings from memory	-	LOAD

Examples:

"http://192.168.0.205/control?ETH_MODE=DYNAMIC" – "DYNAMIC"

"http://192.168.0.205/control?ETH_IP=192.168.0.223" – "192.168.0.223"

"http://192.168.0.205/control?ETH_IP_MASK=255.255.255.0" – "255.255.255.0"

"http://192.168.0.205/control?ETH_GATEWAY=192.168.0.1" – "192.168.0.1"

"http://192.168.0.205/control?ETH_NAME=Fork_garden" – "Fork_garden"

"http://192.168.0.205/control?ETH_SAVE" – "SAVE"

"http://192.168.0.205/control?ETH_LOAD" – "LOAD"

Other commands

Command	Description	Input value	Return value
G_DEVICE_INFO	Return info about device in next order: firmware version, hardware version, bootloader version, serial number, production date, loading date	-	

Examples:

"http://192.168.0.205/control?G_DEVICE_INFO" -

"1.0,3.0,1.0,01193901528,28.09.2019,30.09.2019"