

ICC-4C-2000 Industrial Current Controller – 4 Channels



Main features:

- Graphic user interface *Optotune Cockpit* for control via USB or Ethernet
- Communication interfaces:
 - USB, UART, I2C, Ethernet with PoE+ capability
 - Analog input (0 – 10V)
 - General purpose I/O, alternative usage as SPI
 - Trigger OUT
 - Trigger IN
 - 5 x Status LEDs (R/G)
- Software SDKs for Python and C# available
- RoHS, REACH and CE certified

Mechanical specifications

| | | |
|------------------------------------|--|---------|
| Dimensions (L x W x H) | 164 x 105 x 44 | mm |
| Weight | 700 | g |
| USB connector | Micro B, USB 2.0 | |
| Max. USB cable length ¹ | 5 | m |
| Accepted DC Barrel Plug | 2.1 I.D. x 5.5 O.D. x 10.0 | mm |
| Output connector | Standard rectangular header, 100" (2.54mm) pitch | 34 pins |
| I/O connector | Standard rectangular header, 100" (2.54mm) pitch | 20 pins |
| Mounting | T-slots for standard M4 nut | |

Electrical specifications

| | | |
|-------------------------------|--------------|-----|
| Supply voltage range | 24 to 48 | VDC |
| PoE specification | PoE+ 803.3at | |
| Total power consumption (max) | 65 | W |

Thermal specifications

| | | |
|-----------------------|-----------|----|
| Operating temperature | 0 to 60 | °C |
| Storage temperature | -40 to 85 | °C |

Driver outputs

| | | |
|---|---|------|
| Maximum current (4Ch., simultaneously) | ±2000 | mA |
| Minimum current step | 0.25 | mA |
| Resolution | 14 | bits |
| Front-end voltage (configurable) ² | 6 – 18 (factory default = 18) | V |
| Output voltage limit | 4 – 16 | V |
| Output stage topology | Full bridge, filtered PWM (load not grounded) | |
| Digital communication | Dedicated I2C bus for each channel, 400 kHz max. | |
| I2C logic level (pull-ups implemented) | 3.3 | V |
| Power for I2C (each channel separately) | 3.3 | V |
| Power for logics, max. current | 4 x typ. 100 (each channel overcurrent protected) | mA |
| Status LED (main + each channel) | Red, Green, Orange LED | |

¹ Maximum tested cable length in lab conditions, actual performance may depend on electromagnetic environment. Longer lengths are possible using active USB cables.

² The front-end voltage supplies the output stage. The output voltage is slightly lower than the front-end voltage.

Driver inputs

| | | | |
|--|---|------------------------|------|
| Analog inputs level (4 channels) | | 0 – 10 | V |
| Analog input resolution (each channel) | | 16 | Bits |
| Analog input BW | | 40 | kHz |
| Digital inputs | 4x GPIO (configurable as SPI), I2C, UART, SYNC_A, SYNC_B, MCU RESET | | |
| Digital inputs logic level | | 3.3 (NOT 5 V tolerant) | V |

Standard products

| | |
|---------------------------|---|
| ICC-4C-2000 | ICC-4C-2000 Controller only |
| ICC-4C-2000 Extension Kit | Extension kit containing: <ul style="list-style-type: none"> • Adapter board for XPRs, BSW and FMR (with 20 cm extension cable) • Micro-USB cable • DIN Rail clamp kit |
| ICC-4C Power Adapter | Power adapter (EU, CH, US, UK, CN main cables available) |

Mechanical layout

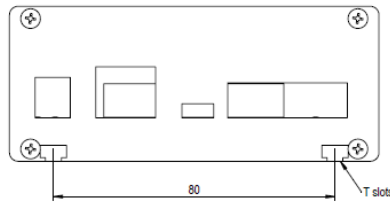
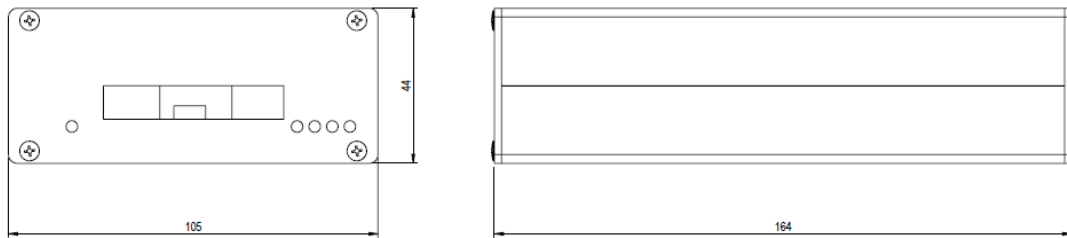


Figure 1: Mechanical dimensions of the ICC-4C-2000.

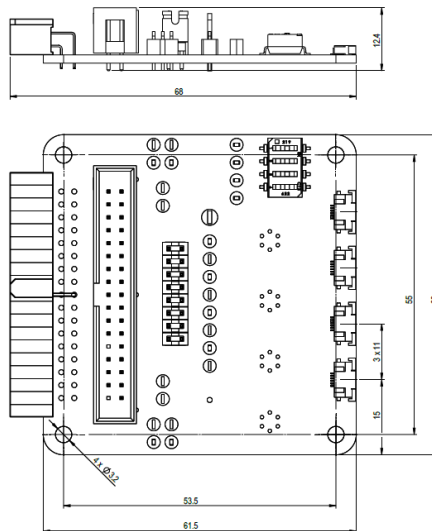


Figure 2: Mechanical dimension of adapter board.

| Connector | X21 | X22 | X30 | X31 | X32 | X33 |
|--------------------|--------|--------|-------|-------|-------|-------|
| Compatible Devices | FMR-20 | FMR-20 | | | | |
| | BSW-20 | BSW-20 | | | | |
| | XPR-20 | XPR-20 | XPR-9 | XPR-9 | XPR-9 | XPR-9 |
| | XPR-33 | XPR-33 | | | | |
| | LSR-30 | LSR-30 | | | | |
| Current Channels | 0 & 1 | 2 & 3 | 0 | 1 | 2 | 3 |



Do not use same current channels for more than one device.

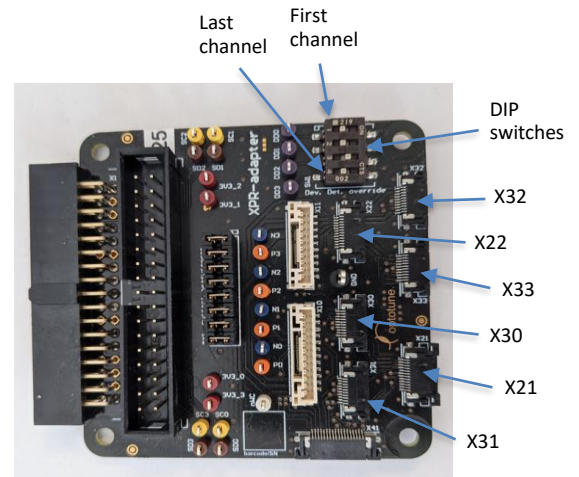


Figure 3: Adapter board pin description and table of compatible devices.

Input/Output connector pinout

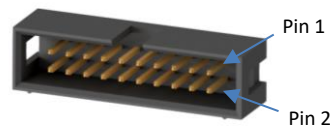


Figure 4: Connector³: Würth Electronics header 100 mil 90deg.

| Pin | Signal | Value | Function |
|-----|----------------|----------------------|--|
| 1 | AnalogIn2 | 0..10 V | Analog input signal #2 |
| 2 | AnalogIn3 | 0..10 V | Analog input signal #3 |
| 3 | AnalogIn0 | 0..10 V | Analog input signal #0 |
| 4 | AnalogIn1 | 0..10 V | Analog input signal #1 |
| 5 | nRESET | 0..3.3V (CMOS level) | Driver reset signal (inverted level) |
| 6 | AGND | - | Analog signal ground |
| 7 | USART_EXT_TX | 0..3.3V (CMOS level) | UART transmit line |
| 8 | SYNC_B | 0..3.3V (CMOS level) | Synchronization input B |
| 9 | USART_EXT_RX | 0..3.3V (CMOS level) | UART receive line |
| 10 | SYNC_A | 0..3.3V (CMOS level) | Synchronization input A |
| 11 | GND | - | Digital ground |
| 12 | GPIO3_SPI_MOSI | 0..3.3V (CMOS level) | General purpose digital IO #3 or SPI MOSI signal (Trigger OUT/IN) |
| 13 | I2C_HOST_SDA | 0..3.3V (CMOS level) | I2C data line from host |
| 14 | GPIO2_SPI_MISO | 0..3.3V (CMOS level) | General purpose digital IO #2 or SPI MISO signal (Trigger OUT/IN) |
| 15 | I2C_HOST_SCL | 0..3.3V (CMOS level) | I2C clock line from host |
| 16 | GPIO1_SPI_NSS | 0..3.3V (CMOS level) | General purpose digital IO #1 or SPI SlaveSelect signal (Trigger OUT/IN) |
| 17 | GND | - | Digital ground |
| 18 | GPIO0_SPI_SCK | 0..3.3V (CMOS level) | General purpose digital IO #0 or SCK signal (Trigger OUT/IN) |
| 19 | VCC_EXT_IN | 24...48 V | External DC supply voltage positive input |
| 20 | GND_EXT_IN | - | External DC supply voltage negative input |

³ First samples of the ICC-4C might have a different connector. Please contact us if the connector doesn't look like in the figure above.

Connecting the adapter board

The adaptor board can be connected directly to the 34-pin output connector or with extension cable (Figure 5).



Figure 5: Adapter board connection options.

Connecting to a device with FPC cable

The flex cable can be plugged directly into the Molex connector of the adapter board (Figure 6). Open the connector clamp with tweezers, then insert the flex cable fully with the copper pads facing upwards. Close the black clamp.

IMPORTANT: Do NOT hot plug the flex cable from the adapter board, it might damage the controller!

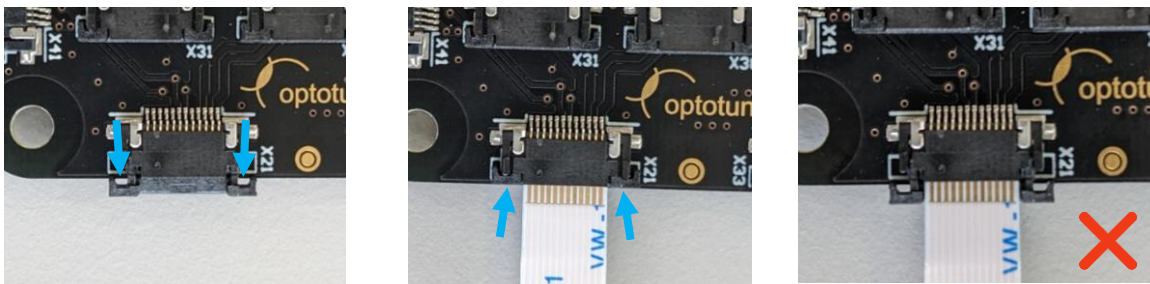


Figure 6: Proper connection to a device with flex cable and wrong connection without fully enclosed connector.

ICC-4C-2000 Main status and Channel status LEDs

| LED | Color | Legend |
|----------------|--------|-------------------------------|
| Main Status | Red | Power on, no connection |
| | Orange | Operation OK (possible error) |
| | Green | Operation OK |
| Channel Status | Red | Device error |
| | Green | Device detected; operation OK |

Input/Output signals

Output trigger

The signal generator of the ICC-4C outputs a trigger signal from the GPIOx pins (0,1,2,3 depending on the channel, see **Input/Output connector pinout** for pin assignment). The trigger signal is HIGH (3.3V, max. 5 mA) at phase 0° of the selected waveform and goes LOW in the middle of the period (Figure 7). For pulse pattern, it reflects the duty cycle.

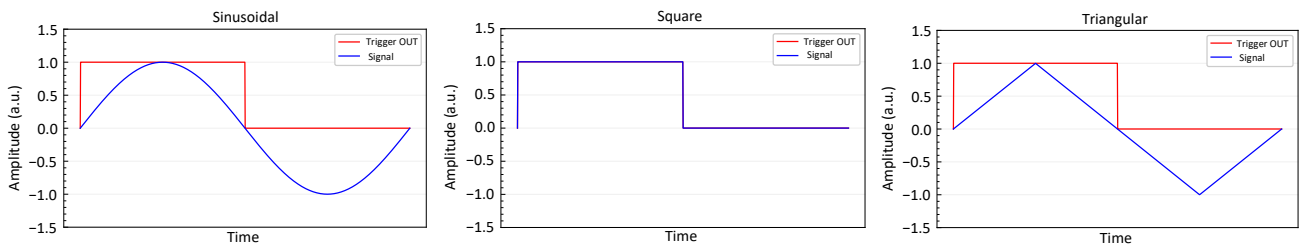


Figure 7: Different waveforms overlapped with the corresponding Trigger OUT signal.

Input trigger

The signal generator of the ICC-4C can be synchronized with an external input trigger. When the trigger input signal goes HIGH (max 3.3V), the selected waveform starts off at phase 0° (Figure 8). There are two options (see **Input/Output connector pinout** for pin assignment):

- SYNC_A pin: an input signal to this pin synchronizes all channels simultaneously.
- GPIOx: by default, GPIOx pins are programmed to be used as trigger output (see previous paragraph). They can be reprogrammed to be used as trigger input for single channels (0, 1, 2, 3). In this case, trigger input function is overridden by GPIOx and hence any signal on SYNC_A is ignored.

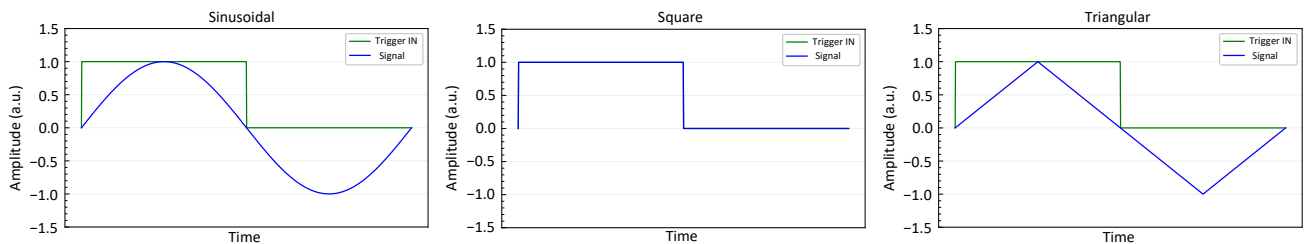


Figure 8: Different waveforms overlapped with an exemplary Trigger IN signal.

Analog Input

Each individual channel of the ICC-4C-2000 can be controlled via a dedicated 0-10 V analog input (see **Input/Output connector pinout** for pin assignment). The resolution of the ADC is 16 bits. The analog input can be mapped to a certain current range on the corresponding channel. Always respect the RMS and peak current limitations of the attached device. For additional information on how to set up the analog mapping, please refer to the Optotune Cockpit software manual.

Simple mode communication

Simple mode communication is RS-232 based serial communication interface that can be used to communicate to the device via a Serial Terminal (e.g., *Termite*, *Putty*. Baud rate: 256000, Parity: N, Stop bits: 1, Data bits: 8) or via *Telnet Client* (on Windows OS), if Ethernet communication is used. It employs a set of ASCII characters commands and answers to interact with ICC-4C driver features. Commands and replies are terminated by character sequence CR, LF (resp. 0x0D, 0x0A). The protocol is not case sensitive and white spaces are ignored.

List of simple mode replies:

| Simple communication mode reply | Description |
|---------------------------------|--|
| OK[CR][LF] | Command accepted and performed without limits. |
| NO[CR][LF] | Command not accepted, for any reason. |
| OL[CR][LF] | Command not accepted, because parameter reached lower limit. |
| OU[CR][LF] | Command not accepted, because parameter reached upper limit. |

| | |
|---------------|------------------------|
| ERROR[CR][LF] | Command not available. |
|---------------|------------------------|

List of simple mode commands:

| Simple mode command | Description |
|---------------------------|---|
| START[CR][LF] | Driver answers "OK" if driver is ready to use and device is detected. Otherwise "ERROR" is received. |
| STATUS[CR][LF] | Driver answers with status encoded within 4 bytes information. Example: "0x00015000[CR][LF]" See next section for further description of the status bytes. |
| RESET[CR][LF] | Restarts driver's firmware. Note: no answer is sent via serial line. |
| GOPRO[CR][LF] | Starts binary protocol-based mode of serial communication. Serial message CRC is not checked. |
| GOPROCRC[CR][LF] | Starts binary protocol-based mode of serial communication. Serial message CRC is checked. |
| GETID[CR][LF] | Answers with firmware serial number. Example: "14352500-00-A[CR][LF]" |
| GETVERSION[CR][LF] | Answers with firmware version number. Example: "1.0.740706[CR][LF]" |
| GETGITSHA1 | Answers with 40 bytes hexadecimal GIT build identification. Example : "eb8115e6b04814f0c37146bbe3dbc35f3e8992e0[CR][LF]" |
| GETSN[CR][LF] | Answers with board and device serial number. Example: "Board: CDAA0057, Device: ANAA722[CR][LF]" |
| GETDEVICESN[CR][LF] | Answers with serial number of a device connected depending on active channel. Example: "Device: ANAA722[CR][LF]" |
| DETECTDEVICE[CR][LF] | Runs device autodetection on active channel. When detection is successful, answers with device name. Example: "XPR[CR][LF]" |
| SETCHANNEL=%int[CR][LF] | Set active channel. Channel index of ICC-4C is 0-3. Command is used as predecessor of other commands applying on active channel of ICC-4C driver. |
| GETCHANNEL[CR][LF] | Get active channel returns an integral number of active/selected channel, 0-3 for ICC-4C driver. Example: "2[CR][LF]" |
| SETCURRENT=%float[CR][LF] | Set current value for active channel. Command supports decimal parameter value in mA units. Current value is limited either by power capabilities of ICC driver itself or connected device. |
| GETCURRENT[CR][LF] | Answers with value of active channel current setpoint. Returned value is decimal number in units of milliamperes. Example: "15.6[CR][LF]" |
| GETTEMP[CR][LF] | Answers with actual temperature of device connected to active channel. Returned temperature is a decimal value in units of degrees of Celsius. Example : "27.54[CR][LF]" |
| SETTEMPLIM=%f[CR][LF] | Set operational temperature limit for selected channel in degrees of Celsius. |

Driver status (refers to "STATUS" command)

The driver "error status" is a 32 bits information about driver itself or a device connected to a certain channel.

The "error status" might have active state, when error status is present or inactive, when error status was present, but currently is inactive.

| Byte | Status bit | Driver status |
|------|------------|---|
| 0. | 0. | Fault on channel 0 – active |
| | 1. | Fault on channel 0 – inactive |
| | 2. | Fault on channel 1 – active |
| | 3. | Fault on channel 1 – inactive |
| | 4. | Fault on channel 2 – active |
| | 5. | Fault on channel 2 – inactive |
| | 6. | Fault on channel 3 – active |
| 1. | 7. | Fault on channel 3 – inactive |
| | 0. | Driver overheat status – active |
| | 1. | Driver overheat status – inactive |
| | 2. | Channel 0: device not detected – active |
| | 3. | Channel 0: device not detected – inactive |
| | 4. | Channel 1: device not detected – active |

| | | |
|----|----|--|
| | 5. | Channel 1: device not detected – inactive |
| | 6. | Channel 2: device not detected – active |
| | 7. | Channel 2: device not detected – inactive |
| 2. | 0. | Channel 3: device not detected – active |
| | 1. | Channel 3: device not detected – inactive |
| | 2. | Channel 0: Device supply 3.3V power overcurrent – active |
| | 3. | Channel 0: Device supply 3.3V power overcurrent – inactive |
| | 4. | Channel 1: Device supply 3.3V power overcurrent – active |
| | 5. | Channel 1: Device supply 3.3V power overcurrent – inactive |
| | 6. | Channel 2: Device supply 3.3V power overcurrent – active |
| | 7. | Channel 2: Device supply 3.3V power overcurrent – inactive |
| 3. | 0. | Channel 3: Device supply 3.3V power overcurrent – active |
| | 1. | Channel 3: Device supply 3.3V power overcurrent – inactive |
| | 2. | I2C communication error – active |
| | 3. | I2C communication error – inactive |
| | 4. | EEPROM reading error -active |
| | 5. | EEPROM reading error – inactive |
| | 6. | Hall sensor out of range – active |
| | 7. | Hall sensor out of range - inactive |

Pro mode communication protocol

Pro Mode communication protocol with additional functionalities is available on request.

Safety and compliance

The product fulfills the RoHS, REACH and CE compliance standards. The customer is solely responsible for complying with all relevant safety regulations for integration and operation.

For more information on optical, mechanical, and electrical parameters, please contact sales@optotune.com

Supporting documentation (available on request)

The following support documentation is available on request:

- How to connect ICC-4C to a PC via direct Ethernet connection
- How to configure ICC-4C for Analog Input Mode with Optotune Cockpit and Python