

IO-Warrior28

Generic universal I/O Controller for USB



Code Mercenaries

1.0 Features

- USB interface
- USB V2.0 compliant full speed device
- USB HID 1.1 compliant
- 19 general purpose I/O pins
- 1000 Hz polling rate for I/O
- IIC master up to 1000 kBit/s
- Supports HD44780 compatible display modules and several graphic display modules
- Easy to use starter kit
- No USB knowledge necessary to use
- Single +3.3V power supply
- Available in QFN28 package, or as SOIC24 and DIL28 module

1.1 Variants

IO-Warrior is available in a couple standard package options as chip or module. Custom modifications are possible.

IO-Warrior28

- 19 general purpose I/O pins, typ. 1 kHz read
- IIC master up to 1 MHz
- Alphanumeric and graphic display support
- Legacy IIC for IOW24 compatibility
- Minimal external component count
- Identifies as either low power (100mA) or high power (500mA) bus powered device. Pin selectable
- Easy to use starter kit for immediate access
- QFN28 package, DIL28 module, and SO24 legacy module to replace IOW24-S

IO-Warrior40 & IO-Warrior56

Please check the respective data sheets for those chips.

1.2 Custom variants

Custom adaptations are available on request. If necessary complex functions can be added to the standard chips to directly control specific circuits.

2.0 Functional overview

IO-Warrior offers a simple access to the USB. Many projects that formerly used the parallel port or some other kind of direct I/O interface today face the problem that getting simple things to work on the USB is quite complex.

With a serial or parallel port you needed only a simple circuit to control an external relay or read a single switch. With USB you need a microcontroller that handles all the protocol work to do the same thing. Several standard commands need to be supported to get a device accepted as a USB device, let alone implementing any real function.

IO-Warrior brings simplicity to the USB. The protocol is encapsulated in the IO-Warrior Chip. You only have to care about the I/O pins and have to write only a few simple lines of code to access them.

IO-Warrior also supports a range of industrial standard interfaces to simplify interfacing to certain chips or modules. These interfaces are handled internally in IO-Warrior removing the bandwidth wasting controlling of individual pins.

IO-Warrior28

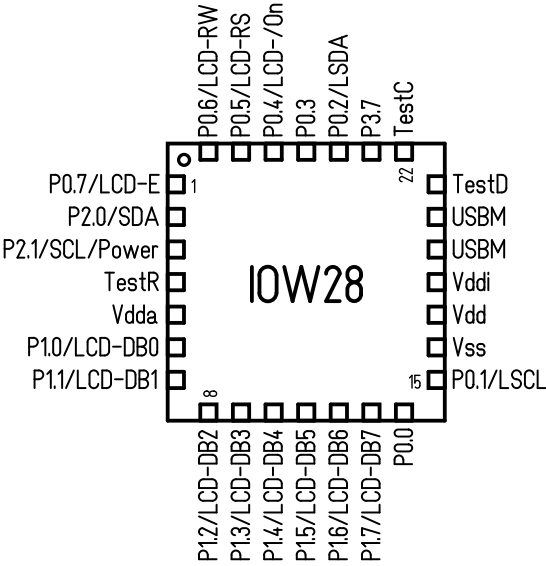
2.1 Product selection matrix

Type	I/O Pins	LCD	IIC	SPI	RC5 IR	Keys	LEDs	Timer	DIL40	SSOP48	DIL24	SOIC24	Starter Kit
IO-Warrior40	32	√	√			√	√		√	√			√
IO-Warrior24 (obsolete)	16	√	√	√	√		√	√			√	√	√
IO-Warrior24PV (obsolete)	12		√	√							√	√	
IO-Warrior28	19	√	√									√	coming
IO-Warrior56	50	√	√	√		√	√						√

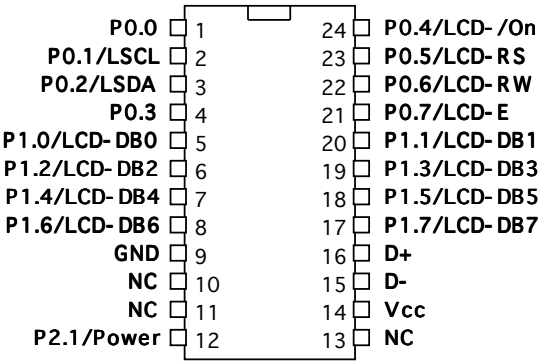
2.2 Starter Kits

IO-Warrior28

3.0 Pin Configurations IO-Warrior28-Q28 QFN28 Chip

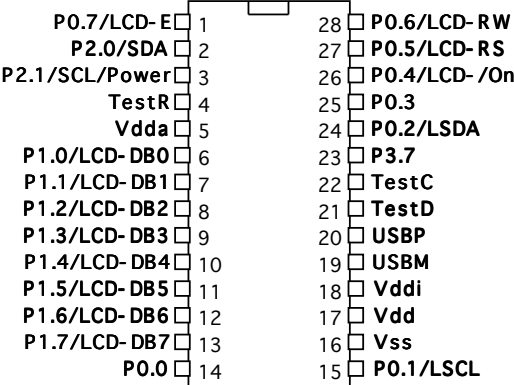


IO-Warrior28-S24 SOIC24 replacement module



All drawings: TOP VIEW!

IO-Warrior28-DIL28 DIL28 module



IO-Warrior28

4.0 Pin Descriptions IO-Warrior28-Q28/-DIL28

Name	I/O	Type	Pins	Description
USBP, USBM	I/O	special	20, 19	USB differential data lines
P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7	I/O	I/O open drain, internal pullup	14, 15, 24, 25, 26, 27, 28, 1	First I/O Port.
P1.0, P1.1, P1.2, P1.3, P1.4, P1.5, P1.6, P1.7	I/O	I/O open drain, internal pullup	6, 7, 8, 9, 10, 11, 12, 13	Second I/O Port
P2.0, P2.1	I/O	I/O open drain, internal pullup	2, 3	Third I/O Port. Also native IIC port. 5 V tolerant. P2.1 is used to determine power setting at power up.
P3.7	I/O	I/O open drain, internal pullup	23	Fourth I/O Port
Vss		power supply	16	Ground
Vdd, Vdda, Vddi		power supply	17, 5, 18	Supply voltage, connect to 3.3 V
TestR, TestD, TestC		special	4, 21, 22	Used during manufacturing, do not connect

4.1 Pin Descriptions IO-Warrior28-S24

Name	I/O	Type	Pins	Description
D+, D-	I/O	special	16, 15	USB differential data lines
P0.0, P0.1, P0.2, P0.3, P0.4, P0.5, P0.6, P0.7	I/O	I/O open drain, internal pullup	1, 2, 3, 4, 24, 23, 22, 21	First I/O Port.
P1.0, P1.1, P1.2, P1.3, P1.4, P1.5, P1.6, P1.7	I/O	I/O open drain, internal pullup	5, 20, 6, 19, 7, 18, 8, 17	Second I/O Port
P2.1/Power	I/O	I/O open drain, internal pullup	12	Third I/O Port. used to determine power setting during power up
GND		Power supply	9	Ground
Vcc		Power supply	14	Supply voltage
NC	-	no connection	10, 11, 13	No internal connection, don't care

IO-Warrior28

4.2 Pin descriptions

USBP, USBM

Differential data lines of USB. Connect these signals direct to a USB cable. ESD protection may be added.

P0.0..P0.7

First I/O port of the chip. These pins correspond with the lowest 8 bits of the input or output (bits 0..7).

P1.0..P1.7

Second I/O Port. Corresponds to the bits 8..15.

P2.0..P2.1

Third I/O Port. Corresponds to the bits 16..17.

P2.1 is used to determine the power setting at power up. A weak pull up should be used to set high power (500mA) mode and a weak pull down for low power mode (100 mA). This determines at reset how much power the IOW28 will demand from the host. Checking the power setting happens before the internal pull-up resistor is activated.

P3.7

Fourth I/O Port. Corresponds to bit 31.

Vss

Power supply ground.

Vdd, Vddi, Vdda

Supply voltage. Requires 3.3 V

Vcc, GND (IOW28-S24)

Power supply pins for SOIC24 replacement module. Vcc is 5 V.

TestC, TestD, TestR

These pins are used during production of the IO-Warrior chips, do not connect.

IO-Warrior28

4.3 Special mode pin functions

IO-Warrior supports driving IIC compatible chips and HD44780 compatible display modules and some other display modules direct. Handling IIC via the normal generic I/O would be very slow as each edge of data and clock would have to be transmitted separately. At a rate of 1000 such transactions per second (which is the maximum IO-Warrior is allowed by USB specifications) the maximum bit rate would be around 500 bits/sec.

To make IIC and other devices usable, IO-Warrior implements the special mode functions. By handling the IIC inside IO-Warrior the actual data rate is increased by some orders of magnitude.

When any of the special mode functions is activated a couple pins will no longer respond as generic I/O pins but are under control of the activated special mode function.

4.3.1 IIC Mode pins

IO-Warrior28 can act as an IIC master. It does have two IIC ports. A legacy port is implemented in software to allow direct pin compatibility with some IOW24-S applications. This legacy IIC should be used only for legacy designs where pin compatibility is required.

For new applications use the native IIC port for IO-Warrior28. It offers speeds from 10 kBit/s to 1000 kBit/s. The native IIC port has been optimized for maximum performance, allowing transfers of > 60 kByte/s without interfering with other functions of the IOW28. This allows to get close to the theoretical maximum of IIC data transfer except for the fastest speed at 1 MBit/s, where about 60% of the theoretical throughput is possible.

The following pins get reassigned when the IIC function is enabled:

Function	Legacy	Native
SCL	P0.1	P2.1
SDA	P0.2	P2.0

These pins will no longer be affected by the data sent via the normal port setting command. Both pins have internal pull up resistors and can be connected direct to IIC compatible chips.

4.3.2 LCD Mode pins

IO-Warrior has support for controlling alpha-numeric display modules based on or compatible with HD44780 as well as some graphic displays.

For simplicity the function is referred to as the LCD function, but there are also OLED and vacuum fluorescence displays that are compatible with this interface.

The following pins get reassigned when the LCD function is enabled:

Function	IOW28
/On	P0.4
RS	P0.5
R/W	P0.6
E	P0.7
Data0	P1.0
Data1	P1.1
Data2	P1.2
Data3	P1.3
Data4	P1.4
Data5	P1.5
Data6	P1.6
Data7	P1.7

When the LCD function is enabled these pins will no longer be affected by the normal port setting command.

/On should be used to enable power supply to LCD modules that have high current demand or backlighting. The /On signal is low when the LCD function is enabled, it does go high when IO-Warrior enters suspend mode or when the LCD function is disabled.

IO-Warrior28

5.0 Device Operation

Due to the fact that all current operating systems offer an especially easy access to devices in the HID class, IO-Warrior was designed as a generic HID device. While this does not exactly fit the device it makes using it a lot easier.

By identifying as a generic HID class device IO-Warrior avoids being controlled by any of the higher level system drivers, which makes it possible to access IO-Warrior from application level.

5.1 Accessing IO-Warrior

A common misconception with people new to the USB is that they think they can "talk to the USB port". The truth is that you do that as likely as you are going to directly talk to your Ethernet port.

Communication on the USB is always with a specific device attached to the USB. The USB itself is only the medium through which you communicate.

To get access to a certain device you have to look for the VendorID and ProductID of that device. The specific mechanisms for doing so depend on the individual operating system.

For details refer to our sample code.

5.2 IO-Warrior communication

IO-Warrior28 has six USB endpoints for the I/O communication. Endpoints are like virtual communication ports into or out of the device.

An endpoint can be assigned to an interface. Interfaces are like virtual devices or subsystems within a device. IO-Warrior28 uses interface 0 to talk to the pins directly, interface 1 to talk to the special mode functions, and interface 2 for the native IIC function.

A SetReport request sending a four byte output report to interface 0 sets the port pins. SetReport requests to interface 1 with a length of 64 bytes are used to control the special mode functions.

For input data IO-Warrior is using endpoint 1 as an interrupt-in endpoint. Interrupt in this case is a bit misleading. For USB interrupt means that data is sent when there is new data available, the host computer is periodically asking the device for new data, the device itself can not initiate the data transfer. IO-Warrior sends a new report any time it detects a change to the input pin status.

Reactions to commands to the special mode functions are sent via endpoint 2, also in interrupt transfer mode.

5.3 IO-Warrior input behaviour

IO-Warrior scans the status of all pins once every millisecond. If it detects a change from the last status a new report via endpoint 1 is issued. Pins which are currently used by a special mode function are not checked and are always reported as being high.

5.5 IO-Warrior output behaviour

Upon receiving a SetReport request IO-Warrior writes the new data to the output pins. There is a small timing offset between the ports getting their new status, which is about $1/4 \mu\text{s}$.

5.6 Using pins as inputs or outputs

All I/O pins on IO-Warrior can be used as input or output pins.

Basically all pins act as inputs all of the time. When receiving an input report from IO-Warrior you always get the current status on the pins.

Writing a 0 as output value to any pin causes it to drive the pin low with an open drain driver. Usually this will result in this pin being read as a zero input as well, unless so much current has to be drained by the pin that the voltage at it gets above the threshold level.

Writing a 1 to a pin causes the open drain driver to be turned off. The pin will be pulled high by an internal pull up resistor. Now the pin acts either as an output with a high level, or can be used as an input.

5.7 Power supply

USB does allow a device to be "Bus Powered". This means the device does get its power off the USB port. To avoid overloading on the USB ports devices need to advertise their power requirements. There are two power classes for devices: Low power and high power. Low power devices may draw up to 100mA off the USB, high power devices up to 500mA.

Likewise there are high power and low power ports. Usually high power ports are those on the motherboard and on hubs with external power supply or hubs in a monitor. Low power ports are typically on hubs that get their power off the USB, like hubs in keyboards.

If the system decides that there is not sufficient power to supply a high power device that device does not get enabled.

IO-Warrior can operate either as a high power or low power device. The power setting is

IO-Warrior28

determines by checking the status of the P2.1/ Power pin on powering up the IO-Warrior. Since the pin is also used as a I/O pin a relatively high resistor value should be used to pull the pin high or low. 270 k Ω or more is recommended.

5.8 Suspend

All devices on the USB port need to support the suspended state. When the host computer stops to periodically access the USB, like when it goes to sleep, all devices need to enter the suspended state and drop their power draw to less than 500 μ A for low power devices or less than 2.5mA for high power devices.

When entering suspended state IO-Warrior pulls all pins high. Care must be taken in designing external circuits so that they will draw no more than the allowed suspend power rating while all pins of IO-Warrior are high.

5.9 Remote Wakeup

IO-Warrior28 does not support remote wakeup.

IO-Warrior28

5.10 Special mode I/O

To enable IO-Warrior to talk to devices that have more complex demands we have added the special mode functions. When any of these functions is enabled a couple pins of IO-Warrior turn into special function pins.

Talking to the special mode functions is handled via the USB interface 1, which is also configured as generic HID compliant.

Commands to the functions are sent via endpoint 0 as SetReport requests to interface 1. Replies from the functions are returned as interrupt in reports via endpoint 2.

To talk to these functions and to handle different requests to them ReportIDs are used which enable multiple functions to use the same endpoint. All reports to and from special mode interfaces are always eight bytes long, including the ReportID.

The following chapters describe the individual special mode functions.

5.10.1 Legacy IIC Special mode function

Use the legacy IIC function only in applications where a replacement of the IOW24 is required. For any new designs use the native IIC function of IO-Warrior28 which offers a superior performance while interfering less with other functions. The legacy IIC is only intended for backwards compatibility.

The legacy IIC function is enabled and disabled by sending a report with the following structure with ReportID=1 to interface 1:

ReportID	1	2	3	4	...	62	63
\$01 out	enable	flags	timeout	\$00	\$00	\$00	\$00

enable=\$01 enables the IIC function, \$00 disables it. Other values are reserved for future use.

Upon enabling IIC the SDA and SCL pins are pulled high and are no longer under control of SetReport requests to interface 0. Disabling IIC does return the pins under control of interface 0 and pulls them high initially.

flags is a legacy from IOW24 and not used on IOW28, the content of this byte is "don't care". Disabling internal pullups and the Sensibus protocol are not available on IOW28.

The timeout byte sets the maximum time IOW waits for SCL to be released by the slave when clock stretching handshake is used. Time is timeout x 500 μ sec, writing a zero sets timeout to 256x500 μ sec.

If the timeout expires due to the slave pulling down SCL too long, IO-Warrior will return an error, just like when the slave generates no Ack signal.

A write request to the IIC is send with ReportID=2 and has the following format:

ReportID	1	2	3	4	...	62	63
\$02 out	flags	data	data	data	data	data	data

flags contains the following bits:

- 7 - Generate Start
- 6 - Generate Stop
- 5 - unused, zero
- 4 - unused, zero
- 3 - unused, zero
- 2 - data count MSB
- 1 - data count
- 0 - data count LSB

If bit 7 - "Generate Start" is set a start signal (SDA falling edge while SCL is high) is generated on the IIC prior to sending out the first data byte.

Bit 6 - "Generate Stop" causes a stop signal (SDA goes high while SCL is high) to be generated after sending the last valid data byte of this report.

"data count" gives the number of valid data bytes in the report. The number may range from zero to six, higher values cause the report to be ignored.

To do write transactions that are longer than six bytes, send the first report with just the "Generate Start" bit set, then send additional reports with neither bit 6 or 7 set until the report with the last bytes is send which has the "Generate Stop" bit set.

Any write transactions are acknowledged by a report via interrupt-in endpoint 2:

ReportID	1	2	3	4	...	62	63
\$02 in	flags	\$00	\$00	\$00	\$00	\$00	\$00

flags contains the following bits:

- 7 - Error bit, 1=error
- 6 - unused, zero
- 5 - unused, zero
- 4 - unused, zero
- 3 - unused, zero
- 2 - data count MSB
- 1 - data count
- 0 - data count LSB

"data count" indicates the last byte that was successfully transferred and acknowledged by the slave (if any). An error is indicated when the slave does not acknowledge a transfer.

IOWarrior28

Reading data off the IIC is initiated with a ReportID=3. The initiating report has the following format:

ReportID	1	2	3	4	...	62	63
\$03 out	count	command	\$00	\$00	\$00	\$00	\$00

"command" holds the command byte to be send to the IIC. "count" is the number of bytes that should be read off the IIC after sending the command byte, values 1 to 62 are valid.

A start signal is automatically generated before sending the command byte and a stop is generated after the last data byte is received.

Data is returned in an input report with ID=3 (which is different from the output ReportID=3 used to initiate the read transaction) via endpoint 2. This restricts a read to fewer bytes than on IOW24.

ReportID	1	2	3	4	...	62	63
\$03 in	flags	data	data	data	data	data	data

flags contains the following bits:

- 7 - error, set if slave does not ack command byte
- 6 - unused, zero
- 5 - data count MSB
- 4 - data count
- 3 - data count
- 2 - data count
- 1 - data count
- 0 - data count LSB

Should the IIC slave fail to acknowledge the command byte the error flag will be set and the transaction aborted. IIC does not have an error condition during the actual reading of data after the command byte was sent.

IO-Warrior28

5.10.2 LCD Special mode function

The LCD special mode function supports display modules that are compatible with the HD44780 controller and several graphic display controllers that use a compatible interface. This controller is made by Hitachi and has set the de-facto standard for alphanumeric LCD modules. There are also OLED and vacuum fluorescence displays that use this type of interface.

The display modules come in various configurations with up to 80 characters total in any kind of arrangement from single line to four lines. Displays with more than 80 characters typically use more than one HD44780. IO-Warrior28 does not directly support modules with more than a single HD44780, some additional hardware may be required for this.

It is recommended to read the HD44780 manual for using the LCD function. Also have a look at our application note AN5: Driving Display Modules with IO-Warrior.

The LCD function is enabled by sending an output report with ID 4 to the USB interface 1:

ReportID	1	2	3	4	...	62	63
\$04 out	enable	\$00	\$00	\$00	\$00	\$00	\$00

enable = \$00 disables the LCD function, enable = \$01 enables the LCD function, other values are reserved.

Upon enabling the LCD function the Pins are put under control of the LCD function and can no longer be controlled by SetReport to interface 0.

The /On pin is pulled low when the LCD function is enabled, it will go high when the IO-Warrior enters suspend state.

To write data to the connected LCD module an output report with ReportID 5 is written with the following format:

ReportID	1	2	3	4	...	62	63
\$05 out	flags	data	data	data	data	data	data

"flags" contains the following bits:

- 7 - RS, Register Select bit
- 6 - unused, zero
- 5 - data count MSB
- 4 - data count
- 3 - data count
- 2 - data count
- 1 - data count
- 0 - data count LSB

The status of the RS bit is used to set the RS line to the LCD module. This allows access to the Instruction register (RS=0) or Data Register (RS=1) of the LCD module.

With "data count" the number of bytes to be written is specified. IO-Warrior will write up to 62 data bytes to the register specified by the RS bit. The Busy bit of the LCD module is automatically checked and data written only when the LCD module is ready to accept it. There is a 16 ms timeout when waiting for the Busy bit, if this time has passed the transaction is canceled, no special feedback is provided.

To read data from the LCD module an output report with ID 6 is sent to interface 1:

ReportID	1	2	3	4	...	62	63
\$06 out	flags	\$00	\$00	\$00	\$00	\$00	\$00

"flags" contains the following bits:

- 7 - RS, Register Select bit
- 6 - unused, zero
- 5 - data count MSB
- 4 - data count
- 3 - data count
- 2 - data count
- 1 - data count
- 0 - data count LSB

RS specifies which register is to be accessed. Data count sets the number of bytes to be read off the LCD (will be ignored if RS=0, only a single read will be done then).

Up to 62 bytes can be read with one request. The data read from the LCD module is returned in input reports with ID 6:

ReportID	1	2	3	4	...	62	63
\$06 in	count	data	data	data	data	data	data

"count" specifies the number of bytes returned in this report.

IO-Warrior28

5.10.3 Native IIC Special mode function

The native IIC function of IO-Warrior28 offers a much higher performance than the legacy function. It uses a separate interface, so the bandwidth of the two endpoints is not shared with any other functions.

Fast mode and fast mode plus are supported, allowing data rates up to 1 MBit/s and a throughput of > 60.000 bytes/s.

The native IIC function is enabled and disabled by sending a report with the following structure with ReportID=1 to interface 2:

ReportID	1	2	3	4	...	62	63
\$01 out	enable	flags	timeout	\$00	\$00	\$00	\$00

enable=\$01 enables the IIC function, \$00 disables it. Other values are reserved for future use.

Upon enabling IIC the SDA and SCL pins are pulled high and are no longer under control of SetReport requests to interface 0. Disabling IIC does return the pins under control of interface 0 and pulls them high initially.

flags selects the speed of the IIC interface and sets the drivers on the SCL/SDA pins accordingly.

The values for flags are as follows:

- 0 - 100 kbit/s
- 1 - 400 kbit/s
- 2 - 50 kbit/s
- 3 - 10 kbit/s
- 4 - 1000 kbt/s

timeout is not yet implemented.

A write request to the IIC is send with ReportID=2 and has the following format:

ReportID	1	2	3	4	...	62	63
\$02 out	flags	data	data	data	data	data	data

flags contains the following bits:

- 7 - Generate Start
- 6 - Generate Stop
- 5 - data count MSB
- 4 - data count
- 3 - data count
- 2 - data count
- 1 - data count
- 0 - data count LSB

If bit 7 - "Generate Start" is set a start signal (SDA falling edge while SCL is high) is generated on the IIC prior to sending out the first data byte.

Bit 6 - "Generate Stop" causes a stop signal (SDA

goes high while SCL is high) to be generated after sending the last valid data byte of this report.

"data count" gives the number of valid data bytes in the report. The number may range from one to 62, other values cause an error to be returned.

To do write transactions that are longer than 62 bytes, send the first report with just the "Generate Start" bit set, then send additional reports with neither bit 6 or 7 set until the report with the last bytes is send which has the "Generate Stop" bit set.

Any write transactions are acknowledged by a report via interrupt-in endpoint 3:

ReportID	1	2	3	4	...	62	63
\$02 in	flags	code	\$00	\$00	\$00	\$00	\$00

flags contains the following bits:

- 7 - Error bit, 1=error
- 6 - Arb Loss, 1 = lost arbitration
- 5 - data count MSB
- 4 - data count
- 3 - data count
- 2 - data count
- 1 - data count
- 0 - data count LSB

"data count" indicates the last byte that was successfully transfered and acknowledged by the slave (if any). An error is indicated when the slave does not acknowledge a transfer.

code contains an additional error code to indicate what happend. The following values are defined:

- 0 - no special code
- 1 - wrong number of bytes requested
- 2 - transaction without Satrt requested
- 3 - NACK received
- 4 - Bus Error

Reading data off the IIC is initiated with a ReportID=3. The initiating report has the following format:

ReportID	1	2	3	4	...	62	63
\$03 out	count	addr	Mcount	\$00	\$00	\$00	\$00

"addr" holds the slave address to be send to the IIC. "count" is the number of bytes that should be read off the IIC after sending the command byte, Mcount is the most significant byte for the count.

A start signal is automatically generated before sending the command byte and a stop is generated after the last data byte is received.

Data is returned in input reports with ID=3 (which is different from the output ReportID=3 used to

IO-Warrior28

initiate the read transaction) via endpoint 3. If more than 62 bytes are requested the data is returned in multiple reports.

ReportID	1	2	3	4	...	62	63
\$03 in	flags	data	data	data	data	data	data

flags contains the following bits:

- 7 - error, set if slave does not ack command byte
- 6 - unused, zero
- 5 - data count MSB
- 4 - data count
- 3 - data count
- 2 - data count
- 1 - data count
- 0 - data count LSB

Should the IIC slave fail to acknowledge the command byte the error flag will be set and the transaction aborted. IIC does not have an error condition during the actual reading of data after the command byte was sent.

5.10.4 Getting current pin status

Due to the way Windows implements HID support IO-Warrior is unable to continuously send its status. HID class devices do have a function that allows the host to set the rate at which reports should be repeated if there is no change to the data. Windows does this rate to zero for IO-Warrior, which means IO-Warrior may send data only if there are changes.

To be able to get the current status from IO-Warrior it does support a Special Mode Function that always returns the current status of all pins.

To get the port status just send a report with ID \$FF to interface 1:

ReportID	1	2	3	4	5	6	7
\$FF out	\$00	\$00	\$00	\$00	\$00	\$00	\$00

This will result in the current pin status to be returned immediately in an input report with ID \$FF with the following format:

ReportID	1	2	3	4	5	6	7
\$FF in	Port0	Port1	Port2	Port3	\$00	\$00	\$00

IO-Warrior28

6.0 Absolute Maximum Ratings

Storage Temperature	-65°C to +150°C
Ambient Temperature with power applied	-40°C to +85°C
Supply voltage on Vcc relative to Gnd	-0.3V to +4V
DC input voltage	-0.3V to +4V
Maximum current into all ports	80mA
Power Dissipation	max. 170mW
Static discharge voltage	>2000V
Latch-up current	>200mA

6.1 DC Characteristics

	Parameter	Min	Max	Units	Remarks
V _{cc}	Operating Voltage	2.0	3.6	V	typ. 3.3 V
I _{cc}	Operating Supply Current		25	mA	
I _{sb}	Suspend mode current		350	μA	internally active
I _{ol}	Sink current on interface pins		25	mA	max. combined all pins 80 mA
I _{olen}	Sink current on /En pin		25	mA	max. combined all pins 80 mA
V _{ol8}	Output low voltage		0.4	V	I = 8 mA
V _{oh8}	Output high voltage	V _{ddi} -0.4		V	I = 8 mA
V _{ol20}	Output low voltage		1.3	V	I = 20 mA
V _{oh20}	Output high voltage	V _{ddi} -1.3		V	I = 20 mA
R _{up}	Pull up/down resistors	25	55	kΩ	typ. 40 kΩ
V _{ith}	Input Threshold Voltage	0.7 x V _{ddi}		V	

IO-Warrior28

7.0 Ordering information

Partname	Order Code	Description	Package
IO-Warrior28	IOW28-Q28	IO-Warrior 19 I/O	QFN28
IO-Warrior28	IOW28-DIL28	IO-Warrior 19 I/O, pinned module	DIL28
IO-Warrior28	IOW28-S24	IO-Warrior module for partial backward compatibility to IOW24	SOIC24
IO-Warrior28 Starter kit	IOW28KIT	not yet available	

The chips listed here are standard products. Customized chips are available on request.

7.1 Packaging info

QFN28 chips come in trays - TBD

DIP28 modules - TBD

SOIC24 replacement modules - TBD

7.2 USB VendorID and ProductID

By default all IO-Warrior chips are shipped with the USB VendorID of Code Mercenaries (\$7C0 or decimal 1984) and a fixed ProductID.

On request chips can be equipped with the customers VendorID and ProductID. VendorIDs can be obtained from the USB Implementers Forum <www.usb.org>

Customized chips may be subject to minimum order quantities, contact <sales@codemerchs.com> for details.

Following are the ProductIDs for the IO-Warrior controllers:

IO-Warrior28 \$1504

ProductIDs are independent of the package type.

7.3 Serial numbers

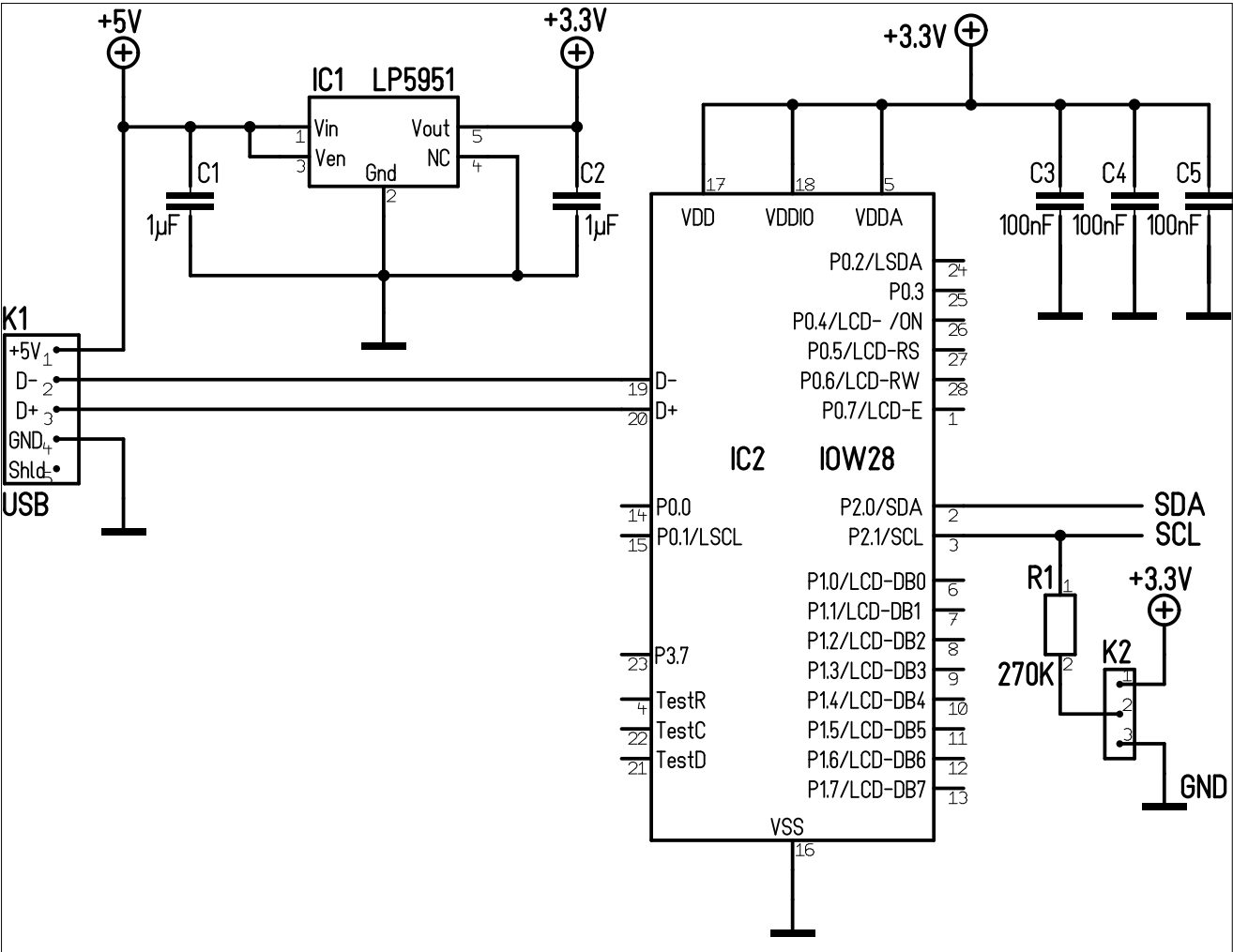
All IO-Warrior28 chips have unique serial numbers in their device descriptors. These serial numbers can be used to simplify programming for multiple IO-Warriors connected to a single computer.

The serial numbers are factory programmed and can not be changed. Serial numbers are 8 digit hexadecimal numbers. No two chips of a type will be produced with identical serial numbers.

It is not possible to order chips with a specific serial number unless they are ordered as custom chips which are subject to minimum order volumes and setup charges.

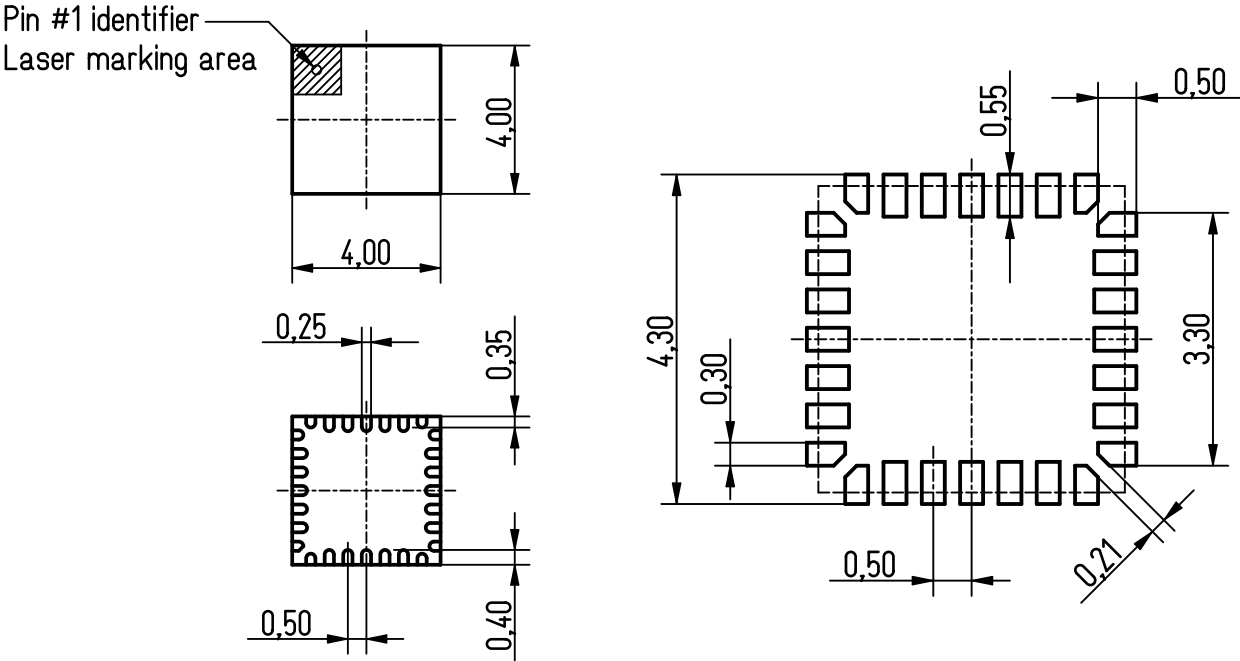
IO-Warrior28

8.0 IO-Warrior28 basic circuit



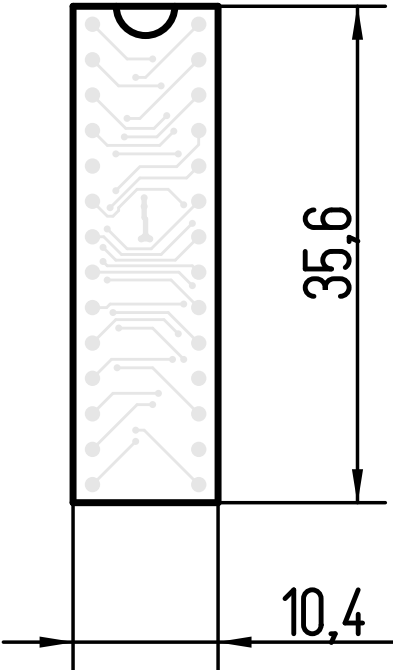
IO-Warrior28

9. Package Dimensions 28 Pin QFN - UQFPN28 - 4x4 mm with 0.5 mm pitch and recommended footprint



Package thickness: 0.55 mm ±0.05 mm
Outer contour tolerance: ±0.1 mm

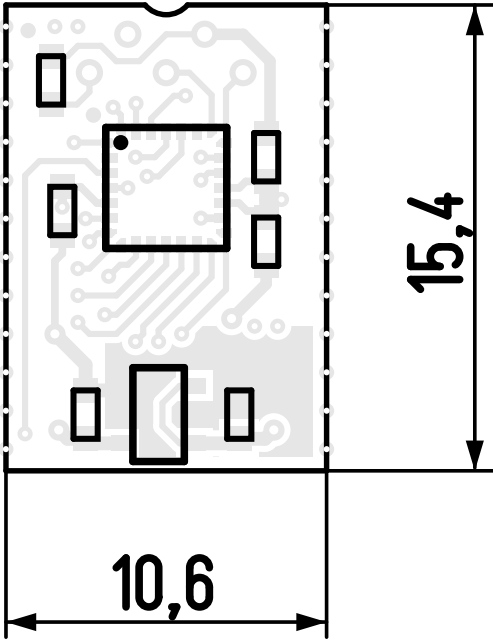
28 Pin DIL module



All dimensions: mm

IO-Warrior28

24 Pin SO24 Replacement Module



Height at thickest point: max. 2.2 mm

IO-Warrior28

10. ESD Considerations

IO-Warrior has an internal ESD protection to withstand discharges of more than 2000V without permanent damage. However ESD may disrupt normal operation of the chip and cause it to exhibit erratic behaviour.

For the typical office environment the 2000V protection is normally sufficient. Though for industrial use additional measures may be necessary.

When adding ESD protection to the signals special care must be taken on the USB signal lines. The USB has very low tolerance for additional resistance or capacitance introduced on the USB differential signals.

10.1 EMC Considerations

IO-Warrior uses relatively low power levels and so it causes few EMC problems.

To avoid any EMC problems the following rules should followed:

- Put the 100nF ceramic capacitors right next to the power supply pins of the chip and make sure the PCB traces between the chips power pins and the capacitor are as short as possible.
- Run the power supply lines first to the capacitor, then to the chip.
- Make the matrix lines only as long as absolutely necessary.
- Keep the two USB signal lines close to each other, route no other signal between them. USB uses differential signalling so the best signal quality with lowest RF emission is achieved by putting these lines very close to each other.

11.0 Revision History

Current shipping version of IO-Warrior28 is V1.0.0.3 preview.

1.0.0.3

- First public version. Preview for initial testing.

11.1 Document Revision History

Still in draft!

12.0 Legal Stuff

This document is ©1999-2018 by Code Mercenaries.

The information contained herein is subject to change without notice. Code Mercenaries makes no claims as to the completeness or correctness of the information contained in this document.

Code Mercenaries assumes no responsibility for the use of any circuitry other than circuitry embodied in a Code Mercenaries product. Nor does it convey or imply any license under patent or other rights.

Code Mercenaries products may not be used in any medical apparatus or other technical products that are critical for the functioning of lifesaving or supporting systems. We define these systems as such that in the case of failure may lead to the death or injury of a person. Incorporation in such a system requires the explicit written permission of the president of Code Mercenaries.

Trademarks used in this document are properties of their respective owners.

Code Mercenaries
 Hard- und Software GmbH
 Karl-Marx-Str. 147a
 12529 Schönefeld OT Grossziethen
 Germany
 Tel: +49-3379-20509-20
 Mail: support@codemercs.com
 Web: www.codemercs.com

HRB 9868 CB
 Geschäftsführer: Guido Körber, Christian Lucht