

IO Expander Datashheet

Features

- Easy to use and easy to understand I/O Interfacing for any demands
- Powerful plattform for the simplest of requirements
- Giant footprint to improve weight distribution in YOUR product
- Simple UART interfacing for near instant IO operations

Applications

- Super secure systems in family businesses
- Highly specialized Linux systems doing ethernet stuff
- Usage in Hackathons because you can

SCK	1	30	NC
3.3V	2	29	NC
Ref	3	28	NC
res.	4	27	D8
res.	5	26	D7
res.	6	25	D6
res.	7	24	D5
res.	8	23	D4
res.	9	22	D3
res.	10	21	D2
res.	11	20	D1
5V	12	19	GND
RST	13	18	RST
GND	14	17	RX
VIN	15	16	TX

Figure 1: Pinout and internal circuit

General Description

This **datasheet** enables you to understand the basic concepts of this sophisticated and utterly overdimensioned I/O Expander.

Pin Description

Pin description is done using the table below. (Pinout should really be sufficient though...)

Table 1: Pin Description

Pin	Name	Function
1	SCK	Clock
2	3.3V	3.3 V Power supply
3	Ref	Reference
4 - 11	Res	Reserved - Nothing to see here
12	5V	5V Power supply
13	RST	Reset on high
14	GND	Ground
15	VIN	Voltage In
16	TX	UART Transmit - Connect to RX of given controller
17	RX	Uart Receive - Connect to TX of given controller
18	RST	Reset on high
19	GND	Ground
20	D1	Digital I/O Pin
21	D2	Digital I/O Pin
22	D3	Digital I/O Pin
23	D4	Digital I/O Pin
24	D5	Digital I/O Pin
25	D6	Digital I/O Pin
26	D7	Digital I/O Pin
27	D8	Digital I/O Pin
28	NC	Not Connected
29	NC	Not Connected
30	NC	Not Connected

¹ Errors in pin assignment are users fault.

Detailed Description

This I/O Expander is a general-purpose input/output (GPIO) peripheral that provides 8 I/O ports, D1 to D8, controlled through a high-speed serial interface. These 8 I/O ports can be used as inputs or open-drain outputs in any combination.

Communication

The communication with this device is done using predefined messages. There are three distinct messages to control the I/O Expander and one response message which announces the levels of defined input pins. Those concepts are described in the next sections.

Setting Output Ports

The communication with this device is as simple as it gets. To set or reset I/O ports, one must send the SET command and supply 8 bit of data to the device. The command to set ports is defined as 0x0F. The protocol expects the application to identify the ports which shall change state to be identified with four LSB according to ???. The Mapping is shown in ??.

I/O direction definition

Identification of pins as Input / output is done with the I/O definition message. One must send the DIR command followed by the corresponding The command to specify the direction of ports is defined as 0xF0. Pins identified with a 1 are set to be output's and pins set to 0 are set as inputs.

Reading Inputs

Similar to the I/O direction definition, the read request message is a special case. For this request, command is specified as 0xAA.

When the read request message is received, the I/O Expander returns with a port status message. This Message identifies the direction of all ports in the first byte. If a port is set as output with high level, the corresponding bit in the second byte is set to high (1). If the port is defined as input, the corresponding Bit in the second byte represents high state with a high level (1) or low state with a low level (0).

Table 2: Pin Address Map

Pin	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Hex
D1	0	0	0	0	0	0	0	1	0xX1
D2	0	0	0	0	0	0	1	0	0xX2
D3	0	0	0	0	0	1	0	0	0xX4
D4	0	0	0	0	1	0	0	0	0xX8
D5	0	0	0	1	0	0	0	0	0x1X
D6	0	0	1	0	0	0	0	0	0x2X
D7	0	1	0	0	0	0	0	0	0x4X
D8	1	0	0	0	0	0	0	0	0x8X

¹ x means dont care.

² Multiple Pins can be set to high or low at the same time by addressing them at the same time.

Table 3: Setting I/O's to high

Pin	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	Hex
D1	0	0	0	0	0	0	0	1	0xX1
D2	0	0	0	0	0	0	1	0	0xX2
D3	0	0	0	0	0	1	0	0	0xX4
D4	0	0	0	0	1	0	0	0	0xX8
D5	0	0	0	1	0	0	0	0	0x1X
D6	0	0	1	0	0	0	0	0	0x2X
D7	0	1	0	0	0	0	0	0	0x4X
D8	1	0	0	0	0	0	0	0	0x8X

¹ x means dont care.

² Multiple Pins can be set to high or low at the same time by addressing them at the same time.