

## NAME

i2cset - set I2C registers

## SYNOPSIS

```
i2cset [-f] [-y] [-m mask] [-r] i2cbus chip-address data-address [value] ... [mode]  
i2cset -V
```

## DESCRIPTION

i2cset is a small helper program to set registers visible through the I2C bus.

## OPTIONS

- V Display the version and exit.
- f Force access to the device even if it is already busy. By default, i2cset will refuse to access a device which is already under the control of a kernel driver. Using this flag is dangerous, it can seriously confuse the kernel driver in question. It can also cause i2cset to silently write to the wrong register. So use at your own risk and only if you know what you're doing.
- y Disable interactive mode. By default, i2cset will wait for a confirmation from the user before messing with the I2C bus. When this flag is used, it will perform the operation directly. This is mainly meant to be used in scripts.
- m **mask**  
The *mask* parameter, if specified, describes which bits of *value* will be actually written to *data-address*. Bits set to 1 in the mask are taken from *value*, while bits set to 0 will be read from *data-address* and thus preserved by the operation. Please note that this parameter assumes that the read and write operations for the specified mode are symmetrical for the device you are accessing. This may or may not be the case, as neither I2C nor SMBus guarantees this.
- r Read back the value right after writing it, and compare the result with the value written. This used to be the default behavior. The same limitations apply as those of option **-m**.

There are three required options to i2cset. *i2cbus* indicates the number or name of the I2C bus to be scanned. This number should correspond to one of the busses listed by *i2cdetect -l*. *chip-address* specifies the address of the chip on that bus, and is an integer between 0x03 and 0x77. *data-address* specifies the address on that chip to write to, and is an integer between 0x00 and 0xFF.

The *value* parameter, if specified, is the value to write to that location on the chip. If this parameter is omitted, then a short write is issued. For most chips, it simply sets an internal pointer to the target location, but doesn't actually write to that location. For a few chips though, in particular simple ones with a single register, this short write is an actual write. If the mode parameter is **s** or **i**, multiple values can be specified.

The *mode* parameter, if specified, is one of the letters **b**, **w**, **s**, or **i**, corresponding to a write size of a single byte, a 16-bit word, a SMBus block write, or an I2C block write, respectively. For SMBus and I2C block writes, the write size is determined by the number of *value* parameters. Except for I2C block writes, a **p** can also be appended to the *mode* parameter to enable PEC. If the *mode* parameter is omitted, i2cset defaults to byte mode without PEC. The *value* provided must be within range for the specified data type (0x00-0xFF for byte and block writes, 0x0000-0xFFFF for words). Another possible mode is **sc**, which doesn't write any value (so-called short write). You usually don't have to specify this mode, as it is the default when no value is provided, unless you also want to enable PEC.

## WARNING

i2cset can be extremely dangerous if used improperly. It can confuse your I2C bus, cause data loss, or have more serious side effects. Writing to a serial EEPROM on a memory DIMM (chip

addresses between 0x50 and 0x57) may DESTROY your memory, leaving your system unbootable!  
Be extremely careful using this program.

**SEE ALSO**

[i2cdump\(8\)](#), [isaset\(8\)](#)

**AUTHOR**

Frodo Looijaard, Mark D. Stuebaker and Jean Delvare

This manual page was originally written by David Z Maze <dmaze@debian.org> for the Debian GNU/Linux system.