

**NAME**

`i2c-stub-from-dump` - feed `i2c-stub` with dump files

**SYNOPSIS**

**`i2c-stub-from-dump`** *address*[,*address*,...] *dump-file* [*dump-file* ...]

**DESCRIPTION**

`i2c-stub-from-dump` is a small helper script for the `i2c-stub` kernel driver. It lets you setup one or more fake I2C chips on the `i2c-stub` bus based on dumps of the chips you want to emulate.

`i2c-stub-from-dump` requires `i2cdetect` and `i2cset` to be installed and reachable through the user's `PATH`. The former is used to find out the `i2c-stub` bus number, while the latter is used to write to the fake I2C chips.

**EXAMPLE**

You have an I2C chip on system A. You would like to do some development on its driver on system B. Here are the few steps you have to follow.

On system A, use `i2cdump` to capture a dump from the chip. Assuming that the chip in question lives at address `0x4c` on I2C bus 0, you would run:

```
i2cdump -y 0 0x4c b > chip.dump
```

Adjust the bus number and chip address for your case. `i2cdetect` can help you find out their values. If the device uses word (16-bit) register access instead of the traditional byte (8-bit) access, use mode `w` instead of `b`.

Copy the dump file to system B.

On system B, run:

```
i2c-stub-from-dump 0x4c chip.dump
```

This will load the required `i2c-dev` and `i2c-stub` kernel drivers if needed, then write all the register values to the emulated I2C chip at address `0x4c`. Again, adjust the address as needed.

**LIMITATIONS**

There are some limitations to the kind of devices that can be handled:

- Device must not have banks (as most Winbond devices do).

**SEE ALSO**

[i2cdump\(8\)](#), [i2cdetect\(8\)](#), [i2cset\(8\)](#)

**AUTHOR**

Jean Delvare