

.. alebo programujem cez grafický výstup

Už sa vám stalo, že ste súrne potrebovali naprogramovať I2C EEPROM pamäť, ale programátor nebol po ruke? Ale bol, len ste si to neuvedomili.

Má ho prakticky každý počítač v sebe. I2C zbernica je dostupná je na grafickom výstupe pre externý monitor - v prevedení VGA, DVI alebo HDMI.

Zbernica DP (Display port) ju neobsahuje.

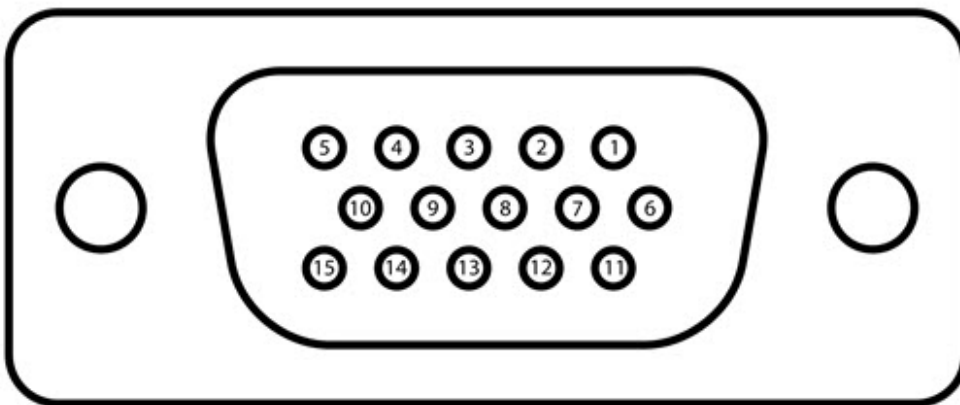
I2C zbernica na grafickom rozhraní, je nazývaná DDC kanál (Display Data Channel), pomocou ktorého monitor posieľa operačnému systému svoje parametre a operačný systém na ich základe nastavuje optimálne rozlíšenie (plus časovanie videosignálov na grafickej karte).

Balík parametrov je v definovanej štruktúre (definovaná vo VESA Standard), ktorá sa nazýva EDID (Extended Display Identification Data). EDID je teda blok údajov uložených v EEPROM pamäti v monitore s HW adresou 0x50, pripojenej k I2C zbernici. EEPROM je napájaný z +5V priamo z PC (súčasť graf. výstupov) a teda pamäť komunikuje aj pri vypnutom monitore.

Signály I2C zbernice (Data, Clock) a napájanie +5 a GND sa nachádzajú na nasledujúcich vývodoch:

Na konektoroch pripájajúcich externý monitor sa nachádza tu:

VGA konektor:



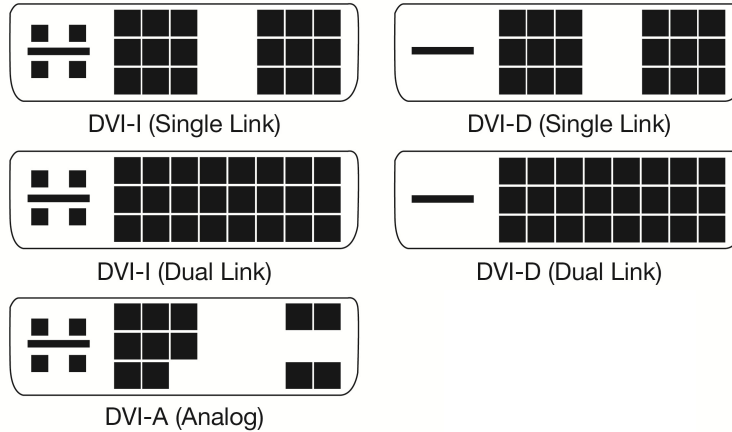
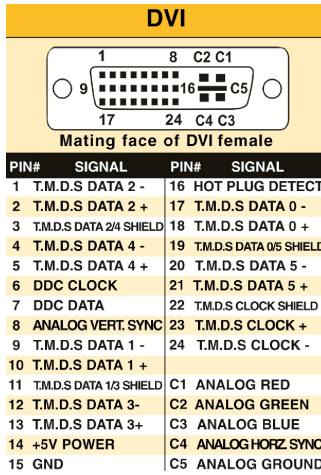
Pin no	Name	Description	Pin no	Name	Description
Pin 1	RED	Red video	Pin 9	KEY/PWR	formerly key, now +5V DC
Pin 2	GREEN	Green video	Pin 10	GND	Ground (VSync, DDC)
Pin 3	BLUE	Blue video	Pin 11	ID0/RES	formerly Monitor ID bit 0, reserved since E-DDC
Pin 4	ID2/RES	formerly Monitor ID bit 2, reserved since E-DDC	Pin 12	ID1/SDA	formerly Monitor ID bit 1, I <sup>2</sup> C data since DDC2
Pin 5	GND	Ground	Pin 13	HSync	Horizontal sync
Pin 6	REDGND Red GND		Pin 14	VSync	Vertical sync
Pin 7	GREENGND	Green GND	Pin 15	ID3/SCL	formerly Monitor ID bit 3, I <sup>2</sup> C clock since DDC2
Pin 8	BLUEGND	Blue GND			

img src: [\[url\]](#)

I2C zbernica je na pinoch č. 15 a 12 (15 - SCL, 12 - SDA), +5V (pin č. 9)

Staršie VGA konektory I2C neobsahovali, ale tie v dnešnej dobe už asi ani nenájdeme.

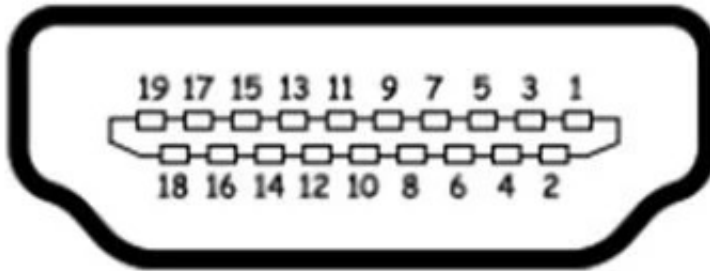
DVI konektor:



img src: [\[url\]](#)

I2C zbernica je na pinoch č. 6 a 7 (6 - SCL, 7 - SDA), +5V (pin č. 14). Je prítomná pri všetkých typoch DVI, teda pri DVI-A, DVI-D a pri DVI-I.

HDMI konektor:



Pin#	Signal	Pin#	Signal
1	TMDS data 2+	11	TMDS clock shield
2	TMDS data 2 shield	12	TMDS clock-
3	TMDS data 2-	13	CEC
4	TMDS data 1+	14	No connected
5	TMDS data 1 shield	15	DDC clock
6	TMDS data 1-	16	DDC data
7	TMDS data 0+	17	Ground
8	TMDS data 0 shield	18	+5V power
9	TMDS data 0-	19	Hot plug detect
10	TMDS clock+		

img src: [\[url\]](#)

I2C zbernica je na pinoch č. 15 a 16 (15 - SCL, 16 - SDA), +5V (pin č. 18)

Na identifikované vývody SDA,SCL, GND a +5V si teda môžeme pripojiť svoj EEPROM čip.

### Software

Ak operačný systém chce načítať EDID informácie cez DDC kanál, musí teda vedieť ovládať I2C zbernicu (stačí čítanie). No z princípu I2C zbernice je však čítanie a zápis podobné (nevyžaduje rozdielny HW, a ani dodatočné signály), takže každý OS vie (by mal) čítať aj zapisovať dáta pomocou I2C.

No nie každý OS jednoducho sprístupňuje túto vlastnosť užívateľovi.

Vo Windows-e si to treba naprogramovať a hotovú utilitu na zápis som nenašiel.

Preto som zvolil linux.

Operačný systém - LINUX

Je nutné doinštalovať balíky i2c-tools a read-edid, budem ich potrebovať.

1.

```
sudo apt install i2c-tools read-edid
```

Okrem grafických výstupov, sa I2C zbernica používa v počítačoch na viacerých miestach - napríklad pri komunikácii s tepelnými senzormi na základnej doske, pri identifikácii RAM modulov atď.

Preto musím identifikovať, ktorá I2C zbernica je na mojom grafickom výstupe (konektore)

Najskôr sa pozriem, aké I2C zbernice mám v PC:

1.

```
sudo i2cdetect -l
```

```
root@edizontn-pc:/home/edizontn/Stiahnuté# sudo i2cdetect -l
i2c-3  i2c      nvkm-0000:01:00.0-bus-0003  I2C adapter
i2c-1  i2c      nvkm-0000:01:00.0-bus-0001  I2C adapter
i2c-8  i2c      nvkm-0000:01:00.0-aux-000a  I2C adapter
i2c-6  i2c      nvkm-0000:01:00.0-aux-0008  I2C adapter
i2c-4  i2c      nvkm-0000:01:00.0-bus-0004  I2C adapter
i2c-2  i2c      nvkm-0000:01:00.0-bus-0002  I2C adapter
i2c-0  i2c      nvkm-0000:01:00.0-bus-0000  I2C adapter
i2c-9  i2c      nvkm-0000:01:00.0-aux-000b  I2C adapter
i2c-7  i2c      nvkm-0000:01:00.0-aux-0009  I2C adapter
i2c-5  i2c      nvkm-0000:01:00.0-bus-0005  I2C adapter
```

Vidím, že mám 10 I2C zberníc označenie používajú 0-9.

Ďalej potrebujem zistiť, na ktorej zbernici sa nachádza EEPROM pamäť s adresou 0x50.

Môžem použiť príkaz `i2cdetect -y n`, kde `n` je číslo I2C zbernice, ktorého výstupom je zoznam adries ktoré ACK-ovali I2C dopyt (odpovedali na adresovanie) a vo výpise hľadať len tie, kde je adresa 0x50, ale toto je zdĺhavé.

Ľahšie je použiť utilitu `get-edid`, ktorá robí presne to isté postupne po všetkých dostupných I2C zberniciach, ale hľadá iba potvrdenú adresu 0x50.

1.

```
sudo get-edid
```

```

root@edizontn-pc:/home/edizontn/Stiahnuté# sudo get-edid
This is read-edid version 3.0.2. Prepare for some fun.
Attempting to use i2c interface
No EDID on bus 0
No EDID on bus 1
No EDID on bus 3
No EDID on bus 4
No EDID on bus 6
No EDID on bus 7
No EDID on bus 8
No EDID on bus 9
2 potential busses found: 2 5
Will scan through until the first EDID is found.
Pass a bus number as an option to this program to go only for that one.
128-byte EDID successfully retrieved from i2c bus 2
00000000S@!x0E0XS0($PT020`a@XDK00*0a@XDK00
20
Looks like i2c was successful. Have a good day.
(2
0LP154WE2-TLA50root@edizontn-pc:/home/edizontn/Stiahnuté# █

```

Z výpisu sa dozvedáme, že toto PC obsahuje naozaj desať I2C zberníc a EEPROM pamäť reagujúca na adresu 0x50 sa nachádza na zbernici číslo 2 a 5, pričom na zbernici 2 program prečítal 128 bytov EDID informácií. To je monitor ktorý práve používam pri práci.

Teraz pripojím na grafický konektor, pomocou ktorého chcem programovať moju I2C pamäť, funkčný LCD monitor a znova spustím get-edid

```

This is read-edid version 3.0.2. Prepare for some fun.
Attempting to use i2c interface
No EDID on bus 0
No EDID on bus 3
No EDID on bus 4
No EDID on bus 6
No EDID on bus 7
No EDID on bus 8
No EDID on bus 9
3 potential busses found: 1 2 5
Will scan through until the first EDID is found.
Pass a bus number as an option to this program to go only for that one.
Bus 1 doesn't really have an EDID...
128-byte EDID successfully retrieved from i2c bus 2
00000000S@!x0E0XS0($PT020`a@XDK00*0a@XDK00
20
Looks like i2c was successful. Have a good day.
(2
0LP154WE2-TLA50root@edizontn-pc:/home/edizontn/Stiahnuté# █

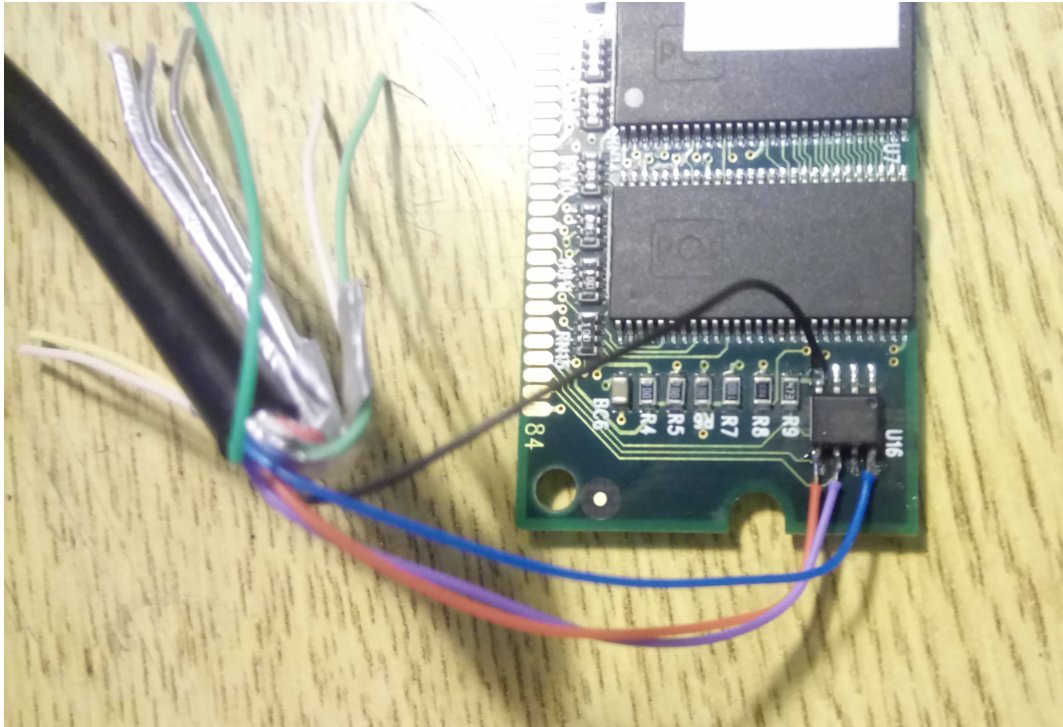
```

Na výpise následne vidím, že pribudla pamäť 0x50 na zbernici č.1 (to je práve pripojený LCD monitor). Keďže som ho pripojil na HDMI rozhranie, I2C zbernica č.1 sa nachádza teda na HDMI. Postup opakujem ale na VGA rozhraní. Tam je pre zmenu zbernica č.0

Tento postup som robil len z toho dôvodu, aby som spároval čísla I2C zberníc z fyzickou zbernicou na grafických konektoroch. **Je treba mať istotu, že zbernica s ktorou pracujem je naozaj tá, na ktorú mám pripojený svoj EEPROM čip, aby nedošlo k poškodeniu (prepísaniu) iného I2C čipu v PC.**

Teraz odpojím LCD monitor a na jeho miesto si pripojím svoj EEPROM čip, s ktorým chcem pracovať.

Použil som poškodený HDMI kábel, ktorý mal jeden koniec dobrý, tak som ho strihol a prepípaním identifikoval signály I2C. Tie som naspájkoval na svoj EEPROM čip. Vývody A0 (prípadne aj A1 a A2) pamäte nastavujem tak, aby I2C adresa bola 0x50 (není nutné-ale treba ju poznať)



Starý RAM modul som použil lebo nič lepšie nebolo po ruke 😊

Momentálne je I2C EEPROM pamäť s adresou 0x50 pripojená k zbernici 1.

Samotný zápis a čítanie je v možné pomocou príkazov `i2cget` a `i2cset`

1. `i2cget -a 1 0x50 0x00 b`

načíta byte z adresy 0x00 z EEPROM s adresou 0x50 zo zbernice č.1

1. `i2cset -a 1 0x50 0x00 0x01 b`

zapiše byte s hodnotou 0x01 na adresu 0x00 do EEPROM s adresou 0x50 na zbernici č.1

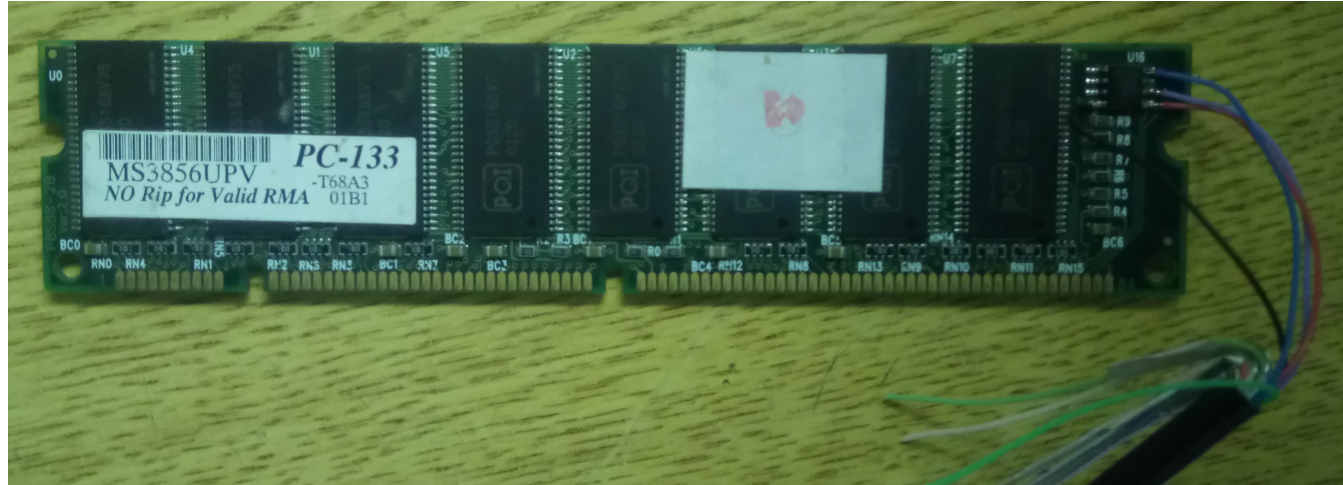
1.

i2cdump -a 1 0x50 b

vyčíta blok 256 byte z adresy 0x00 z EEPROM s adresou 0x50 zo zbernice č.1:

```
0 1 2 3 4 5 6 7 8 9 a b c d e f 0123456/89abcdef
00: 80 08 04 0c 0a 02 40 00 01 70 54 00 80 08 00 01  ??????@. ?pT.???.?
10: 8f 04 04 01 01 00 0e 00 00 00 00 0f 0e 0f 2a 20  ??????.?....???*
20: 15 08 15 08 00 00 00 00 00 00 00 00 00 00 00 00  ??????.....
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 12 9b  .....??
40: 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  .....?
50: 50 39 30 35 30 31 37 30 00 00 00 00 00 00 00 00  P9050170.....
60: 00 00 00 1d 53 33 38 35 36 55 50 56 2d 54 36 38  ...MS3856UPV-T68
70: 41 33 00 00 00 00 00 00 00 00 00 00 00 00 64 fd  A3.....d?
80: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
90: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
a0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
b0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
c0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
d0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
e0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
f0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
```

pre porovnanie - štítok na RAM module:



Príkazom i2ctransfer sa dajú kombinovať viaceré správy a poslať v jednom prenose (read+write+read+write...).

Teraz som v stave, keď viem modifikovať (čítať alebo zapísať) jednu bunku v mojej EEPROM pamäti.

Lenže, ja chcem pracovať s celou pamäťou. Linux bohužiaľ nemá (neviem o nich) nástroje na prenos ľubovoľne dlhého bloku do/z EEPROM a jeho uloženie/načítanie v binárnom súbore - čo by sa pre programovanie a načítanie obsahu EEPROM hodilo.

Použijem script!

Po hľadaní na internete som ale taký script nenašiel. Niečo tam síce je, ale nespĺňa to vyššie uvedené.

Preto som požiadal nášho člena [Fabosk](http://Fabosk) a ten mi so scriptom pomohol. Vychádzali sme zo scriptu [write-edid](http://write-edid), ktorý sme modifikovali.

Tu je: <https://github.com/Mikrozone-Server/rw-i2c>

Script poskytuje aj návratové kódy. Tak ho môžete použiť pri svojej práci sofisticovanejšie.

Napríklad - pripojenie malého LCD displeja priamo na VGA konektor a "kŕmenie" živými dátami o teplote CPU....

To musí vyzerať fakt srandovne.

Paráda. čo viac treba....

Ďakujem členovi MZ - [Fabo.sk](https://fabo.sk) za práce na úprave scriptu.

Script (ešte sa na ňom pracuje! ) je uložený na GitHub-e: <https://github.com/Mikrozone-Server/rw-i2c>

Ak by ste radi niečo doplniť, alebo v prípade problémov s ním, použite [issues](#) priamo na githube.