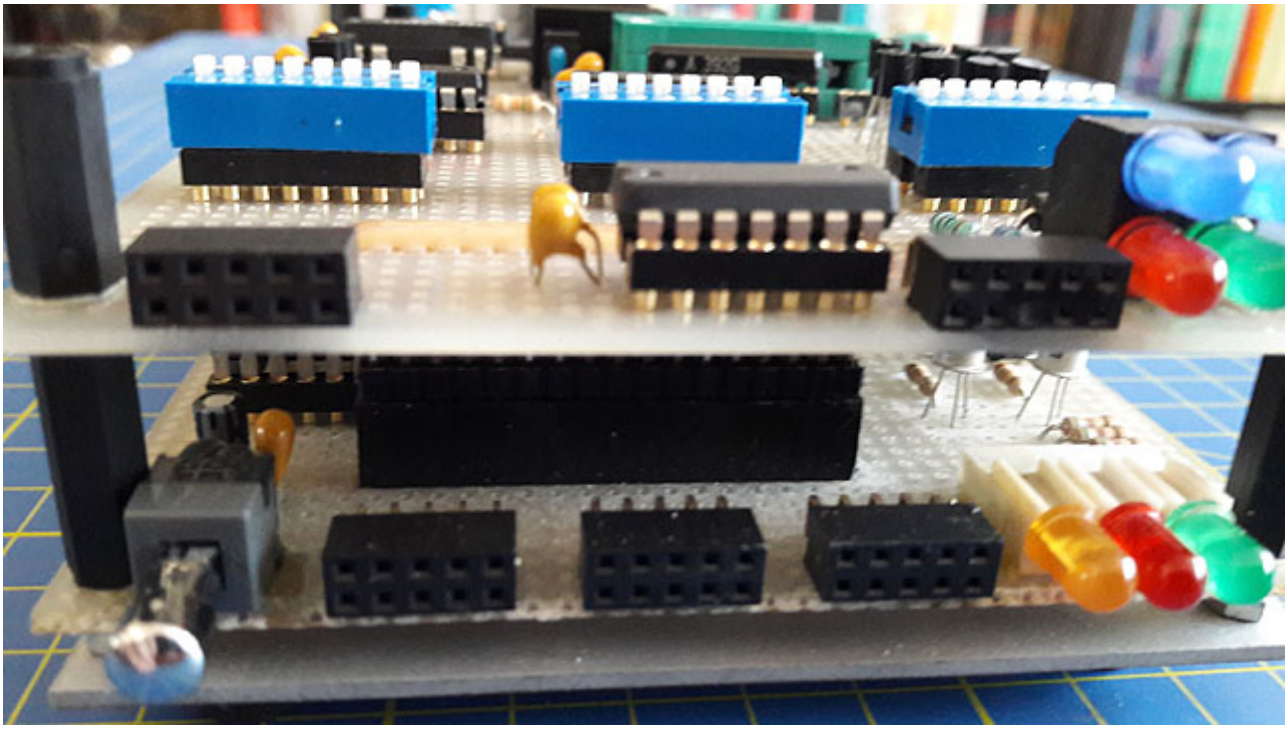
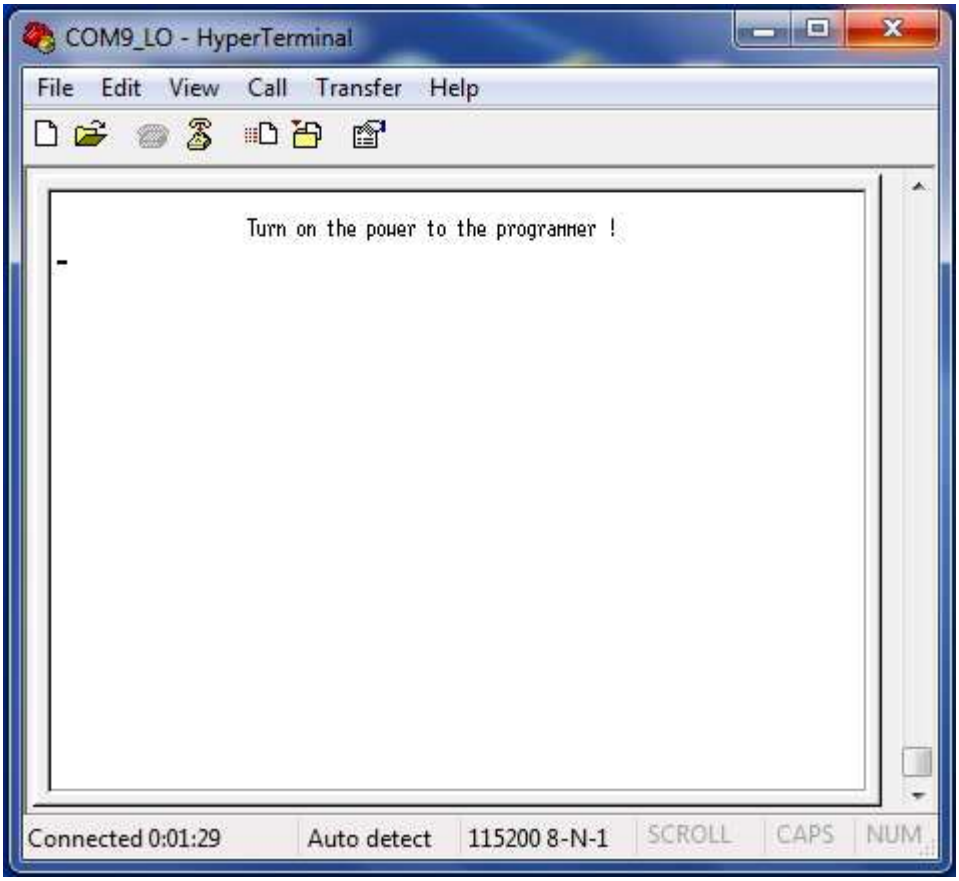
PROM BURNER 74188 - manuál:

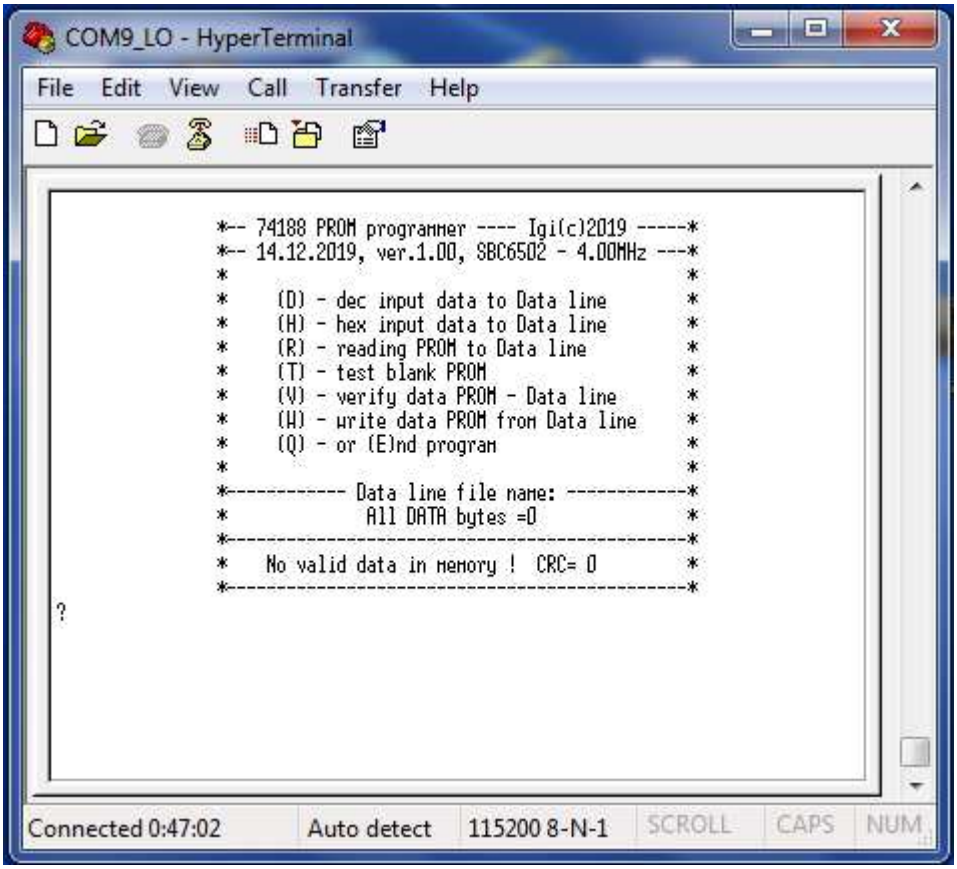


No a teraz už idem k používaniu vlastného programu, nazvem to "mini" manuálom:

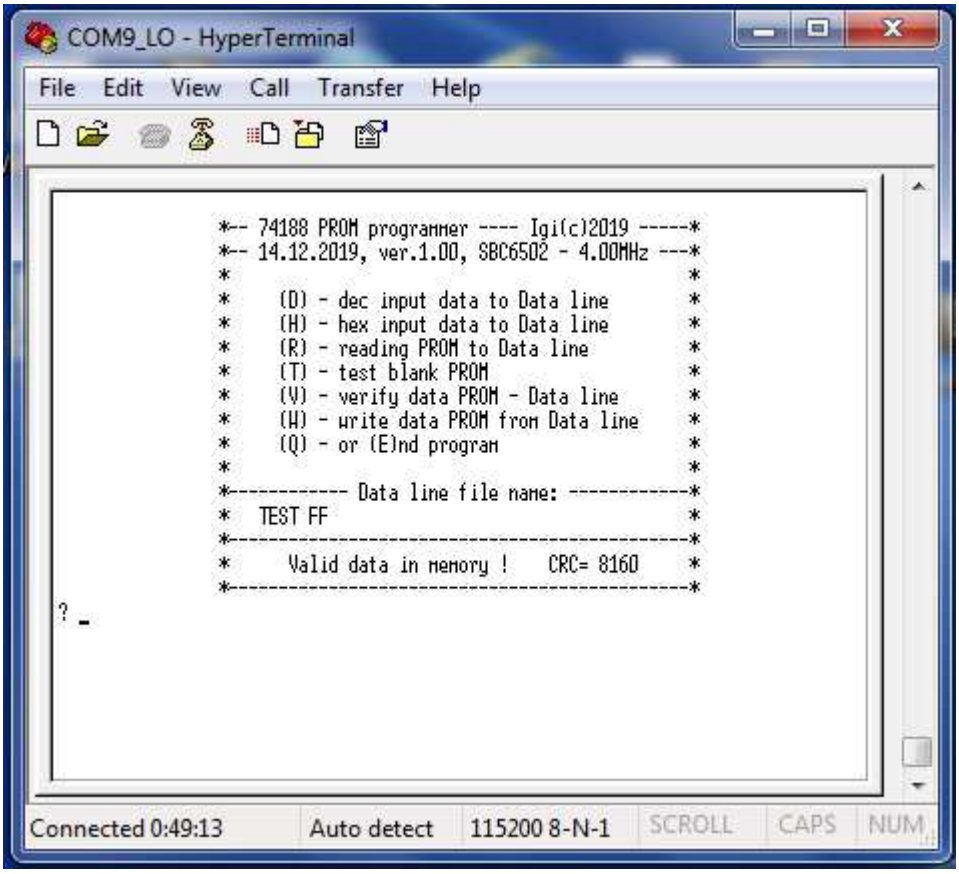


Úvodná obrazovka hneď po spustení programu, už je aktivovaný čip **8255A**, potom už môžeme zapnúť napájanie programátoru.

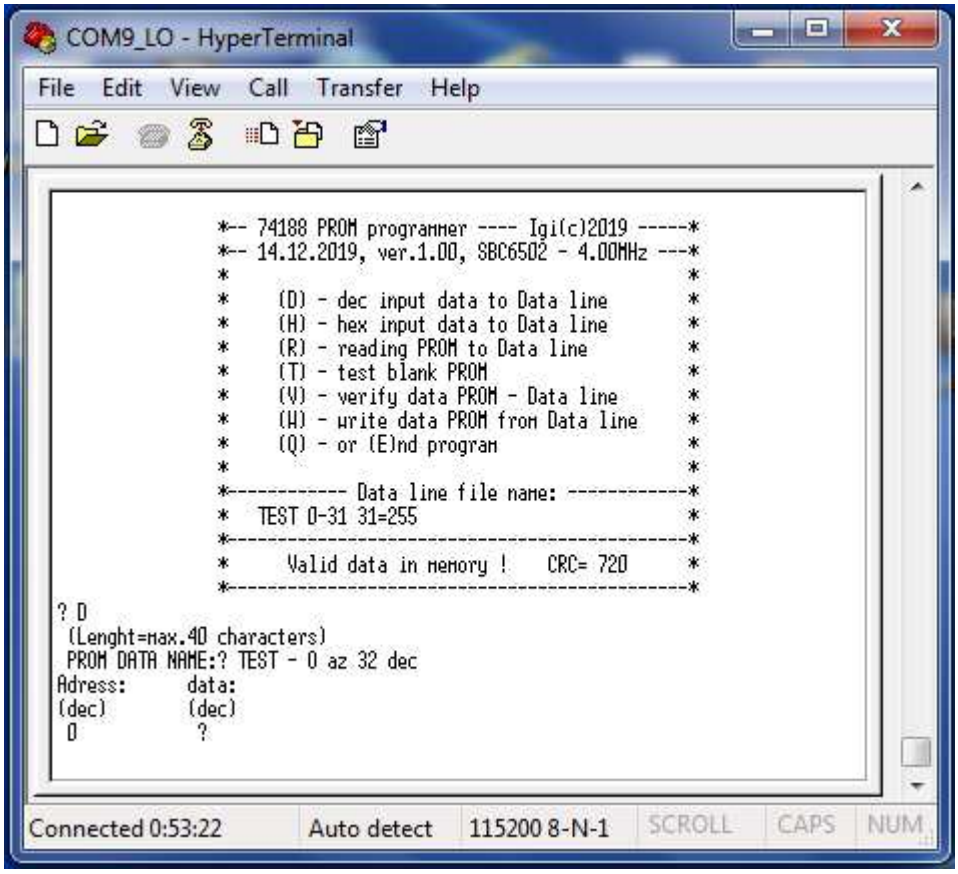
Ak budeme pracovať len s voľbou (**D**) alebo (**H**) vôbec nie je potrebné zapnúť napájanie programátoru. (V obidvoch prípadoch len ručne vkladáme data a príde k vygenerovaniu DATA riadkov.)



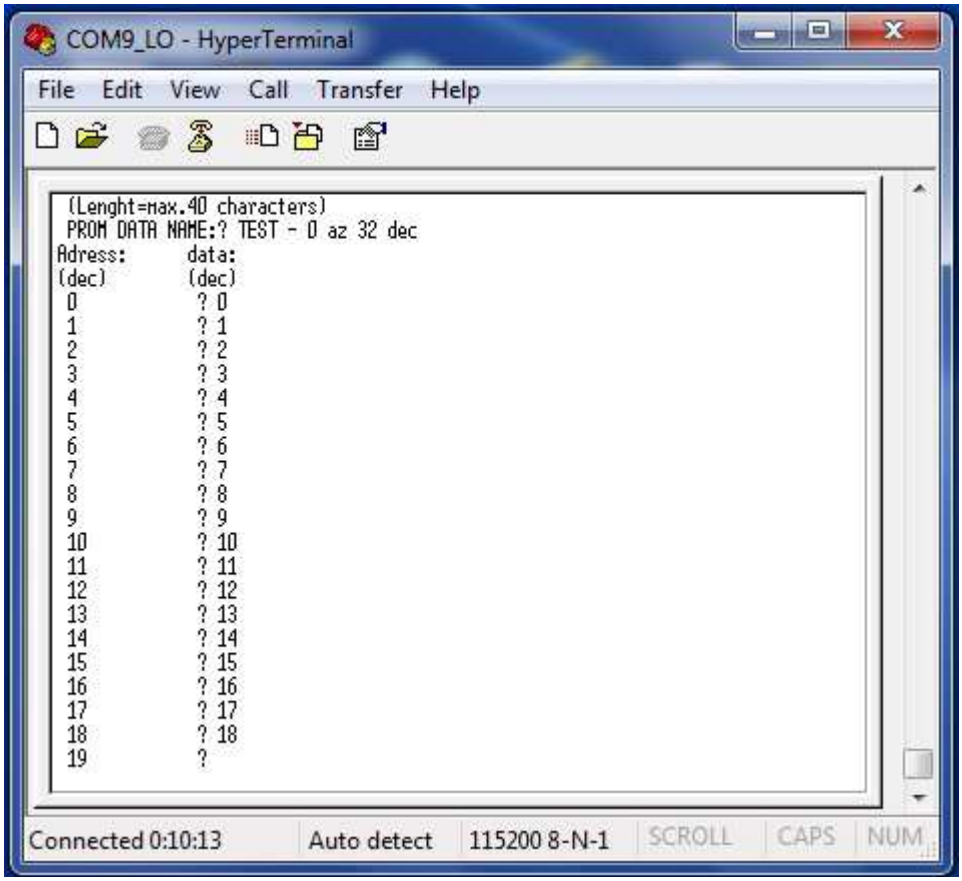
Vlastné menu, všimnite si že teraz **DATA** riadky v programe obsahujú samé **nuly (0)**, čiže v programe nie je prítomné nič na napálenie, nakoniec - vypíše sa to na displej.



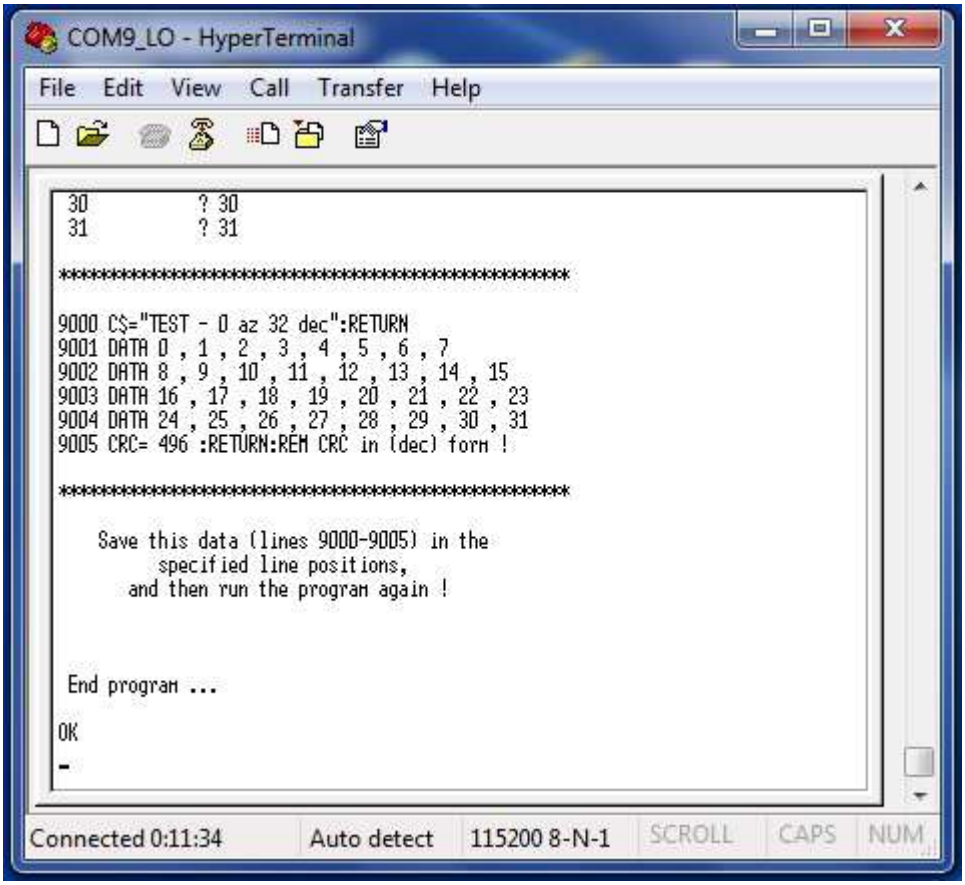
Tu som už nahral nejaký obsah do **Data** riadkov, čiže tu už máme "živé" **Data**, program na to reaguje a upozorní na to.



Zvolená možnosť (**D**), vkladáme vždy kompletne celých **32 byte**, všetky údaje sa vypisujú (aj zadávajú) v **dec** tvare, na začiatku zadáme názov. Maximálna dĺžka názvu pre **DATA** je **40** znakov, po prekročení dĺžky na to program upozorní a ponúkne možnosť nového názvu. Toto obmedzenie dĺžky názvu som si určil sám, myslím si že 40 znakov na popis dát by malo stačiť.



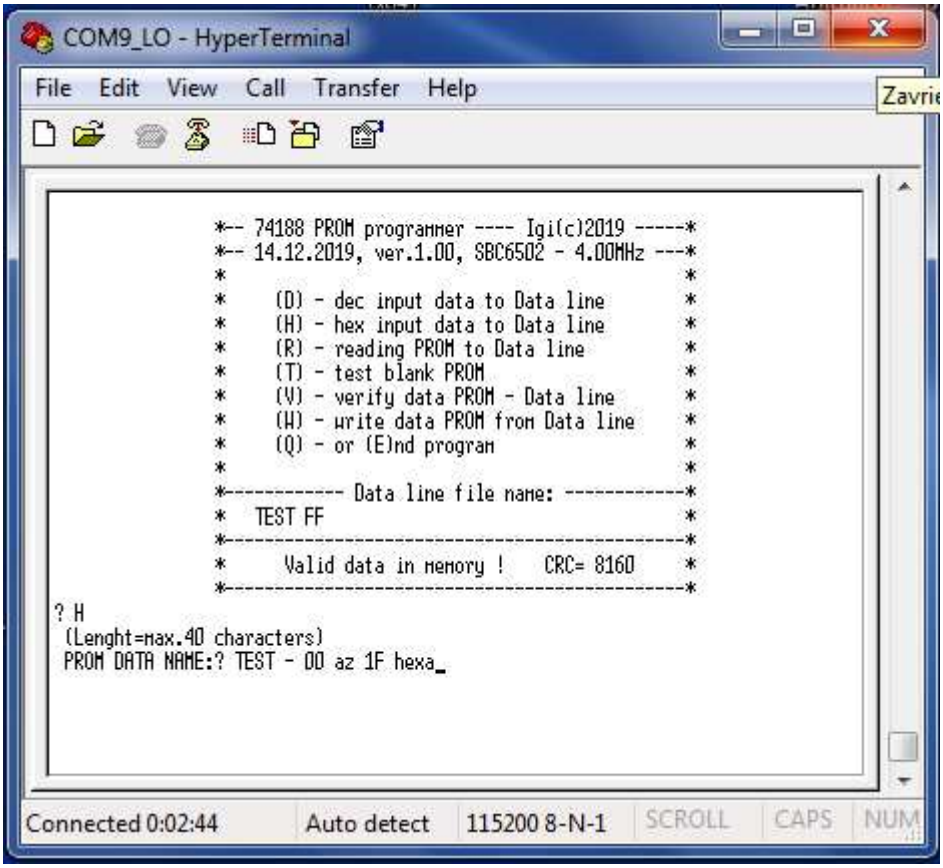
Pokračujeme vo vkladaní dát (tu je uvedený príklad) na dané adresy **PROM** až nakoniec prídeme na posledný byte č.31(dec).



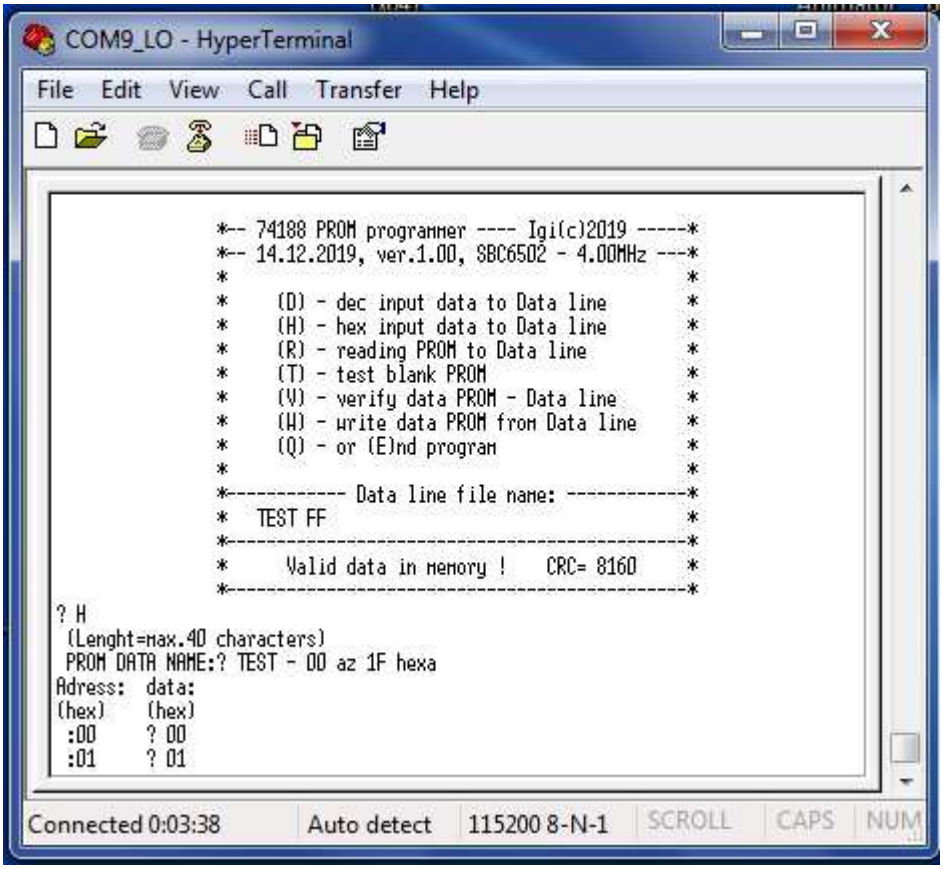
Tu už program hneď vygeneruje výsledné **DATA** riadky. Do vlastného tela programu potom vkladáme riadky **9000** až **9005** (tak ako je to popísané na displeji).

Vyrátané jednoduché **CRC** sa tak isto prenáša, pri spätnom nahratí do **SBC6502** je ***vždy*** nanovo vypočítané a kontrolované s uloženou hodnotu **CRC** - ak nesedí = koniec programu (síce je to krajná možnosť pretože to značí že je niekde naozaj problém, ale kontroluje sa to, štandartne to samozrejme zbehne bez najmenších problémov).

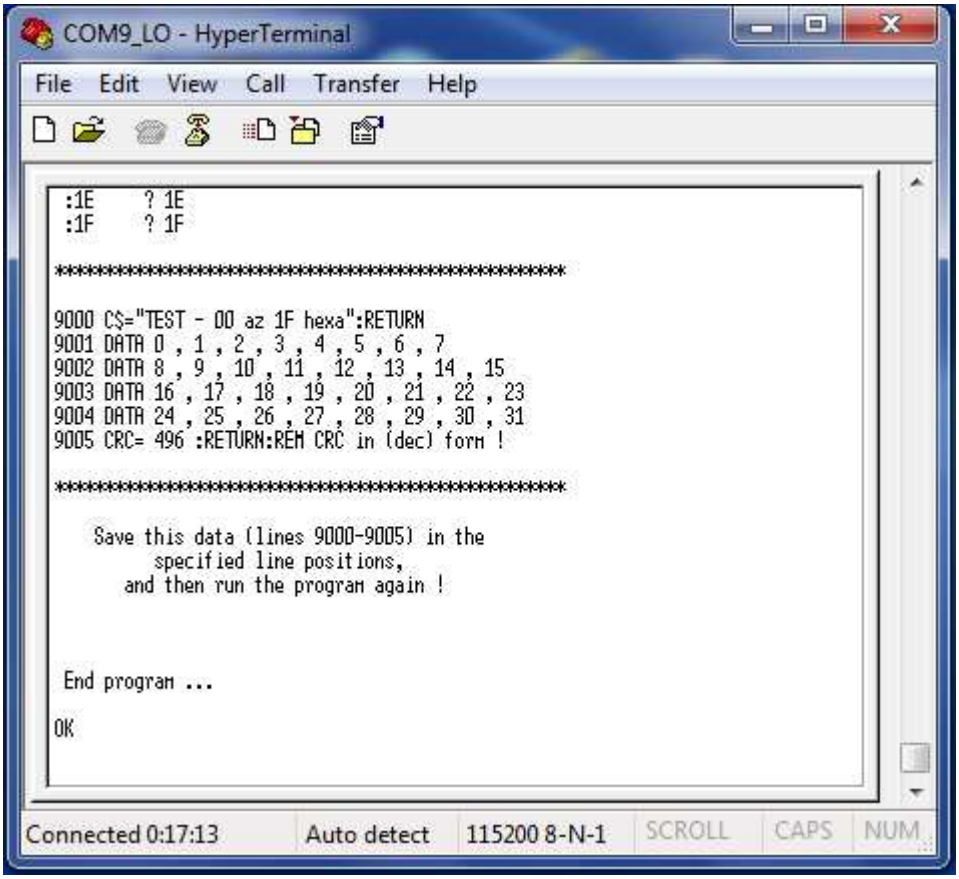
Ak je všetko O.K. program sa po vygenerovaní **DATA** riadkov ukončí a cez **CAPTURE** môžeme **DATA** nahrať buď do programu, alebo si ich odložíme pre neskoršie použitie. Výhodou je to že k tomu už ukladáme aj názov týchto dát, dá sa v tom potom ľahko orientovať.



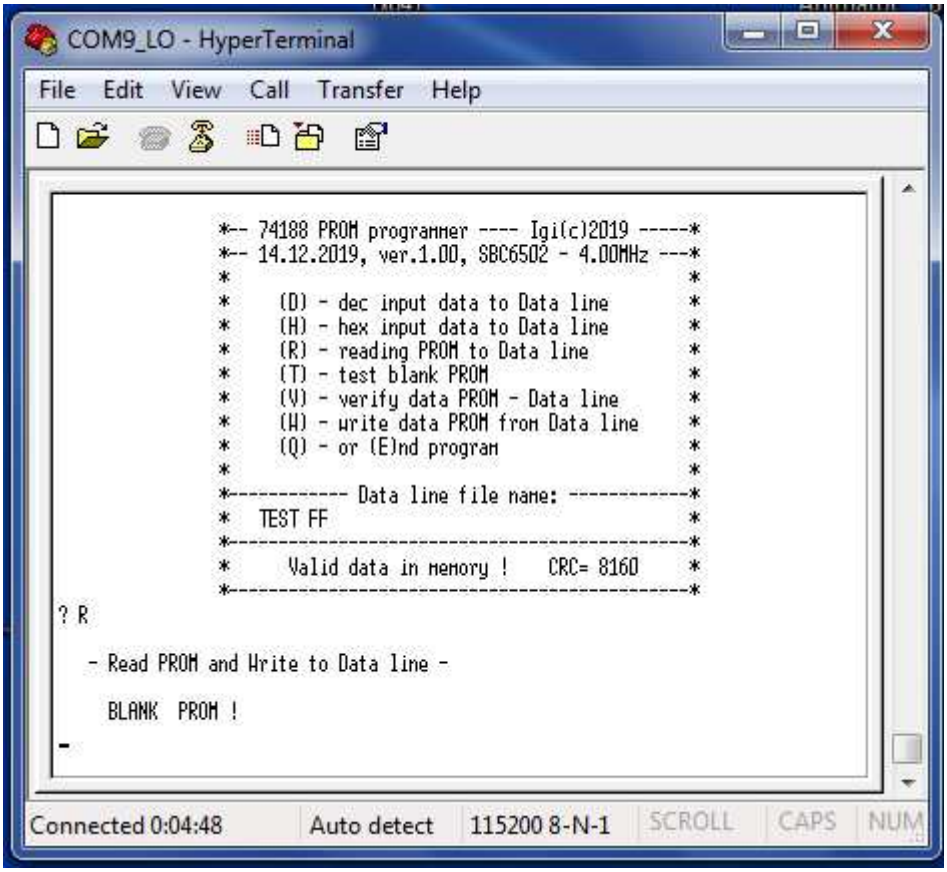
Teraz ideme ukladať data - voľba (**H**) z inými vstupnými údajmi - teraz sú všetky údaje v hexa tvare. Niekedy sú údaje na vkladanie dostupné len v tomto tvare (a my ich nemusíme prácne prepisovať do (dec) tvaru, program pri tejto voľbe to urobí za Vás. Tak isto zase ukladáme celých **32 byte** (0÷1Fh).



Možná dĺžka názvu sa nemení, adresa a data sú ale teraz zadávané v hexa tvare.

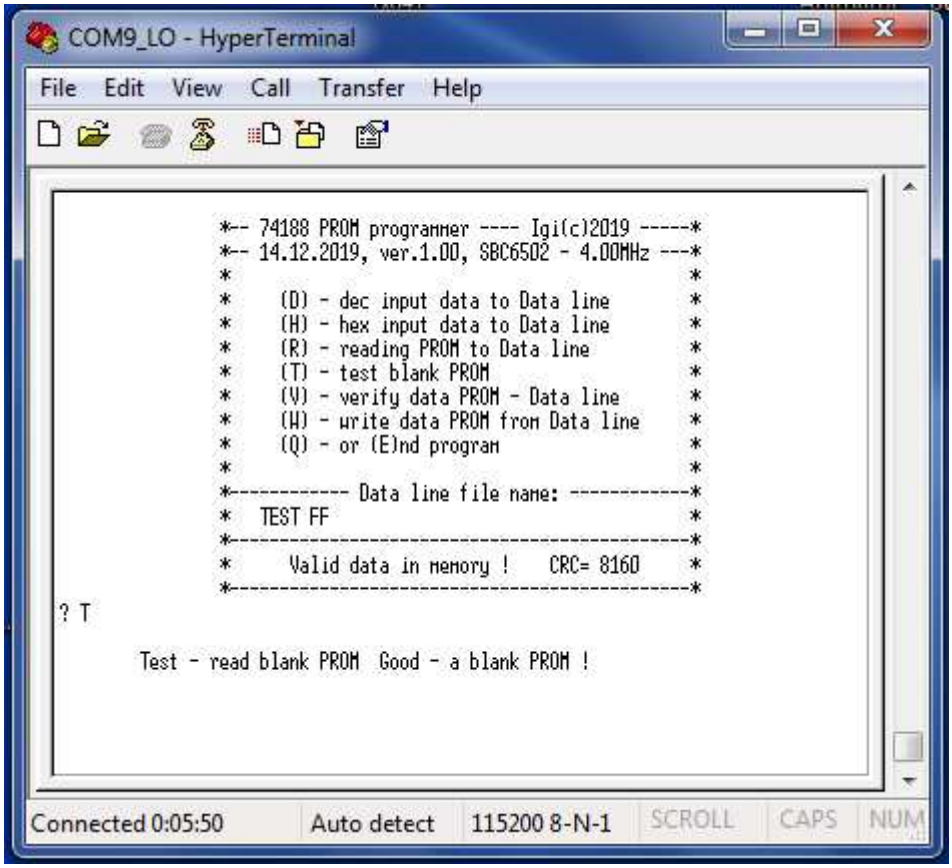


Po kompletnom načítaní údajov z klávesnice sú zase vygenerované **Data** riadky, tie sú **<u>vždy</u>** v (dec) tvare. Program sa ukončí a my môžeme cez **CAPTURE** vlastné **DATA** riadky buď niekde odložiť alebo ich nahrajeme priamo do programu.



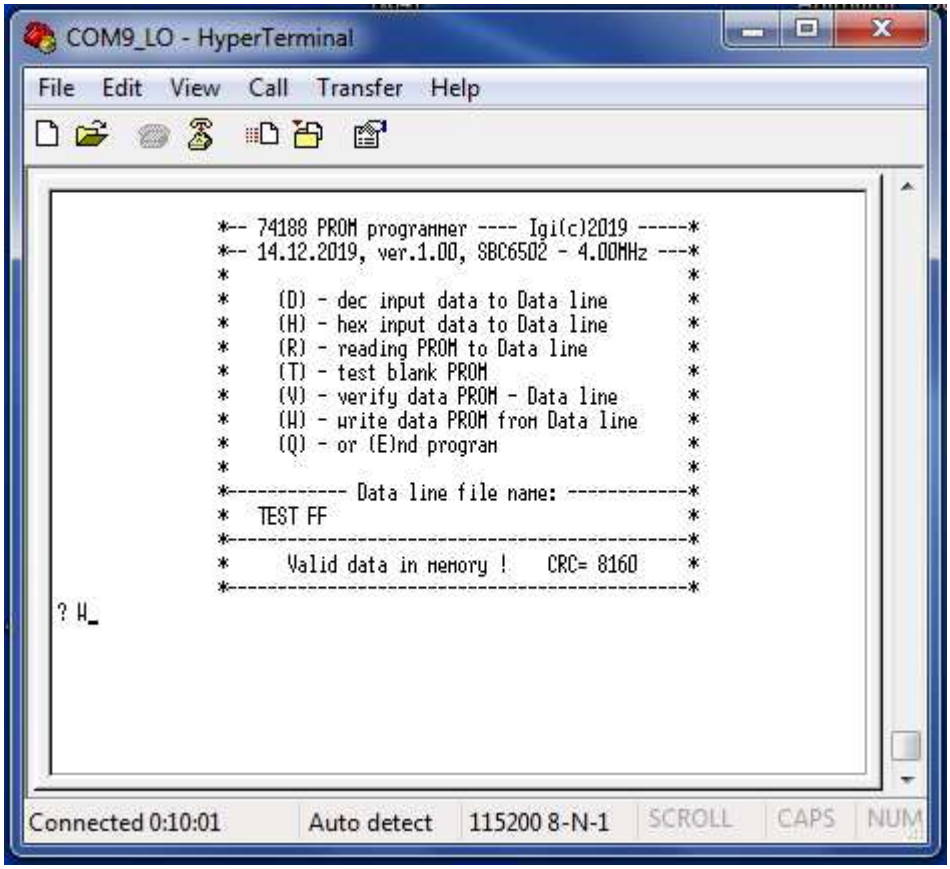
Tu pomocou voľby (**R**) sa pokúšame načítať obsah **PROM**, ak je **PROM** aktívna (naprogramovaná) tak sú vygenerované **DATA** riadky s príslušným obsahom dát, čiže ak je živý obsah - vzniknú **DATA** riadky s následným ukončením programu.

Ak je **PROM** čistá, nepoužitá, program nás na to upozorní a nevytvorí zbytočne **DATA** riadky obsahujúce samé nuly a vráti sa naspäť do menu.



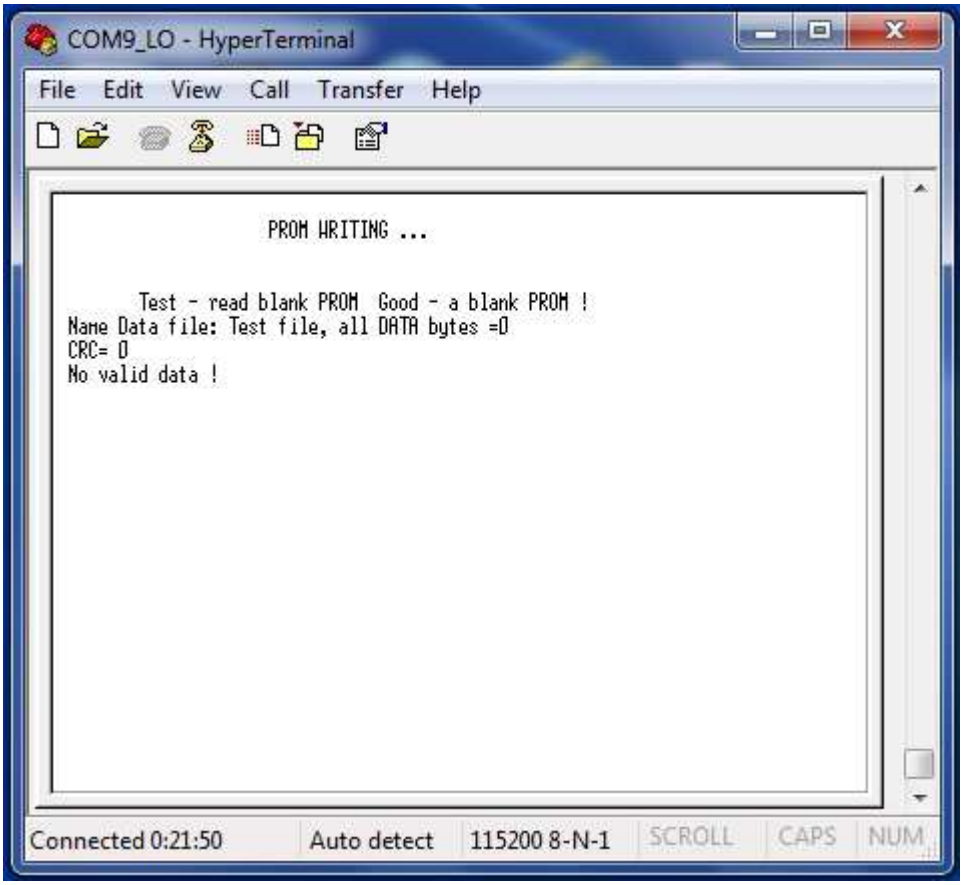
Táto voľba (**T**) nerobí nič iného - len kontroluje či máme k dispozícii čistú, nepoužitú **PROM**. Čistá **PROM** (obsah = samé nuly). Nič viac nič menej. Po požadovanej informácii sa program vráti do menu.

Ak je založená "živá" **PROM** program zase na to upozorní (tu by nebolo dobré do nej niečo zapisovať).

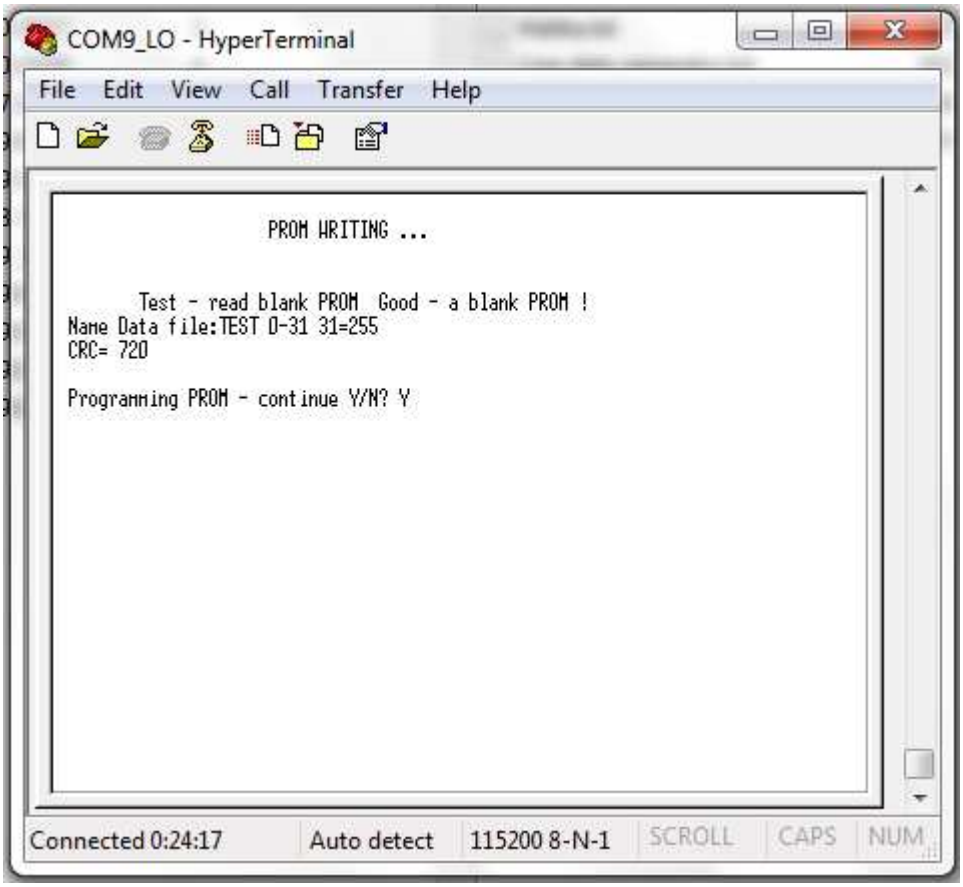


Voľbu (**V**) verifikáciu vynechávam, pretože k verifikácii prichádza tak isto vždy po napálení. Tak isto ak si nahráme potrebné **DATA** riadky tak ich vieme verifikovať zo založenou **PROM** (kontrola obsahu PROM = DATA). Podmienkou pre verifikáciu je založená **PROM** a k tomu nahraté príslušné **DATA** riadky zo žiadaným obsahom..

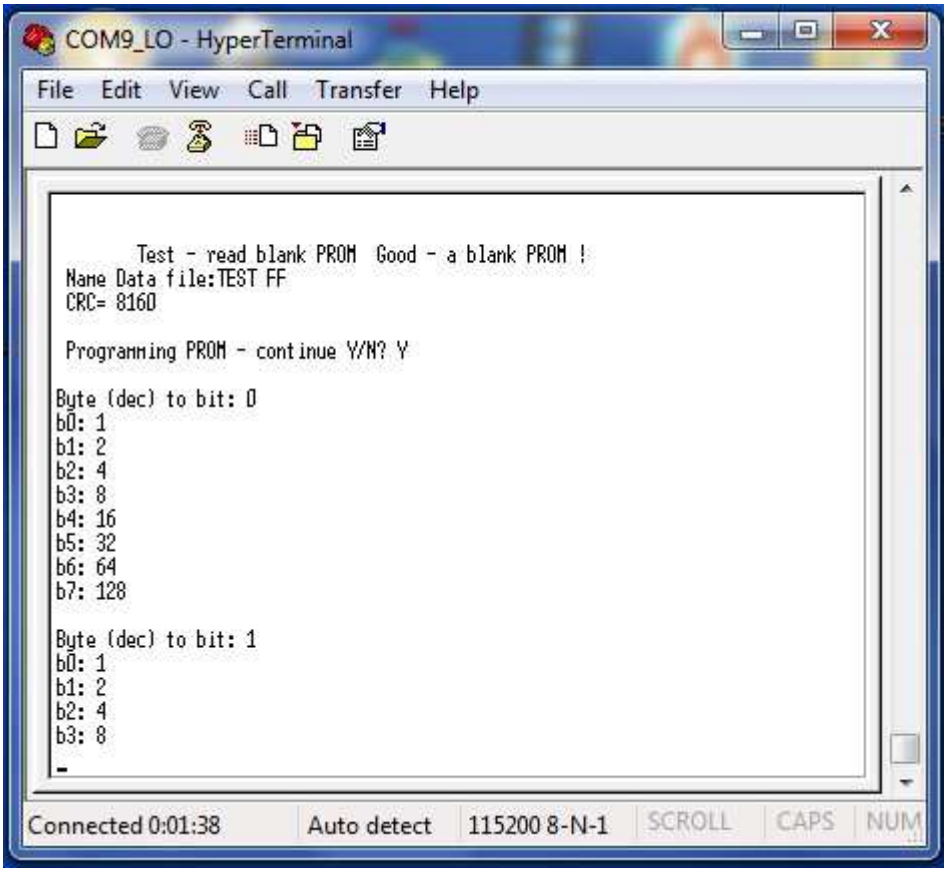
Ale poďme späť na voľnu (**W**) - zápis. Tu je viacero možností. Najprv vždy je skontrolovaná **PROM** či je čistá (t.j. musí obsahovať samé nuly), ak je teda **PROM O.K.** ide sa v programe ďalej.



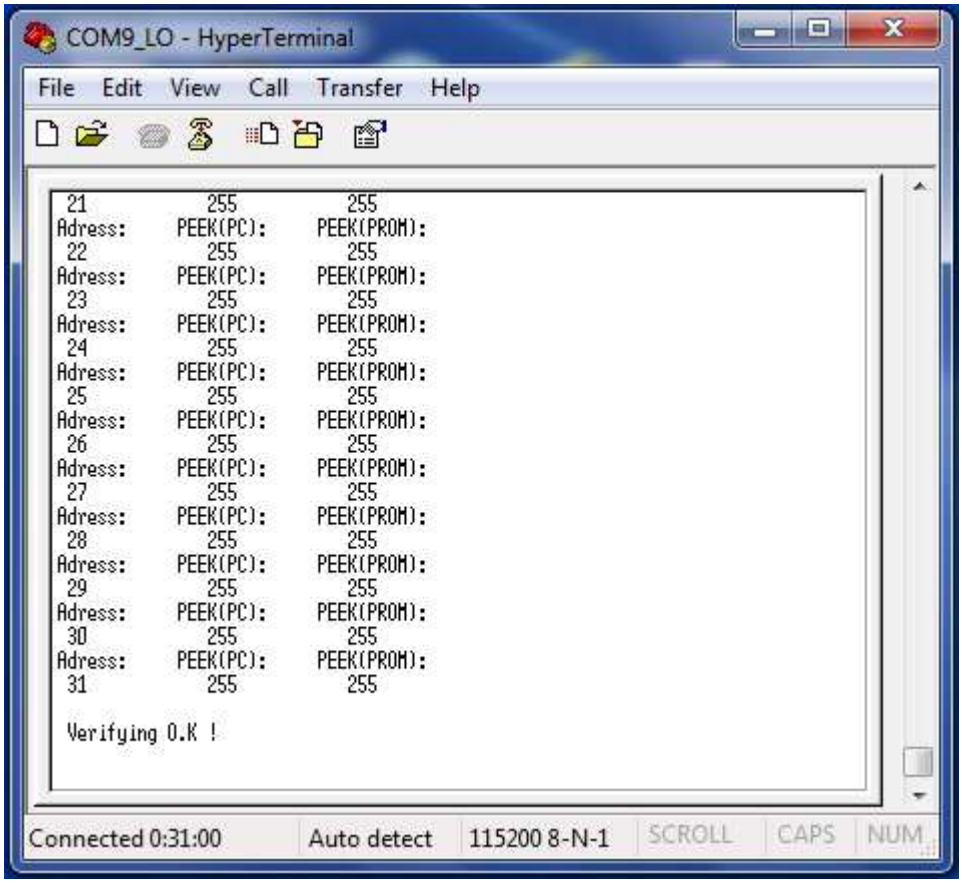
Program si sám skontroluje či ste nezaložili aktívnu (t.j. naprogramovanú) **PROM**, ak áno nedovolí pokračovať. Nedovolí tiež pokračovať ak je **PROM** čistá a nemáte v **DATA** riadkoch už uložené "živé" údaje (náš prípad na obrazovke). Vráti nás do menu. Možno sa to zdá zbytočné, ale naozaj je to treba skontrolovať.



Ak máme čistú **PROM** a tak isto máme živé **DATA**, program sa už len opýta - idete ďalej ?
N - nás vráti do menu, **Y** - začne vlastné napaľovanie.



Tu je už vidieť priebeh napaľovania, postupne sa napaľujú bity **0** až **7** v jednotlivých byte, časové slučky sú zvolené tak aby sa **PROM 74188** po programových pulzoch neprehriala, takže chvíľku to napaľovanie predsa len trvá. Ale to pohodlie oproti ručnému "dlapaniu" na jednoduchom programátore je naozaj neporovnateľné ...

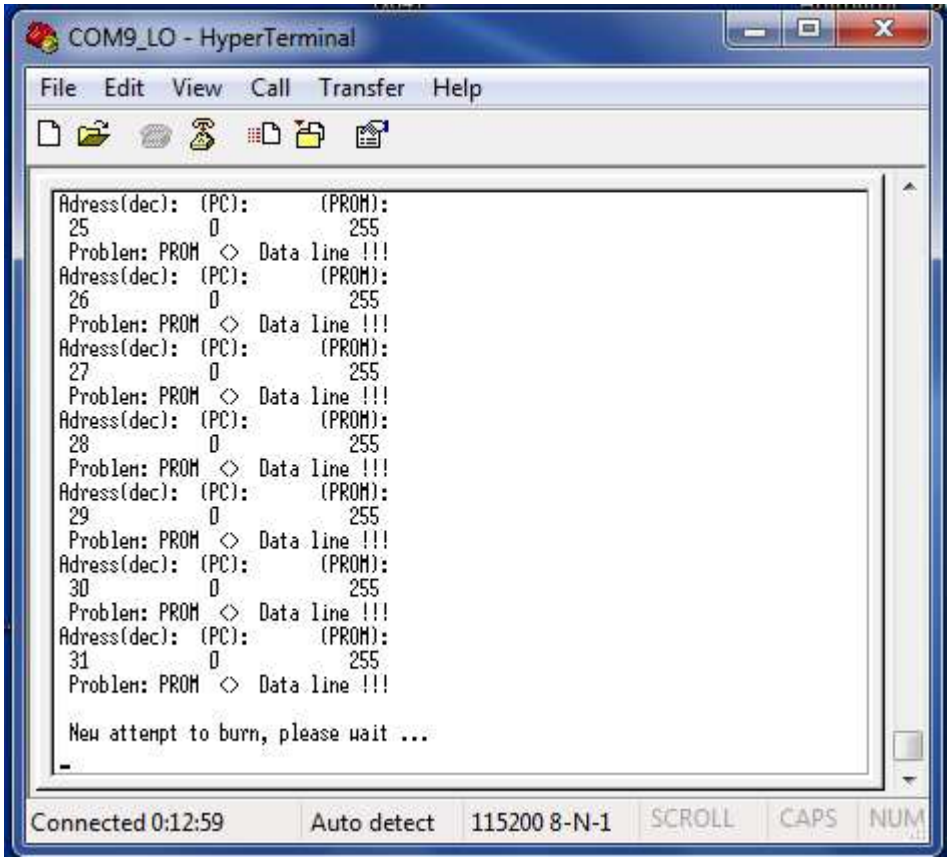


No a na záver - ak máme dobrú **PROM** a napálenie zbehlo v poriadku, tiež zbehla aj jej verifikácia - tak uvidíte túto hlášku. Po pár sekundách zmizne a vráti nás naspäť do menu. Že táto hláška po chvíli zmizne nie je na závadu, pretože ak **PROM** nie je v poriadku program sa ukončí, čiže vždy rozpoznáte či to zbehlo aj keby ste práve odišli z miestnosti. **PC** neznamená skratku pre **P**ersonal Computer ale adresu **p**ortu **C** (podrobnejšie priamo v programe). Ak uvedený spôsob nevyhovuje, stačí si príslušne upraviť programové vybavenie.

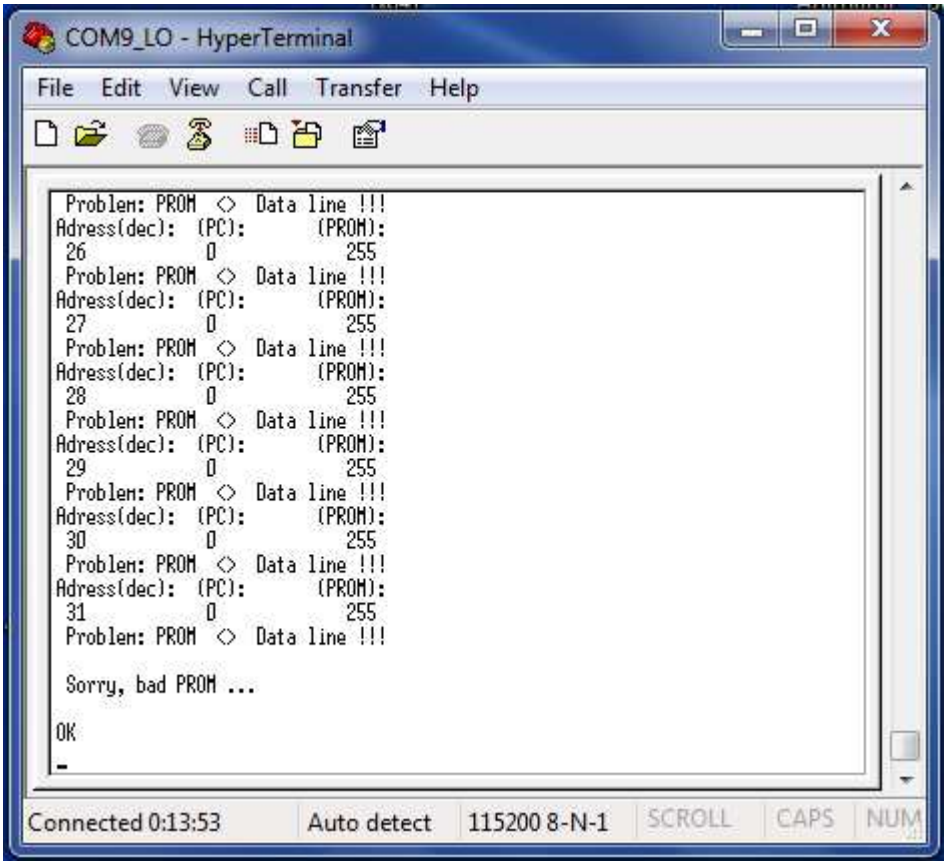
Takto to pracuje ak je všetko O.K.

No a tu by sa mohlo skončiť, lenže život občas môže priniesť aj nepríjemné prekvapenia - napálenie sa nemusí podariť, pritom zbehol kompletný celý cyklus zápisu **32byte** - čo teraz ? Programové vybavenie na to pamätá, cyklus ešte 1x nanovo zopakuje.

Ukážka čo sa stane ak napálime **PROM** a nezbehne v poriadku verifikácia:



Ukážka čo sa stane ak napálime **PROM** a nezbehne verifikácia, príde k zopakovaniu napaľovania s ďalšou následnou verifikáciou (tá je na poslednom obrázku). Vypisuje sa rozdiel na danej adrese toho čo je v počítači a čo je v **PROM**. Ak pri druhom pokuse napaľovanie zbehne a následne zbehne bez chyby aj verifikácia je **PROM** považovaná za dobrú (potvrdí to aj oznam na obrazovke) a program sa vráti do menu. Ak nezbehne verifikácia tak sa dostávame k poslednému obrázku.

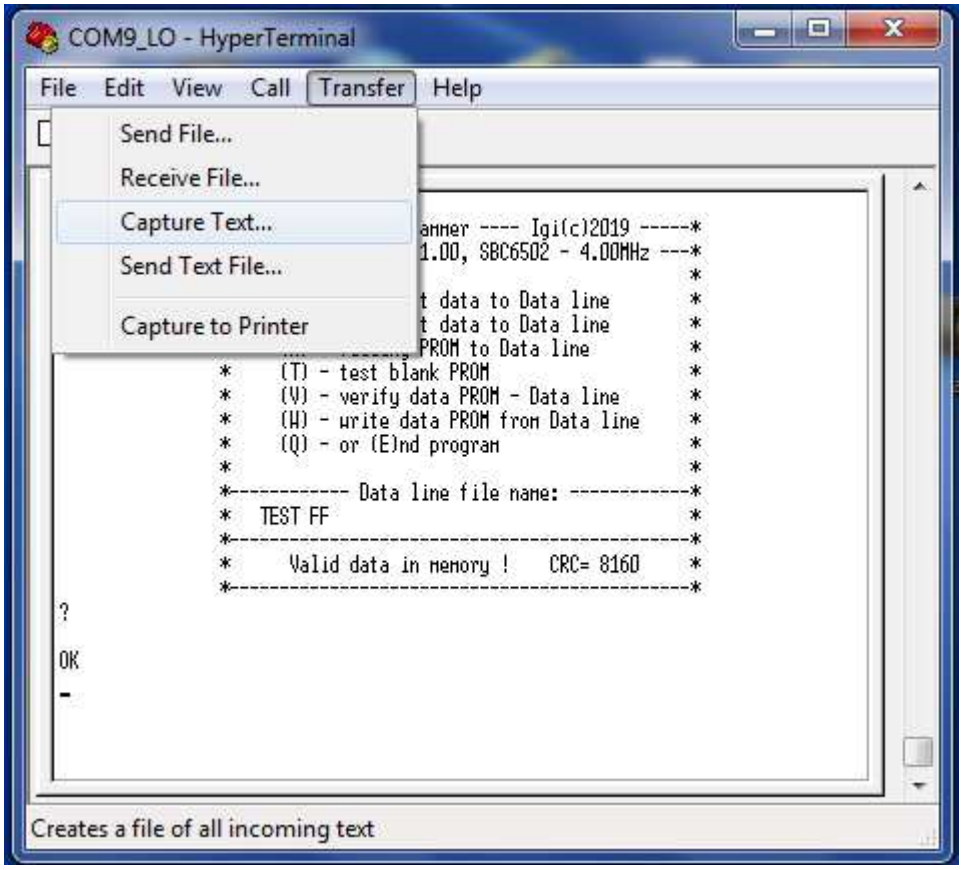


Ak nezbehne ani po druhom pokuse o napálenie správne verifikácia - **PROM** pamäť je vyhlásená za vadnú a program končí. Je to síce výnimočný stav, ale je dobré pripomenúť že ovládací program počíta aj s takouto alternatívou.

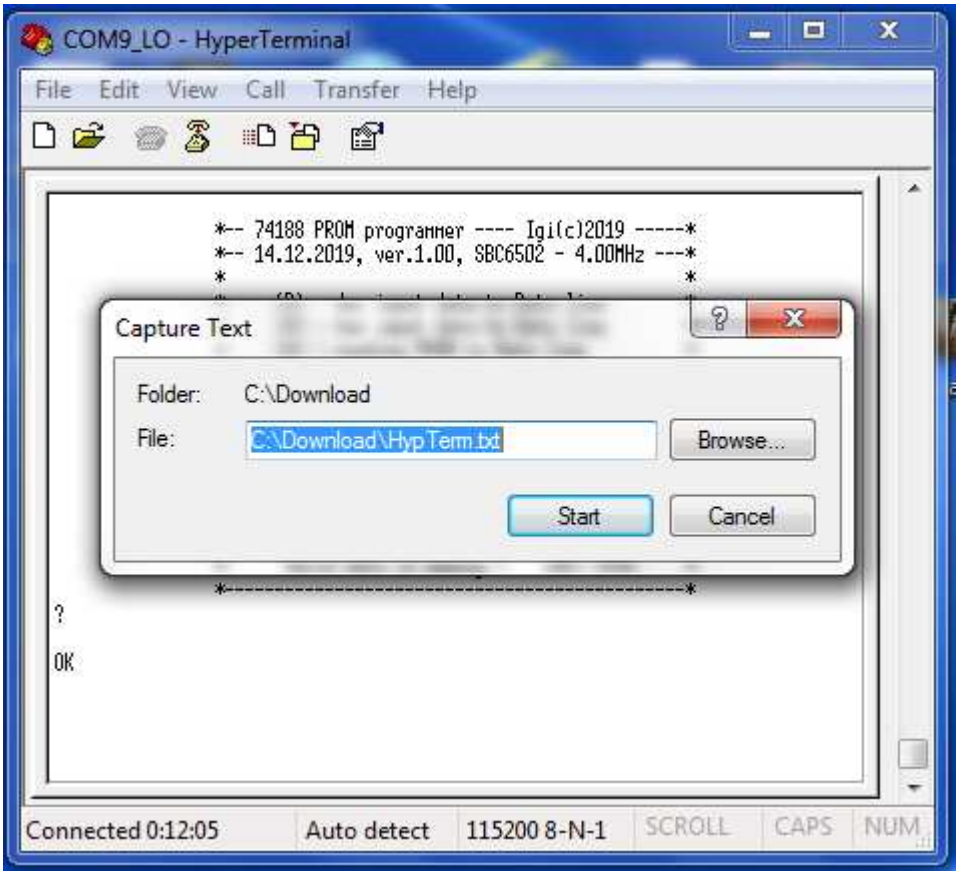
(Síce toto tam mám, ale zatiaľ som sa reálne k tejto hláške nedostal - nechýba mi, túto konkrétnu chybu "simulujem" len vybratím **PROM** z pätičky).

Doplnková informácia - ako vkladať DATA riadky do programu:

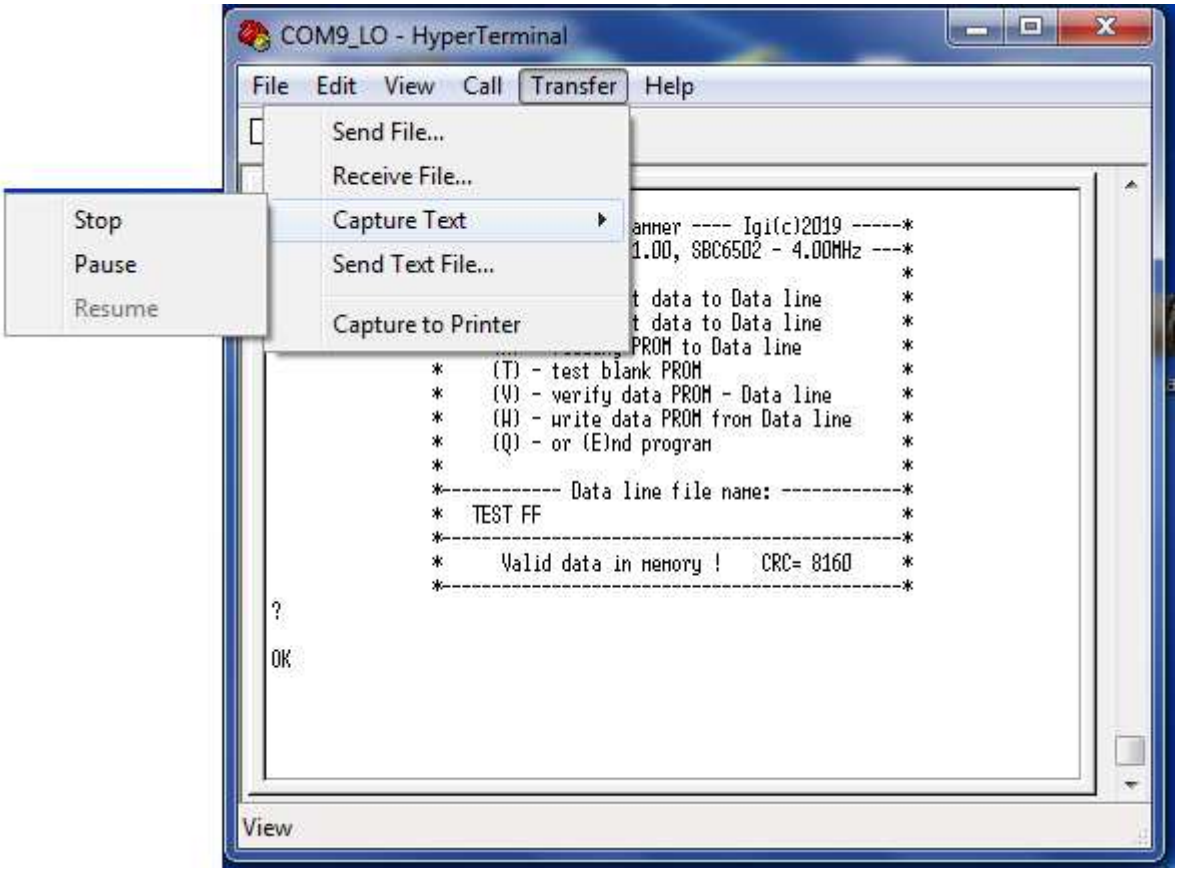
Ešte sa raz vrátim k tomu ako si zaznamenať obsah toho s čím sa robí v **HyperTerminál**i, je to funkcia **Capture** - toto nám umožní si zaznamenať aj **Data** riadky ktoré program vygeneruje. Už som spomínal že **SBC6502** a jeho **BASIC** nemá implementované príkazy **LOAD** a **SAVE**, tak tento "nedostatok" musím nejako obísť. Ako ? Takto ...



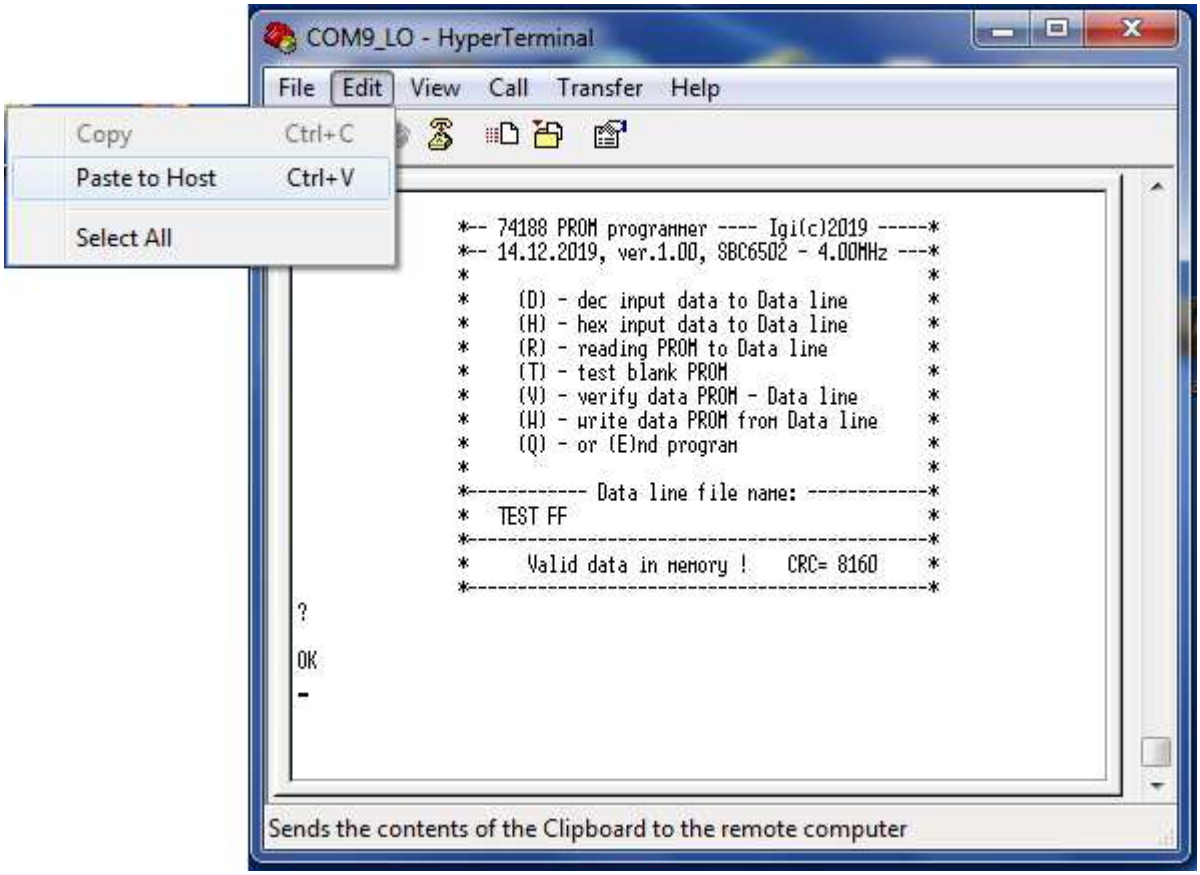
Pred spustením funkcie z **MENU** programu, napríklad **D** alebo **H**, tak isto **R** spustíme záznam do txt súboru cez **CAPTURE**.



Zvolíme adresár kde sa bude zapisovať a meno vytváraného súboru. Mnou používanú cestu kde sa vytvorí txt súbor vidíte na samotnom obrázku.



Ak zbehne nami požadovaný záznam tak ho následne môžeme stopnúť, otvoríme (napríklad V Notepade) nami vytvorený .txt súbor a môžeme **DATA** riadky priamo skopírovať do nášho programu - alebo si záznam odložíte pre neskoršie použitie. **DATA** riadky obsahujú názov - čiže nie je problém neskôr vedieť k čomu sa **DATA** riadky vzťahujú.



Ak **DATA** riadky chceme vložiť do programu - v textovom súbore (zo záznamu **Capture**) označíme požadované **REM** riadky a pomocou **CTRL+C** si ich zapamätáme a následne pomocou **Paste to Host** vložíme priamo do programu. Presne takto isto môžeme aj nahrať celé programové vybavenie jedným ťahom. Tento zložitejší spôsob vkladania **DATA** riadkov je daný tým že **SBC6502** nemá v **BASIC**u príkazy **LOAD** a **SAVE**, tak som to musel nejako obísť.

No a to je asi tak všetko.

Igi (c) 2020