

# Orodje za upravljanje z Margento terminali

Matej Kseneman<sup>1,2</sup>, Franc Horvat<sup>1</sup>, Amor Chowdhury<sup>1,2</sup>

<sup>1</sup> Margento R & D d.o.o, Gosposvetska c. 84, 2000 Maribor

<sup>2</sup> Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor, Slovenija

E-pošta: [matej.kseneman@gmail.com](mailto:matej.kseneman@gmail.com)

## Automatic Margento terminals programming environment

*The following article presents a novel approach in a methodology for programming terminals via serial or USB port. The main objective of this software is to incorporate all data into a single file, so the users can quickly program the end-point terminals. This way it's possible to exchange terminal data between developers and end users. This software also has the ability to produce a file that is used for remote flash procedure, thus not needing a physical access to the terminal. Modular structure of the firmware enables us to select only important and crucial elements, therefore saving up a lot of precious flash memory space.*

### 1 Uvod

V podjetju Margento stremimo k univerzalnemu orodju, ki bo sposobno operirati z vsemi obstoječimi terminali, tako na razvojnem, kot tudi uporabniškem nivoju. To smo ločili z namenom, kajti enako orodje uporabljajo tudi končni kupci, saj lahko sami spreminjajo določene tekstovne nize, menjavajo raznorazne logotipe, itd. Po drugi strani pa želimo razvijalcem omogočiti karseda modularno zasnovo, saj s tem zelo poenostavimo in avtomatiziramo proces izdelave različic za raznorazne potrošniške trge (npr. Egipt, Šri Lanka, Savdska Arabija, Slovenija, itd.). Modularna zasnova seveda zahteva razdrobljenost pomnilnika, kajti različni moduli imajo dostop do skupnega prostora, ki je skupen vsem, a hkrati želimo, da se specifična različnih modulov shranjuje v zato rezerviran prostor. Vse skupaj povežemo v neko smiselno celoto v obliki datoteke, ki je prenosljiva in kjer se s povezavo na terminal lahko konfigurira končni produkt. Tako je vzporedno z razvojem novih terminalov nastajal tudi programski produkt Margento Terminal Tool.

Terminali, ki so plod razvoja podjetja Margento, temeljijo na arhitekturi DSP (TMS320F28235 [1]), vendar novejša različica že vnašajo ARM arhitekturo, ki jo najdemo dobesedno povsod (mobilni telefoni, tablice, mikrokontrolerji, itd.). Z novejšo ARM arhitekturo (Cortex™-M3 [2]) vnašamo velike spremembe v delovanje terminalov, vsaj kar se tiče prenosa podatkov med računalnikom in terminalom, saj ti terminali komunicirajo preko USB serijskega protokola.

Članek predstavlja orodje Terminal Tool namenjeno Margento terminalom, v katerem ima uporabnik ali razvijalec možnost ustvariti modularno zasnovo, kar

nam omogoči boljšo sledenje revizijam modulov in konec koncev prihranimo tudi na prostoru pomnilnika. Uporabnik lahko spreminja nastavitve terminala, ki jih združimo v posamezne segmente: nastavitve terminala v obliki parametrov, zvoki terminala, raznorazni logotipi in izpisi na zaslonu, itd. Poleg tega je uporabniku na voljo še izdelava datoteke, ki se posreduje na MP3 strežnik, ki omogoča posodabljanje terminalov na daljavo [3].

### 2 Strojna oprema

Med novejših terminalov spadajo predvsem terminali z oznako mPOS T4500 (TMS320F28335 [4]), Add-on nPOS T5000 in vendPOS (LM3S9B96 [5]), kasneje se pojavi tudi USB dongle, temelječ na ST32F415RG [6]. Slik3 1-3 prikazujejo novejša tipa terminalov.



Slika 1. Margento mPOS Terminal T4500



Slika 2. Margento Terminal Add-on



Slika 3. Margento VendPOS

Terminali se razlikujejo po uporabi, tako je mPOS terminal namenjen predvsem mobilnemu plačevanju, od katerih omenimo najpomembnejše, in sicer plačevanje z NFC karticami, mobilno plačevanje, program zvestobe in bonusa v trgovinah, plačevanje računov, mestni promet, itd. Margento Terminal Add-On (nPOS T5000) je namenjen predvsem kot dodatek POS terminalom, s tem pa naročnikom ni potrebna menjava trenutnega POS terminala, saj Add-on terminal nanj le priključi. Na ta način odjemalec pridobi dodatne funkcije, ki so podobne mPOS terminalu, in sicer: brez-kontaktne plačila, program zvestobe in bonusa, polnjene mobilnih računov, plačevanje računov, itd. VendPOS je namenjen avtomatom s pijačo, Dongle pa bo skrbel predvsem kot dodatna naprava, saj preko USB OTG razširi delovanje pametnih telefonov in tablic, s tem da jim doda možnost branja RFID kartic itd. Namenjena bo predvsem taksistom, ki bodo lahko prevažali potnike tudi z uporabo Urbana ali kakšnih drugih plačilnih kartic.

### 3 Programska oprema

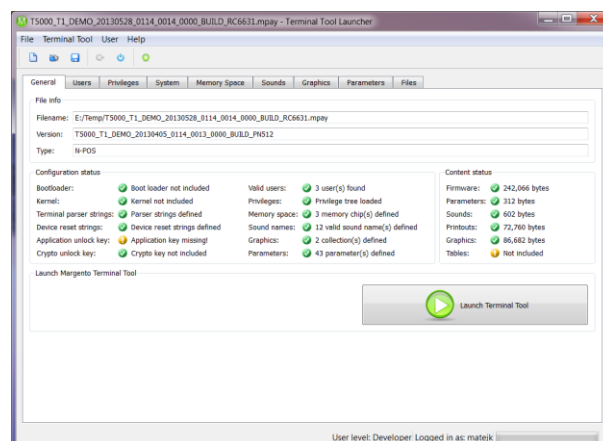
#### 3.1 Kratak opis programskega okolja

Orodje za upravljanje Margento terminalov temelji na knjižnici Qt 5.3.1[8], ki skrbi za uporabniški vmesnik cele palete operacijskih sistemov (Linux, Windows, Mac), vendar smo se pri razvoju orodja omejili predvsem na Windows okolje. Povezava med orodjem Terminal Tool in terminalom se vrši preko serijske RS232 povezave (za starejše tipe terminalov; do T4500) in preko USB 2.0 serijskega vmesnika, na katerem temeljijo vsi novejši terminali (T5000, vendPOS in Dongle).

Vzpostavljane povezave poteka med zagonim nalagalnikom in orodjem, in sicer smo v ta namen implementirali API funkcije, ki nam omogočajo komunikacijo in seveda tudi neko akcijo na strani terminala, npr. programiranje ali brisanje določenega sektorja pomnilnika. S povezovanjem želimo čim manj obremenjevati glavno programsko nit, zato ustvarimo dinamične knjižnice (.dll datoteke), ki v svoji niti skrbijo za povezavo in komunikacijo med API terminala in programskim orodjem. Med API funkcije spadajo

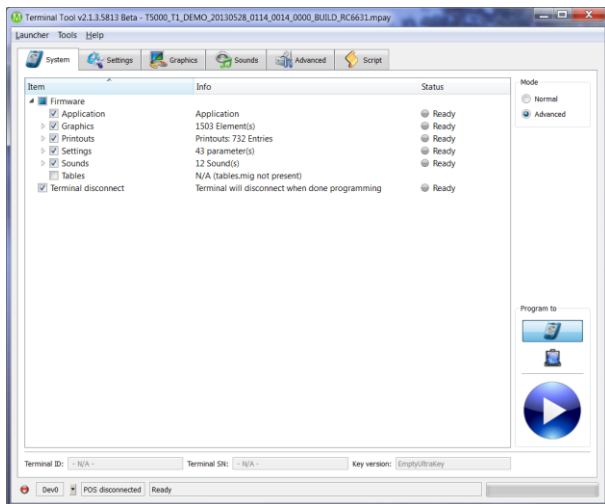
operacije kot so brisanje Flash pomnilnika, pisanje na Flash pomnilnik, verifikacija zapisanih podatkov, odklepanje in zaklepanje procesorja, itd.

Za lažjo upravljanje z vsemi nastavitvami in komunikacijo je vpeljan uporabniški vmesnik, v katerem je mogoče spreminjati docela vse parametre terminala. Programsko orodje je razdeljeno na dva povezana sklopa, in sicer sta to Terminal Tool Launcher in Terminal Tool. Prvi je namenjen osnovnim nastavitvam datoteke in konec koncev tudi programskega dela terminala, saj združuje lastnosti, kot so dovoljeni uporabniki, njihovi privilegiji, sistemski parametri s terminalskimi ključi, razporeditev pomnilnika (še posebej pomembno pri modularni zasnovi), zvoki in slike terminala ter parametri, ki skrbijo za pravilno delovanje modulov (npr. IP naslov centra). Omenjen del si lahko predstavljamo kot nekakšno lupino, saj definira le naslove pomnilnika in za kakšne tipe podatkov gre, nič pa nam ne pove, kaj je zajeto v njih. Za slednje pa poskrbi drugi sklop, tj. Terminal Tool. Slika 4 prikazuje izgled prvega sklopa orodja Terminal Tool Launcher.



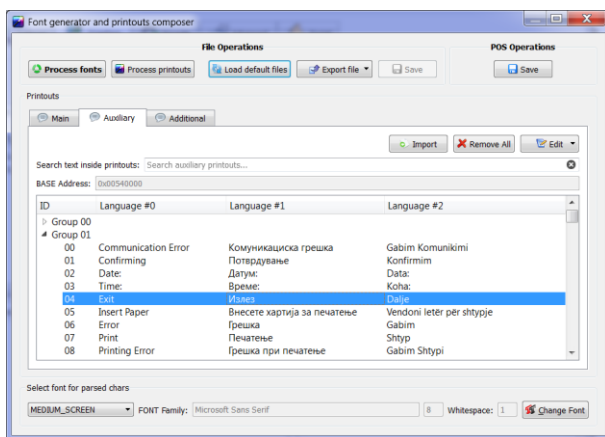
Slika 4. Margento Terminal Tool Launcher

Drugi sklop programskega orodja skrbi za to, da kreirane pomnilniške segmente dejansko napolni s podatki, ki jih kasneje uporablja terminal. Kot primer vzemimo npr. eno črko slovenske abecede. Seveda moramo na nek način to črko, ki lahko ima različne tipe pisav, različne velikosti, itd. predstaviti tudi na terminalski strani. V ta namen v programskem okolju iz izbrane pisave in velikosti črki generiramo polje zlogov, ki vsebujejo postavljene bite v matrici, s katero se predstavi ta črka. Torej vse kar je potrebno storiti grafičnemu gonilniku terminala, je, da iz omenjenega polja pravilno pobarva slikovne točne na zaslonu terminala. Po enakem kopitu so definirani tudi parametri terminala, melodije, ki se zaigrajo ob določenih stanjih transakcije, grafični logotipi in tekstovni nizi. Slika 5 prikazuje orodje Terminal Tool, kjer vidimo velik modri gumb, s katerim naložimo celotno vsebino na priključen terminal ali izdelamo datoteko, namenjeno daljinskemu posodabljanju terminalov.



Slika 5. Margento Terminal Tool

Ker je programsko orodje prisotno na več različnih tržiščih, ki imajo zelo različne pisave, smo končnim uporabnikom olajšali delo z vnosom znakovnih nizov, tako da lahko le-te spreminjajo v samem programskem orodju. Slednje nato poskrbi, da se med grafične elemente dodajo vsi potrebni znaki, ki so nujni za izpis omenjenih nizov (v primeru, ko je problem prostor na terminalu) ali vseh iz določenega kodnega podsistema Unicode. Znakovni niz se nato lahko predstavi le kot niz ID številke teh slikic znakov. Navadno se uporabi kar Unicode ID, vendar v vseh primerih to ni mogoče. Tipičen primer je npr. Šri Lanka, kjer so šele leta 2007 dobili vsega okoli 60 znakov v tabeli Unicode, a ti znaki se kombinirajo, tako da na koncu nastane okoli 300 znakov, ki pa jih je potrebno pretvoriti v zapis terminala.



Slika 6. Vnos znakovnih nizov v več različnih jezikih

### 3.2 Modularna zasnova in povezovanje z novejšimi terminali

Tekom razvoja terminalov smo že opravili prehod na modularno sestavo aplikacij [7], s katero dosežemo boljše upravljanje aplikacije in daljinsko posodabljanje terminalov. Kot primer lahko povemo, da imamo v celotni aplikaciji na voljo več različnih modulov, ki jih s parametri lahko omogočimo ali onemogočimo. Tako

lahko ob naročilu dodatne funkcionalnosti stranke enostavno spremenimo en parameter in preko centra posodobimo vse terminale, ki se navezujejo na to stranko, in že imamo na voljo novo funkcionalnost.

Pri modularni zasnovi ima vsak modul ločeno dodeljene logotipe in parametre, ki so potrebni za delovanje tega modula. Temu je dodeljen tudi fizično ločen prostor v pomnilniku, ki ga razdelimo v podsklope glede na uporabljeni modul. Seveda je postavitve modulov v pomnilniški prostor stvar izbire, paziti moramo le, da se le-ti ne prekrivajo in da ne zmanjka prostora. Tukaj nam je v veliko pomoč orodje, se nam avtomatizira proces preverjanja in urejanja po naslovih, tako da uporabniku ni potrebno neprestano računati in iskati praznega prostora. Slika 7 prikazuje, kako je določen modul zastopan v pomnilniku terminala T4500.

Selection	Base	Size	Type
Memory Space			
EEPROM in Margento	0x00000000	0x00004000	EEPROM
M95256	0x00004000	0x00400000	Flash
Maintenance	0x00004000	0x00190000	
Main subsection	0x00204000	0x00050000	
EPIN SA subsection	0x00254000	0x00010000	
Deposit SA subsection	0x00264000	0x00010000	
Prepaid SA subsection	0x00274000	0x00010000	
SC SA subsection	0x00284000	0x00010000	
Maintain POS7 subsection	0x00294000	0x00010000	
Reports SA subsection	0x002a4000	0x00008000	
Bill Payment SA Logotypes	0x002b4000	0x00008000	
BLP SA	0x002bc000	0x00008000	
UFDS subsection	0x002c4000	0x00010000	
UFDS Maintain Backup subsection	0x002d4000	0x00010000	
SIM Activation Logotypes	0x002e4000	0x00010000	
Other Payment SA Logotypes	0x002f4000	0x00010000	
Firmware Application	0x00300000	0x0003fff7f	Flash
25P80 on programing card	0x00500000	0x00080000	Flash

Slika 7. Modularna zasnova notranjega Flash pomnilnika

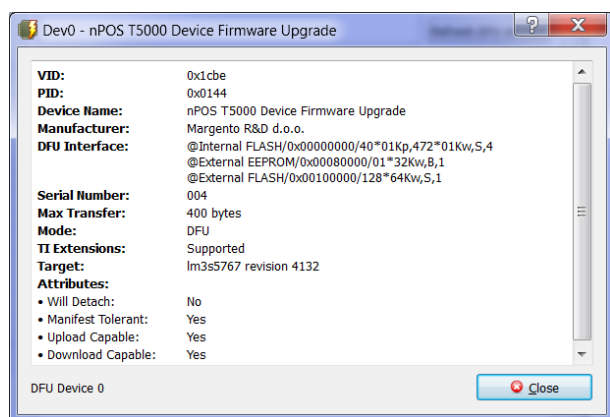
Edino kar nam preostane pri sestavljanju celotne datoteke okolja Terminal Tool, je vnos strojno-programске opreme (ang. Firmware), ki vsebuje operacijski sistem in delovanje zajetih modulov. Ker gre pri tem za vgrajene sisteme, uporabljamo namenska razvojna okolja, in sicer Code Composer Studio za DSP mikrokrmilnike in IAR Embedded Workbench za ARM mikrokrmilnike. Torej je potrebno storiti še vmesni korak, s katerim prevedemo in pretvorimo zgrajeno strojno-programsko kodo v obliko, ki je primerna za pošiljanje preko serijskih povezav do priključenega terminala.

V nadaljevanju bomo povzeli, kako poteka komunikacija med terminalom T5000 in programskim orodjem. USB protokol dela na način, da se ob priključitvi naprave le-ta predstavi operacijskemu sistemu Windows, saj kot verjetno že veste iz lastnih izkušenj, nas vedno ko priključimo novo napravo, pričaka oblaček s podatki. Da operacijski sistem sploh lahko komunicira s terminalom, je potreben določen gonilnik, v katerem je zapisano na kakšen način bosta ti dve napravi medsebojno izmenjevali podatke in kakšna vsa stanja so mogoča v tej napravi.

Terminal T5000, vendPOS in Dongle imajo na USB serijskemu protokolu možna dva stanja, in sicer stanje, v katerem se izvaja aplikacija (zagonski nalagalnik zažene

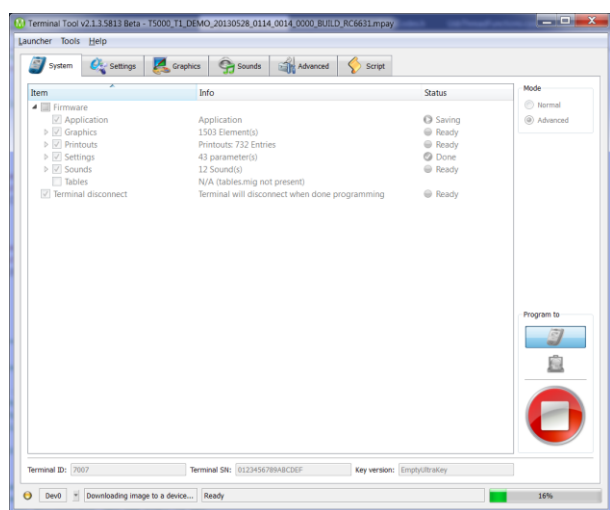
aplikacijo na določenem naslovu notranjega Flash pomnilnika) in stanje posodobitve strojno-programске opreme naprave (ang. DFU – Device Firmware Upgrade). Programiranje terminala je mogoče samo v slednjem načinu, tako da če se terminal nahaja v drugem načinu, je potrebno preko gonilnika napravo prisilno preklopiti v DFU režim delovanja.

Ob priključitvi naprave na miniUSB port, se nam le ta predstavi v obliki svojega imena, iz katerega lahko ugotovimo, kakšne vrste je naprava. Slika 8 prikazuje podroben odziv naprave, ki ga dobimo na zahtevo uporabnika v programski opremi Terminal Tool.



Slika 8. Podrobnejši odziv naprave na zahtevo

Prenos poteka na zahtevo programske opreme, kjer vse poteka preko standarda USB protokola. Torej, programska oprema mora v obliki paketov pošiljati zahteve za operacije, ki so definirane kot API funkcije zagnanskega nalagalnika. Preko USB protokola se pošiljajo zahteve, npr. za prenos proti napravi, status pa se preverja s GET\_STATUS bloki. Slika 9 prikazuje primer prenosa podatkov v smeri terminala.



Slika 9. Programiranje ARM terminala

## 4 Sklep

Pri razvoju novih naprav stremimo k izdelavi izboljšanih, hitrejših, kompaktnjših, cenejših naprav in kar je najpomembnejše k napravam, ki jih z majhnimi spremembami dopolnimo ali omejimo pri njihovem delovanju. Za lažje upravljanje terminalov smo razvili programsko orodje Margento Terminal Tool, s katerim je mogoče programirati vse obstoječe terminale. V povezavi z MPSA sistemom, lahko z orodjem generiramo tudi datoteko, ki poskrbi za posodabljanje terminalov na daljavo. V tem primeru ima uporabnik možnost izbire točno določenih modulov znotraj vseh možnih v okviru datoteke orodja. Z izgradnjo izbranih modulov znotraj datoteke bo nastala aplikacija optimalna s stališča uporabljenih modulov in tudi njena posodobitev na daljavo bo zahtevala manj prenesenih podatkov do terminala.

Modularnost in možnost izgradnje je privedla predvsem do hitrejših sprememb pri posodabljanju terminalov, hkrati pa hitrejša povezava USB terminalov prihrani še na času peke terminalov.

## Literatura

- [1] Texas Instruments DSP TMS320F28235, <http://focus.ti.com/lit/ds/symlink/tms320f28235.pdf>
- [2] ARM, uvod v ARM Cortex-M3 procesor, <http://www.arm.com/files/pdf/IntroToCortex-M3.pdf>
- [3] M. Kseneman, F. Horvat, A. Chowdhury, Avtomatizirano posodabljanje Margento terminalov na daljavo, Zbornik sedme konference AIG'11, Maribor, Slovenija, 31. Marec – 1. April 2011.
- [4] C. Golob, Z. Mezgec, A. Chowdhury: TMS320F28335 razvojna plošča, Zbornik sedemnajste mednarodne Elektrotehniške in računalniške konference ERK 2008, 29. september - 1. oktober 2008, Portorož, Slovenija, zv. A, str. 39-42.
- [5] Stellaris LM3S9B96 Microcontroller Data Sheet, [www.ti.com/lit/ds/symlink/lm3s9b96.pdf](http://www.ti.com/lit/ds/symlink/lm3s9b96.pdf)
- [6] ST32F415RG, <http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN1035/PF252143?sc=internet/mcu/product/252143.jsp#>
- [7] M. Kseneman, F. Horvat, A. Chowdhury, Modularna zasnova programske opreme plačilnega terminala - mPOS T4500, Zbornik devetnajste mednarodne Elektrotehniške in računalniške konference ERK 2010, Portorož, Slovenija, 20.-22. september 2010.