

# KROŽEK C-PROGRAMIRANJA ARM MIKROKRMILNIKOV NA LINUX

## Pupils club for C-programming of ARM microcontrollers on Linux

Žiga Lausegger, prof.  
Šolski center Kranj, Srednja tehniška šola  
[ziga.lausegger@gmail.com](mailto:ziga.lausegger@gmail.com)

### **Povzetek**

*Profesor na srednji tehniški šoli naj bi dijakom predal znanje, ki ga bodo lahko vsaj še nekaj let uporabljali s pridom tudi na svojih delovnih mestih v podjetjih. Kljub temu se na tehniških šolah v okviru izobraževalnega programa elektronik še vedno uporabljajo zastareli 8-bitni mikrokontrolerji, ki so npr. prepočasni za procesiranje ultrazvoka ali poganjanje operacijskega sistema. Novejši mikrokontrolerji so dovolj hitri, da poganjajo okleščene različice Linux operacijskih sistemov. Ker naj bi dober tehnik dandanes obvladal Linux operacijski sistem in ARM mikrokontrolerje, slednje pa večinoma programiramo v C-programskem jeziku, na Šolskem centru Kranj vsako leto 16 dijakom ponudimo sodelovanje v krožku Linux/C/ARM.*

**Ključne besede:** Linux, ARM, LPC3141, LPC4088, C.

### **Abstract**

*On a technical secondary school, professors are supposed to teach useful knowledge, which will be later used by his pupils at their workplaces across companies. However, we still teach electronics using outdated 8-bit microcontrollers, which are too slow to even process ultrasound or to run an operating system. That's not the case with newer ARM microcontrollers, fast enough to run lighter Linux operating systems. Nowadays, good technicians need to know about Linux operating systems and ARM microcontrollers, mostly programmed in C-programming language. For this reason Šolski center Kranj offers Linux/C/ARM club to 16 pupils every year.*

**Keywords:** Linux, ARM, LPC3141, LPC4088, C.

## 1 Uvod

Poučevanje tehnike je dokaj zahtevno, saj od predavateljev zahteva stalno spremljanje tehnološkega napredka, selekcijo znanj, usvajanje ter prenos le-teh na študente oz. dijake. Primer, ki zahteva stalno spremljanje so mikroprocesorji, katerih razvoj iz 8-bitnih na novejše 32-bitne se je zgodil dokaj hitro. Tako hitremu preskoku šolstvo ni uspelo slediti in kot rezultat se na tehničnih šolah večinoma še vedno poučuje 8-bitne mikroprocesorje in 8-bitne mikrokontrolerje zgrajene okoli njih (Lausegger, 2013).

Prehod na 32-bitne mikroprocesorje ni enostaven. V kolikor se zanj odločimo, moramo poskrbeti, da se čim dlje časa ne ponovi, kar dosežemo z izbiro popularnih mikroprocesorskih družin, npr. ARM (ARM, 2015). Za lažji prehod na 32-bitne ARM mikroprocesorje se priporoča Linux operacijski sistem, ki zaradi svoje prilagodljivosti omogoča brezplačno programiranje 32-bitnih ARM mikroprocesorjev. Poleg tega ravno Linux dominira v svetu vgrajenih sistemov, kjer Microsoft in Apple zaradi neprilagodljivosti svojih operacijskih sistemov nimata kaj iskati. Linux se torej splača naučiti že zato, ker ga lahko prilagodimo in naložimo na vgrajene sisteme.

Zaradi odprtokodnosti Linux jedra obstaja ogromno število Linux operacijskih sistemov, izmed katerih je zelo malo profesionalnih. Eden boljših je Arch Linux (Arch Linux, 2015), ki od uporabnika zahteva, da si operacijski sistem v celoti postavi sam skozi terminal. To pomeni, da uporabnik operacijski sistem zelo dobro pozna, ga posledično zna prilagoditi za uporabo na namiznem računalniku ali za uporabno na vgrajenem sistemu, kjer vedno obstaja omejitev prostora, procesorske moči, RAM, števila priključkov... Takim omejitvam so se enostavno izognili Arch Linux uporabniki, ki so ta operacijski sistem prevedli iz arhitektur x86-32 oz. x86-64, namenjenim procesorjem v namiznih računalnikih, v arhitekturo ARM ter ga ponovno izdali pod imenom Arch Linux ARM. Linux je torej prava izbira, v kolikor se želimo učiti moderne mikroprocesorje oz. mikrokontrolerje.

Previdno je potrebno izbrati tudi programski jezik, v katerem želimo programirati ARM mikrokontrolerje. V zadnjem času programiranje ARM mikrokontrolerjev večinoma učijo z uporabo višjenivojskih jezikov Basic, Python in C++. Toda, ali se z uporabo naštetih jezikov resnično učimo programiranja ARM mikrokontrolerjev? Odgovor je ne, toda zakaj? Omenjeni jeziki so odvisni od knjižnic oz. gonilnikov, ki jih je predhodno napisal nekdo drug v ARM zbirniku in C-programskem jeziku. Jeziki Basic, Python in C++ nimajo konstruktorjev, ki bi omogočali programiranje registrov, postavljanje sklada v spominskih enotah... Omogočajo zgolj branje prednarejenih funkcij, zaradi česar se z njihovo uporabo učimo le uporabe programskega jezika, ničesar pa o programiranju mikrokontrolerjev oz. mikroprocesorjev. Drugače je z uporabo C-programskega jezika, ki zaradi svoje vsestranskosti, moči, uporabnosti in hitrosti izvršljive kode po postopku prevajanja velja za stalnico v svetu programiranja mikroprocesorskih naprav. Tako je že od samega izida C-programskega jezika v katerem je spisano celo Linux jedro (Linux, 2015).

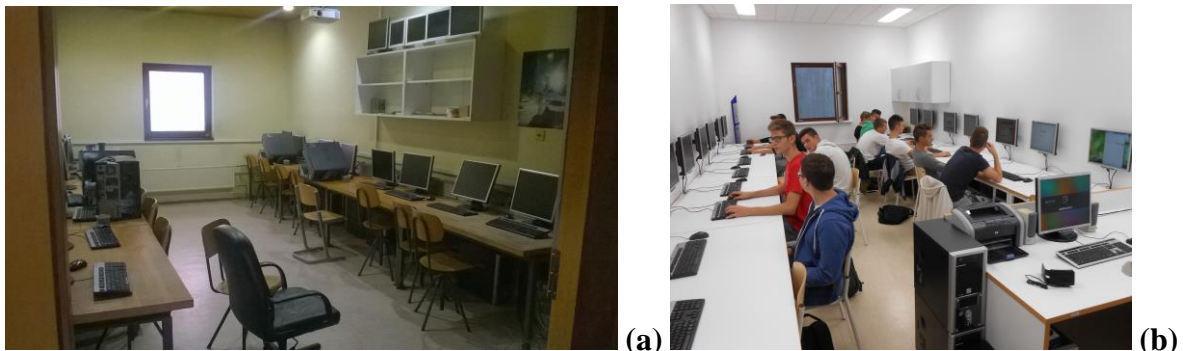
Linux, C-programski jezik in ARM mikrokontrolerji tvorijo zaokroženo celoto, ki jo na Šolskem centru Kranj ponujamo dijakom in zunanjim interesentom v obliki krožka Linux/C/ARM.

## 2 Osnovna filozofija dela

Krožek Linux/C/ARM sledi angleškemu univerzitetnemu modelu, ki se osredotoča na delitev družbe na majhne zaokrožene celote. Primer so angleške knjižnice, ki niso razdeljene na velike oddelke, kot npr. fizika, matematika, zemljepis ..., temveč na manjše oddelke. Dober primer je fizika, ki se najprej deli na termodinamiko, klasično fiziko, moderno fiziko..., le-ta pa naprej na pododdelke kvantna fizika polja, kvantna mehanika, splošna teorija relativnosti, posebna teorija relativnosti ... Vsakemu pododdelku v knjižnici namenijo majhen prostor, kjer se pogosto zadržuje več strokovnjakov s podobnimi interesi. To pripelje do različnih pogovorov, idej in razvoja.

## 3 Učilnica

Krožek že od začetka poteka v učilnici 347, slika 1 (a), ki je majhna, a z ugodno postavitvijo delovnih mest – predvsem zaslonov, ki so postavljeni tako, da omogočajo nadzor nad delom vseh udeležencev hkrati. V primeru zapletov predavatelj lahko hitro pomaga, saj so vsa delovna mesta dosegljiva že z enim samim sprehodom po sredini učilnice. To je bistvenega pomena za tekoče izvajanje krožka, saj je zapletov veliko. Učilnica je bila pred kratkim prenovljena, slika 1 (b). Montaža zaslonov na stene se je izkazala za odlično potezo, s katero dijaki na mizah pridobijo dovolj prostora za vgrajene sisteme z ARM mikrokrmilniki.

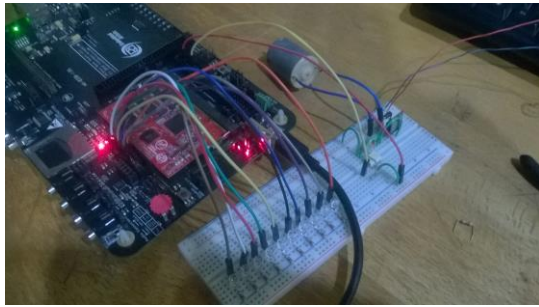


Slika 1: (a) Učilnica 347 pred in (b) po prenovi

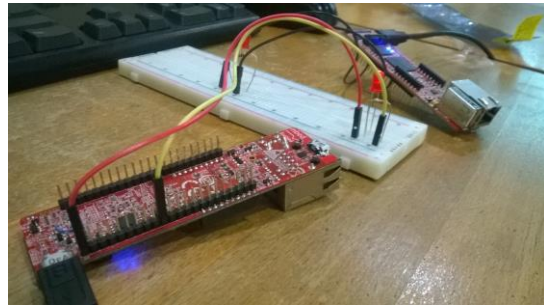
## 4 Finance in z njimi povezan razvoj

Krožek uspešno poteka že tretje leto. Prvo šolsko leto (2013 - 2014) smo izvedli obsežno pilotno različico krožka, s katero smo preverjali, kakšna bo udeležba, primernost vsebin in koliko ur na teden so udeleženci pripravljene nameniti sebi oz. temi, ki jih zanima. Ker smo se izvajalci in organizatorji prvo leto še "lovili", je bil krožek brezplačen. Drugo leto smo krožek za vse dijake financirali iz šolskega sklada, medtem ko je danes krožek plačljiv (Šolski center Kranj, 2015) - stroške dijaki poravnajo po položnici pred začetkom krožka.

Prvi dve leti smo programirali mikrokrmilnik LPC3141, zgrajen okoli mikroprocesorja ARM9. Za učenje mikrokrmilnika smo uporabljali vgrajeni sistem "EA3141 developer's kit" (Embedded Artists, 2015), slika 2 (a), ki brez DDV stane 179€. Ker šola takrat ni imela dovolj sredstev za nakup šestnajstih kompletov, je večina udeležencev na pobudo izvajalca kupila omenjeni vgrajeni sistem in ga naslednje leto kot rabljen kos opreme prodala drugim udeležencem.



(a)



(b)

Slika 2: (a) Starejši vgrajeni sistem "EA3141 developer's kit" in (b) novejši vgrajeni sistem "LPC4088 quick start board"

Nekaj izmed udeležencev si vgrajenega sistema ni moglo privoščiti, zato so v paru sodelovali s tistimi, ki so sistem imeli. Problem je šola rešila z nakupom šestnajstih novejših in cenejših vgrajenih sistemov. Odločili smo se za nakup "LPC4088 quick start board" (Embedded Artists 2015), slika 2 (b), ki stane 59€, je manjši in posledično lažje vgradljiv v modelarski avto, gliser, helikopter... Jedro novega vgrajenega sistema je mikrokontroler LPC4088, ki vsebuje ARM Cortex-M4 mikroprocesor.

## 5 Vodeni del krožka

Programiranje vgrajenih sistemov, mikrokontrolerov in mikroprocesorjev je zapleten postopek, ki zahteva veliko količino predpriprave. Posledično vsebinsko ne začnemo s programiranjem vgrajenih sistemov, temveč z osnovami - operacijskim sistemom Arch Linux, ki ga večina udeležencev preko virtualnega računalnika namesti na šolske računalnike, slika 3 (a), tisti z lastnimi prenosniki pa neposredno na disk, slika 3 (b).



(a)



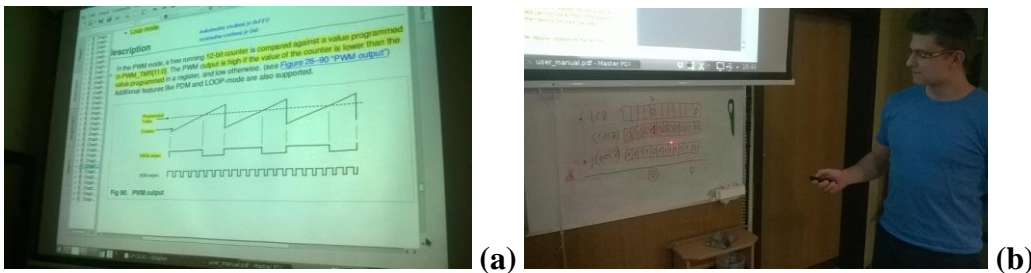
(b)

Slika 3: (a) Nameščanje Arch Linux na šolski računalnik in (b) prenosni računalnik

Ko je nameščanje operacijskega sistema končano, je potrebno še naložiti manjkajoče module, vzpostaviti spletne povezave, namestiti strežnik za namizje, namizje po lastni izbiri, upravljalnik datotek, spletni brskalnik, prevajalno verigo za mikrokontrolerke ARM, tekstovni urejevalnik in IDE s prepoznavanjem C sintakse. Sledi demonstracija Linux ukazne vrstice in s tem povezanih osnovnih Linux ukazov, ki so tudi sami binarni programi, prvotno napisani v C-programskem jeziku - najdemo jih v mapi "/bin". Nekaj izmed teh preprostih programov v istem jeziku sprogramiramo tudi z udeleženci krožka.

Ko udeleženci dobijo občutek za programiranje preprostih C-programov, se preusmerimo na pisanje programov za mikrokontroler LPC4088. Najprej skupaj z udeleženci preletimo in z markerjem označimo pomembna dejstva o zgradbi mikrokontrolerke, slika 4 (a), ki jih najdemo v LPC4088 uporabniškem priročniku - PDF različica je dosegljiva v spletu (NXP 2014). Nejasnosti dodatno razložimo na tabli in razlago vpišemo v PDF. Da bi zagotovili boljši

prenos znanja, včasih razlago prevzamejo tudi udeleženci, ki krožek obiskujejo že več let in tematiko bolje obvladajo, slika 4 (b).



**Slika 4: (a) Označevanje pomembnih stvari v LPC4088 uporabniškem priročniku in (b) razlaga s strani boljših udeležencev**

V končni fazi z udeleženci ugotovimo, da je mikrokrmilnik v bistvu računalnik, sestavljen iz perifერიj. Ker so nekatere perifერიje težje od drugih, jih obravnavamo in programiramo po vrsti od lažjih proti težjim. Pri vsaki perifერიji najprej skupaj z udeleženci pobarvamo pomembne stavke, bite in registre, ter razmislimo, kako bi za to perifერიjo napisali C-program. Sledi pisanje programa v IDE KDevelop, nalaganje na vgrajeni sistem in testiranje. Pri prvih perifერიjah so udeleženci dokaj nesamostojni, a se kmalu privadijo na način dela, ki je pri težjih perifერიjah že tako utečen, da predavatelj le občasno nudi pomoč.

## 6 Samostojni del krožka

V končni fazi se udeleženci razdelijo v skupine dveh ali treh udeležencev, saj so se večje skupine izkazale za neučinkovite. Vsaka izmed skupin si zada cilj in ga poiskuje doseči s sintezo pridobljenih znanj ali celo s študijem neobravnavanih perifერიj. Slednje zahteva veliko truda, a predstavlja tudi doprinos k razvoju krožka. Študij katerekoli nove perifერიje je lahko tudi iztočnica za zaključno nalogo pri četrti poklicni enoti.

## 7 Zaključek

Krožek Linux/C/ARM se je do sedaj obnesel izvrstno, a na žalost ni poznan izven Šolskega centra Kranj. V prihodnosti je potrebno narediti več za promocijo krožka in ga preko CPI ponuditi tudi profesorjem na srednjih šolah. Slednji bi tako kvalitetno znanje lahko prenesli na svoje dijake. Krožek lahko preko Medpodjetniškega izobraževalnega centra (MIC) (Šolski center Kranj, 2015) na Šolskem centru Kranj ali podjetja OpenLab (OpenLab, 2015) ponudimo tudi zunanjim interesentom.

Pridobljeno znanje se uporablja tudi npr. v razvojnem oddelku ISKRA, kjer dva bivša dijaka Šolskega centra Kranj opravljata počitniško delo programiranja ARM mikrokrmilnikov. Poleg tega v podjetju ISKRA množično prehajajo na operacijski sistem Linux in bodo v prihodnosti potrebovali dodatne strokovnjake - po možnosti bivše dijake Šolskega centra Kranj.

## 8 Viri

Arch Linux, *A simple, lightweight distribution* (online). 2015 (citirano 20. 09. 2015). Dostopno na naslovu: <https://www.archlinux.org/>.

ARM, *The Architecture for the Digital World* (online). 2015. (citirano 20. 09. 2015). Dostopno na naslovu: <http://www.arm.com/index.php>.

Embedded Artists, *LPC3141 Developer's kit* (online). 2015. (citirano 20. 09. 2015). Dostopno na naslovu: [http://www.embeddedartists.com/products/kits/lpc3141\\_kit.php](http://www.embeddedartists.com/products/kits/lpc3141_kit.php).

Embedded Artists, *LPC4088 Quickstart board* (online). 2015. (citirano 20. 09. 2015). Dostopno na naslovu: [http://www.embeddedartists.com/products/boards/lpc4088\\_qsb.php](http://www.embeddedartists.com/products/boards/lpc4088_qsb.php)

Lausegger, Ž. *Diplomska naloga: C programiranje ARM9 v prostodostopnem okolju za Linux*. Ljubljana: Univerza v Ljubljani. 2013. Dosegljivo na povezavi: <http://ziga-lausegger.com/strani/univerza-diploma.php>.

NXP, *LPC408x/407x User manual* (online). 2014. (citirano 20. 09. 2015). Dostopno na naslovu: [http://www.nxp.com/documents/user\\_manual/UM10562.pdf](http://www.nxp.com/documents/user_manual/UM10562.pdf).

OpenLab, *Dobrodošli v OpenLabu* (online). 2015. (citirano 20. 09. 2015). Dostopno na naslovu: <http://www.openlab.si/>.

The Linux Kernel Organization, *The Linux Kernel Archives* (online). 2015. (citirano 20. 09. 2015). Dostopno na naslovu: <https://www.kernel.org/>.

Šolski center Kranj, *Katalog PPIV* (online). 2015. (citirano 20. 09. 2015). Dostopno na naslovu: [http://www.sckr.si/tsc/documents/upload/2014\\_2015/123/katalogPPIV1516.pdf](http://www.sckr.si/tsc/documents/upload/2014_2015/123/katalogPPIV1516.pdf).

Šolski center Kranj, *Medpodjetniški izobraževalni center* (online). 2015. (citirano 20. 09. 2015). Dostopno na naslovu: <http://www.sckr.si/tsc/mic/>.