



Gábor Kreinicker &lt;kreinickergabor@gmail.com&gt;

---

**CSFK Csillagászati Intézet demonstrátori pályázat eredménye 2020**

102 üzenet

**Róbert Szabó** <szabo.robert@csfk.mta.hu>

2020. október 3. 23:49

Címzett: Gábor Kreinicker &lt;kreinickergabor@gmail.com&gt;

Másolatot kap: András Pál &lt;pal.andras@csfk.mta.hu&gt;

Kedves Gábor!

Örömmel értesítelek, hogy 2020. október 1-től 5 hónapra demonstrátori pozíciót ajánlunk Neked a Csillagászati Intézetben. A szükséges papírmunkával kapcsolatban hamarosan értesítést küldünk. A pozíció kölcsönös megelégedettség esetén 2021. augusztus 31-ig meghosszabbítható. Neked és témavezetődnek (cc-ben) eredményes munkát kívánok!

Üdvözlettel:  
Szabó Róbert  
igazgató

--  
Robert Szabo  
Director, Konkoly Observatory  
ELKH CSFK, Research Centre for Astronomy and Earth Sciences

MTA CSFK Lendület Near-Field Cosmology Research Group leader

<http://www.konkoly.hu/staff/rszabo.html><http://www.konkoly.hu/KIK/>

*I believe in a sensible work-life balance. I don't expect an answer to this email outside of your normal working hours.*

---

**Andras Pal** <apal@szofi.net>

2020. október 12. 12:03

Címzett: Gábor Kreinicker &lt;kreinickergabor@gmail.com&gt;

Szia Gábor!

Koszi megegyeszer a jelentkezest, atfutottuk kozben mi is itten a jelentkezesedet es majd itten a virusos-homeofficeos-karantenes idoszak lehetosegeihez kepest mihamarabb meg is probalhatjuk elkezdeni a munkalatokat.

Elso korben azirant erdeklodnek, hogy a motivacios leveledben is említett "programozas & mikrovezzerlok" tema az pontoabban mit takar, milyen rendszereket (architektura, toolchain, ...) ismersz, tanulsz, tanulgatsz, miben csinalod a fejleszteseket? Mi is tobb fajta vonalon mozgunk mind műhold-temaban, mind tavcsoves-automatizalos (azaz kicsit szoszerint "földhözragadtabb" témákhoz kapcsolodoan is), kerdes az igy az elejen hogy megtalaljuk a "közös metszeetet" vagy "legnagyobb kozos osztot" vagy ilyesmit ami alapjan el tudunk indulni.

Udv, A.

On Sat, 3 Oct 2020, Róbert Szabó wrote:

Kedves Gábor!

Örömmel értesítelek, hogy 2020. október 1-től 5 hónapra

demonstrátori pozíciót ajánlunk Neked a Csillagászati Intézetben.

A szükséges papírmunkával kapcsolatban hamarosan értesítést küldünk.

A pozíció kölcsönös megelégedettség esetén 2021. augusztus 31-ig meghosszabbítható.

Neked és témavezetődnek (cc-ben) eredményes munkát kívánok!

Üdvözlettel:

Szabó Róbert

igazgató

--

Robert Szabo

Director, Konkoly Observatory

ELKH CSFK, Research Centre for Astronomy and Earth Sciences

MTA CSFK Lendület Near-Field Cosmology Research Group leader

[ <http://www.konkoly.hu/staff/rszabo.html> | <http://www.konkoly.hu/staff/rszabo.html> ]

[ <http://www.konkoly.hu/KIK/> | <http://www.konkoly.hu/KIK/> ]

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. október 12. 12:40

Címzett: Andras Pal <apal@szofi.net>

Tisztelt András Pál!

Úgy kezdődött minden, hogy szereztem egy commodore 64-et, és ebben tanulgattam a basic (2.0) nyelvet, majd miután elkezdtem tanulmányaimat gimnáziumban, kezdtem el órai szinten java-t tanulni, amiben több, érettségi szempontjából hasznos gyakorlóprogramot írtam.

El kezdett érdekelni a robotika, így beszereztem egy Arduino Uno-t, egy rakat szenzort és alkatrészt. Ezzel kezdtem el c++ nyelvvel foglalkozni. Ezzel egy időben kezdtem el Fusion 360-ban megtervezni saját alkatrészeim, amiket ki tudtam 3d nyomtatni / nyomtattatni. Ezzel a tudással több projektemet is sikerült elkészítenem, mint pl. laboratópegység, vagy robotkar.

Neveztem a 29. Ifjúsági Tudományos és Innovációs Tehetségkutató Versenyre egy járás diagnosztikai eszközzel, amely mintavételezési sebességéhez nem volt elegendő egy Arduino, így szereztem be egy Raspberry-t. A működéshez szükséges összes programot pythonban írtam, melyhez az idő rövidsége miatt felületesen értek csak, de azóta is fejlesztem tudásom.

Az egyetemen C-vel kezdtünk el foglalkozni, valamint korábban kipróbálási szinten HTML-eztem és többit olvastam PLC-kről, valamint próbálkoztam játékok elkészítésével Unity-ben is, de azt is csak felületesen.

Nyáron kezdtem el Inventorban modellezni, ami egyszerre volt könnyű és okozott nehézséget a Fusionnal való hasonlóságok miatt, de már kellően rugalmasan tudom használni.

Szerintem körülbelül ennyi, amit tudok.

Üdvözlettel:

Kreinicker Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2020. október 12. 13:14

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Koszi az összefoglalót! Na, akkor nezzuk sorban:

- az Arduinio az határozottan jó, mert mi elég sokat dolgozunk AVR alapokon. Sok elektronikánk kimonodttan ugyanerre az MCU-ra (ATmega328P) épül, mint ami az Arduino Uno-ban van. Illetve most mostanában egyre többet használnak AVR architektúrájú, saját fejlesztésű soft processzorokat FPGA-kon belül (pl ilyen repül majd most a műholdakban is). Annyi a különbség hogy mi ezeket teljesen "bare metal" módon, saját könyvtárakkal használjuk.
- PLC-k helyett mi FPGA-akat használunk. Alapvető digitális logikában mennyire vagy jártas (kombinációs vs. szekvenciális logika, ilyesmik)? HDL-eket mennyire tanulmányoztad akarcsak alapszinten is? Ilyesmi hogy Verilog, esetleg VHDL?
- az Inventor az határozottan jó, mi is abban dolgozunk! Beleértve a 3d-nyomtatásos projekteket és/vagy prototípusokat is.

Egyéb kérdések: linuxot mint olyat mennyire hasznalsz? Az inventorozást leszámítva mi mindent linuxos (full free & open source) környezetben használunk, így ennek ismerete és/vagy rutinszerű használata a későbbiekben határozottan előnyös lehet!

A C programozás is határozottan jó, itten házon belül gyk minden abban megy, főleg ami a beagyazott celhardvereken fut. C++-t már nem igazán használunk, de persze az tök jó ha abban van tapasztalatod. Illetve neha-neha előkerül inline assembly is, de az tényleg ritka és csak tényleg specialisabb területeken jön elő (pl amikor hatékonyan kell összeszoroznunk egy 32 bites és egy 8 bites számot, vagy precízebb időzítésekkor, ilyesmik).

Meg egy fontos téma/kérdés: kommunikációs dolgokat mennyire ismered? UART, RS485, CAN, I2C, SPI, ... ezek közül eddig mi jött szembe és milyen szinten?

udv, Andras

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. október 12. 13:52

Címzett: Andras Pal <apal@szofi.net>

Tisztelt András Pál!

Korábban egy már meglevő ötlet alapján építettem egy bináris számológépet TTL logikai kapuk használatával. Ezáltal a kombinációs logikával szerintem képpen vagyok, a szekvenciálisról is már olvastam többet, de arról mélyebb tudásom nincs. HDL-ekhez nem igazán értek, a nyáron szerettem volna FPGA-król tanulni, de ez sajnos nem így alakult, mivel már a bevezető rész olyan tudásra épített, ami akkor még nem állt rendelkezésemre.

Linuxot nem igazán használtam eddig, amit használtam, az a Raspberry Debian alapú operációs rendszere, ami, ha jól tudom egy módosított linux rendszer.

Annak nagyon örülök, hogy ebből a demonstrátori lehetőségből hasznos nyelvet tanulok (C), de mivel most tanulom így még nem hiszem, hogy hatékonyan tudnám használni.

A felsoroltak közül az I2C-t alkalmaztam már többször is, a legutóbb a járási diagnosztikai eszköz 7db szenzorának olvasásakor jelentett megoldást a számomra. Viszont az SPI működési elve alapján gyorsabbnak bizonyult, mely az eszköz számára nagyon fontos lenne, így arra fogunk áttérni. Az I2C-vel ellentétben az SPI-t még nem alkalmaztam gyakorlatban, de a működését már tanulmányoztam többször is. A többről csak hallottam.

Üdvözléssel:

Kreinicker Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2020. október 19. 16:56

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

No, bocsánat a kesei reakcióért, közbejött egy mátrai kiruccanas.

Nulladik kérdés: most csütörtökön (okt 22) össze tudnak-e futni fent az intézetben, a Normafán? En leszek akkor

mar fent (így a határfeltételek mellett is) és akkor tudunk egy kicsit diskurálni közvetlenül is. Vagyis, pont a matrai kiruccanasnak meg mindig van egy kis bizonytalansága, de nagyon bízom benne hogy már szívesen visszatérek Budapestre.

Első körben ami lehet majd mint egy onnálló feladat az lekepezo infraszenzorok (MLX90640 típusúak) kiolvasása + adatfeldolgozása STM32F0 sorozatu mikrokontrollerek segítségével. Egy ilyen projektbe vágnánk bele mi is a következő hónapokban, aminek a célja az hogy egy kombinált Nap- és horizont-szenzor rendszerként tudjuk használni kicsi muholdaknál. Ez az infraszenzor is I2C-n keresztül működtethető, egy buszra többet is rá tudunk aggatni, és a mikrokontroller vezérelné a körkörös kiolvasást + adatfeldolgozást.

A másik, ami nem konkrét feladat hanem inkább fejlesztés-önképzés az az FPGA-kon a szoft processzoros környezet tesztelése-fejlesztése, mostmár immáron a perifériákra fókuszálva és a későbbiekben egy 4-stage instruction pipeline-ra épülő AVR processzor készítése. Ezek már értelemszerűen kicsit haladabb témák így első olvasatra de egyszerűbb perifériák (pl PWM, timerek, systick-ek, input caputre, ...) tervezése az kellemesen jól elsajátítható mind a C tanulással, mind a HDL/Verilog tanulással parhuzamosan is - úgy hogy közben már akár produktív alkalmazása is lenne előbbutóbb. Itt viszont az egy határozott előny hogy van Arduino-s tapasztalatod - mert maga a soft CPU is teljesen hasonló felépítésű. A filozófiai különbség úgyé az hogy míg az Arduino/ATmega vonalon adottak a perifériák ("ez van, ezt kell szeretni" módon jönnek ezek az ic-k), a soft megközelítésnek az az előnye hogy csak azokat a perifériákat tesszük bele ami kell, azokat viszont olyan darabszámban ami epp' kell nekünk.

A linuxos dologgak kapcsolatban meg egy kérdés: az mennyire megoldható hogy az általad használt gép(ek)en részben az fusson? Második oprendszer, dual boot, ilyesmi? Onmagában a virtualis gép sem lenne rossz, de az elektronika miatt jobb ha a linux maga is hardverközelibb.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. október 19. 17:41

Tisztelt Pál András!

Először is bocsásson meg, de egyre gyanúsabb, hogy nem tudom melyik a vezeték és melyik a keresztnéve, amely minden megszólításnál apró problémát okoz.

A nulladik kérdésre, sajnos a csütörtök nem jó, jövőhéten hétfő délelőtt, kedd délután, vagy csütörtök lenne a legszuperebb. A személyes tárgyalásnak / beszélgetésnek én is nagyon örülnék.

Amiket leírt, azok nagyon felcsigáztak, alig várom, hogy belekezdjünk. Azonban, ha nem probléma, szeretném megérdeklődni, hogy a fényútváltó rendszer tervezésével mi lesz a továbbiakban? Úgy vettem észre, hogy jelentősen kezdünk eltérni a SPICA témától. Ne értsen félre, természetesen nagyon érdekel és nagyon szívesen csinálom ezt is, különösen az FPGA és műhold témák nagyon érdekelnek, melyeket a korábbi email-jeiben említett meg, de érdekel, mi lesz annak a projektnek a jövője.

A linux rendszer természetesen megoldható, korábban már szerettem volna, hogy legyen a gépemen, habár a szerzés módja még nem világos.

Köszönettel:  
Kreinicker Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. október 19. 20:55

Szia Gábor!

Igen, ez neha már előfordult, semmi gond :)

Oke, akkor a jövő hetet még diskuráljuk le! Alapjaron vagyok, leszek, szóval meg tudjuk oldani. Inkább az "otthoni munkavégzésre való törekvés" optimalizálása lesz a kérdés.

Ah, értem. Marmint igen, megkerdeztem közben hogy hogysmint voltak a téma-kijelölések és mondta Szabo Robi (igazgató) hogy az esetekben az elektronikai hatter miatt a mi témánkat tarsítanánk hozzád. Szóval remelem nem gond ;) A konkrét temafelvetést idemasolom mindjárt (és akkor úgy már lesz is értelme a kérdéseimnek és/vagy

az elobbi tema-felvetesnek, jobban latod hogy mit miért irtam, stbstb).

Illetve kozben az van hogy ettol teljesen fuggetlenül a SPICA-t mint olyat joesellyel toroltek es/vagy halasztottak (nem tudom mikori hir, en is most nemreg tudtam meg). Igy az eredeti temafelvetes ottan vsz szinten modositásra kenyszerul. Mi meg jelenleg kismuholdas dolgokat csinalgatunk - amiknek persze ugyanugy vannak elonyei is mint hatranyai is.

Udv, Andras

Ps: az eredeti kiiras amit/ahogy mi bekuldtuk:

Cím: Kisműholdak fedélzeti programozása

Rövid ismertetés: A Csillagászati Intézet az elmúlt években vezető szerepet játszott két kisműholdas kísérlet detektor-rendszerének fejlesztésében. A fejlesztési feladatok egy része a detektor elektronikákhoz, valamint az elektronikákban levő mikrokontrollerek programozásához kapcsolódik. Az érdeklődők a további projekteken keresztül tudnak bekapcsolódni az érintett területek egy részébe: digitális jelfeldolgozás, műhold egyes egységei közötti stabil kommunikáció, adattárolás és telemetria-előkészítés, tudományos alkalmazásoknak megfelelő adat-management. Valamint kitérünk ezen feladatok háttérét biztosító technológiákra is: beágyazott programozás, digitális elektronikai alapelemek absztrakciója és megvalósítása, szoft processzorok tervezése és tesztelése, szoft perifériák tervezése és tesztelése, távoli - akár repülés közbeni - szoftver-frissítések lehetőségei.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. október 19. 23:13

Címzett: Andras Pal <apal@szofi.net>

Tisztelt Pál András!

Ez teljesen érthető, köszönöm szépen!

Üdvözléssel:

Kreinicker Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2020. október 20. 22:00

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Oke. Kerdes hogy addig, a köv ~1 hetes idotavra kuldjek-e valami kicsit reszletesebb barmit es/vagy olvasgatnivalot a tema(k)ban?

Andras

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. október 21. 1:16

Címzett: Andras Pal <apal@szofi.net>

Tisztelt Pál András!

Igen, az szuper lenne!

Üdv. Kreinicker Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2020. október 23. 19:04

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Oke, igen, ez kimaradt. Szoval aminek erdemes lenne utananezned első kanyarban az lehetne:

- az STM32Fx-es mikrokontrollerek lelki világa, miket tudnak, milyen perifériai vannak, milyen toolchain-ek erhetőek el (lásd pl: [https://en.wikipedia.org/wiki/STM32#STM32\\_F0](https://en.wikipedia.org/wiki/STM32#STM32_F0), arm-none-eabi-gcc);

- kis olvasgatas a kulonbozo periferialrol ([https://en.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface), [https://en.wikipedia.org/wiki/CAN\\_bus](https://en.wikipedia.org/wiki/CAN_bus), <https://en.wikipedia.org/wiki/RS-485>);

- illetve kis Verilog-bevezető, melyhez a mellékelt könyvecske szerintem nagyon jó alapot ad keves hatterismeret nélkül is.

Ez így nyilván elsore elég sok. Szóval nem az a cél hogy mindet azonnal megtanulj és ezekből holnap zéhá (mint a viccben), hanem kicsit megalapozni azokat az - első körben hardverközelibb - dolgokat amikkel mi dolgozunk itten. Illetve ezekhez a fenti dolgokhoz vannak nekünk mindenféle játszos boardjaink (boltiak: STM32 Nucleo-64, iCE40HX8K-B-EVN, illetve saját fejlesztésű breadboard-szerű, ezelőbbi breakout boardhoz hasonló, de már a célalkalmazásokhoz is közelebb álló eszközeink), amiket haza tudsz vinni gyakorolni és kipróbálni a fentieket majdnem-teljes-értéken. A "majdnem" az itt a kommunikáció hiányát takarja leginkább, bár pont nemrég zártunk le egy full kombi board design-t, aminek a nyákja is már elkészült és kb az összes fenti dolgot tartalmazza úgy egyszerre hogy "csak bedugod az usb-be és megy". De annak az összeállításáig még nem jutottunk el :/

Ha hardverközelibb a játék akkor első körben szinte mindegy is hogy windozon vagy linuxon kezdesz, de később érdemesebb lesz a linux felé konvergálni. Annak a "beszerzése" egyszerű: <https://www.debian.org/distrib/>. Ha a laptopod bírja (és miért ne bírna) akkor első körben kipróbálhatod az installálás folyamatát virtualis gépen is - de az hosszabb távon azért nem feltétlen jó mert ha hardverközelibbek a dolgok akkor nem baj ha az operációs rendszer natív. De persze egy "full screen" virtualbox-os linux sem rossz már!

Andras

[Az idézett szöveg el van rejtve]



**Introduction\_to\_Logic\_Synthesis\_using\_Verilog\_HDL.pdf**  
8009K

**Andras Pal** <apal@szofi.net>

2020. október 26. 10:13

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Volt/van egy kis vis maior-unk most a műhold-elektronikákkal, így a köv napok kicsit erratikusak nekem. De persze a dolgok parhuzamosítása mindig jó dolog - így kérdés: ha nem késő és/vagy meg tudnád oldani, akkor már korábban (2 korúlre) mennyire tudnál feljönni a normafára?

Kösz, A.

[Az idézett szöveg el van rejtve]

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. október 26. 10:18

Címzett: Andras Pal <apal@szofi.net>

Tisztelt Pál András!

Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó lehet.

Üdv: Gábor

[Az idézett szöveg el van rejtve]

**Andras Pal** <apal@szofi.net>

2020. október 26. 10:24

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Oke, eszben tartom! Szerda délelőttig akkor diskuráljuk le. Az is lehet hogy valahol a city-ben összefutunk, hamar ugyanis hoómooffice van és a normafa mindentől is messze van. Neked a Moszkva-Déli-BAH-Moricz tengely mennyire esik utba?

A.

[Az idézett szöveg el van rejtve]

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. október 26. 10:32

Címzett: Andras Pal <apal@szofi.net>

Ha a Móricz az a körtér, akkor az nagyon közel van. Viszont nekem be kell mennem minden képpen a titkárságra aláírni a szerződést, így ha az is Normafán van, akkor szívesen kimegyek oda is.

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. október 26. 10:38

Igen, a "Normafa" az maga az intézet, [Konkoly-Thege M. ut 15-17](#). Egyesek "Svábhegy"-kent is szeretnek rá hivatkozni, az is geográfiailag kb ugyanarra akkuratus :)

Csak azért kerdek mert ennekem ha nem is 100% a Normafa csutortokon, odalent a városban (~Moricz korter videke) nekem is hatekonyabb az eljutas.

A,  
[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. október 26. 10:44

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm szépen!  
[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. október 26. 23:07

A doksikát/referenciákat mennyire tudtad megnevezni amiket küldtem? Kiindulásnak ezek mennyire jók? Persze kicsit még levegőben lógnak a témák, de ettől függetlenül(?)

A  
[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. október 26. 23:25

Beleolvastam, de a hetedik heti hajrá és a TDK határidő közelsége nem engedett komolyan elmélyedni még benne. A téma nagyon bejött, a jövőhétre már felkészültebb leszek  
[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. október 28. 18:44

Oke! Akkor viszont lehet hogy diskuráljunk megiscsak jövő heten, mert közben engem is nagyon bekapott az eksziz es/vagy a daráló a műhold kapcsán. Most már szerda van, és meg mindig nem tudom hogy a holnap az hogy lesz. És attól felek hogy csak delelőtt derül ki hogy már delután fel kell mennem vagy csak pénteken vagy hogy.

Neked mikor vannak ezek a határidők es/vagy mikor záródnak le a dolgok (melyik is pl az a hetedik hét, meg mi a TDK határideje, ilyesmik)?  
[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. október 29. 9:01

Rendben, az nekem is most előnyös lesz.  
TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz minden, a hetedik hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét elején.  
Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha minden igaz, az is meglesz.

Üdv: Gábor  
[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. október 30. 16:22

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás céljából?

Köszönettel: Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2020. október 30. 16:56

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Igen, en itten az Icarus Verilog (iverilog) nevű programot hasznalom (<http://iverilog.icarus.com/>). Ennek ha jól latom van windozos portja is: <http://bleyer.org/icarus/>.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás céljából?

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020. okt. 29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.  
TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz minden, a hetedik hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét elején.  
Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha minden igaz, az is meglesz.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 28., Sze, 18:44):

Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo heten, mert kozben engem is nagyon bekapott az ekszij es/vagy a daralo a műhold kapcsan. Most mar szerda van, es meg mindig nem tudom hogy a holnap az hogy lesz. Es attol felek hogy csak delelott derul ki hogy mar delutan fel kell mennem vagy csak penteken vagy hogy.

Neked mikor vannak ezek a hataridok es/vagy mikor zarodnak le a dolgok (melyik is pl az a hetedik het, meg mi a TDK hatarideje, ilyesmik)?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Beleolvastam, de a hetedik heti hajrá és a TDK határidő közelsége nem engedett komolyan elmélyedni még benne. A téma nagyon bejött, a jövőhétre már felkészültebb leszek

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 23:07):



A doksikat/referenciakat mennyire tudta megnevezni amiket küldtem?  
Kiindulásnak  
ezek mennyire jók? Persze kicsit még levegőben lógnak a témák, de ettől függetlenül(?)

A

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm szépen!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 10:38):

Igen, a "Normafa" az maga az intézet, Konkoly-Thege M. ut 15-17.  
Egyesek

"Svábhegy"-kent is szeretnek rá hivatkozni, az is geográfiai kb  
ugyannyira  
akkuratus :)

Csak azért kérdelem mert ennekem ha nem is 100% a Normafa csütörtökon,  
oda lent a  
városban (~Móricz korter vidéke) nekem is hatékonyabb az eljutás.

A,

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Ha a Móricz az a körtér, akkor az nagyon közel van. Viszont nekem be  
kell  
mennem minden képpen a titkárságra aláírni a szerződést, így ha az  
is  
Normafán van, akkor szívesen kimegyek oda is.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 10:24):

Oke, esz bentartom! Szerda delelöttig akkor diskuráljuk le. Az is  
lehet  
hogymint  
valahol a city-ben összefutunk, hamár ugyanis hoómmooffice van és a  
normafa  
mindentől is messze van. Neked a Moszkva-Déli-BAH-Móricz tengely  
mennyire  
esik  
utba?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó lehet.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 10:13):

Szia Gábor!

Volt/van egy kis vis maior-unk most a műhold-elektronikákkal,  
így a  
köv  
napok

kicsit erraticusak nekem. De persze a dolgok parhuzamosítása mindig jó

dolog -  
 így kérdés: ha nem késő es/vagy meg tudnád oldani, akkor már kora-közép délután (2 korulre) mennyire tudnál feljönni a normafára?

Koszi, A.

On Mon, 19 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Először is bocsásson meg, de egyre gyanúsabb, hogy nem tudom melyik a vezeték és melyik a keresztnéve, amely minden megszólításnál apró problémát okoz.

A nulladik kérdésre, sajnos a csütörtök nem jó, jövőhéten hétfő délelőtt, kedd délután, vagy csütörtök lenne a legszuperebb. A személyes tárgyalásnak / beszélgetésnek én is nagyon örülnék.

Amiket leírt, azok nagyon felcsigáztak, alig várom, hogy belekezdjünk.

Azonban, ha nem probléma, szeretném megérdeklődni, hogy a fényútváltó rendszer tervezésével mi lesz a továbbiakban? Úgy vettem észre, hogy jelentősen kezdünk eltérni a SPICA témától. Ne értsen félre, természetesen nagyon érdekel és nagyon szívesen csinálom ezt is, különösen az FPGA és műhold témák nagyon érdekelnek, melyeket a korábbi email-jeiben említett meg, de érdekel, mi lesz annak a projektnek a jövője.

A linux rendszer természetesen megoldható, korábban már szerettem volna, hogy legyen a gépemen, habár a szerzés módja még nem világos.

Köszönettel:  
 Kreinicker Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 19., 16:56):

Szia Gábor!

No, bocsanat a kesei reakcióért, közbejött egy mátraí kiruccanas.

Nulladik kérdés: most csütörtökön (okt 22) össze tudnak-e futni fent az intézetben, a Normafán? En leszek akkor már fent (így a

hatarfeltetelek  
mellett  
is) es akkor tudunk egy kicsit diskuralni kozvetlenul is.

Vagyis,  
pont a  
matrai  
kiruccanasnak meg mindig van egy kis bizonytalansaga, de nagyon  
bizom  
benne  
hogy mar szerdan visszatek Budapestre.

Elso korben ami lehet majd mint egy onnallo feladat az lekepezo  
infraszenzorok  
(MLX90640 tipusuak) kiolvasasa + adatfeldolgozasa STM32F0  
sorozatu  
mikrokontrollerek segitsegevel. Egy ilyen projektbe vagnank  
bele  
mi  
is a  
kovetkezo honapokban, aminek a celja az hogy egy kombinalt  
Nap- es  
horizont-szenzor rendszerkent tudjuk hasznalni kicsi  
muholdaknal.  
Ez  
az  
infraszenzor is I2C-n keresztul mukodtetheto, egy buszra  
tobbet is  
ra  
tudunk  
aggatni, es a mikrokontroller vezernelne a korkoros kiolvasasat  
+  
adatfeldolgozast.

A masik, ami nem konkret feladat hanem inkabb  
fejlesztes-onkepzes  
az  
az  
FPGA-kon a szoft processzoros kornyezet tesztelese-fejleszte,  
mostmar  
immaron  
a periferiakra fokuszalva es a kesobbiekben egy 4-stage  
instruction  
pipeline-ra  
epulo AVR processzor keszitese. Ezek mar ertelemszeruen kicsit  
haladobb  
temak  
igy elso olvasatra de egyszerubb periferiak (pl PWM, timerek,  
systick-ek,  
input  
caputre, ...) tervezese az kellemesen jól elsajatithato mind a  
C  
tanulassal,  
mind a HDL/Verilog tanulassal parhuzamosan is - ugy hogy kozben  
mar  
akar  
produktiv alkalmazasa is lenne elobbutobb. Itt viszont az egy  
hataroztt  
elony  
hogy van Arduinio-s tapasztalatod - mert maga a soft CPU is  
teljesen  
hasonlo  
felepitesu. A filozofiai kulonbseg ugye az hogy mig az  
Arduinio/ATmega  
vonalon  
adottak a periferiak ("ez van, ezt kell szeretni" modon jonnek  
ezek

az  
ic-k), a  
soft megközelítésnek az az előnye hogy csak azokat a  
perifériákat  
tesszük  
bele  
ami kell, azokat viszont olyan darabszámban ami epp' kell  
nekünk.  
A linuxos dologgak kapcsolatban meg egy kérdés: az mennyire  
megoldható  
hogy az  
általad használt gép(ek)en részben az fusson? Második  
operációs rendszer,  
dual  
boot,  
ilyenmi? Onmagában a virtualis gép sem lenne rossz, de az  
elektronika  
miatt  
jobb ha a linux maga is hardverközelibb.  
udv, András  
On Mon, 12 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt András Pál!  
Korábban egy már meglevő ötlet alapján építettem egy bináris  
számológépet  
TTL logikai kapuk használatával. Ezáltal a kombinációs  
logikával  
szerintem  
képpen vagyok, a szekvenciálisról is már olvastam többet, de  
arról  
mélyebb  
tudásom nincs. HDL-ekhez nem igazán értek, a nyáron szerettem  
volna  
FPGA-król tanulni, de ez sajnos nem így alakult, mivel már a  
bevezető  
rész  
olyan tudásra épített, ami akkor még nem állt rendelkezésemre.  
Linuxot nem igazán használtam eddig, amit használtam, az a  
Raspberry  
Debian  
alapú operációs rendszere, ami, ha jól tudom egy módosított  
linux  
rendszer.  
Annak nagyon örülök, hogy ebből a demonstrátori lehetőségből  
hasznos  
nyelvet tanulok (C), de mivel most tanulom így még nem hiszem,  
hogy  
hatékonyan tudnám használni.  
A felsoroltak közül az I2C-t alkalmaztam már többször is, a  
legutóbb  
a  
járásdiagnosztikai eszköz 7db szenzorának olvasásakor  
jelentett  
megoldást a  
számomra. Viszont az SPI működési elve alapján gyorsabbnak  
bizonyult,  
mely  
az eszköz számára nagyon fontos lenne, így arra fogunk  
átterni.

Az I2C-vel ellentétben az SPI-t még nem alkalmaztam gyakorlatban, de a működését már tanulmányoztam többször is. A többiről csak hallottam.

Üdvözlettel:  
Kreinicker Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 12., H, 13:14):

Szia Gábor!

Koszi az összefoglalót! Na, akkor nezzuk sorban:  
- az Arduino az határozottan jó, mert mi elég sokat dolgozunk AVR alapokon.  
Sok elektronikánk kimonodttan ugyanerre az MCU-ra (ATmega328P) épül, mint ami az Arduino Uno-ban van. Illetve most mostanában egyre többet használunk AVR architektúrájú, saját fejlesztetű soft processzorokat FPGA-kon belül (pl. ilyen repül majd most a műholdakban is). Annyi a különbség hogy mi ezeket teljesen "bare metal" módon, saját könyvtarakkal használjuk.  
- PLC-k helyett mi FPGA-akat használunk. Alapvető digitális logikában mennyire vagy jártas (kombinációs vs. szekvencialis logika, ilyesmit)?

HDL-eket mennyire tanulmányoztad akarcsak alapszinten is? Ilyesmi hogy Verilog, esetleg VHDL?

- az Inventor az határozottan jó, mi is abban dolgozunk! Beleértve a 3d-nyomtatásos projekteket és/vagy prototípusokat is.

Egyéb kérdések: linuxot mint olyat mennyire használod? Az inventorozást leszámítva mi mindent linuxos (full free & open source) környezetben használunk, így ennek ismerete és/vagy rutinszerű használata a későbbiekben határozottan előnyös lehet!

A C programozás is határozottan jó, itten hazon belül gyk minden abban megy, főleg ami a beagyazott celhardvereken fut. C++-t már nem igazan

hasznalunk, de  
persze az tok jo ha abban van tapasztalatod. Illetve  
neha-neha  
elokerul  
inline  
assembly is, de az tenyleg ritka es csak tenyleg specialisabb  
teruleteken  
jon  
elo (pl amikor hatekonyan kell osszeszoroznunk egy 32bites es  
egy  
8bites  
számot, vagy precizebb időzítésekkor, ilyesmik).  
Meg egy fontos téma/kerdes: kommunikacios dolgokat mennyire  
ismered?  
UART,  
RS485, CAN, I2C, SPI, ... ezek kozul eddig mi jott szembe es  
milyen  
szinten?  
udv, Andras  
On Mon, 12 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt András Pál!  
Úgy kezdődött minden, hogy szereztem egy commodore 64-et, és  
ebben  
tanulgattam a basic (2.0) nyelvet, majd miután elkezdtem  
tanulmányaimat  
gimnáziumban, kezdtem el órai szinten java-t tanulni, amiben  
több,  
érettségi szempontjából hasznos gyakorlóprogramot írtam.  
El kezdett érdekelni a robotika, így beszereztem egy Arduino  
Uno-t,  
egy  
rakat szenzort és alkatrészt. Ezzel kezdtem el c++ nyelvvel  
foglalkozni.  
Ezzel egy időben kezdtem el Fusion 360-ban megtervezni saját  
alkatrészeim,  
amiket ki tudtam 3d nyomtatni / nyomtattatni. Ezzel a  
tudással  
több  
projektemet is sikerült elkészítenem, mint pl.  
labortápegység,  
vagy  
robotkar.  
Neveztem a 29. Ifjúsági Tudományos és Innovációs  
Tehetségkutató  
Versenyre  
egy járás diagnosztikai eszközzel, amely mintavételezési  
sebességéhez  
nem  
volt elegendő egy Arduino, így szereztem be egy  
Raspberry-t. A  
működéshez  
szükséges összes programot pythonban írtam, melyhez az idő  
rövidsége  
miatt  
felületesen értek csak, de azóta is fejlesztem tudásom.  
Az egyetemen C-vel kezdtünk el foglalkozni, valamint  
korábban  
kipróbálási

szinten HTML-eztem és többet olvastam PLC-kről, valamint  
 próbálkoztam játékok elkészítésével Unity-ben is, de azt is csak  
 felületesen.  
 Nyáron kezdtem el Inventorban modellezni, ami egyszerre volt  
 könnyű és  
 okozott nehézséget a Fusionnal való hasonlóságok miatt, de

**Andras Pal** <apal@szofi.net>

2020. november 4. 16:45

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

No, tul vagy kozben a mindenben? Illetve a het hetralevo resze akkor mikor lenne jo? En holnap delutan tervezek feljoni a normafára elso kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás céljából?

Köszönettel: Gábor

Gábor Kreinicker <kreinickergabor@gmail.com> ezt írta (időpont: 2020. okt. 29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.

TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz minden, a hetedik hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét elején. Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha minden igaz, az is meglesz.

Üdv: Gábor

Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. okt. 28., Sze, 18:44):

Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo heten, mert kozben engem is nagyon bekapott az ekszij es/vagy a daralo a műhold kapcsan. Most mar szerda van, es meg mindig nem tudom hogy a holnap az hogy lesz. Es attol felek hogy csak delelott derul ki hogy mar delutan fel kell mennem vagy csak penteken vagy hogy.

Neked mikor vannak ezek a hataridok es/vagy mikor zarodnak le a dolgok (melyik is pl az a hetedik het, meg mi a TDK hatarideje, ilyesmik)?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Beleolvastam, de a hetedik heti hajrá és a TDK határidő közelsége nem engedett komolyan elmélyedni még benne. A téma nagyon bejött, a jövőhétre már felkészültebb leszek

Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. okt. 26., H

23:07):

A doksikát/referenciákat mennyire tudtad megnezni amiket küldtem?  
Kiindulásnak  
ezek mennyire jók? Persze kicsit még levegőben lógnak a témák, de ettől függetlenül(?)

A

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm szépen!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H

10:38):

Igen, a "Normafa" az maga az intézet, Konkoly-Thege M. út 15-17.

Egyesek

"Svábhegy"-kent is szeretnek rá hivatkozni, az is geografiaiilag kb  
ugyannyira  
akkuratus :)

Csak azért kérdelem mert ennekem ha nem is 100% a Normafa csütörtökön,  
odalent a  
városban (~Móricz korter vidéke) nekem is hatékonyabb az eljutás.

A,

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Ha a Móricz az a körtér, akkor az nagyon közel van. Viszont nekem be

kell

mennem minden képpen a titkárságra aláírni a szerződést, így ha az

is

Normafán van, akkor szívesen kimegyek oda is.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H

10:24):

Oke, eszbe tartom! Szerda délelőttig akkor diskuráljuk le. Az is

lehet

hogyan

valahol a city-ben összefutunk, hamár ugyanis hoómooffice van és a

normafa

mindentől is messze van. Neked a Moszkva-Déli-BAH-Móricz tengely

mennyire

esik

utba?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó lehet.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H

10:13):

Szia Gábor!

Volt/van egy kis vis maior-unk most a műhold-elektronikákkal,

így a



köv  
napok  
kicsit erraticusak nekem. De persze a dolgok parhuzamositasa

mindig  
jo  
dolog -  
igy kerdes: ha nem keső es/vagy meg tudnad oldani, akkor mar

kora-kozep  
delutan  
(2 korulre) mennyire tudnal feljönni a normafára?

Koszi, A.

On Mon, 19 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Először is bocsásson meg, de egyre gyanúsabb, hogy nem tudom

melyik a  
vezeték és melyik a keresztnéve, amely minden megszólításnál

apró  
problémát  
okoz.

A nulladik kérdésre, sajnos a csütörtök nem jó, jövőhéten hétfő  
délelőtt,  
kedd délután, vagy csütörtök lenne a legszuperebb. A személyes  
tárgyalásnak  
/ beszélgetésnek én is nagyon örülnék.

Amiket leírt, azok nagyon felcsigáztak, alig várom, hogy  
belekezdjünk.

Azonban, ha nem probléma, szeretném megérdeklődni, hogy a  
fényútváltó rendszer tervezésével mi lesz a továbbiakban? Úgy

vettem  
észre,  
hogy jelentősen kezdünk eltérni a SPICA témától. Ne értsen

félre,  
természetesen nagyon érdekel és nagyon szívesen csinálom ezt is,  
különösen  
az FPGA és műhold témák nagyon érdekelnek, melyeket a korábbi  
email-jeiben említett meg, de érdekel, mi lesz annak a

projektnek a  
jövője.

A linux rendszer természetesen megoldható, korábban már

szerettem  
volna,  
hogy legyen a gépemen, habár a szerzés módja még nem világos.

Köszönettel:  
Kreinicker Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 19.,  
H,  
16:56):

Szia Gábor!

No, bocsanat a kesei reakcioert, kozbejott egy mátrai

kiruccanas.

Nulladik kerdes: most csutortokon (okt 22) ossze tudnak-e futni

fent

az  
intezetben, a Normafan? En leszek akkor mar fent (igy a  
hatarfeltetelek  
mellett  
is) es akkor tudunk egy kicsit diskuralni kozvetlenul is.

Vagyis,  
pont a  
matrai  
kiruccanasnak meg mindig van egy kis bizonytalansaga, de nagyon  
bizom  
benne  
hogy mar szerdan visszaterek Budapestre.

Elso korben ami lehet majd mint egy onnallo feladat az lekepezo  
infraszenzorok  
(MLX90640 tipusuak) kiolvasasa + adatfeldolgozasa STM32F0  
sorozatu  
mikrokontrollerek segitsegevel. Egy ilyen projektbe vagnank  
bele  
mi  
is a  
kovetkezo honapokban, aminek a celja az hogy egy kombinalt  
Nap- es  
horizont-szenzor rendszerkent tudjuk hasznalni kicsi  
muholdaknal.  
Ez  
az  
infraszenzor is I2C-n keresztul mukodtetheto, egy buszra  
tobbet is  
ra  
tudunk  
aggatni, es a mikrokontroller vezernelne a korkörös kiolvasasat  
+  
adatfeldolgozast.

A masik, ami nem konkret feladat hanem inkabb  
fejlesztés-önképzés  
az  
az  
FPGA-kon a szoft processzoros kornyezet tesztelese-fejlesztese,  
mostmar  
immaron  
a periferiakra fokuszalva es a kesobbiekben egy 4-stage  
instruction  
pipeline-ra  
epulo AVR processzor keszitese. Ezek mar ertelemszeruen kicsit  
haladobb  
temak  
igy elso olvasatra de egyszerubb periferiak (pl PWM, timerek,  
systick-ek,  
input  
caputre, ...) tervezese az kellemesen jól elsajatithato mind a  
C  
tanulassal,  
mind a HDL/Verilog tanulassal parhuzamosan is - ugy hogy kozben  
mar  
akar  
produktiv alkalmazasa is lenne elobbutobb. Itt viszont az egy  
hataroztt  
elony  
hogy van Ardunio-s tapasztalatod - mert maga a soft CPU is  
teljesen  
hasonlo  
felepitesu. A filozofiai kulonbseg ugye az hogy mig az  
Arduino/ATmega  
vonalon

adottak a periferiak ("ez van, ezt kell szeretni" módon jönnek ezek az ic-k), a soft megközelítésnek az az előnye hogy csak azokat a periferiakat tesszük bele ami kell, azokat viszont olyan darabszámban ami epp' kell nekünk.

A linuxos dologgak kapcsolatban meg egy kérdés: az mennyire megoldható hogy az általad használt gép(ek)en reszben az fusson? Második oprendszer, dual boot, ilyesmi? Onmagában a virtualis gép sem lenne rossz, de az elektronika miatt jobb ha a linux maga is hardverközelibb.

udv, Andras

On Mon, 12 Oct 2020, Gábor Kreinicker wrote:

Tisztelt András Pál!

Korábban egy már meglevő ötlet alapján építettem egy bináris számológépet TTL logikai kapuk használatával. Ezáltal a kombinációs logikával szerintem képpen vagyok, a szekvenciálisról is már olvastam többet, de arról mélyebb tudásom nincs. HDL-ekhez nem igazán értek, a nyáron szerettem volna FPGA-król tanulni, de ez sajnos nem így alakult, mivel már a bevezető rész olyan tudásra épített, ami akkor még nem állt rendelkezésemre.

Linuxot nem igazán használtam eddig, amit használtam, az a Raspberry Debian alapú operációs rendszere, ami, ha jól tudom egy módosított linux rendszer.

Annak nagyon örülök, hogy ebből a demonstrátori lehetőségből hasznos nyelvet tanulok (C), de mivel most tanulom így még nem hiszem, hogy hatékonyan tudnám használni.

A felsoroltak közül az I2C-t alkalmaztam már többször is, a legutóbb a járási diagnosztikai eszköz 7db szenzorának olvasásakor jelentett megoldást a számomra. Viszont az SPI működési elve alapján gyorsabbnak bizonyult, mely

az eszköz számára nagyon fontos lenne, így arra fogunk áttérni.

Az I2C-vel ellentétben az SPI-t még nem alkalmaztam gyakorlatban, de a működését már tanulmányoztam többször is. A többiről csak hallottam.

Üdvözlettel:  
Kreinicker Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 12., H, 13:14):

Szia Gábor!

Koszi az összefoglalót! Na, akkor nezzuk sorban:  
- az Arduino az határozottan jó, mert mi elég sokat dolgozunk AVR alapokon.  
Sok elektronikánk kimonodttan ugyanerre az MCU-ra (ATmega328P) epül, mint ami az Arduino Uno-ban van. Illetve most mostanában egyre többet hasznalunk AVR architektúrájú, saját fejlesztű soft processzorokat FPGA-kon belül (pl ilyen repül majd most a műholdakban is). Annyi a különbség hogy mi ezeket teljesen "bare metal" módon, saját könyvtarakkal használjuk.  
- PLC-k helyett mi FPGA-akat használunk. Alapvető digitális logikában mennyire vagy jártas (kombinációs vs. szekvencialis logika, ilyesmi)?  
HDL-eket mennyire tanulmányoztad akarcsak alapszinten is? Ilyesmi hogy Verilog, esetleg VHDL?  
- az Inventor az határozottan jó, mi is abban dolgozunk!  
Beleértve a 3d-nyomtatásos projekteket és/vagy prototipizálásokat is.  
Egyéb kérdések: linuxot mint olyat mennyire használod? Az inventorozást leszámítva mi mindent linuxos (full free & open source) környezetben használunk, így ennek ismerete és/vagy rutinszerű használata a későbbiekben határozottan előnyös lehet!

A C programozás is határozottan jó, itten hazon belül gyk minden abban megy,

igazan  
 neha-neha  
 egy  
 ismered?  
 milyen  
 ebben  
 több,  
 Uno-t,  
 tudással  
 több  
 labortápegység,  
 vagy  
 Tehetségkutató  
 sebességéhez  
 nem  
 Raspberry-t. A  
 rövidsége  
 miatt

főleg ami a beagyazott celhardvereken fut. C++-t már nem  
 használunk, de  
 persze az tök jó ha abban van tapasztalatod. Illetve  
 elokerul  
 inline  
 assembly is, de az tenyleg ritka es csak tenyleg specialisabb  
 teruleteken  
 jon  
 elo (pl amikor hatekonyan kell osszeszoroznunk egy 32bit-es  
 8bit-es  
 számot, vagy precizebb időzítésekkel, ilyesmi).  
 Meg egy fontos téma/kérdés: kommunikációs dolgokat mennyire  
 UART,  
 RS485, CAN, I2C, SPI, ... ezek közül eddig mi jött szembe és  
 szinten?  
 udv, András  
 On Mon, 12 Oct 2020, Gábor Kreinicker wrote:  
 Tisztelt András Pál!  
 Úgy kezdődött minden, hogy szereztem egy Commodore 64-et, és  
 tanulgattam a BASIC (2.0) nyelvet, majd miután elkezdtem  
 tanulmányaimat  
 gimnáziumban, kezdtem el órai szinten Java-t tanulni, amiben  
 érettségi szempontjából hasznos gyakorlóprogramot írtam.  
 El kezdett érdekelni a robotika, így beszereztem egy Arduino  
 egy  
 rakat szenzort és alkatrészt. Ezzel kezdtem el C++ nyelvvel  
 foglalkozni.  
 Ezzel egy időben kezdtem el Fusion 360-ban megtervezni saját  
 alkatrészeim,  
 amiket ki tudtam 3D nyomtatni / nyomtattatni. Ezzel a  
 projektemet is sikerült elkészítenem, mint pl.  
 robotkar.  
 Neveztem a 29. Ifjúsági Tudományos és Innovációs  
 Versenyre  
 egy járás diagnosztikai eszközzel, amely mintavételezési  
 volt elegendő egy Arduino, így szereztem be egy  
 működéshez  
 szükséges összes programot Pythonban írtam, melyhez az idő  
 felületesen értek csak, de azóta is fejlesztem tudásom.  
 Az egyetemen C-vel kezdtünk el foglalkozni, valamint

korábban kipróbálási szinten HTML-eztem és többet olvastam PLC-kről, valamint próbálkoztam játékok elkészítésével Unity-ben is, de azt is csak felületesen. Nyáron kezdtem el Inventorban modellezni, ami egyszerre volt könnyű és okozott nehézséget a Fusionnal való hasonlóságok miatt, de már kellően

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. november 4. 17:05

Címzett: Andras Pal <apal@szofi.net>

Tisztelt Pál András!

A holnap délután részemről szuper!

Mikor tájtra számítsak?

Üdv: Gábor

[Az idézett szöveg el van rejtve]

[Az idézett szöveg el van rejtve]

>>>>>>>>>> és

>>>>>>>>>> okozott nehézséget a Fusionnal való hasonlóságok miatt, de

>>> már

>>>>>>>>> kellõen

>>>>>>>>>>> rugalmasan tudom használni.

>>>>>>>>>>>>>>>

[illegible]

>>>>>>>>>>>>>>>

>>>>>>>>>>>> Üdvözlettel:

>>>>>>>>>> Kreinicker Gábor

>>>>>>>>>>>>>>>>>

>>>>>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

>>> 12.,

>>>>>> H,

[Az idézett szöveg el van rejtve]

**Andras Pal** <apal@szofi.net>

2020. november 4. 17:09

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, es akkor utana indulok fel. Szoval fel 2 korultol mar leszek itt. Ha errefele jarsz akkor hivj fel: (20)3962151, es akkor kimegyek a kapuhoz. Merthogy mi itten egy masik epulet masik szekciojaban vagyunk.

udv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper!

Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 16:45):

Szia Gábor!

No, tul vagy kozben a mindenen? Illetve a het hetralevo resze akkor mikor

lenne  
jo? En holnap delutan tervezek feljonni a normafára elso kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás céljából?

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020.  
okt.  
29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.  
TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz minden, a hetedik  
hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét elején.  
Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha minden

igaz,  
az is meglesz.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 28., Sze,  
18:44):

Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo heten, mert  
kozben  
engem is nagyon bekapott az ekszij es/vagy a daralo a műhold kapcsan.  
Most mar  
szerda van, es meg mindig nem tudom hogy a holnap az hogy lesz. Es

attol

felek  
hogy csak delelott derul ki hogy mar delutan fel kell mennem vagy csak  
penteken  
vagy hogy.

Neked mikor vannak ezek a hataridok es/vagy mikor zarodnak le a dolgok  
(melyik  
is pl az a hetedik het, meg mi a TDK hatarideje, ilyesmik)?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Beleolvastam, de a hetedik heti hajrá és a TDK határidő közelsége nem  
engedett komolyan elmélyedni még benne. A téma nagyon bejött, a  
jövőhétre  
már felkészültebb leszek

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H  
23:07):

A doksikat/referenciakat mennyire tudtad megnezni amiket kuldtem?  
Kiindulasnak  
ezek mennyire jok? Persze kicsit még levegoben lognak a temak, de

ettol

fuggetlenul(?)

A

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm szépen!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 10:38):

Igen, a "Normafa" az maga az intézet, Konkoly-Thege M. ut 15-17.

Egyesek

"Svábhegy"-kent is szeretnek rá hivatkozni, az is geografiailag kb ugyanarra akkuratus :)

csütörtökön,

Csak azért kérdezem mert énnekem ha nem is 100% a Normafa

odalent a városban (~Móricz korter vidéke) nekem is hatékonyabb az eljutás.

A,

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

be

Ha a Móricz az a körtér, akkor az nagyon közel van. Viszont nekem

kell

is

mennem minden képpen a titkárságra aláírni a szerződést, így ha az

Normafán van, akkor szívesen kimegyek oda is.

10:24):

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H

lehet

Oke, eszbe tartom! Szerda délelőttig akkor diskuráljuk le. Az is

hogyan

normafa

valahol a city-ben összefutunk, hamar ugyanis hóomoffice van és a

mindentől is messze van. Neked a Moszkva-Déli-BAH-Móricz tengely

menyire esik utba?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó lehet.

Üdv: Gábor

H

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26.,

10:13):

Szia Gábor!

így a

Volt/van egy kis vis maior-unk most a műhold-elektronikákkal,

köv

napok

mindig

jó

kicsit erraticusak nekem. De persze a dolgok parhuzamosítása

dolog -

így kérdés: ha nem késő és/vagy meg tudnád oldani, akkor már kora-közép



delután  
(2 körülre) mennyire tudnál feljönni a normafára?

Koszi, A.

On Mon, 19 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Először is bocsásson meg, de egyre gyanúsabb, hogy nem tudom

vezeték és melyik a keresztnéve, amely minden megszólításnál

problémát  
okoz.

A nulladik kérdésre, sajnos a csütörtök nem jó, jövőhéten

délelőtt,  
kedd délután, vagy csütörtök lenne a legszuperebb. A személyes  
tárgyalásnak  
/ beszélgetésnek én is nagyon örülnék.

Amiket leírt, azok nagyon felcsigáztak, alig várom, hogy  
belekezdjünk.

Azonban, ha nem probléma, szeretném megérdeklődni, hogy a  
fényútváltó rendszer tervezésével mi lesz a továbbiakban? Úgy

észre,  
hogy jelentősen kezdünk eltérni a SPICA témától. Ne értsen

természetesen nagyon érdekel és nagyon szívesen csinálom ezt

különösen  
az FPGA és műhold témák nagyon érdekelnek, melyeket a korábbi  
email-jeiben említett meg, de érdekel, mi lesz annak a

jövője.

A linux rendszer természetesen megoldható, korábban már

volna,  
hogy legyen a gépemen, habár a szerzés módja még nem világos.

Köszönettel:  
Kreinicker Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

16:56):

Szia Gábor!

No, bocsanat a kesei reakcioert, kozbejott egy mátraí

Nulladik kerdes: most csutortokon (okt 22) ossze tudnak-e

az  
intezetben, a Normafan? En leszek akkor mar fent (igy a  
hatarfeltetelek  
mellett

is) es akkor tudunk egy kicsit diskurálni közvetlenül is.

Vagyis, pont a matrai kiruccanasnak meg mindig van egy kis bizonytalansága, de nagyon bizom benne hogy már szívesen visszatérek Budapestre.

Elso körben ami lehet majd mint egy önálló feladat az infraszenzorok (MLX90640 típusúak) kiolvasása + adatfeldolgozása STM32F0 sorozatú mikrokontrollerek segítségével. Egy ilyen projektbe vágunk bele mi is a következő hónapokban, aminek a célja az hogy egy kombinált Nap- és horizont-szenzor rendszerként tudjuk használni kicsi műholdaknál. Ez az infraszenzor is I2C-n keresztül működtethető, egy buszra többet is rakhatunk tudunk aggregálni, és a mikrokontroller vezérelné a körkörös kiolvasást + adatfeldolgozást.

A másik, ami nem konkrét feladat hanem inkább fejlesztés-önképzés az az FPGA-kon a szoft processzoros környezet tesztelése-fejlesztése, mostmár immáron a perifériákra fókuszálva és a későbbiekben egy 4-stage instruction pipeline-ra épülő AVR processzor készítése. Ezek már értelemszerűen kicsit haladobb témák így első olvasatra de egyszerűbb perifériák (pl PWM, timer, systick-ek, input caputre, ...) tervezése az kellemesen jól elsajátítható mind a C tanulóssal, mind a HDL/Verilog tanulóssal párhuzamosan is - úgy hogy közben már akar produktív alkalmazása is lenne előbbutóbb. Itt viszont az egy határozott elony hogy van Arduino-s tapasztalatod - mert maga a soft CPU is teljesen hasonló

felepitesu. A filozofiai kulonbseg ugye az hogy mig az  
Arduino/ATmega  
vonalon  
adottak a periferiak ("ez van, ezt kell szeretni" modon  
jonnek  
ezek  
az  
ic-k), a  
soft megkozelitesnek az az elonye hogy csak azokat a  
periferiakat  
tesszuk  
bele  
ami kell, azokat viszont olyan darabszamban ami epp' kell  
nekunk.  
megoldhato  
A linuxos dologgak kapcsolatban meg egy kerdes: az mennyire  
hogyan az  
altalad hasznalt gep(ek)en reszben az fusson? Masodik  
oprendszer,  
dual  
boot,  
ilyesmi? Onmagaban a virtualis gep sem lenne rossz, de az  
elektronika  
miatt  
jobb ha a linux maga is hardverkozelibb.  
udv, Andras  
On Mon, 12 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt András Pál!  
Korábban egy már meglevő ötlet alapján építettem egy bináris  
számológépet  
TTL logikai kapuk használatával. Ezáltal a kombinációs  
logikával  
szerintem  
képben vagyok, a szekvenciálisról is már olvastam többet, de  
arról  
mélyebb  
tudásom nincs. HDL-ekhez nem igazán értek, a nyáron  
szerettem  
volna  
bevezető  
rész  
olyan tudásra épített, ami akkor még nem állt  
rendelkezésemre.  
Linuxot nem igazán használtam eddig, amit használtam, az a  
Raspberry  
Debian  
alapú operációs rendszere, ami, ha jól tudom egy módosított  
linux  
rendszer.  
Annak nagyon örülök, hogy ebből a demonstrátori lehetőségből  
hasznos  
nyelvet tanulok (C), de mivel most tanulom így még nem  
hiszem,  
hogy  
hatékonyan tudnám használni.  
A felsoroltak közül az I2C-t alkalmaztam már többször is, a  
legutóbb

a  
járásdiagnosztikai eszköz 7db szenzorának olvasásakor  
jelentett megoldást a számomra. Viszont az SPI működési elve alapján gyorsabbnak bizonyult, mely az eszköz számára nagyon fontos lenne, így arra fogunk áttérni.  
Az I2C-vel ellentétben az SPI-t még nem alkalmaztam gyakorlatban, de a működését már tanulmányoztam többször is. A többiről csak hallottam.  
Üdvözlettel:  
Kreinicker Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 12., H, 13:14):  
Szia Gábor!  
Koszi az összefoglalót! Na, akkor nezzuk sorban:  
- az Arduinio az határozottan jó, mert mi elég sokat dolgozunk AVR alapokon.  
(ATmega328P) Sok elektronikánk kimonodttan ugyanerre az MCU-ra epul, mint ami az Arduino Uno-ban van. Illetve most mostanában egyre tobbet hasznalun AVR architekturaju, saját fejlesztű soft processzorokat FPGA-kon belül (pl ilyen repül majd most a műholdakban is). Annyi a kulonbseg hogy mi ezeket teljesen "bare metal" modon, saját konyvtarakkal hasznaljuk.  
digitális logikában mennyire vagy jártas (kombinacios vs. szekvencialis logika, ilyesmik)? HDL-eket mennyire tanulmanyoztad akarcsak alapszinten is? Ilyesmi hogy Verilog, esetleg VHDL?  
Beleertve a - az Inventor az határozottan jó, mi is abban dolgozunk! 3d-nyomtatatos projekteket es/vagy prototipizalasokat is.

7/7/21, 18:08



- az FPGA az ez: iCE40HX8K-BG121, ezt kell kiválasztanod majd.
- mellekelem a tegnapi tesztelt kis verilog-kódot is, abban megtalálod a PCF-et: ezt a PCF-et megveszi az iccube2 is! (a verilog-ot meg nyilván)

Illetve az ARM-es dolgokkal kapcsolatban is akkor pár info:

- mellekelem a tegnapi példaprogramot (UART: kisbetű -> nagybetű)
- a toolchain (ami a \*.c fileokból megcsinálja a \*.bin és/vagy \*.hex image-t, amit aztán fel kell tölteni) az `arm-none-eabi` néven fut. Ennek több komponense van:
  - \* A C/C++/asm fordító maga: ez a gcc-arm-none-eabi
  - \* Eszközök amikkel a lefordított fileokat lehet manipulálni (pl a \*.hex-et elolvasni a \*.o/\*.elf-ből): binutils-arm-none-eabi
  - \* És az előre lefordított könyvtárak, hogy a jól megszokott C-s (stdio: printf, stb) meg C++-os (STL és tarsai) dolgokat tudd használni: libnewlib-arm-none-eabi, libstdc++-arm-none-eabi-newlib.

Ez kb ugyanilyen néven fut Windows alatt is, egy gyors kereséssel pl ezt találtam: <https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>. A lényeg hogy ez a "gcc-arm-none-eabi for windows" dologra keress rá!

A programozó az ST-Link V2 (-1), ezt lehet OpenOCD-vel programozni. Az OpenOCD van Linux alá is, de van Windowsos prortja is: <https://gnutoolchains.com/arm-eabi/openocd/>. Ettől függetlenül az ST saját cuccait is lehet használni valószínűleg: <https://www.st.com/en/development-tools/stsw-link009.html>, de az OpenOCD annyira jó hogy Windowsos használat kollegától is azt hallom hogy inkább az OpenOCD-t preferáljak az STM32-es MCU-khoz.

A.

On Fri, 6 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Említett nekem segédleteket a lattice boardhoz. Továbbá említette, hogy az fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is szeretnék kérni segítséget.

Köszönettel: K. Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020. nov. 4., Sze, 17:28):

Rendben!

Szerintem én is megebédelek és indulok

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 17:09):

Szia Gábor!

Megebédelek odalent, kb olyan fel 1-1-ig bezárolag, és akkor utána indulok fel. Szóval fel 2 korútlól mar leszek itt. Ha erre fele jársz akkor hívj fel: (20)3962151, és akkor kimegyek a kapuhoz. Merthogy mi itten egy másik épület másik szekciójában vagyunk.  
üdv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper!  
Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 16:45):

Szia Gábor!

No, tul vagy kozben a mindenen? Illetve a het hetralevo resze akkor mikor lenne jo? En holnap delutan tervezek feljonnai a normafára elso kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás céljából?

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020. okt. 29., Cs 9:01):

Rendben, az nekem is most előnyös lesz. TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz minden, a hetedik hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét elején. Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha minden igaz, az is meglesz.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 28., Sze, 18:44):

Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo heten, mert kozben engem is nagyon bekapott az ekszija es/vagy a daralo a műhold

kapcsan. Most mar szerda van, es meg mindig nem tudom hogy a holnap az hogy lesz. Es attol

felek hogy csak delelott derul ki hogy mar delutan fel kell mennem vagy csak

penteken vagy hogy.

Neked mikor vannak ezek a hataridok es/vagy mikor zarodnak le a dolgok

(melyik is pl az a hetedik het, meg mi a TDK hatarideje, ilyesmik)?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Beleolvastam, de a hetedik heti hajrá és a TDK határidő közelsége nem engedett komolyan elmélyedni még benne. A téma nagyon bejött, a jövőhétre már felkészültebb leszek

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H



23:07):

A doksikat/referenciákat mennyire tudtad megnezni amiket küldtem?  
Kiindulásnak  
ezek mennyire jók? Persze kicsit még levegőben lógnak a témák, de

ettől

függetlenül(?)

A

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm szépen!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H  
10:38):

Igen, a "Normafa" az maga az intézet, Konkoly-Thege M. ut 15-17.

Egyesek

"Svábhegy"-kent is szeretnek rá hivatkozni, az is geografiaiilag

kb

ugyannyira  
akkuratus :)

Csak azért kérdelem mert ennekem ha nem is 100% a Normafa

csütörtökön,

odalent a  
városban (~Móricz korter vidéke) nekem is hatékonyabb az eljutás.

A,

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Ha a Móricz az a körtér, akkor az nagyon közel van. Viszont

nekem  
be

kell

mennem minden képpen a titkárságra aláírni a szerződést, így

ha az

is

Normafán van, akkor szívesen kimegyek oda is.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

26., H

10:24):

Oke, eszbe tartom! Szerda délelőttig akkor diskuráljuk le. Az

is

lehet

hogya

valahol a city-ben összefutunk, hamar ugyanis hóomoffice van és

a

normafa

mindentől is messze van. Neked a Moszkva-Déli-BAH-Móricz

tengely

mennyire  
esik  
útba?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó lehet.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

26.,

H

10:13):

Szia Gábor!

Volt/van egy kis vis maior-unk most a

műhold-elektronikákkal,

így a

köv

napok

kicsit erraticusak nekem. De persze a dolgok parhuzamositasa

mindig

jo

dolog -

így kérdés: ha nem késő es/vagy meg tudnad oldani, akkor mar

kora-kozep

delutan

(2 korulre) mennyire tudnal feljönni a normafára?

Koszi, A.

On Mon, 19 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Először is bocsásson meg, de egyre gyanúsabb, hogy nem

tudom

melyik a

vezeték és melyik a keresztnéve, amely minden

megszólításnál

apró

problémát

okoz.

A nulladik kérdésre, sajnos a csütörtök nem jó, jövőhéten

hétfő

délelőtt,

kedd délután, vagy csütörtök lenne a legszuperebb. A

személyes

tárgyalásnak

/ beszélgetésnek én is nagyon örülnék.

Amiket leírt, azok nagyon felcsigáztak, alig várom, hogy

belekezdjünk.

Azonban, ha nem probléma, szeretném megérdeklődni, hogy a

fényűtváltó rendszer tervezésével mi lesz a továbbiakban?

Úgy

vettem

észre,

hogy jelentősen kezdünk eltérni a SPICA témától. Ne értsen

félre,

természetesen nagyon érdekel és nagyon szívesen csinálom

ezt

is,

különösen

az FPGA és műhold témák nagyon érdekelnek, melyeket a

korábbi

email-jeiben említett meg, de érdekel, mi lesz annak a

projektnek a

jövője.

A linux rendszer természetesen megoldható, korábban már szerettem volna, hogy legyen a gépemen, habár a szerzés módja még nem világos.

Köszönettel:  
Kreinicker Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 19., H, 16:56):

Szia Gábor!

No, bocsanat a kesei reakcioert, kozbejott egy mátrai kiruccanas.

Nulladik kerdes: most csutortokon (okt 22) ossze tudnak-e futni fent az intezetben, a Normafan? En leszek akkor mar fent (igy a hatarfeltetelek mellett is) es akkor tudunk egy kicsit diskuralni kozvetlenul is.

Vagyis, pont a mátrai kiruccanasnak meg mindig van egy kis bizonytalansaga, de nagyon bizom benne hogy mar szerdan visszaterek Budapestre.

Elso korben ami lehet majd mint egy onnallo feladat az infraszenzorok (MLX90640 tipusuak) kiolvasasa + adatfeldolgozasa STM32F0 sorozatu mikrokontrollerek segitsegevel. Egy ilyen projektbe vagnank bele mi is a kovetkezo honapokban, aminek a celja az hogy egy kombinált Nap- es horizont-szenzor rendszerkent tudjuk hasznalni kicsi muholdaknal.

Ez az infraszenzor is I2C-n keresztul mukodtetheto, egy buszra tobbet is tudunk aggatni, es a mikrokontroller vezernelne a körkörös kiolvasasat + adatfeldolgozast.

A masik, ami nem konkret feladat hanem inkabb fejlesztés-önképzés az

az  
tesztelese-fejlesztese,  
mostmar  
immaron  
a periferiakra fókuszálva és a későbbiekben egy 4-stage  
instruction  
pipeline-ra  
épülő AVR processzor készítése. Ezek már értelemszerűen  
kicsit  
haladabb  
temak  
így első olvasatra de egyszerűbb perifériák (pl PWM,  
timerek,  
systick-ek,  
input  
caputre, ...) tervezése az kellemesen jól elsajátítható  
mind  
a  
C  
tanulással,  
mind a HDL/Verilog tanulással parhuzamosan is - úgy hogy  
közben  
már  
akar  
produktív alkalmazása is lenne előbbutobb. Itt viszont az  
egy  
határozott  
elony  
hogy van Arduino-s tapasztalatod - mert maga a soft CPU is  
teljesen  
hasonlo  
felepitesu. A filozofiai kulonbseg ugye az hogy mig az  
Arduino/ATmega  
vonalon  
adottak a periferiak ("ez van, ezt kell szeretni" módon  
jonnek  
ezek  
az  
ic-k), a  
soft megkozelitesnek az az elonye hogy csak azokat a  
periferiakat  
tesszuk  
bele  
ami kell, azokat viszont olyan darabszamban ami epp' kell  
nekunk.  
A linuxos dologgak kapcsolatban meg egy kerdes: az  
mennyire  
megoldhato  
hogy az  
altalad hasznalt gep(ek)en reszben az fusson? Masodik  
oprendszer,  
dual  
boot,  
ilyenmi? Onmagaban a virtualis gep sem lenne rossz, de az  
elektronika  
miatt  
jobb ha a linux maga is hardverkozelibb.  
udv, Andras  
On Mon, 12 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt András Pál!

bináris

számológépet

TTL logikai kapuk használatával. Ezáltal a kombinációs

logikával

szerintem

képpen vagyok, a szekvenciálisról is már olvastam

többet, de

arról

mélyebb

tudásom nincs. HDL-ekhez nem igazán értek, a nyáron

szerettem

volna

FPGA-król tanulni, de ez sajnos nem így alakult, mivel

már a

bevezető

rész

olyan tudásra épített, ami akkor még nem állt

rendelkezésemre.

Linuxot nem igazán használtam eddig, amit használtam, az

a

Raspberry

Debian

alapú operációs rendszere, ami, ha jól tudom egy

módosított

linux

rendszer.

Annak nagyon örülök, hogy ebből a demonstrátori

### 3 melléklet



**iCE40HX8K-B-EVN.pdf**

1594K



**payload-hx8k-bg121.tgz**

1K



**stm32f072-test.tgz**

4K

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. november 7. 14:31

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos tanulmányozásának.

Egy közeli ismerősöm ajánlásából elkezdtük föltelepíteni az ubuntu-t, mivel a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem volt zökkenőmentes a telepítés. Először a pendrive amin keresztül telepíteni akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt föl. Majd miután sikerült, a Grub nem települt így nem tudtam elindítani a rendszert. Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és egyebek. Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált adataimat el kell fogadják, de az is hamarosan meglesz.

A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött pdf-eket át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni. Szerencsére az is teljesen témába vág, hiszen C programozás lesz. Nagyon le voltam nyűgözve a találkozókör és remélem minden rendben lesz!

Köszönettel:

K. Gábor

[Az idézett szöveg el van rejtve]

[Az idézett szöveg el van rejtve]

[Az idézett szöveg el van rejtve]

>>>>>>>>>>>>>>> Kreinicker Gábor

[illegible]

>>>>>>>>>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.

>>> okt.

[Az idézett szöveg el van rejtve]

>>>>>>>>>>>>>>> udv, Andras

[illegible][illegible][illegible]

>>>>>>>>>>>>> Tisztelt András Pál!

[illegible]

[Az idézett szöveg el van rejtve]

[illegible][illegible]

>>>>>>>>>>>>>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.

>>> okt.

[Az idézett szöveg el van rejtve]

**Andras Pal** <apal@szofi.net>

2020. november 7. 16:02

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper lesz, azon nagyjából mindent is meg lehet csinálni amit debian-on, és hogyha standalone és/vagy desktop üzemben használad akkor tényleg nem lehet gond! Headless üzemben már nem használnék egyáltalán ubuntu-t, de az más kérdés ;)

Linux alá viszont van teljesen free & open source Lattice FPGA toolchain: <http://www.clifford.at/icestorm/>, ezt erdemes lehet feltelepíteni. Mi is használjuk ezt (yosys-0.9, nextpnr-ice40, icestorm-utils), ezekkel a csomagokkal közvetlenül is tudod linux alatt mindenféle licenc-problema meg nyűg nélkül használni a barmit! Cserebe a forrás(kod)bol való telepítés nem annyira egyszerű, elég sok a függőség, és nehezen derül ki hogy mi az. De ebben viszont a windowsos cuccokkal ellentétben tudok segíteni :) Ugyanakkor természetesen én is nézem a linuxos icecube2-t, és az is működik persze, szóval hajrá azzal is. Mondjuk az nekem kicsit nehezkés, a "bare metal" meg a klasszikus "unix filozofia" dolgok után nekem ezeket az IDE-ket elég nehéz megszoknom :) De az más kérdés ;)

A.

On Sat, 7 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos tanulmányozásának. Egy közeli ismerősöm ajánlásából elkezdtük föltelepíteni az ubuntu-t, mivel a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem volt zökkenőmentes a telepítés. Először a pendrive amin keresztül telepíteni akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt föl. Majd miután sikerült, a Grub nem települt így nem tudtam elindítani a rendszert. Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és egyebek. Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált adataimat el kell fogadják, de az is hamarosan meglesz. A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött pdf-eket át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni. Szerencsére az is teljesen témába vág, hiszen C programozás lesz. Nagyon le voltam nyűgözve a találkozókór és remélem minden rendben lesz!

Köszönettel:

K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6., P. 9:44):

Szia Gábor!

Persze, kuldom akkor az infokat:

- mellekelve az evaluation board doksija, ebben benne kell legyen minden referencia ami hasznos lehet - bar ugye a mi boardunk az masmilyen.
- a fejlesztokeszlet az ez: <https://www.latticesemi.com/iCEcube2>, van minden

oprendszer ala (linux, windoz).

- az FPGA az ez: iCE40HX8K-BG121, ezt kell kiválasztanod majd.
- mellekelem a tegnapi tesztelt kis verilog-kodot is, abban megtalálod a PCF-et: ezt a PCF-et megcsí az icecube2 is! (a verilog-ot meg nyilván)

Illetve az ARM-es dolgokkal kapcsolatban is akkor pár info:

- mellekelem a tegnapi példaprogramot (UART: kisbetű -> nagybetű)
- a toolchain (ami a \*.c fileokból megcsinálja a \*.bin és/vagy \*.hex image-t,

amit aztán fel kell tölteni) az `arm-none-eabi` néven fut. Ennek több komponense van:

- \* A C/C++/asm fordító maga: ez a gcc-arm-none-eabi
- \* Eszközök amikkel a lefordított fileokat lehet manipulálni (pl a \*.hex-et eloallítani a \*.o/\*elf-ből): binutils-arm-none-eabi
- \* Es az előre lefordított könyvtárak, hogy a jól megszokott C-s (stdio: printf, stb) meg C++-os (STL és társai) dolgokat tud használni: libnewlib-arm-none-eabi,

libstdc++-arm-none-eabi-newlib.

Ez kb ugyanilyen néven fut windoz alatt is, egy gyors kereséssel pl ezt találtam:

<https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>.

A lényeg hogy ez a "gcc-arm-none-eabi for windows" dologra keress rá!

A programozó az ST-Link V2 (-1), ezt lehet OpenOCD-vel programozni. Az OpenOCD

van linux alá is, de van windozos prortja is:

<https://gnutoolchains.com/arm-eabi/openocd/>. Ettől függetlenül az ST saját

cuccait is lehet használni valószínűleg:

<https://www.st.com/en/development-tools/stsw-link009.html>, de az OpenOCD annyira jó hogy windozt használó kollegától is azt hallom hogy inkább az OpenOCD-t preferálják az STM32-es MCU-khoz.

A.

On Fri, 6 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Említett nekem segédleteket a lattice boardhoz. Továbbá említette, hogy az

fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is szeretnék kérni segítséget.

Köszönettel: K. Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020. nov.

4., Sze, 17:28):

Rendben!

Szerintem én is megebédelek és indulok

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 17:09):

Szia Gábor!

Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, és akkor utána indulok

fel. Szóval fel 2 korultól már leszek itt. Ha erre fele jársz akkor hívj

fel:

(20)3962151, és akkor kimegyek a kapuhoz. Merthogy mi itten egy másik

epulet  
masik szekciojaban vagyunk.  
udv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper!  
Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 16:45):

Szia Gábor!

No, tul vagy kozben a mindenen? Illetve a het hetralevo resze akkor  
mikor lenne  
jo? En holnap delutan tervezek feljonni a normafára elso  
kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás  
céljából?

Köszönettel: Gábor

2020. Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont:

okt.  
29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.  
TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz minden, a  
hetedik hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét elején.  
Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha minden  
igaz,  
az is meglesz.

Üdv: Gábor

Sze, Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 28.,  
18:44):

mert Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo heten,

kozben engem is nagyon bekapott az ekszija es/vagy a daralo a műhold  
kapcsan.

Most mar szerda van, es meg mindig nem tudom hogy a holnap az hogy lesz. Es

attol felek  
hogy csak delelőtt derül ki hogy mar delutan fel kell mennem vagy

csak penteken



vagy hogy.

Neked mikor vannak ezek a határidők és/vagy mikor záródnak le a

dolgok

(melyik is pl az a hetedik hét, meg mi a TDK határideje, ilyesmik)?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Beleolvastam, de a hetedik heti hajrá és a TDK határidő közelsége

nem

engedett komolyan elmélyedni még benne. A téma nagyon bejött, a jövőhétre már felkészültebb leszek

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 23:07):

A doksikát/referenciákat mennyire tudtad megnevezni amiket

küldtem?

Kiindulásnak ezek mennyire jók? Persze kicsit még levegőben lógnak a témák,

de

ettől

függetlenül(?)

A

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm szépen!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

26., H

10:38):

Igen, a "Normafa" az maga az intézet, Konkoly-Thege M. ut

15-17.

Egyesek

"Svábhegy"-kent is szeretnek rá hivatkozni, az is

geografiaiilag

kb

ugyannyira akkuratus :)

Csak azért kérdelem mert ennekem ha nem is 100% a Normafa

csütörtökön,

odalent a városban (~Móricz korter vidéke) nekem is hatékonyabb az

eljutás.

A,

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Ha a Móricz az a körtér, akkor az nagyon közel van. Viszont

nekem

be

kell

mennem minden képpen a titkárságra aláírni a szerződést, így

ha az

is

Normafán van, akkor szívesen kimegyek oda is.

26., H 10:24):  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.  
is lehet hogy  
valahol a city-ben összefutunk, hamar ugyis hoómmooffice van  
es  
a normafa  
tengely mindentol is messze van. Neked a Moszkva-Déli-BAH-Moricz  
mennyre  
esik  
utba?  
A.  
On Mon, 26 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt Pál András!  
Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó lehet.  
Üdv: Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.  
26., H 10:13):  
Szia Gábor!  
Volt/van egy kis vis maior-unk most a  
műhold-elektronikákkal,  
igy a köv  
napok  
kicsit erraticusak nekem. De persze a dolgok  
parhuzamositasa  
mindig  
jo  
dolog -  
igy kerdes: ha nem keső es/vagy meg tudnad oldani, akkor  
mar  
kora-kozep  
delutan  
(2 korulre) mennyire tudnal feljönni a normafára?  
Koszi, A.  
On Mon, 19 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt Pál András!  
Először is bocsásson meg, de egyre gyanúsabb, hogy nem  
tudom melyik a  
vezeték és melyik a keresztnéve, amely minden  
megszólításnál  
apró  
problémát  
okoz.

A nulladik kérdésre, sajnos a csütörtök nem jó, jövőhéten  
hétfő délelőtt,  
kedd délután, vagy csütörtök lenne a legszuperebb. A  
személyes tárgyalásnak  
/ beszélgetésnek én is nagyon örülnék.  
Amiket leírt, azok nagyon felcsigáztak, alig várom, hogy  
belekezdjünk.  
Azonban, ha nem probléma, szeretném megérdeklődni, hogy a  
fényútváltó rendszer tervezésével mi lesz a továbbiakban?  
Úgy vettem észre,  
hogy jelentősen kezdünk eltérni a SPICA témától. Ne  
értsen félre,  
természetesen nagyon érdekel és nagyon szívesen csinálom  
ezt is,  
különösen  
az FPGA és műhold témák nagyon érdekelnek, melyeket a  
korábbi email-jeiben említett meg, de érdekel, mi lesz annak a  
projektnek a jövője.  
A linux rendszer természetesen megoldható, korábban már  
szerettem volna,  
hogy legyen a gépemen, habár a szerzés módja még nem  
világos.  
Köszönettel:  
Kreinicker Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.  
okt. 19., H, 16:56):  
Szia Gábor!  
No, bocsanat a kesei reakcioert, kozbejott egy mátrai  
kiruccanas.  
Nulladik kerdes: most csutortokon (okt 22) ossze  
tudnak-e futni fent az  
intezetben, a Normafan? En leszek akkor mar fent (igy a  
hatarfeltetelek mellett  
is) es akkor tudunk egy kicsit diskuralni kozvetlenul  
is.  
Vagyis, pont a mátrai  
nagyon kiruccanasnak meg mindig van egy kis bizonytalansaga, de  
bizom

					benne	
					hogy mar szerdan visszaterek Budapestre.	
					Elso korben ami lehet majd mint egy onnallo feladat az	
				lekepezo		
					infraszenzorok	
					(MLX90640 tipusuak) kiolasasa + adatfeldolgozasa	
STM32F0						
				sorozatu		
					mikrokontrollerek segitsegevel. Egy ilyen projektbe	
				vagnank		
				bele		
				mi		
					is a	
					kovetkezo honapokban, aminek a celja az hogy egy	
kombinalt						
				Nap- es		
				muholdaknal.		horizont-szenzor rendszerkent tudjuk hasznalni kicsi
				Ez		
					az	
				tobbet is		infraszenzor is I2C-n keresztul mukodtetheto, egy buszra
					ra	
					tudunk	
					aggatni, es a mikrokontroller vezernelne a korkoros	
				kiolasasat		
				+		
					adatifdolgozast.	
					A masik, ami nem konkret feladat hanem inkabb	
				fejlesztos-onkcpzs		
				az		
					az	
				FPGA-kon a szoft processzos kornyezet		
				tesztelese-fejlesztese,		
				mostmar		
					immaron	
					a periferiakra fokukszalva es a kesobbiekben egy 4-stage	
				instruction		
					pipeline-ra	
					epulo AVR processzor keszitese. Ezek mar ertelemszeruen	
				kicsit		
					haladobb	
					temak	
					igy elso olvasatra de egyszerubb periferiak (pl PWM,	
				timerek,		
					systick-ek,	
					input	
					caputre, ...) tervezese az kellemesen jol elsajatithato	
				mind		
				a		
				C		
					tanulassal,	
					mind a HDL/Verilog tanulassal parhuzamosan is - ugy hogy	
				kozben		
				mar		
					akar	
					produktiv alkalmazasa is lenne elobbutobb. Itt viszont	
az						
				egy		
					hataroztt	
					elony	
					hogy van Arduino-s tapasztalatod - mert maga a soft CPU	
is						
				teljesen		

hasonlo  
felepitesu. A filozofiai kulonbseg ugye az hogy mig az  
Arduino/ATmega  
vonalon  
adottak a periferiak ("ez van, ezt kell szeretni" modon  
jonnek  
ezek  
az  
ic-k), a  
soft megkozelitesnek az az elonye hogy csak azokat a  
periferiakat  
tesszuk  
bele  
ami kell, azokat viszont olyan darabszamban ami epp'

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. november 16. 13:52

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a küldött dolgokat olvastam. Sajnos sem az openocd-t, sem az icestorm toolchain-t nem sikerült "éltre keltenem". Elvileg a yosys-0.9, nextpnr-ice40 és icestorm-utlis csomagokat is telepítettem a megadott módon. Tud esetleg küldeni valamilyen segítséget, hogyan tudnám programozni ezeket a lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék segítséget kérni, telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai eszközünkkel TDK csoportunkban első lettünk, így a Semmelweis Egyetem TDK mellett készülhetünk az országos TDK-ra. Mivel már kész van az eszköz, és csak bemutatni kell, így nem vesz el jelentős időt.

Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy két szabadsági fokból álló állványon levő parabola-antenna fogja "bescannelni" az eget, és a vett jelerősséget fogja egy skála szerint osztályozni, majd ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni. Fő célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell érzékelnünk. Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő érzékelővel. Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb lenne) csinálni?

Köszönettel: K. Gábor

[Az idézett szöveg el van rejtve]

[Az idézett szöveg el van rejtve]

[Az idézett szöveg el van rejtve]

[illegible][illegible]

>>>>>>>>>>>>>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.

>>>>> okt.

[Az idézett szöveg el van rejtve]

[illegible][illegible]

```
>>>>> okt
```

[Az idézett szöveg el van rejtve]

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. november 16. 18:34

Szia Gábor!

Ah, értem, ez így akkor elsore tenyleg nem egyszerű. Viszont ennél specifikusabban kene tudni hogy hol is akadtál el. Tehat:

- telepites ment? Minden fent van?
- az arm-es toolchain az ezen csomagokbol all:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi
  - \* libnewlib-arm-none-eabi,
  - \* libstdc++-arm-none-eabi-newlib.
- ami meg hasznos csomag es kellhet:
  - \* build-essential (ebben van helyi C fordito is + a `make` program)
  - \* srecord
- openocd az csak siman ez:
  - \* openocd

Mindegyiket siman `apt-get install ...` modon tudod telepiteni! Ezek sikerultek? Ha ezek mind fent vannak, akkor az altalam mellekelt forrast mar le tudod fordítani es fel kell tudnod tölteni a kis board-ra. Ami meg kellhet a forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom mellekelve. En konkretan sima rendszergazdakent az `/usr/local/arm-none-eabi/include` alkovnytár alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt tesztprogramban levő Makefile-ban az elejen (kb 12-15ik sorban) levő

ARM=/usr/local/arm-none-eabi

sort modositd oda ahova betetted! Lehet siman nálad is, a `/home-od` alatt! De ugyis panaszkodik a fordito hogyha nem találja ;) Szoval legjobb emlekeim szerint ennyi kell, es ami meg hianyzik az ezeknek a csomagoknak a fuggosege - szoval jo kell hogy legyen, legjobb emlekeim szerint.

Az FPGA az sajnos tenyleg egy fokkal bonyolultabb. Elobb nezzuk meg ezt az ARM + STM32 témát, es akkor utana. Valojaban azokat sem nehez feltenni, de kellhet par olyan development header (pl libboost-csomag) amiket feltetelez hogy fent van de alapbol szinte biztos hogy nincs fent nálad. Nalam konkretan ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-filesystem-dev:amd64	install
libboost-filesystem1.62-dev:amd64	install
libboost-filesystem1.62.0:amd64	install
libboost-iostreams-dev:amd64	install
libboost-iostreams1.62-dev:amd64	install
libboost-iostreams1.62.0:amd64	install
libboost-program-options-dev:amd64	install
libboost-program-options1.62-dev:amd64	install
libboost-program-options1.62.0:amd64	install
libboost-python-dev	install
libboost-python1.62-dev	install
libboost-python1.62.0	install
libboost-regex1.62-dev:amd64	install
libboost-regex1.62.0:amd64	install
libboost-serialization1.62-dev:amd64	install
libboost-serialization1.62.0:amd64	install
libboost-system1.62-dev:amd64	install
libboost-system1.62.0:amd64	install
libboost-thread-dev:amd64	install
libboost-thread1.62-dev:amd64	install
libboost-thread1.62.0:amd64	install
libboost1.62-dev:amd64	install

szoval ha azt mondod hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev libboost-chrono-dev libboost-date-time-dev libboost-dev libboost-filesystem-dev libboost-iostreams-dev libboost-program-options-dev libboost-python-dev libboost-regex-dev libboost-serialization-dev libboost-system-dev libboost-thread-dev
```

(igen, ez igy egy jó hosszú parancssor :)), akkor mindent is feltesz ami kell a libboosthoz (es a sima lib-eket is felteszi ami kell a \*-dev csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De tenyleg ez kicsit macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi az ami hianyzik. Hanyas verzioju milyen linuxot is tettél fel pontosan?

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a küldött dolgokat olvasgattam. Sajnos sem az openocd-t, sem az icedstorm toolchain-t nem sikerült "éltre keltenem". Elvileg a yosys-0.9, nextpnr-ice40 és icedstorm-utils csomagokat is telepítettem a megadott módon. Tud esetleg küldeni valamilyen segítséget, hogyan tudnám programozni ezeketül a lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék segítséget kérni, telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai eszközünkkel TDK csoportunkban elsőkké lettünk, így a Semmelweis Egyetem TDK mellett készülhetünk az országos TDK-ra. Mivel már kész van az eszköz, és csak bemutatni kell, így nem vesz el jelentős időt.

Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy két szabadsági fokból álló állványon levő parabola-antenna fogja "bescannelni" az eget, és a vett jelerősséget fogja egy skála szerint osztályozni, majd ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni. Fő célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell érzékelnünk. Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő érzékelővel. Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb lenne) csinálni?

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo, 16:02):

Szia Gábor!

Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper lesz, azon nagyjából mindent is meg lehet csinálni amit debian-on, es hogyha standalone es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond! Headless üzemben mar nem használnék egyáltalán ubuntu-t, de az mas kerdes ;)

Linux alá viszont van teljesen free & open souce Lattice FPGA toolchain: <http://www.clifford.at/icedstorm/>, ezt erdemes lehet feltelepíteni. Mi is hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icedstorm-utils), ezekkel a csomagokkal kozvetlenül is tudod linux alatt mindenféle licenc-problema megnyűg nélkül használni a barmit! Cserebe a forras(kod)bol valo telepites nem annyira egyszerű, eleg sok a függőség, es nehezen derül ki hogy mi az. De ebben viszont a windozos cuccokkal ellentétben tudok segiteni :) Ugyanakkor természetesen en is neztem a linuxos icecube2-t, es az is mukodik persze, szoval hajra azzal is. Mondjuk az nekem kicsit nehezkes, a "bare metal" meg a klasszikus "unix filozofia" dolgok utan nekem ezeket az IDE-ket eleg nehez megszoknom :/ De az mas kerdes ;)

A.

On Sat, 7 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos tanulmányozásának.

Egy közeli ismerősöm ajánlásából elkezdtük föltelepíteni az ubuntu-t, mivel

a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem volt zökkenőmentes a telepítés. Először a pendrive amin keresztül telepíteni akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt föl. Majd miután sikerült, a Grub nem települt így nem tudtam elindítani a rendszert.

Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és egyebek.

Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált adataimat el

kell fogadják, de az is hamarosan meglesz.

A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött pdf-eket

át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.

Szerencsére

az is teljesen témába vág, hiszen C programozás lesz.

Nagyon le voltam nyugözve a találkozókor és remélem minden rendben lesz!

Köszönettel:

K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6., P, 9:44):

Szia Gábor!

Persze, kuldom akkor az infokat:

- mellekelve az evaluation board doksija, ebben benne kell legyen minden

referencia ami hasznos lehet - bar ugye a mi boardunk az masmilyen.

- a fejlesztokeszlet az ez: <https://www.latticesemi.com/iCEcube2>, van minden

oprendszer ala (linux, windoz).

- az FPGA az ez: iCE40HX8K-BG121, ezt kell kivlasztanod majd.

- mellekelem a tegnap tesztelt kis verilog-kodot is, abban

megtalalad a

PCF-et: ezt a PCF-et megeszti az icecube2 is! (a verilog-ot meg nyilván)

Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:

- mellekelem a tegnapi peldaprogramot (UART: kisbetu -> nagybetu)

- a toolchain (ami a \*.c fileokbol megcsinalja a \*.bin es/vagy \*.hex

image-t,

amit aztan fel kell tolneni) az `arm-none-eabi` néven fut. Ennek tobb

komponense van:

\* A C/C++/asm fordito maga: ez a gcc-arm-none-eabi

\* Eszkozok amikkel a leforditott fileokat lehet manipualni (pl

a

\*.hex-et eloallitani a \*.o/\*.elf-ből): binutils-arm-none-eabi

\* Es az elore leforditott konyvtaraok, hogy a jól megszokott C-s

(stdio: printf, stb) meg C++-os (STL es tarsai) dolgokat tudd

hasznalni: libnewlib-arm-none-eabi,

libstdc++-arm-none-eabi-newlib.

Ez kb ugyanilyen néven fut windoz alatt is, egy gyors keresessel pl ezt talaltam:

<https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>



A lenyeg hogy ez a "gcc-arm-none-eabi for windows" dologra keress rá!

A programozó az ST-Link V2 (-1), ezt lehet OpenOCD-vel programozni. Az OpenOCD van linux alá is, de van windowsos prortja is:  
<https://gnutoolchains.com/arm-eabi/openocd/>. Ettől függetlenül az ST saját cuccait is lehet használni valószínűleg:  
<https://www.st.com/en/development-tools/stsw-link009.html>, de az

OpenOCD

annyira jó hogy windowst használó kollegáktól is azt hallom hogy inkább az

OpenOCD-t preferálják az STM32-es MCU-khoz.

A.

On Fri, 6 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Említett nekem segédleteket a lattice boardhoz. Továbbá említette, hogy az fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is szeretnék kérni segítséget.

Köszönettel: K. Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020. nov.

4., Sze, 17:28):

Rendben!

Szerintem én is megebédelek és indulok

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 17:09):

Szia Gábor!

Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, és akkor utána indulok fel. Szóval fel 2 korultól már leszek itt. Ha erre fele jász akkor

hívj

fel:  
 (20)3962151, és akkor kimegyek a kapuhoz. Merthogy mi itten egy másik épület másik szekciójában vagyunk.  
 udv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper!  
 Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 16:45):

Szia Gábor!

No, túl vagy közben a mindenen? Illetve a hét hetralevo része akkor mikor lenne

jo? En holnap delutan tervezek feljonni a normafára elso  
kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás  
céljából?

Köszönettel: Gábor

2020. Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont:

okt.  
29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.  
TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz minden, a  
hetedik hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét

elején.  
Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha  
minden

igaz,  
az is meglesz.

Üdv: Gábor

Sze, Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 28.,

18:44):

mert Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo heten,

kozben  
engem is nagyon bekapott az ekszij es/vagy a daralo a műhold

kapcsan.

Most mar  
szerda van, es meg mindig nem tudom hogy a holnap az hogy lesz.

Es

attol  
felek  
hogy csak delelott derul ki hogy mar delutan fel kell mennem

vagy

csak  
penteken  
vagy hogy.

Neked mikor vannak ezek a hataridok es/vagy mikor zarodnak le a

dolgok

(melyik  
is pl az a hetedik het, meg mi a TDK hatarideje, ilyesmik)?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Beleolvastam, de a hetedik heti hajrá és a TDK határidő

közelsége

nem

engedett komolyan elmélyedni még benne. A téma nagyon bejött, a  
jövőhétre  
már felkészültebb leszek

26., H Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

23:07):

kuldtem?

A doksikat/referenciákat mennyire tudtad megnevezni amiket

de

Kiindulásnak

ezek mennyire jók? Persze kicsit még levegőben lognak a témák,

ettől

függetlenül(?)

A

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

szépen!

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm

26., H

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

10:38):

15-17.

Igen, a "Normafa" az maga az intézet, Konkoly-Thege M. ut

Egyesek

"Svábhegy"-kent is szeretnek rá hivatkozni, az is

geografailag

kb

ugyannyira

akkuratus :)

csutortokon,

Csak azért kérdezem mert ennem ha nem is 100% a Normafa

odalent a

városban (~Móricz korter vidéke) nekem is hatékonyabb az

eljutás.

A,

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Ha a Móricz az a körtér, akkor az nagyon közel van. Viszont

nekem

be

kell

mennem minden képpen a titkárságra aláírni a szerződést,

így

ha az

is

Normafán van, akkor szívesen kimegyek oda is.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

26., H

10:24):

Az

Oke, esz bentartom! Szerda délelőttig akkor diskuráljuk le.

is

lehet

hogyan

valahol a city-ben összefutunk, hamar ugyanis hóomoffice van

es

a

normafa

tengely

mindentől is messze van. Neked a Moszkva-Déli-BAH-Moricz

mennyire  
esik  
utba?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó lehet.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 10:13):

Szia Gábor!

Volt/van egy kis vis maior-unak most a műhold-elektronikákkal, így a köv napok kicsit erraticusak nekem. De persze a dolgok parhuzamosítása mindig jó dolog - így kérdés: ha nem késő és/vagy meg tudnád oldani, akkor mar kora-közép délután (2 körülre) mennyire tudnál feljönni a normafára?

Kösz, A.

On Mon, 19 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Először is bocsásson meg, de egyre gyanúsabb, hogy nem tudom melyik a vezeték és melyik a keresztnéve, amely minden megszólításnál apró problémát okoz.

A nulladik kérdésre, sajnos a csütörtök nem jó, jövőhéten hétfő délelőtt, kedd délután, vagy csütörtök lenne a legszuperebb. A személyes tárgyalásnak / beszélgetésnek én is nagyon örülnék.

Amiket leírt, azok nagyon felcsigáztak, alig várom,

hogy belekezdjünk.  
Azonban, ha nem probléma, szeretném megérdeklődni,  
hogy a  
továbbiakban? fényűtváltó rendszer tervezésével mi lesz a  
Úgy  
vettem észre,  
értsen hogy jelentősen kezdünk eltérni a SPICA témától. Ne  
félre,  
természetesen nagyon érdekel és nagyon szívesen  
csinálok  
ezt is,  
különösen  
korábbi az FPGA és műhold témák nagyon érdekelnek, melyeket a  
projektnek a email-jeiben említett meg, de érdekel, mi lesz annak a  
jövője.  
szerettem A linux rendszer természetesen megoldható, korábban már  
volna, hogy legyen a gépemen, habár a szerzés módja még nem  
világos.  
Köszönettel:  
Kreinicker Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.  
okt.  
19.,  
H,  
16:56):  
Szia Gábor!  
No, bocsanat a kesei reakcioert, kozbejott egy mátrai  
kiruccanas.  
Nulladik kerdes: most csutortokon (okt 22) össze  
tudnak-e  
futni  
fent  
az  
intezetben, a Normafan? En leszek akkor mar fent (igy  
a  
hatarfeltetelek  
mellett  
is) es akkor tudunk egy kicsit diskuralni kozvetlenul  
is.  
Vagyis,  
pont a  
matrai  
de kiruccanasnak meg mindig van egy kis bizonytalansaga,  
nagyon  
bizom  
benne  
hogy mar szerdan visszaterek Budapestre.



**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent telepítettem. Ami még nem sikerült, az az openocd életre bírása. Csatolok egy képet, és bízom benne, hogy tud segíteni. Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az st-link csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a feltöltésig nem jutottam el.

Köszönettel: K. Gábor

```
kreinicker@kreinicker:~$ cd Desktop
kreinicker@kreinicker:~/Desktop$ dir
stm32
kreinicker@kreinicker:~/Desktop$ cd stm32
kreinicker@kreinicker:~/Desktop/stm32$ dir
config.h crt0.c delay_basic.h main.c main.ld Makefile stm32_gpi
kreinicker@kreinicker:~/Desktop/stm32$ openocd main.c
Open On-Chip Debugger 0.10.0
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
embedded:startup.tcl:60: Error: Can't find openocd.cfg
in procedure 'script'
at file "embedded:startup.tcl", line 60
Error: Debug Adapter has to be specified, see "interface" command
embedded:startup.tcl:60: Error:
in procedure 'script'
at file "embedded:startup.tcl", line 60
kreinicker@kreinicker:~/Desktop/stm32$
```

[Az idézett szöveg el van rejtve]

[Az idézett szöveg el van rejtve]

[Az idézett szöveg el van rejtve]

>>>>>>> okt.

[Az idézett szöveg el van rejtve]

>> 2020.

>>>>>>> okt.

[Az idézett szöveg el van rejtve]

**Andras Pal** <apal@szofi.net>

2020. november 17. 20:36

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a board-ra, nem a main.c-t eteted meg kozvetlenul. Probald ki a `make upload` parancsot, az az ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a Makefile bogaraszasabol rajossz hogy hogy is van ez pontosan.

Szoval a workflow az itten a kovetkezo:

- A \*.c fileokból \*.o (object) fileokat fordítunk (arm-none-eabi-gcc)
- Az egészet összelinkeljük (arm-none-eabi-ld)
- Ez a típusu mikrokontroller a 0x000000004-es címen található utasításra ugrik először, ott kezdi a program futtatását
- Ez a típusu mikrokontroller olyan hogy a 0x000000000 tartományra a 0x08000000 tartományt kepezi le alapértelmezésben

- A linkelésnél így garantálnunk kell hogy legyen "valami" ami a 0x08000004-es címen ami értelmes \_és\_ később meghívja a main() függvényt - merthogy minden C program a main()-nal "kezdődik". Na, ezt a feladatot látja el a crt.c - plusz még pár egyebet is, de azzal nem igazán kell foglalkozkodni, azzal majd később ;)
- A linkeles után ki kell szednünk a keletkező \*.elf fileból a "lenyeget". Ezt csinálja a arm-none-eabi-objcopy, és ez hozzá létre a \*.hex-et.
- Ezt a \*.hex-et pedig az openocd-vel beleírhatjuk a mikrokontroller flash-ébe

Ha megnezed, a Makefile az ebből mindent megcsinál, megfelelő sorrendben. Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen játék. És az openocd konfigurációja sem annyira egyszerű, mert egyrészt meg kell mondani neki hogy milyen típusú programozót használsz, másrészt milyen típusú a target mikrokontroller (azaz a programozónak tudnia kell hogy a programozási protokollon keresztül hogyan éri el a mikrokontroller memóriáját és ez függhet a mikrokontroller konkrét típusától - bár egy családon belül ez elég egyseges, ezért elég csak azt mondani hogy "stm32f0x", és nem azt hogy stm32f072c6t6). Plusz meg vannak egyéb finomságok is, pl a flash-t úgy minden írás előtt törölni kell és csak utána írhatod bele, stb. Ezeket mind-mind megcsinálja a Makefile-be elrejtett parancs-*valami*.

Egyébként ez a fenti sema nagyjából minden beagyazott cuccra ugyanígy megy, csak itt ez a "bare metal" módszer ezt is szüksegesse teszi hogy lépésről lépésre menj(unk) végig a lépéseken.

Az arduino-s projektnél is vsz ugyanígy volt, csak gondolom ott használtak valami fejlesztőkörnyezetet ami mindent is megcsinált, nem? Vagy hogy?

A.

On Tue, 17 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent telepítettem. Ami még nem sikerült, az az openocd életre bírása. Csatolok egy képet, és bízom benne, hogy tud segíteni. Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az st-link csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a feltöltésig nem jutottam el.

Köszönettel: K. Gábor

[image: Screenshot from 2020-11-17 19-53-48.jpg]

Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 16., H, 18:35):

Szia Gábor!

Ah, értem, ez így akkor elsore tenyleg nem egyszerű. Viszont ennél specifikusabban kéne tudni hogy hol is akadtál el. Tehát:

- telepítés ment? Minden fent van?
- az arm-es toolchain az ezen csomagokból áll:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi
  - \* libnewlib-arm-none-eabi,
  - \* libstdc++-arm-none-eabi-newlib.
- ami meg hasznos csomag és kellhet:
  - \* build-essential (ebben van helyi C fordító is + a `make` program)
  - \* srecord
- openocd az csak simán ez:
  - \* openocd

Mindegyiket simán `apt-get install ...` módon tudod telepíteni! Ezek sikerültek? Ha ezek mind fent vannak, akkor az általam mellekelt forrást már le

tudod fordítani és fel kell tudnod tölteni a kis board-ra. Ami meg kellhet a

forgatáshoz azok az STM32-specifikus headerek. Ezeket elküldöm mellekelve.

En

konkrétan sima rendszergazdaként az /usr/local/arm-none-eabi/include alkönyvtár

alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt  
teszprogramban  
levő Makefile-ban az elejen (kb 12-15ik sorban) levő

```
ARM=/usr/local/arm-none-eabi
```

sort modosítsd oda ahova betetted! Lehet siman nálad is, a /home-od alatt!  
De  
ugyis panaszkodik a fordító hogyha nem találja ;) Szóval legjobb emlékeim  
szerint ennyi kell, és ami meg hiányzik az ezeknek a csomagoknak a  
függősege -  
szóval jó kell hogy legyen, legjobb emlékeim szerint.

Az FPGA az sajnos tényleg egy fokkal bonyolultabb. Elobb nezzük meg ezt az  
ARM + STM32 témát, és akkor utána. Valójában azokat sem nehéz feltenni, de  
kellhet pár olyan development header (pl libboost-csomag) amiket  
feltételez  
hogy fent van de alából szinte biztos hogy nincs fent nálad. Nálam  
konkrétan  
ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-filesystem-dev:amd64	install
libboost-filesystem1.62-dev:amd64	install
libboost-filesystem1.62.0:amd64	install
libboost-iostreams-dev:amd64	install
libboost-iostreams1.62-dev:amd64	install
libboost-iostreams1.62.0:amd64	install
libboost-program-options-dev:amd64	install
libboost-program-options1.62-dev:amd64	install
libboost-program-options1.62.0:amd64	install
libboost-python-dev	install
libboost-python1.62-dev	install
libboost-python1.62.0	install
libboost-regex1.62-dev:amd64	install
libboost-regex1.62.0:amd64	install
libboost-serialization1.62-dev:amd64	install
libboost-serialization1.62.0:amd64	install
libboost-system1.62-dev:amd64	install
libboost-system1.62.0:amd64	install
libboost-thread-dev:amd64	install
libboost-thread1.62-dev:amd64	install
libboost-thread1.62.0:amd64	install
libboost1.62-dev:amd64	install

szóval ha azt mondod hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev  
libboost-chrono-dev libboost-date-time-dev libboost-dev  
libboost-filesystem-dev libboost-iostreams-dev libboost-program-options-dev  
libboost-python-dev libboost-regex-dev libboost-serialization-dev  
libboost-system-dev libboost-thread-dev
```

(igen, ez így egy jó hosszú parancssor :)), akkor mindent is feltesz ami  
kell a  
libboosthoz (és a sima lib-eket is felteszi ami kell a \*-dev csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De tényleg ez  
kicsit  
macerasabb, szóval jobban kellene tudnom hogy hol akad el, és mi az ami



hiányzik. Hányas verzióju milyen linuxot is tettél fel pontosan?

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a küldött dolgokat olvastam. Sajnos sem az openocd-t, sem az icedstorm toolchain-t

nem sikerült "éltre keltenem". Elvileg a yosys-0.9, nextpnr-ice40 és icedstorm-utils csomagokat is telepítettem a megadott módon. Tud esetleg küldeni valamilyen segítséget, hogyan tudnám programozni ezeket a lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék segítséget kérni,

telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis Egyetem TDK

mellett készülhetünk az országos TDK-ra. Mivel már kész van az eszköz, és csak bemutatni kell, így nem vesz el jelentős időt.

Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy két szabadsági fokból álló állványon levő parabola-antenna fogja "bescannelni"

az eget, és a vett jelerősséget fogja egy skála szerint osztályozni, majd ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni. Fő célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell érzékelnünk.

Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő érzékelővel. Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb lenne) csinálni?

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo, 16:02):

Szia Gábor!

Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper lesz, azon nagyjából mindent is meg lehet csinálni amit debian-on, és hogyha standalone es/vagy desktop üzemben használsz akkor tényleg nem lehet gond! Headless üzemben már nem használnék egyáltalán ubuntu-t, de az más kérdés ;)

Linux alá viszont van teljesen free & open source Lattice FPGA toolchain: <http://www.clifford.at/icedstorm/>, ezt érdemes lehet feltelepíteni. Mi

is

hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icedstorm-utils), ezekkel a csomagokkal közvetlenül is tudod linux alatt mindenféle licenc-problema megnyűg nélkül használni a barmit! Cserebe a forrás(kod)bol való telepítés nem annyira egyszerű, elég sok a függőség, és nehezen derül ki hogy mi az.

De

ebben

viszont a windozos cuccokkal ellentétben tudok segíteni :) Ugyanakkor természetesen én is néztem a linuxos icecube2-t, és az is működik persze, szóval hajra azzal is. Mondjuk az nekem kicsit nehezkés, a "bare metal" meg a klasszikus "unix filozofia" dolgok után nekem ezeket az IDE-eket elég

nehéz

megszoknom :/ De az más kérdés ;)

A.

On Sat, 7 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos tanulmányozásának. Egy közeli ismerősöm ajánlásából elkezdjük föltelepíteni az ubuntut,

mivel

a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem volt zökkenőmentes a telepítés. Először a pendrive amin keresztül telepíteni akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt föl.

Majd

miután sikerült, a Grub nem települt így nem tudtam elindítani a rendszert.

Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és egyebek.

Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált adataimat

el

kell fogadják, de az is hamarosan meglesz.

A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött pdf-eket

át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.

Szerencsére

az is teljesen témába vág, hiszen C programozás lesz.

Nagyon le voltam nyűgözve a találkozókör és remélem minden rendben lesz!

Köszönettel:

K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6., P, 9:44):

Szia Gábor!

Persze, kuldom akkor az infokat:

- mellekelve az evaluation board doksija, ebben benne kell legyen minden

referencia ami hasznos lehet - bár ugye a mi boardunk az más módon.

- a fejlesztőeszköz az ez: <https://www.latticesemi.com/iCEcube2>,

van

minden

operációs rendszer alá (linux, windoz).

- az FPGA az ez: iCE40HX8K-BG121, ezt kell kiválasztanod majd.

- mellekelem a tegnapi tesztelt kis verilog-kódot is, abban

megtalálod a

PCF-et: ezt a PCF-et megeszi az icecube2 is! (a verilog-ot meg nyilván)

Illetve az ARM-es dolgokkal kapcsolatban is akkor pár info:

- mellekelem a tegnapi példaprogramot (UART: kisbetű -> nagybetű)

- a toolchain (ami a \*.c fileokból megcsinálja a \*.bin és/vagy \*.hex

image-t,

amit aztán fel kell tölteni) az `arm-none-eabi` néven fut. Ennek több

komponense van:

\* A C/C++/asm fordító maga: ez a gcc-arm-none-eabi

\* Eszközök amikkel a lefordított fileokat lehet manipualni

(pl

a

\*.hex-et eloallítani a \*.o/\*.elf-ből):

binutils-arm-none-eabi

\* Es az elore lefordított konyvtaraok, hogy a jól megszokott

C-s

(stdio: printf, stb) meg C++-os (STL es tarsai) dolgokat

tudd

hasznalni: libnewlib-arm-none-eabi,  
libstdc++-arm-none-eabi-newlib.

Ez kb ugyanilyen néven fut windoz alatt is, egy gyors keresessel pl

ezt

talaltam:

<https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>

rá!

A lenyeg hogy ez a "gcc-arm-none-eabi for windows" dolologra keress

A programozo az ST-Link V2 (-1), ezt lehet OpenOCD-vel progrmaozni. Az  
OpenOCD

van linux ala is, de van windozos prortja is:

<https://gnutoolchains.com/arm-eabi/openocd/>. Ettol fuggetlenul az ST  
sajat

cuccait is lehet hasznalni valoszinuleg:

<https://www.st.com/en/development-tools/stsw-link009.html>, de az

OpenOCD

annyira jo hogy windozt hasznalo kollegaktol is azt hallom hogy inkabb

az

OpenOCD-t preferaljak az STM32-es MCU-khoz.

A.

On Fri, 6 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Említett nekem segédleteket a lattice boardhoz. Továbbá említette,

hogy

az

fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is

szeretnék

kérni segítséget.

Köszönettel: K. Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont:

2020.

nov.

4., Sze, 17:28):

Rendben!

Szerintem én is megebédelek és indulok

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze  
17:09):

Szia Gábor!

Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, es akkor

utana

indulok  
fel. Szoval fel 2 korultól mar leszek itt. Ha errefele jarsz akkor  
hivj  
fel:  
(20)3962151, es akkor kimegyek a kapuhoz. Merthogy mi itten egy  
masik  
epulet  
masik szekciojaban vagyunk.  
udv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper!  
Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 16:45):

Szia Gábor!

No, tul vagy kozben a mindenben? Illetve a het hetralevo resze  
akkor  
mikor  
lenne  
jo? En holnap delutan tervezek feljonni a normafára elso  
kozelitesben.  
A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás céljából?

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020. okt. 29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.  
TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz minden, a  
hetedik  
hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét  
elején.  
Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha  
minden  
igaz,  
az is meglesz.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 28., Sze, 18:44):

Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo  
heten,  
mert

közben engem is nagyon bekapott az ekszija és/vagy a daráló a műhold  
 kapcsolom.  
 Most már szerda van, és még mindig nem tudom hogy a holnap az hogy lesz.  
 És attól felek hogy csak delelőtt derül ki hogy mármint delután fel kell mennem vagy csak pénteken vagy hogy.  
 Neked mikor vannak ezek a határidők és/vagy mikor záródnak le a dolgok?  
 (melyik is pl az a hetedik hét, meg mi a TDK határideje, ilyesmik)?  
 A.  
 On Mon, 26 Oct 2020, Gábor Kreinicker wrote:  
 | Beolvastam, de a hetedik heti hajrá és a TDK határidő közelsége nem engedett komolyan elmélyedni még benne. A téma nagyon bejött, a jövőhétre már felkészültebb leszek.  
 Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 23:07):  
 | A doksikát/referenciákat mennyire tudtad megnevezni amiket küldtem? Kiindulásnak ezek mennyire jók? Persze kicsit még levegőben lógna a tematika,  
 de ettől függetlenül(?)  
 A.  
 On Mon, 26 Oct 2020, Gábor Kreinicker wrote:  
 | Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm szépen!  
 Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 26., H 10:38):  
 Igen, a "Normafa" az maga az intézet, Konkoly-Thege Miksa 15-17. Egyesek "Svábhegy"-kent is szeretnek rá hivatkozni, az is geográfiaiilag kb ugyanarra akkuratus :)

csütörtökön, Csak azért kerdelem mert ennekem ha nem is 100% a Normafa  
odalent a  
városban (~Moricz korter videke) nekem is hatékonyabb az  
eljutás.

A,  
On Mon, 26 Oct 2020, Gábor Kreinicker wrote:  
Ha a Móricz az a körtér, akkor az nagyon közel van.

Viszont nekem be kell mennem minden képpen a titkárságra aláírni a szerződést,  
így ha az is Normafán van, akkor szívesen kimegyek oda is.  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.  
okt. 26., H 10:24):  
Oke, eszbentartom! Szerda delelőtig akkor diskuráljuk  
le. Az is lehet hogy valahol a city-ben összefutunk, hamár ugyis hoómmooffice  
van es a normafa mindentől is messze van. Neked a Moszkva-Déli-BAH-Moricz  
tengely mennyire esik útba?  
A.  
On Mon, 26 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt Pál András!  
Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó  
lehet.  
Üdv: Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.  
okt. 26., H 10:13):  
Szia Gábor!  
Volt/van egy kis vis maior-unk most a műhold-elektronikákkal,  
így a köv

napok  
kicsit erraticusak nekem. De persze a dolgok  
parhuzamosítása  
mindig  
jó  
dolog -  
akkor így kérdés: ha nem késő es/vagy meg tudnad oldani,  
mar  
kora-kozep  
delutan  
(2 korulre) mennyire tudnal feljönni a normafára?  
Koszi, A.  
On Mon, 19 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt Pál András!  
Először is bocsásson meg, de egyre gyanúsabb, hogy  
nem  
tudom  
melyik a  
vezeték és melyik a keresztnéve, amely minden  
megszólításnál  
apró  
problémát  
okoz.  
A nulladik kérdésre, sajnos a csütörtök nem jó,  
jövőhéten  
hétfő  
délelőtt,  
kedd délután, vagy csütörtök lenne a legszuperebb. A  
személyes  
tárgyalásnak  
/ beszélgetésnek én is nagyon örülnék.  
Amiket leírt, azok nagyon felcsigáztak, alig várom,  
hogyan  
belekezdjünk.  
Azonban, ha nem probléma, szeretném megérdeklődni,  
hogyan a  
fényűtváltó rendszer tervezésével mi lesz a  
továbbiakban?  
Úgy  
vettem  
észre,  
hogyan jelentősen kezdünk eltérni a SPICA témától. Ne  
értsen  
félre,  
természetesen nagyon érdekel és nagyon szívesen

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. november 19. 21:09

Kedves András Pál!

Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül megoldanom, mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a boardra. Amit leírt a korábbiakban, mindent többször átfutottam és ellenőriztem, de kifog rajtam egyelőre a Linux.

El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a "st-info --probe" sort a terminálba. A headereket sikeresen bemásoltam a megfelelő helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a következő sorokat kapom a "make upload" esetén:

```
arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o stm32_rcc.o /usr/lib/arm-none-eabi/
```

7/7/21, 18:08



mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a boardra. Amit leírt a korábbiakban, mindent többször átfutottam és ellenőriztem, de kifog rajtam egyelőre a Linux.  
 El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a "st-info --probe" sort a terminálba. A headereket sikeresen bemásoltam a megfelelő helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a következő sorokat kapom a "make upload" esetén:

```
arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or directory
arm-none-eabi-ld: cannot find
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or directory
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or directory
make: *** [Makefile:54: main.elf] Error 1
```

Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt, ezért elnézést!

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 17., K, 20:36):

Al nem! Az openocd az a binaris image-t (main.hex) tölti fel a board-ra, nem a main.c-t eteted meg kozvetlenul. Probald ki a `make upload` parancsot, az az ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a Makefile bogaraszasabol rajossz hogy hogy is van ez pontosan.

Szoval a workflow az itten a kovetkezo:

- A \*.c fileokbol \*.o (object) fileokat forditunk (arm-none-eabi-gcc)
- Az egeszet osszelinkeljük (arm-none-eabi-ld)
- Ez a tipusu mikrokontroller a 0x000000004-es címen található

utasításra

ugrik először, ott kezdi a program futtatását

- Ez a tipusu mikrokontroller olyan hogy a 0x000000000 tartomanyra a 0x08000000 tartományt kepezi le alapertelmezesben
- A linkelésnél így garantálnunk kell hogy legyen "valami" ami a 0x080000004-es címen ami értelmes \_és\_ kesobb meghivja a main() fuggvenyt
- 

merthogy minden C program a main()-nal "kezdodik". Na, ezt a feladatot latja el

a crt.c - plusz még pár egyebet is, de azzal nem igazan kell foglalatzkodni, azzal majd kesobb ;)

- A linkeles utan ki kell szednunk a keletkezo \*.elf filebol a "lenyeget". Ezt

csinalja a arm-none-eabi-objcopy, es ez hozza létre a \*.hex-et.

- Ezt a \*.hex-et pedig az openocd-vel beleirhatjuk a mikrokontroller flash-ébe

Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo sorrendben.

Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es az openocd

konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani neki hogy

milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target mikrokontroller (azaz a programozonak tudnia kell hogy a programozasi protokollon keresztul hogyan eri el a mikrokontroller memoriajat es ez fugghet

a mikrokontroller konkret tipusától - bar egy csaladon belül ez elegge

egyseg, ezért elég csak azt mondani hogy "stm32f0x", és nem azt hogy stm32f072c6t6). Plusz meg vannak egyéb finomságok is, pl a flash-t ugye minden iras előtt törölni kell és csak utána írhatod bele, stb. Ezeket mind-mind megcsinálja a Makefile-be elrejtett parancs- valami.

Egyébként ez a fenti sema nagyjából minden beagyazott cuccra ugyanígy megy, csak itt ez a "bare metal" módszer ezt is szükségessé teszi hogy lépésről lépésre menj(unk) végig a lépéseken.

Az arduino-s projektnél is vsz ugyanígy volt, csak gondolom ott használtak valami fejlesztőkörnyezetet ami mindent is megcsinált, nem? Vagy hogy?

A.

On Tue, 17 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent telepítettem. Ami még nem sikerült, az az openocd életre bírása. Csatolok egy képet, és bízom benne, hogy tud segíteni.

Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az st-link csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a feltöltésig nem jutottam el.

Köszönettel: K. Gábor

[image: Screenshot from 2020-11-17 19-53-48.jpg]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 16., H, 18:35):

Szia Gábor!

Ah, értem, ez így akkor elsore tényleg nem egyszerű. Viszont ennél specifikusabban kéne tudni hogy hol is akadtál el. Tehát:

- telepítés ment? Minden fent van?
- az arm-es toolchain az ezen csomagokból all:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi
  - \* libnewlib-arm-none-eabi,
  - \* libstdc++-arm-none-eabi-newlib.
- ami még hasznos csomag és kellhet:
  - \* build-essential (ebben van helyi C fordító is + a `make`

program)

- \* srecord
- openocd az csak simán ez:
  - \* openocd

Mindegyiket simán `apt-get install ...` módon tudod telepíteni! Ezek sikerültek? Ha ezek mind fent vannak, akkor az általam mellekelt forrást már le

tudod fordítani és fel kell tudnod tölteni a kis board-ra. Ami meg kellhet

a forgatáshoz azok az STM32-specifikus headerek. Ezeket elküldöm mellekelve.

En konkrétan sima rendszergazdaként az /usr/local/arm-none-eabi/include alkönyvtár alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt

teszprogramban  
levő Makefile-ban az elejen (kb 12-15ik sorban) levő

ARM=/usr/local/arm-none-eabi

sort modosítsd oda ahova betetted! Lehet simán nálad is, a /home-od  
alatt!

De  
ugyis panaszkodik a fordító hogyha nem találja ;) Szóval legjobb  
emlekeim

szerint ennyi kell, és ami meg hiányzik az ezeknek a csomagoknak a  
függősége -  
szóval jó kell hogy legyen, legjobb emlekeim szerint.

Az FPGA az sajnos tényleg egy sokkal bonyolultabb. Elobb nezzük meg ezt  
az  
de ARM + STM32 témát, és akkor utána. Valójában azokat sem nehéz feltenni,

kellhet pár olyan development header (pl libboost-csomag) amiket  
feltetelezz  
hogy fent van de alából szinte biztos hogy nincs fent nálad. Nálam  
konkrétan  
ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-filesystem-dev:amd64	install
libboost-filesystem1.62-dev:amd64	install
libboost-filesystem1.62.0:amd64	install
libboost-iostreams-dev:amd64	install
libboost-iostreams1.62-dev:amd64	install
libboost-iostreams1.62.0:amd64	install
libboost-program-options-dev:amd64	install
libboost-program-options1.62-dev:amd64	install
libboost-program-options1.62.0:amd64	install
libboost-python-dev	install
libboost-python1.62-dev	install
libboost-python1.62.0	install
libboost-regex1.62-dev:amd64	install
libboost-regex1.62.0:amd64	install
libboost-serialization1.62-dev:amd64	install
libboost-serialization1.62.0:amd64	install
libboost-system1.62-dev:amd64	install
libboost-system1.62.0:amd64	install
libboost-thread-dev:amd64	install
libboost-thread1.62-dev:amd64	install
libboost-thread1.62.0:amd64	install
libboost1.62-dev:amd64	install

szóval ha azt mondod hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev  
libboost-chrono-dev libboost-date-time-dev libboost-dev  
libboost-filesystem-dev libboost-iostreams-dev  
libboost-program-options-dev  
libboost-python-dev libboost-regex-dev libboost-serialization-dev  
libboost-system-dev libboost-thread-dev
```

(igen, ez így egy jó hosszú parancssor :)), akkor mindent is feltesz ami  
kell a  
libboosthoz (és a sima lib-eket is felteszi ami kell a \*-dev

csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De tenyleg ez kicsit macerásabb, szóval jobban kellene tudnom hogy hol akad el, és mi az ami hiányzik. Hányas verzióju milyen linuxot is tettél fel pontosan?

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a küldött

dolgokat olvasgattam. Sajnos sem az openocd-t, sem az icestorm toolchain-t nem sikerült "életre keltenem". Elvileg a yosys-0.9, nextpnr-ice40 és icestorm-utils csomagokat is telepítettem a megadott módon. Tud esetleg küldeni valamilyen segítséget, hogyan tudnám programozni ezekentúl a lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék segítséget kérni, telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis Egyetem TDK

és mellett készülhetünk az országos TDK-ra. Mivel már kész van az eszköz, csak bemutatni kell, így nem vesz el jelentős időt.

két Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy szabadsági fokból álló állványon levő parabola-antenna fogja

"bescannelni" az eget, és a vett jelerősséget fogja egy skála szerint osztályozni, majd

ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni. Fő célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell érzékelnünk.

érzékelővel. Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő

Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb lenne) csinálni?

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo, 16:02):

Szia Gábor!

lesz, Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper

azon nagyjabol mindent is meg lehet csinálni amit debian-on, és hogyha standalone es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!

Headless

üzemben már nem használnék egyáltalán Ubuntu-t, de az más kérdés ;)

Linux alá viszont van teljesen free & open source Lattice FPGA

toolchain:

<http://www.clifford.at/icestorm/>, ezt érdemes lehet feltételezni. Mi

is

hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icestorm-utils), ezekkel a csomagokkal közvetlenül is tudod Linux alatt mindenféle

licenc-problema

még

nyűg nélkül használni a barmit! Cserebe a forrás(kod)bol való

telepítés

nem

annyira egyszerű, elég sok a függőség, és nehezen derül ki hogy mi az.

De

ebben

viszont a Windowsos cuccokkal ellentétben tudok segíteni :) Ugyanakkor természetesen én is néztem a Linuxos Icecube2-t, és az is működik

persze,

szóval

hajrá azzal is. Mondjuk az nekem kicsit nehezebb, a "bare metal" meg a klasszikus "Unix filozofia" dolgok után nekem ezeket az IDE-ket elég

nehéz

megszoknom :/ De az más kérdés ;)

A.

On Sat, 7 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos

tanulmányozásának.

Egy közeli ismerősem ajánlásából elkezdtek föltelepíteni az Ubuntu-t,

mivel

a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem volt zökkenőmentes a telepítés. Először a pendrive amin keresztül

telepíteni

akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt föl.

Majd

miután sikerült, a Grub nem települt így nem tudtam elindítani a rendszert.

Ezt ma sikerült megoldani, úgyhogy kezdődhet a Linux tanulás és egyébek.

Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált adataimat

el

kell fogadják, de az is hamarosan meglesz.

A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött pdf-eket

át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.

Szerencsére

az is teljesen témába vág, hiszen C programozás lesz.

Nagyon le voltam nyűgözve a találkozókör és remélem minden rendben

lesz!

Köszönettel:

K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6., P, 9:44):

Szia Gábor!

Persze, küldöm akkor az infokat:

- mellekelve az evaluation board doksija, ebben benne kell legyen

minden  
referencia ami hasznos lehet - bar ugye a mi boardunk az masmilyen.  
- a fejlesztokeszlet az ez: <https://www.latticesemi.com/iCEcube2>,  
van  
minden  
oprendszer ala (linux, windoz).  
- az FPGA az ez: iCE40HX8K-BG121, ezt kell kivlasztanod majd.  
- mellekelem a tegnap tesztelt kis verilog-kodot is, abban  
megtalalad a  
PCF-et: ezt a PCF-et megeszti az icecube2 is! (a verilog-ot meg  
nyilvan)  
Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:  
- mellekelem a tegnapi peldaprogramot (UART: kisbetu -> nagybetu)  
- a toolchain (ami a \*.c fileokbol megcsinalja a \*.bin es/vagy  
\*.hex  
image-t,  
amit aztan fel kell tolteni) az `arm-none-eabi` néven fut. Ennek  
tobb  
komponense van:  
\* A C/C++/asm fordito maga: ez a gcc-arm-none-eabi  
\* Eszkozok amikkel a leforditott fileokat lehet manipulalni  
(pl  
a  
\*.hex-et eloallitani a \*.o/\*.elf-ből):  
binutils-arm-none-eabi  
\* Es az elore leforditott konyvtaraok, hogy a jól megszokott  
C-s  
(stdio: printf, stb) meg C++-os (STL es tarsai) dolgokat  
tudd  
hasznalni: libnewlib-arm-none-eabi,  
libstdc++-arm-none-eabi-newlib.  
Ez kb ugyanilyen néven fut windoz alatt is, egy gyors keresessel pl  
ezt  
talaltam:  
  
<https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>  
.  
rá!  
A lenyeg hogy ez a "gcc-arm-none-eabi for windows" dolologra keress  
Az  
A programozo az ST-Link V2 (-1), ezt lehet OpenOCD-vel progrmaozni.  
OpenOCD  
van linux ala is, de van windozos prortja is:  
<https://gnutoolchains.com/arm-eabi/openocd/>. Ettol fuggetlenul az  
ST  
sajat  
cuccait is lehet hasznalni valoszinuleg:  
<https://www.st.com/en/development-tools/stsw-link009.html>, de az  
OpenOCD  
annyira jo hogy windozt hasznalo kollegaktol is azt hallom hogy  
inkabb  
az  
OpenOCD-t preferaljak az STM32-es MCU-khoz.  
A.  
On Fri, 6 Nov 2020, Gábor Kreinicker wrote:  
Kedves Pál András!

Említett nekem segédleteket a lattice boardhoz. Továbbá említette,  
 hogy az fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is  
 szeretnék kérni segítséget.

Köszönettel: K. Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020.  
 nov. 4., Sze, 17:28):

Rendben!

Szerintem én is megebédelek és indulok

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze  
 17:09):

Szia Gábor!

Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, es akkor

utana

indulok

fel. Szoval fel 2 korultól mar leszek itt. Ha errefele jatsz

akkor

hivj

fel:

(20)3962151, es akkor kimegyek a kapuhoz. Merthogy mi itten egy

masik

epulet

masik szekciojaban vagyunk.

udv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper!

Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4.,

Sze

16:45):

Szia Gábor!

No, tul vagy kozben a mindenben? Illetve a het hetralevo resze

akkor

mikor

lenne

jo? En holnap delutan tervezek feljonni a normafára elso

kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás  
 céljából?

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta

(időpont:

2020.

okt.

29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.

TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz

minden, a

hetedik

hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét

elején.

Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha

minden

igaz,

az is meglesz.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

28.,

Sze,

18:44):

Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo

heten,

mert

kozben

engem is nagyon bekapott az eksziz es/vagy a daralo a műhold

kapcsan.

Most mar

szerda van, es meg mindig nem tudom hogy a holnap az hogy

lesz.

Es

attol

felek

hogy csak delelott derul ki hogy mar delutan fel kell mennem

vagy

csak

penteken

vagy hogy.

Neked mikor vannak ezek a hataridok es/vagy mikor zarodnak

le

a

dolgok

(melyik

is pl az a hetedik het, meg mi a TDK hatarideje, ilyesmik)?

A.

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Beleolvastam, de a hetedik heti hajrá és a TDK határidő

közelsége

nem

engedett komolyan elmélyedni még benne. A téma nagyon

bejött, a

jövőhétre

már felkészültebb leszek

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

26., H



23:07):

kuldtem?

A doksikat/referenciákat mennyire tudtad megnezni amiket

Kiindulasnak  
ezek mennyire jok? Persze kicsit még levegoben lognak a

temak,

de

ettol

fuggetlenul(?)

A

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

szépen!

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.

okt.

26., H

10:38):

ut

Igen, a "Normafa" az maga az intezet, Konkoly-Thege M.

15-17.

Egyesek

"Svábhegy"-kent is szeretnek ra hivatkozni, az is

geografailag

kb

ugyannyira  
akkuratus :)

Csak azert kerdem mert ennekem ha nem is 100% a Normafa

csutortokon,

odalent a  
varosban (~Moricz korter videke) nekemis hatekonyabb az

eljutas.

A,

On Mon, 26 Oct 2020, Gábor Kreinicker wrote:

Ha a Móricz az a körtér, akkor az nagyon közel van.

Viszont

nekem  
be

kell

mennem minden képpen a titkárságra aláírni a

szerződést,

így

ha az

is

Normafán van, akkor szívesen kimegyek oda is.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.

okt.

26., H

10:24):

le.

Az

is

lehet

hogy

valahol a city-ben összefutunk, hamar ugyis hoómooffice

van  
es  
a  
normafa  
mindentol is messze van. Neked a  
Moszkva-Déli-BAH-Moricz  
tengely  
mennyire  
esik  
utba?  
A.  
On Mon, 26 Oct 2020, Gábor Kreinicker wrote:  
Tisztelt Pál András!  
Ma sajnos eséjtelen. Csütörtökön viszont bármikor jó  
lehet.  
Üdv: Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.  
okt.  
26., H  
10:13):  
Szia Gábor!  
Volt/van egy kis vis maior-unk most a  
műhold-elektronikákkal,  
igy a  
köv  
napok  
kicsit erraticusak nekem. De persze a dolgok  
parhuzamositasa  
mindig

**Gábor Kreinicker** <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)>  
Címzett: Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)>

2020. november 19. 23:17

Sajnos nem ez a gond, már telepítve van.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 19., Cs 21:33):  
Tedd fel a libnewlib-arm-none-eabi nevű csomagot!

```
apt-get install libnewlib-arm-none-eabi
```

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

> Kedves András Pál!

>

> Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül megoldanom,

> mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a

> boardra. Amit leírt a korábbiakban, mindent többször átfutottam és

> ellenőriztem, de kifog rajtam egyelőre a Linux.

> El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a "st-info

> --probe" sort a terminálba. A headereket sikeresen bemásoltam a megfelelő

> helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a következő

> sorokat kapom a "make upload" esetén:

>

> arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o

> stm32\_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc\_nano.a

> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a

```
> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
> arm-none-eabi-ld: cannot find
> /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or directory
> arm-none-eabi-ld: cannot find
> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or directory
> arm-none-eabi-ld: cannot find
> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or directory
> make: *** [Makefile:54: main.elf] Error 1
>
> Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt, ezért
> elnézést!
>
> Köszönettel: K. Gábor
>
> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 17., K, 20:36):
>
>> A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a board-ra,
>> nem a
>> main.c-t eteted meg kozvetlenul. Probald ki a `make upload` parancsot, az
>> az
>> ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a Makefile
>> bogaraszasabol rajossz hogy hogy is van ez pontosan.
>>
>> Szoval a workflow az itten a kovetkezo:
>> - A *.c fileokbol *.o (object) fileokat forditunk (arm-none-eabi-gcc)
>> - Az egeszet osszelinkeljuk (arm-none-eabi-ld)
>> - Ez a tipusu mikrokontroller a 0x000000004-es címen található
>> utasításra
>> ugrik először, ott kezdi a program futtatását
>> - Ez a tipusu mikrokontroller olyan hogy a 0x000000000 tartomanyra a
>> 0x08000000 tartományt kepezi le alapertelmezesben
>> - A linkelésnél így garantálnunk kell hogy legyen "valami" ami a
>> 0x080000004-es címen ami értelmes _és_ kesobb meghivja a main() fuggvenyt
>> -
>> merthogy minden C program a main()-nal "kezdodik". Na, ezt a feladatot
>> latja el
>> a crt.c - plusz még pár egyebet is, de azzal nem igazan kell
>> foglalatoskodni,
>> azzal majd kesobb ;)
>> - A linkeles utan ki kell szednunk a keletkezo *.elf filebol a
>> "lenyeget". Ezt
>> csinalja a arm-none-eabi-objcopy, es ez hozza létre a *.hex-et.
>> - Ezt a *.hex-et pedig az openocd-vel beleirhatjuk a mikrokontroller
>> flash-ébe
>>
>> Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo sorrendben.
>> Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es az
>> openocd
>> konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani neki
>> hogy
>> milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target
>> mikrokontroller (azaz a programozonak tudnia kell hogy a programozasi
>> protokollon keresztul hogyan eri el a mikrokontroller memoriajat es ez
>> fugghet
>> a mikrokontroller konkret tipusától - bar egy csaladon belül ez elegge
>> egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem azt hogy
>> stm32f072cibt6). Plusz meg vannak egyeb finomsagok is, pl a flash-t ugye
>> minden
>> iras előtt torolni kell es csak utana irhatsz bele, stbstb. Ezeket
>> mind-mind
>> megcsinalja a Makefile-be elrejtett parancs- valami.
>>
>> Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra ugyanigy
>> megy,
>> csak itt ez a "bare metal" modszer ezt is szuksegesse teszi hogy lepesrol
>> lepesre menj(unk) vegig a lepeseken.
```

```

>>
>> Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott
>> hasznaltatok
>> valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy hogy?
>>
>> A.
>>
>> On Tue, 17 Nov 2020, Gábor Kreinicker wrote:
>>
>>> Kedves Pál András!
>>>
>>> Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent
>>> telepítettem. Ami még nem sikerült, az az openocd életre bírása. Csatolok
>>> egy képet, és bízom benne, hogy tud segíteni.
>>> Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az st-link
>>> csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a
>> feltöltésig
>>> nem jutottam el.
>>>
>>> Köszönettel: K. Gábor
>>> [image: Screenshot from 2020-11-17 19-53-48.jpg]
>>>
>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 16., H,
>> 18:35):
>>>
>>>> Szia Gábor!
>>>>
>>>> Ah, ertem, ez így akkor elsore tenyleg nem egyszerű. Viszont ennel
>>>> specifikusabban kene tudni hogy hol is akadtál el. Tehat:
>>>> - telepites ment? Minden fent van?
>>>> - az arm-es toolchain az ezen csomagokbol all:
>>>>   * gcc-arm-none-eabi
>>>>   * binutils-arm-none-eabi
>>>>   * libnewlib-arm-none-eabi,
>>>>   * libstdc++-arm-none-eabi-newlib.
>>>> - ami meg hasznos csomag es kellhet:
>>>>   * build-essential (ebben van helyi C fordito is + a `make`
>> program)
>>>>   * srecord
>>>> - openocd az csak siman ez:
>>>>   * openocd
>>>>
>>>> Mindegyiket siman `apt-get install ...` modon tudod telepiteni! Ezek
>>>> sikerultek? Ha ezek mind fent vannak, akkor az altalam mellekelt forrast
>>>> mar le
>>>> tudod fordítani es fel kell tudnod tölteni a kis board-ra. Ami meg
>> kellhet
>>>> a
>>>> forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom
>> mellekelve.
>>>> En
>>>> konkretan sima rendszergazdakent az /usr/local/arm-none-eabi/include
>>>> alkovnytar
>>>> alatt tárolok ezeket, de barhova felteheted! Akkor a mellekelt
>>>> teszprogramban
>>>> levő Makefile-ban az elejen (kb 12-15ik sorban) levő
>>>>
>>>> ARM=/usr/local/arm-none-eabi
>>>>
>>>> sort modosítsd oda ahova betetted! Lehet siman nálad is, a /home-od
>> alatt!
>>>> De
>>>> ugyis panaszkodik a fordito hogyha nem találja ;) Szoval legjobb
>> emlekeim
>>>> szerint ennyi kell, es ami meg hiányzik az ezeknek a csomagoknak a

```

```
>>>> fuggosege -
>>>> szoval jo kell hogy legyen, legjobb emlekeim szerint.
>>>>
>>>> Az FPGA az sajnos tenyleg egy fokkal bonyolultabb. Elobb nezzuk meg ezt
>> az
>>>> ARM + STM32 témát, es akkor utana. Valojaban azokat sem nehez feltenni,
>> de
>>>> lehet par olyan development header (pl libboost-csomag) amiket
>>>> feltetelez
>>>> hogy fent van de alapbol szinte biztos hogy nincs fent nálad. Nalam
>>>> konkretan
>>>> ezek vannak fent:
>>>>
>>>> libboost-atomic1.62-dev:amd64          install
>>>> libboost-atomic1.62.0:amd64            install
>>>> libboost-chrono1.62-dev:amd64          install
>>>> libboost-chrono1.62.0:amd64            install
>>>> libboost-date-time1.62-dev:amd64        install
>>>> libboost-date-time1.62.0:amd64          install
>>>> libboost-dev:amd64                      install
>>>> libboost-filesystem-dev:amd64           install
>>>> libboost-filesystem1.62-dev:amd64       install
>>>> libboost-filesystem1.62.0:amd64         install
>>>> libboost-iostreams-dev:amd64            install
>>>> libboost-iostreams1.62-dev:amd64        install
>>>> libboost-iostreams1.62.0:amd64          install
>>>> libboost-program-options-dev:amd64      install
>>>> libboost-program-options1.62-dev:amd64  install
>>>> libboost-program-options1.62.0:amd64    install
>>>> libboost-python-dev                     install
>>>> libboost-python1.62-dev                 install
>>>> libboost-python1.62.0                   install
>>>> libboost-regex1.62-dev:amd64            install
>>>> libboost-regex1.62.0:amd64              install
>>>> libboost-serialization1.62-dev:amd64    install
>>>> libboost-serialization1.62.0:amd64      install
>>>> libboost-system1.62-dev:amd64           install
>>>> libboost-system1.62.0:amd64             install
>>>> libboost-thread-dev:amd64               install
>>>> libboost-thread1.62-dev:amd64           install
>>>> libboost-thread1.62.0:amd64             install
>>>> libboost1.62-dev:amd64                  install
>>>>
>>>> szoval ha azt monddod hogy
>>>>
>>>> # apt-get --no-install-recommends install libboost-atomic-dev
>>>> libboost-chrono-dev libboost-date-time-dev libboost-dev
>>>> libboost-filesystem-dev libboost-iostreams-dev
>> libboost-program-options-dev
>>>> libboost-python-dev libboost-regex-dev libboost-serialization-dev
>>>> libboost-system-dev libboost-thread-dev
>>>>
>>>> (igen, ez igy egy jó hosszú parancssor :)), akkor mindent is feltesz ami
>>>> kell a
>>>> libboosthoz (es a sima lib-eket is felteszi ami kell a *-dev
>> csomagokhoz)
>>>>
>>>> Ami meg szinte biztos hogy kell az a: flex bison libtool. De tenyleg ez
>>>> kicsit
>>>> macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi az ami
>>>> hiányzik. Hanyas verzioju milyen linuxot is tettél fel pontosan?
>>>>
>>>> A.
>>>>
>>>> On Mon, 16 Nov 2020, Gábor Kreinicker wrote:
>>>>
```

>>>>> Kedves Pál András!  
>>>>>  
>>>>> Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a  
>> küldött  
>>>>> dolgokat olvasgattam. Sajnos sem az openocd-t, sem az icesform  
>>>> toolchain-t  
>>>>> nem sikerült "életre keltennem". Elvileg a yosys-0.9, nextpnr-ice40 és  
>>>>> icesform-utils csomagokat is telepítettem a megadott módon. Tud esetleg  
>>>>> küldeni valamilyen segítséget, hogyan tudnám programozni ezekentúl a  
>>>>> lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék segítséget  
>>>> kérni,  
>>>>> telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam  
>>>>> kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom  
>> elérni.  
>>>>>  
>>>>> Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a  
>>>>> programozásukig sajnos még nem jutottam el.  
>>>>>  
>>>>> Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai  
>>>>> eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis Egyetem  
>>>> TDK  
>>>>> mellett készülhetünk az országos TDK-ra. Mivel már kész van az eszköz,  
>> és  
>>>>> csak bemutatni kell, így nem vesz el jelentős időt.  
>>>>>  
>>>>> Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy  
>> két  
>>>>> szabadsági fokból álló állványon levő parabola-antenna fogja  
>>>> "bescannelni"  
>>>>> az eget, és a vett jelerősséget fogja egy skála szerint osztályozni,  
>> majd  
>>>>> ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni. Fő  
>>>>> célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell  
>>>> érzékelnünk.  
>>>>> Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő  
>> érzékelővel.  
>>>>> Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg  
>>>>> valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb  
>>>> lenne)  
>>>>> csinálni?  
>>>>>  
>>>>> Köszönettel: K. Gábor  
>>>>>  
>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 7., Szo,  
>>>> 16:02):  
>>>>>  
>>>>>> Szia Gábor!  
>>>>>>  
>>>>>> Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper  
>> lesz,  
>>>>>> azon  
>>>>>> nagyjából mindent is meg lehet csinálni amit debian-on, es hogyha  
>>>>>> standalone  
>>>>>> es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!  
>> Headless  
>>>>>> üzemben mar nem használnék egyatalan ubuntu-t, de az mas kerdes ;)  
>>>>>>  
>>>>>> Linux alá viszont van teljesen free & open souce Lattice FPGA  
>> toolchain:  
>>>>>> <http://www.clifford.at/icesform/>, ezt erdemes lehet feltelepiteni. Mi  
>>>> is  
>>>>>> hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icesform-utils), ezekkel a  
>>>>>> csomagokkal kozvetlenul is tudod linux alatt mindenfele  
>> licenc-problema  
>>>>>> meg  
>>>>>> nyűg nélkül használni a barmit! Cserebe a forras(kod)bol valo

```
>> telepites
>>>>> nem
>>>>> annyira egyszerű, elég sok a függőség, es nehezen derül ki hogy mi az.
>>>> De
>>>>> ebben
>>>>> viszont a windozos cuccokkal ellentétben tudok segíteni :) Ugyanakkor
>>>>> természetesen en is neztem a linuxos icecube2-t, es az is mukodik
>> persze,
>>>>> szoval
>>>>> hajra azzal is. Mondjuk az nekem kicsit nehezkes, a "bare metal" meg a
>>>>> klasszikus "unix filozofia" dolgok utan nekem ezeket az IDE-ket eleg
>>>> nehez
>>>>> megszoknom :/ De az mas kerdes ;)
>>>>>
>>>>> A.
>>>>>
>>>>> On Sat, 7 Nov 2020, Gábor Kreinicker wrote:
>>>>>
>>>>>> Kedves Pál András!
>>>>>>
>>>>>> Köszönök mindent, ma hozzá is állok a dolgok alapos
>> tanulmányozásának.
>>>>>> Egy közeli ismerősöm ajánlásából elkezdtuk föltelepíteni az ubuntut,
>>>>>> mivel
>>>>>> a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem volt
>>>>>> zökkenőmentes a telepítés. Először a pendrive amin keresztül
>> telepíteni
>>>>>> akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt föl.
>>>> Majd
>>>>>> miután sikerült, a Grub nem települt így nem tudtam elindítani a
>>>>>> rendszert.
>>>>>> Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és
>>>> egyebek.
>>>>>> Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált
>>>> adataimat
>>>>>> el
>>>>>> kell fogadják, de az is hamarosan meglesz.
>>>>>> A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött
>>>>>> pdf-eket
>>>>>> át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.
>>>>>> Szerencsére
>>>>>> az is teljesen témába vág, hiszen C programozás lesz.
>>>>>> Nagyon le voltam nyűgözve a találkozókör és remélem minden rendben
>>>> lesz!
>>>>>>
>>>>>> Köszönettel:
>>>>>> K. Gábor
>>>>>>
>>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 6., P,
>>>> 9:44):
>>>>>>
>>>>>>> Szia Gábor!
>>>>>>>
>>>>>>> Persze, kuldom akkor az infokat:
>>>>>>> - mellekelve az evaluation board doksija, ebben benne kell legyen
>>>>>>> minden
>>>>>>> referencia ami hasznos lehet - bar ugye a mi boardunk az masmilyen.
>>>>>>> - a fejlesztokeszlet az ez: https://www.latticesemi.com/iCEcube2,
>>>> van
>>>>>>> minden
>>>>>>> oprendszert ala (linux, windoz).
>>>>>>> - az FPGA az ez: iCE40HX8K-BG121, ezt kell kiválasztanod majd.
>>>>>>> - mellekelem a tegnap tesztelt kis verillog-kodot is, abban
>>>>>>> megtalalad a
>>>>>>> PCF-et: ezt a PCF-et megeszi az icecube2 is! (a verillog-ot meg
>>>> nyilván)
```

```
>>>>>>>
>>>>>>> Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:
>>>>>>> - mellekelem a tegnapi peldaprogramot (UART: kisbetu -> nagybetu)
>>>>>>> - a toolchain (ami a *.c fileokbol megcsinalja a *.bin es/vagy
>> *.hex
>>>>>>> image-t,
>>>>>>> amit aztan fel kell tolteni) az `arm-none-eabi` néven fut. Ennek
>> tobb
>>>>>>> komponense van:
>>>>>>> * A C/C++/asm fordito maga: ez a gcc-arm-none-eabi
>>>>>>> * Eszkozok amikkel a leforditott fileokat lehet manipulalni
>>>> (pl
>>>>>>> a
>>>>>>> *.hex-et eloallitani a *.o/*.elf-ből):
>>>> binutils-arm-none-eabi
>>>>>>> * Es az elore leforditott konyvtaraok, hogy a jól megszokott
>>>> C-s
>>>>>>> (stdio: printf, stb) meg C++-os (STL es tarsai) dolgokat
>>>> tudd
>>>>>>> hasznalni: libnewlib-arm-none-eabi,
>>>>>>> libstdc++-arm-none-eabi-newlib.
>>>>>>>
>>>>>>> Ez kb ugyanilyen néven fut windoz alatt is, egy gyors keresessel pl
>>>> ezt
>>>>>>> talaltam:
>>>>>>>
>>>>>>>
>>>>>>>
>> https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi
>>>>>>> .
>>>>>>>
>>>>>>> A lenyeg hogy ez a "gcc-arm-none-eabi for windows" dologra keress
>>>> rá!
>>>>>>>
>>>>>>> A programozo az ST-Link V2 (-1), ezt lehet OpenOCD-vel progrmaozni.
>> Az
>>>>>>> OpenOCD
>>>>>>> van linux ala is, de van windozos prortja is:
>>>>>>> https://gnutoolchains.com/arm-eabi/openocd/. Ettol fuggetlenul az
>> ST
>>>>>>> saját
>>>>>>> cuccait is lehet hasznalni valoszinuleg:
>>>>>>> https://www.st.com/en/development-tools/stsw-link009.html, de az
>>>>>>> OpenOCD
>>>>>>> annyira jo hogy windozt hasznalo kollegaktol is azt hallom hogy
>> inkabb
>>>>>>> az
>>>>>>> OpenOCD-t preferaljak az STM32-es MCU-khoz.
>>>>>>>
>>>>>>> A.
>>>>>>>
>>>>>>> On Fri, 6 Nov 2020, Gábor Kreinicker wrote:
>>>>>>>
>>>>>>>> Kedves Pál András!
>>>>>>>>
>>>>>>>> Említett nekem segédleteket a lattice boardhoz. Továbbá említette,
>>>> hogy
>>>>>>>> az
>>>>>>>> fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is
>>>> szeretnék
>>>>>>>> kérni segítséget.
>>>>>>>>
>>>>>>>> Köszönettel: K. Gábor
>>>>>>>>
>>>>>>>> Gábor Kreinicker <kreinickergabor@gmail.com> ezt írta (időpont:
```



>>>> 2020.  
>>>>>>> nov.  
>>>>>>> 4., Sze, 17:28):  
>>>>>>>  
>>>>>>> Rendben!  
>>>>>>>  
>>>>>>> Szerintem én is megebédelek és indulok  
>>>>>>>  
>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze  
>>>>>>> 17:09):  
>>>>>>>  
>>>>>>> Szia Gábor!  
>>>>>>> Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, es akkor  
>>>> utana  
>>>>>>> indulok  
>>>>>>> fel. Szoval fel 2 korultól mar leszek itt. Ha errefele jarsz  
>> akkor  
>>>>> hivj  
>>>>>>> fel:  
>>>>>>> (20)3962151, es akkor kimegyek a kapuhoz. Merthogy mi itten egy  
>>>> masik  
>>>>>>> epulet  
>>>>>>> masik szekciojaban vagyunk.  
>>>>>>> udv, A.  
>>>>>>>  
>>>>>>> On Wed, 4 Nov 2020, Gábor Kreinicker wrote:  
>>>>>>>  
>>>>>>> Tisztelt Pál András!  
>>>>>>>  
>>>>>>> A holnap délután részemről szuper!  
>>>>>>> Mikor tájtra számítsak?  
>>>>>>>  
>>>>>>> Üdv: Gábor  
>>>>>>>  
>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4.,  
>> Sze  
>>>>>>> 16:45):  
>>>>>>>  
>>>>>>> Szia Gábor!  
>>>>>>>  
>>>>>>> No, tul vagy kozben a mindenen? Illetve a het hetralevo resze  
>>>> akkor  
>>>>>>> mikor  
>>>>>>> lenne  
>>>>>>> jo? En holnap delutan tervezek feljonni a normafára elso  
>>>>>>> kozelitesben.  
>>>>>>>  
>>>>>>> A.  
>>>>>>>  
>>>>>>> On Fri, 30 Oct 2020, Gábor Kreinicker wrote:  
>>>>>>>  
>>>>>>> Tisztelt Pál András!  
>>>>>>>  
>>>>>>> Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás  
>>>>>>> céljából?  
>>>>>>>  
>>>>>>> Köszönettel: Gábor  
>>>>>>>  
>>>>>>> Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta  
>> (időpont:  
>>>>>>> 2020.  
>>>>>>>> okt.  
>>>>>>>> 29., Cs 9:01):  
>>>>>>>>  
>>>>>>>>> Rendben, az nekem is most előnyös lesz.  
>>>>>>>>>> TDK-t ma beadiuk, onnantól lényegesen egyszerűbb lesz

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

**Andras Pal** <apal@szofi.net>

2020. november 20. 0:22

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

### hű... probald meg hogy a prefixet atirod:

/usr/lib/arm-none-eabi/newlib/thumb/v6-m/

iii.

/usr/lib/gcc/arm-none-eabi/7.3.1/thumb/v6-m/

konyvtarakra!

igen, ugylatom hogy az ujabban debian-ban itten van, szoval nem kizart hogy az ubuntu-n is. lehet hogy nalad nem 7.3.1-es hanem kicsit mas a verzioja, de a logika valami hasonlo lehet. keresd meg hogy merre van "v6-m" alverziós libc nano.a illetve libnosys.a, valamit melyik /lib/gcc/ alatt van szinten "v6-m"-es libgcc.a!

n Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Sajnos nem ez a gond, már telepítve van.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 19., Cs 21:33):

Tedd fel a libnewlib-arm-none-eabi nevű csomagot!

```
apt-get install libnewlib-arm-none-eabi
```

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Kedves András Pál!

Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül megoldanom,  
mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a boardra. Amit leírt a korábbiakban, mindent többször átfutottam és ellenőriztem, de kifog rajtam egyelőre a Linux.  
El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a "st-info --probe" sort a terminálba. A headereket sikeresen bemásoltam a megfelelő helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a következő sorokat kapom a "make upload" esetén:

```
arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or
directory
make: *** [Makefile:54: main.elf] Error 1
```

Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt,  
ezért  
elnézést!

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 17., K,  
20:36):

A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a board-ra,  
nem a  
main.c-t eteted meg kozvetlenul. Probald ki a `make upload` parancsot,  
az

az  
ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a Makefile  
bogaraszasabol rajossz hogy hogy is van ez pontosan.

Szoval a workflow az itten a kovetkezo:

- A \*.c fileokbol \*.o (object) fileokat forditunk (arm-none-eabi-gcc)
- Az egeszet osszelinkeljuk (arm-none-eabi-ld)
- Ez a tipusu mikrokontroller a 0x000000004-es címen található

utasításra

ugrik először, ott kezdi a program futtatását

- Ez a tipusu mikrokontroller olyan hogy a 0x000000000 tartomanyra a 0x08000000 tartományt kepezi le alapertelmezesben
- A linkelésnél így garantálnunk kell hogy legyen "valami" ami a 0x080000004-es címen ami értelmes \_és\_ kesobb meghivja a main()

fuggvenyt

- merthogy minden C program a main()-nal "kezdodik". Na, ezt a feladatot latja el a crt.c - plusz még pár egyebet is, de azzal nem igazan kell foglalatoskodni, azzal majd kesobb ;)
- A linkeles utan ki kell szednunk a keletkezo \*.elf filebol a "lenyeget". Ezt csinálja a arm-none-eabi-objcopy, es ez hozza létre a \*.hex-et.

- Ezt a \*.hex-et pedig az openocd-vel beleírhatjuk a mikrokontroller flash-ébe

Ha megnezed, a Makefile az ebből mindent megcsinál, megfelelő sorrendben.

Kicsit korolmenyesnek/hosszunak tunik, de ez egy ilyen játék. Es az openocd konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani neki hogy milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target mikrokontroller (azaz a programozónak tudnia kell hogy a programozasi protokollon keresztül hogyan eri el a mikrokontroller memoriajat es ez fugghet a mikrokontroller konkret tipusától - bar egy családon belül ez elegge egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem azt hogy stm32f072cbt6). Plusz meg vannak egyeb finomsagok is, pl a flash-t ugye minden iras előtt torolni kell es csak utána írhat sz bele, stbstb. Ezeket mind-mind megcsinálja a Makefile-be elrejtett parancs- valami.

Egyebkent ez a fenti sema nagyjából minden beagyazott cuccra ugyanigy megy, csak itt ez a "bare metal" módszer ezt is szuksegesse teszi hogy

lepesrol

lepesre menj(unk) vegig a lepeseken.

Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott hasznaltatok valami fejlesztokornyezetet ami mindent is megcsinált, nem? Vagy hogy?

A.

On Tue, 17 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent telepítettem. Ami még nem sikerült, az az openocd életre bírása.

Csatolok

egy képet, és bízom benne, hogy tud segíteni. Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az

st-link

csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a feltöltésig nem jutottam el.

Köszönettel: K. Gábor

[image: Screenshot from 2020-11-17 19-53-48.jpg]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 16., H, 18:35):

Szia Gábor!

Ah, értem, ez így akkor elsore tenyleg nem egyszerű. Viszont ennél specifikusabban kene tudni hogy hol is akadtál el. Tehat:

- telepites ment? Minden fent van?
- az arm-es toolchain az ezen csomagokból áll:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi
  - \* libnewlib-arm-none-eabi,
  - \* libstdc++-arm-none-eabi-newlib.
- ami meg hasznos csomag es kellhet:
  - \* build-essential (ebben van helyi C fordító is + a `make`

```
program)
* srecord
- openocd az csak siman ez:
* openocd
```

Mindegyiket siman `apt-get install ...` módon tudod telepíteni! Ezek sikerültek? Ha ezek mind fent vannak, akkor az általam mellekelt

forrast

```
mar le
tudod fordítani és fel kell tudnod tölteni a kis board-ra. Ami meg
kellhet
a
forgatáshoz azok az STM32-specifikus headerek. Ezeket elkuldom
mellekelve.
En
konkretan sima rendszergazdaként az /usr/local/arm-none-eabi/include
alkovnytar
alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt
teszprogramban
levő Makefile-ban az elejen (kb 12-15ik sorban) levő
```

```
ARM=/usr/local/arm-none-eabi
```

sort modosítsd oda ahova betetted! Lehet siman nálad is, a /home-od alatt!

De  
ugyis panaszkodik a fordító hogyha nem találja ;) Szóval legjobb  
emlekeim  
szerint ennyi kell, és ami meg hiányzik az ezeknek a csomagoknak a  
fuggósege -  
szóval jó kell hogy legyen, legjobb emlekeim szerint.

Az FPGA az sajnos tényleg egy fokkal bonyolultabb. Elobb nezzük meg

ezt

az  
ARM + STM32 témát, és akkor utána. Valójában azokat sem nehéz

feltenni,  
de

kellhet pár olyan development header (pl libboost-csomag) amiket  
feltetelez  
hogy fent van de alából szinte biztos hogy nincs fent nálad. Nalam  
konkretan  
ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-filesystem-dev:amd64	install
libboost-filesystem1.62-dev:amd64	install
libboost-filesystem1.62.0:amd64	install
libboost-iostreams-dev:amd64	install
libboost-iostreams1.62-dev:amd64	install
libboost-iostreams1.62.0:amd64	install
libboost-program-options-dev:amd64	install
libboost-program-options1.62-dev:amd64	install
libboost-program-options1.62.0:amd64	install
libboost-python-dev	install
libboost-python1.62-dev	install
libboost-python1.62.0	install
libboost-regex1.62-dev:amd64	install
libboost-regex1.62.0:amd64	install
libboost-serialization1.62-dev:amd64	install



```
libboost-serialization1.62.0:amd64      install
libboost-system1.62-dev:amd64           install
libboost-system1.62.0:amd64             install
libboost-thread-dev:amd64               install
libboost-thread1.62-dev:amd64           install
libboost-thread1.62.0:amd64             install
libboost1.62-dev:amd64                  install
```

szoval ha azt mondod hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev
libboost-chrono-dev libboost-date-time-dev libboost-dev
libboost-filesystem-dev libboost-iostreams-dev
libboost-program-options-dev
libboost-python-dev libboost-regex-dev libboost-serialization-dev
libboost-system-dev libboost-thread-dev
```

(igen, ez így egy jó hosszú parancssor :)), akkor mindent is feltesz

ami

kell a  
libboosthoz (es a sima lib-eket is felteszi ami kell a \*-dev  
csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De tenyleg

ez

kicsit  
macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi az

ami

hiányzik. Hanyas verzioju milyen linuxot is tettél fel pontosan?

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a  
küldött

dolgokat olvastam. Sajnos sem az openocd-t, sem az icesstorm  
toolchain-t  
nem sikerült "életre keltenem". Elvileg a yosys-0.9, nextpnr-ice40 és  
icesstorm-utils csomagokat is telepítettem a megadott módon. Tud

esetleg

küldeni valamilyen segítséget, hogyan tudnám programozni ezekentúl a  
lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék segítséget  
kérni,

telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam  
kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom

elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a  
programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai  
eszközünkkel TDK csoportunkban elsőkké lettünk, így a Semmelweis

Egyetem

TDK

eszköz,  
és

mellett készülhetünk az országos TDK-ra. Mivel már kész van az

csak bemutatni kell, így nem vesz el jelentős időt.

két

Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy

szabadsági fokról álló állványon levő parabola-antenna fogja  
"bescannelni"

majd az eget, és a vett jelerősséget fogja egy skála szerint osztályozni,  
ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni.

Fő

célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell érzékelnünk.  
Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő érzékelővel.  
Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb lenne) csinálni?

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo, 16:02):

Szia Gábor!

Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper lesz,  
azon  
nagyjabol mindent is meg lehet csinálni amit debian-on, es hogyha standalone  
es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!

Headless  
üzemben mar nem használnék egyatalan ubuntu-t, de az mas kerdes ;)

Linux alá viszont van teljesen free & open souce Lattice FPGA toolchain:  
<http://www.clifford.at/icestorm/>, ezt erdemes lehet feltelepíteni.

Mi

is  
hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icestorm-utils), ezekkel a csomagokkal kozvetlenul is tudod linux alatt mindenfele  
licenc-problema  
meg  
nyűg nélkül hasznalni a barmit! Cserebe a forras(kod)bol valo  
telepites  
nem  
annyira egyszerű, eleg sok a függőség, es nehezen derül ki hogy mi

az.

De  
ebben  
viszont a windozos cuccokkal ellentétben tudok segíteni :)

Ugyanakkor

termesztesen en is neztem a linuxos icecube2-t, es az is mukodik  
persze,  
szoval  
hajra azzal is. Mondjuk az nekem kicsit nehezkes, a "bare metal"

meg a

klasszikus "unix filozofia" dolgok utan nekem ezeket az IDE-ket eleg  
nehéz  
megszoknom :/ De az mas kerdes ;)

A.

On Sat, 7 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos tanulmányozásának.

ubuntut, Egy közeli ismerősöm ajánlásából elkezdtük föltelepíteni az

mivel  
a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem  
volt  
zökkenőmentes a telepítés. Először a pendrive amin keresztül  
telepíteni  
akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt  
föl.  
Majd  
miután sikerült, a Grub nem települt így nem tudtam elindítani a  
rendszert.  
Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és  
egyebek.  
Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált  
adataimat  
el  
kell fogadják, de az is hamarosan meglesz.  
A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött  
pdf-eket  
át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.  
Szerencsére  
az is teljesen témába vág, hiszen C programozás lesz.  
Nagyon le voltam nyűgözve a találkozókör és remélem minden rendben  
lesz!  
Köszönettel:  
K. Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6., P,  
9:44):  
Szia Gábor!  
Persze, kuldom akkor az infokat:  
- mellekelve az evaluation board doksija, ebben benne kell  
legyen  
minden  
referencia ami hasznos lehet - bar ugye a mi boardunk az  
másmilyen.  
- a fejlesztokeszlet az ez:  
<https://www.latticesemi.com/iCEcube2>,  
van  
minden  
oprendszer ala (linux, windoz).  
- az FPGA az ez: iCE40HX8K-BG121, ezt kell kiválasztanod majd.  
- mellekelem a tegnap tesztelt kis verilog-kodot is, abban  
megtalalad a  
PCF-et: ezt a PCF-et megeszi az icecube2 is! (a verilog-ot meg  
nyilván)  
Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:  
- mellekelem a tegnapi peldaprogramot (UART: kisbetu ->  
nagybetu)  
- a toolchain (ami a \*.c fileokbol megcsinalja a \*.bin es/vagy  
\*.hex  
image-t,  
amit aztan fel kell tolneni) az `arm-none-eabi` néven fut. Ennek  
több  
komponense van:  
\* A C/C++/asm fordito maga: ez a gcc-arm-none-eabi  
\* Eszkozok amikkel a lefordított fileokat lehet  
manipulálni  
(pl  
a  
\*.hex-et eloallítani a \*.o/\*.elf-ből):  
binutils-arm-none-eabi  
\* Es az elore lefordított konyvtaraok, hogy a jól

megszokott

C-s

(stdio: printf, stb) meg C++-os (STL es tarsai) dolgokat

tudd

hasznalni: libnewlib-arm-none-eabi,  
libstdc++-arm-none-eabi-newlib.

Ez kb ugyanilyen néven fut windoz alatt is, egy gyors keresessel

pl

ezt

talaltam:

<https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>

A lenyeg hogy ez a "gcc-arm-none-eabi for windows" dolologra

keress

rá!

A programozo az ST-Link V2 (-1), ezt lehet OpenOCD-vel

progrmaozni.

Az

OpenOCD

van linux ala is, de van windozos prortja is:

<https://gnutoolchains.com/arm-eabi/openocd/>. Ettol fuggetlenul az

ST

sajat

cuccait is lehet hasznalni valoszinuleg:

<https://www.st.com/en/development-tools/stsw-link009.html>, de az

OpenOCD

annyira jo hogy windozt hasznalo kollegaktol is azt hallom hogy

inkabb

az

OpenOCD-t preferaljak az STM32-es MCU-khoz.

A.

On Fri, 6 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Említett nekem segédleteket a lattice boardhoz. Továbbá

említette,

hogy

az

fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is

szeretnék

kérni segítséget.

Köszönettel: K. Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont:

2020.

nov.

4., Sze, 17:28):

Rendben!

Szerintem én is megebédelek és indulok

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4.,

Sze

17:09):

Szia Gábor!

Megebédek odalent, kb olyan fel 1-1-ig bezarolag, es akkor

utana

indulok

fel. Szoval fel 2 korultól mar leszek itt. Ha errefele jarsz

akkor

hivj

fel:

(20)3962151, es akkor kimegyek a kapuhoz. Merthogy mi itten egy

masik

epulet

masik szekciojaban vagyunk.

udv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper!

Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4.,

Sze

16:45):

Szia Gábor!

No, tul vagy kozben a mindenen? Illetve a het hetralevo resze

akkor

mikor

lenne

jo? En holnap delutan tervezek feljonni a normafára elso

kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort

gyakorlás

céljából?

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta

(időpont:

2020.

okt.

29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.

TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz

minden, a

hetedik

hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét

elején.

Ma megyek Normafára a demonstrátori dolgokat aláírni, így

ha

minden

igaz,



szépen!

Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm

Andras Pal &lt;apal@szofi.net&gt; ezt írta (időpont: 2020.

okt.

26., H

10:38):

Igen, a "Normafa" az maga az intézet, Konkoly-Thege M.

ut

15-17.

Egyesek

**Andras Pal** <apal@szofi.net>

2020. november 20. 19:59

Címzett: Gábor Kreinicker &lt;kreinickergabor@gmail.com&gt;

Szia Gábor!

No, van barmi fejlemeny? :) Pontosán milyen linuxot is hasznalsz? Ugy jobban ra tudunk keresni az adott library (\*a file) pontos helyére.

Ez tenyleg elegge melyviz, szoval absz ne ijedj meg tenyleg ha valami nem megy kapasbol es tenyleg szolj ha elakadsz valahol!

A.

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Sajnos nem ez a gond, már telepítve van.

Andras Pal &lt;apal@szofi.net&gt; ezt írta (időpont: 2020. nov. 19., Cs 21:33):

Tedd fel a libnewlib-arm-none-eabi nevű csomagot!

apt-get install libnewlib-arm-none-eabi

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Kedves András Pál!

Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül megoldanom,  
mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a boardra. Amit leírt a korábbiakban, mindent többször átfutottam és ellenőriztem, de kifog rajtam egyelőre a Linux.  
El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a "st-info --probe" sort a terminálba. A headereket sikeresen bemásoltam a megfelelő helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a következő sorokat kapom a "make upload" esetén:

```
arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or
directory
make: *** [Makefile:54: main.elf] Error 1
```

Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt,  
ezért  
elnézést!

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 17., K, 20:36):

A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a board-ra, nem a main.c-t eteted meg kozvetlenul. Probald ki a `make upload` parancsot,

az

az  
ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a Makefile bogaraszasabol rajossz hogy hogy is van ez pontosan.

Szoval a workflow az itten a kovetkezo:

- A \*.c fileokbol \*.o (object) fileokat forditunk (arm-none-eabi-gcc)
- Az egeszet osszszelinkeljuk (arm-none-eabi-ld)

- Ez a tipusu mikrokontroller a 0x000000004-es címen található utasításra

ugrik először, ott kezdi a program futtatását

- Ez a tipusu mikrokontroller olyan hogy a 0x000000000 tartomanyra a 0x08000000 tartományt kepezi le alapertelmezesben

- A linkelésnél így garantálnunk kell hogy legyen "valami" ami a 0x080000004-es címen ami értelmes \_és\_ kesobb meghivja a main()

fuggvenyt

-  
merthogy minden C program a main()-nal "kezdodik". Na, ezt a feladatot latja el

a crt.c - plusz még pár egyebet is, de azzal nem igazan kell

foglalatzkodni,

azzal majd kesobb ;)

- A linkeles utan ki kell szednunk a keletkezo \*.elf filebol a "lenyeget". Ezt

csinalja a arm-none-eabi-objcopy, es ez hozza letre a \*.hex-et.

- Ezt a \*.hex-et pedig az openocd-vel beleirhatjuk a mikrokontroller flash-ébe

Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo

sorrendben.

Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es az

openocd

konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani neki hogy

milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target

mikrokontroller (azaz a programozonak tudnia kell hogy a programozasi

protokollon keresztul hogyan eri el a mikrokontroller memoriajat es ez

fugghet

a mikrokontroller konkret tipusától - bar egy csaladon belül ez elegge

egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem azt hogy

stm32f072c6t6). Plusz meg vannak egyeb finomsagok is, pl a flash-t ugye

minden

iras előtt torolni kell es csak utana irhatsz bele, stbstb. Ezeket

mind-mind

megcsinalja a Makefile-be elrejtett parancs- valami.

Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra ugyanigy

megy,

csak itt ez a "bare metal" modszer ezt is szuksegesse teszi hogy

lepesrol

lepesre menj(unk) vegig a lepeseken.

Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott

hasznaltatok

valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy hogy?

A.



On Tue, 17 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent telepítettem. Ami még nem sikerült, az az openocd életre bírása.

Csatolok

egy képet, és bízom benne, hogy tud segíteni.

Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az

st-link

csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a feltöltésig nem jutottam el.

Köszönettel: K. Gábor

[image: Screenshot from 2020-11-17 19-53-48.jpg]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 16., H, 18:35):

Szia Gábor!

Ah, értem, ez így akkor elsore tenyleg nem egyszerű. Viszont ennél specifikusabban kene tudni hogy hol is akadtál el. Tehat:

- telepites ment? Minden fent van?
- az arm-es toolchain az ezen csomagokbol all:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi
  - \* libnewlib-arm-none-eabi,
  - \* libstdc++-arm-none-eabi-newlib.
- ami meg hasznos csomag es kellhet:
  - \* build-essential (ebben van helyi C fordito is + a `make`

program)

- \* srecord
- openocd az csak siman ez:
  - \* openocd

Mindegyiket siman `apt-get install ...` modon tudod telepíteni! Ezek sikerultek? Ha ezek mind fent vannak, akkor az altalam mellekelt

forrast

mar le tudod fordítani es fel kell tudnod tölteni a kis board-ra. Ami meg

kellhet

a forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom mellekelve.

En konkretan sima rendszergazdakent az /usr/local/arm-none-eabi/include alkovnytar alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt teszprogramban levő Makefile-ban az elejen (kb 12-15ik sorban) levő

ARM=/usr/local/arm-none-eabi

sort modosítsd oda ahova betetted! Lehet siman nálad is, a /home-od alatt!

De ugyis panaszkodik a fordito hogyha nem találja ;) Szoval legjobb emlekeim

szerint ennyi kell, es ami meg hiányzik az ezeknek a csomagoknak a fuggosege - szoval jo kell hogy legyen, legjobb emlekeim szerint.

Az FPGA az sajnos tenyleg egy fokkal bonyolultabb. Elobb nezzuk meg

ezt

az

ARM + STM32 témát, es akkor utana. Valojaban azokat sem nehez

feltenni,

de

kellhet par olyan development header (pl libboost-csomag) amiket  
feltetelez  
hogy fent van de alapbol szinte biztos hogy nincs fent nálad. Nalam  
konkretan  
ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-filesystem-dev:amd64	install
libboost-filesystem1.62-dev:amd64	install
libboost-filesystem1.62.0:amd64	install
libboost-iostreams-dev:amd64	install
libboost-iostreams1.62-dev:amd64	install
libboost-iostreams1.62.0:amd64	install
libboost-program-options-dev:amd64	install
libboost-program-options1.62-dev:amd64	install
libboost-program-options1.62.0:amd64	install
libboost-python-dev	install
libboost-python1.62-dev	install
libboost-python1.62.0	install
libboost-regex1.62-dev:amd64	install
libboost-regex1.62.0:amd64	install
libboost-serialization1.62-dev:amd64	install
libboost-serialization1.62.0:amd64	install
libboost-system1.62-dev:amd64	install
libboost-system1.62.0:amd64	install
libboost-thread-dev:amd64	install
libboost-thread1.62-dev:amd64	install
libboost-thread1.62.0:amd64	install
libboost1.62-dev:amd64	install

szoval ha azt mondod hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev  
libboost-chrono-dev libboost-date-time-dev libboost-dev  
libboost-filesystem-dev libboost-iostreams-dev
```

```
libboost-program-options-dev
```

```
libboost-python-dev libboost-regex-dev libboost-serialization-dev  
libboost-system-dev libboost-thread-dev
```

(igen, ez igy egy jó hosszú parancssor :)), akkor mindent is feltesz

ami

kell a  
libboosthoz (es a sima lib-eket is felteszi ami kell a \*-dev  
csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De tenyleg

ez

kicsit  
macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi az

ami

hianyzik. Hanyas verzioju milyen linuxot is tettél fel pontosan?

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a küldött

dolgokat olvasgattam. Sajnos sem az openocd-t, sem az icestorm toolchain-t nem sikerült "éltre keltenem". Elvileg a yosys-0.9, nextpnr-ice40 és icestorm-utils csomagokat is telepítettem a megadott módon. Tud

esetleg küldeni valamilyen segítséget, hogyan tudnám programozni ezeket a lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék segítséget kérni, telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai eszközünkkel TDK csoportunkban elsőkké lettünk, így a Semmelweis

Egyetem TDK mellett készülhetünk az országos TDK-ra. Mivel már kész van az eszköz, és

csak bemutatni kell, így nem vesz el jelentős időt.

Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy

két szabadsági fokból álló állványon levő parabola-antenna fogja "bescannelni"

majd az eget, és a vett jelerősséget fogja egy skála szerint osztályozni,

Fő ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni.

célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell érzékelnünk.

Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő érzékelővel.

Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb lenne) csinálni?

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo, 16:02):

Szia Gábor!

Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper lesz,

azon nagyjabol mindent is meg lehet csinálni amit debian-on, és hogyha standalone es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!

Headless üzemben mar nem használnék egyáltalán ubuntu-t, de az más kérdés ;)

Linux alá viszont van teljesen free & open source Lattice FPGA toolchain:

<http://www.clifford.at/icestorm/>, ezt érdemes lehet feltölteni.

Mi is használjuk ezt (yosys-0.9, nextpnr-ice40, icedstorm-utils), ezekkel a csomagokkal közvetlenül is tudod linux alatt mindenféle licenc-problema meg nyűg nélkül használni a barmit! Cserebe a forrás(kod)bol való telepítés nem annyira egyszerű, elég sok a függőség, és nehezen derül ki hogy mi az.

De ebben viszont a windozos cuccokkal ellentétben tudok segíteni :)

Ugyanakkor természetesen én is néztem a linuxos icecube2-t, és az is működik persze, szoval hajra azzal is. Mondjuk az nekem kicsit nehezkés, a "bare metal" meg a klasszikus "unix filozofia" dolgok után nekem ezeket az IDE-ket elég nehéz megszoknom :/ De az más kérdés ;)

A.

On Sat, 7 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos tanulmányozásának.

Egy közeli ismerősöm ajánlásából elkezdtük föltelepíteni az ubuntu-t, mivel a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem volt zökkenőmentes a telepítés. Először a pendrive amin keresztül telepíteni akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt föl.

Majd miután sikerült, a Grub nem települt így nem tudtam elindítani a rendszert.

Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és egyebek.

Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált adataimat el kell fogadják, de az is hamarosan meglesz.

A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött pdf-eket át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.

Szerencsére az is teljesen témába vág, hiszen C programozás lesz.

Nagyon le voltam nyűgözve a találkozókör és remélem minden rendben lesz!

Köszönettel:  
K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6., P, 9:44):

Szia Gábor!

Persze, kuldom akkor az infokat:  
 - mellekelve az evaluation board doksija, ebben benne kell  
 legyen minden  
 referencia ami hasznos lehet - bar ugye a mi boardunk az  
 masmilyen.  
 - a fejlesztokeszlet az ez:  
<https://www.latticesemi.com/iCEcube2>,  
 van minden  
 oprendszer ala (linux, windoz).  
 - az FPGA az ez: iCE40HX8K-BG121, ezt kell kivlasztanod majd.  
 - mellekelem a tegnap tesztelt kis verillog-kodot is, abban  
 megtalalalad a  
 PCF-et: ezt a PCF-et meg teszi az icecube2 is! (a verillog-ot meg  
 nyilván)  
 Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:  
 - mellekelem a tegnap peldaprogramot (UART: kisbetu ->  
 nagybetu)  
 - a toolchain (ami a \*.c fileokbol megcsinalja a \*.bin es/vagy  
 \*.hex  
 image-t,  
 amit aztan fel kell tolteni) az `arm-none-eabi` néven fut. Ennek  
 tobb  
 komponense van:  
 \* A C/C++/asm fordito maga: ez a gcc-arm-none-eabi  
 \* Eszkozok amikkel a leforditott fileokat lehet  
 manipulalni  
 (pl  
 a  
 \*.hex-et eloallitani a \*.o/\*.elf-ből):  
 binutils-arm-none-eabi  
 \* Es az elore leforditott konyvtaraok, hogy a jól  
 megszokott  
 C-s  
 (stdio: printf, stb) meg C++-os (STL es tarsai) dolgokat  
 tudd  
 hasznalni: libnewlib-arm-none-eabi,  
 libstdc++-arm-none-eabi-newlib.  
 Ez kb ugyanilyen néven fut windoz alatt is, egy gyors keresessel  
 pl  
 ezt  
 talaltam:  
<https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>  
 .  
 A lenyeg hogy ez a "gcc-arm-none-eabi for windows" dollogra  
 keress  
 rá!  
 A programozo az ST-Link V2 (-1), ezt lehet OpenOCD-vel  
 progrmaozni.  
 Az  
 OpenOCD  
 van linux ala is, de van windozos prortja is:  
<https://gnutoolchains.com/arm-eabi/openocd/>. Ettol fuggetlenul az  
 ST  
 saját  
 cuccait is lehet hasznalni valoszinuleg:

<https://www.st.com/en/development-tools/stsw-link009.html>, de az OpenOCD annyira jó hogy windozt használó kollegáktól is azt hallom hogy inkább az OpenOCD-t preferálják az STM32-es MCU-khoz.

A.

On Fri, 6 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Említett nekem segédleteket a lattice boardhoz. Továbbá említette, hogy az fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is szeretnék kérni segítséget.

Köszönettel: K. Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2020. nov. 4., Sze, 17:28):

Rendben!

Szerintem én is megebédelek és indulok

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 17:09):

Szia Gábor!

Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, és akkor indulok fel. Szóval fel 2 korultól már leszek itt. Ha errefele jarsz akkor hívj fel: (20)3962151, és akkor kimegyek a kapuhoz. Merthogy mi itten egy másik épület másik szekciojában vagyunk. udv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper! Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 16:45):

Szia Gábor!

No, tud vagy közben a mindenen? Illetve a hét hetralevo része

akkor

mikor

lenne

jo? En holnap delutan tervezek feljenni a normafára elso

kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort

gyakorlás

céljából?

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta

(időpont:

2020.

okt.

29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.

TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz

minden, a

hetedik

hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét

elején.

Ma megyek Normafára a demonstrátori dolgokat aláírni, így

ha

minden

igaz,

az is meglesz.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt.

28.,

Sze,

18:44):

Oke! Akkor viszont lehet hogy diskuraljunk megiscsak jovo

heten,

mert

kozben

engem is nagyon bekapott az ekszija es/vagy a daralo a

műhold

kapcsan.

Most mar

szerda van, es meg mindig nem tudom hogy a holnap az hogy

lesz.

Es

attól

felek

hogy csak delelott derul ki hogy mar delutan fel kell

mennem

vagy

csak

penteken  
vagy hogy.

Neked mikor vannak ezek a hataridok es/vagy mikor zarodnak

le

a

dolgok  
(melyik  
is pl az a hetedik het, meg mi a TDK határideje,  
ilyesmik)?  
A.  
On Mon, 26 Oct 2020, Gábor Kreinicker wrote:  
Beleolvastam, de a hetedik heti hajrá és a TDK határidő  
közelsége  
nem  
bejött, a  
engedett komolyan elmélyedni még benne. A téma nagyon  
jövőhétre  
már felkészültebb leszek  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.  
okt. 26., H 23:07):  
A doksikat/referenciákat mennyire tudta megnevezni amiket  
küldtem?  
Kiindulásnak  
ezek mennyire jók? Persze kicsit még levegőben lógnak a  
temak,  
de  
ettől  
függetlenül(?)  
A  
On Mon, 26 Oct 2020, Gábor Kreinicker wrote:  
Rendben, akkor ezt majd szerdáig megbeszéljük, köszönöm  
szépen!  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020.  
okt. 26., H 10:38):  
Igen, a "Normafa" az maga az intézet, Konkoly-Thege M.  
ut  
15-17.  
Egyesek

**Gábor Kreinicker** <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)>  
Címzett: Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)>

2020. november 21. 11:11

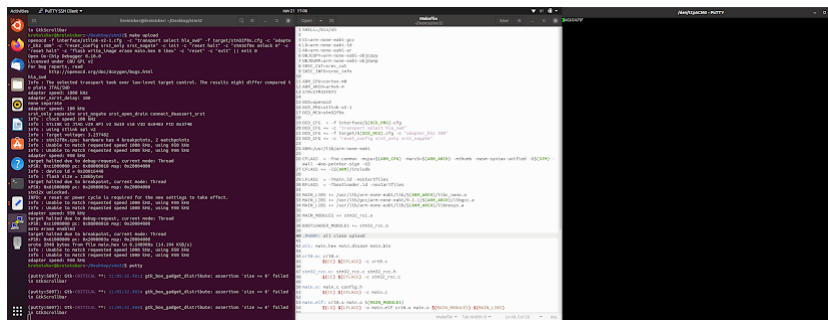
Kedves Pál András! (legutóbb is elrontottam a sorrendet)

Van fejlemény! A Makefile-ban rossz helyek voltak megjelölve forrásnak (középső kép, MAIN\_LIBS sorok 32-34). Így már föltöltötte a kódot ami a csatolt kép bal oldalán látható.

A putty nevű programmal értem el a boardot (jobb oldali kép). Ugye a kód az arról szól, hogy ha nyomok egy gombot, akkor világít az LD2 led, ha pedig megnyomok egy újabb gombot, akkor pedig elalszik. Ez tökéletesen működik. Azonban putty-n keresztül valamiért csak nagybetűket tudok írni, valamint enter nyomás esetén a sor elejére ugrik. Elvileg el kéne tolnia a betűket az abc-ben, de ez nem igazán történik meg. Ennek lehet oka a putty? Emlékszem, hogy ön nem ezt használta, de nem emlékszem, mivel tudott kommunikálni vele.

Köszönöm a segítséget!  
K. Gábor





Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 20., P, 19:59):

Szia Gábor!

No, van barmit fejlemeny? :) Pontosán milyen linuxot is hasznalsz? Ugy jobban ra tudunk keresni az adott library (\*.a file) pontos helyére.

Ez tenyleg elegge melyviz, szoval absz ne ijedj meg tenyleg ha valami nem megy kapasbol es tenyleg szolj ha elakadsz valahol!

A.

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

> Sajnos nem ez a gond, már telepítve van.

>

> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 19., Cs 21:33):

>

>> Tedd fel a libnewlib-arm-none-eabi nevű csomagot!

>>

>> apt-get install libnewlib-arm-none-eabi

>>

>> On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

>>

>>> Kedves András Pál!

>>>

>>> Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül

>> megoldanom,

>>> mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a

>>> boardra. Amit leírt a korábbiakban, mindent többször átfutottam és

>>> ellenőriztem, de kifog rajtam egyelőre a Linux.

>>> El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a "st-info

>>> --probe" sort a terminálba. A headereket sikeresen bemásoltam a megfelelő

>>> helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a következő

>>> sorokat kapom a "make upload" esetén:

>>>

>>> arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o

>>> stm32\_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc\_nano.a

>>> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a

>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a

>>> arm-none-eabi-ld: cannot find

>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libc\_nano.a: No such file or

>> directory

>>> arm-none-eabi-ld: cannot find

>>> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or

>> directory

>>> arm-none-eabi-ld: cannot find

>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or

>> directory

>>> make: \*\*\* [Makefile:54: main.elf] Error 1

>>>

>>> Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt,

>> ezért

>>> elnézést!

>>>

>>> Köszönettel: K. Gábor

>>>

>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 17., K,

```
>> 20:36):
>>>
>>>> A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a board-ra,
>>>> nem a
>>>> main.c-t eteted meg kozvetlenul. Probald ki a `make upload` parancsot,
>> az
>>>> az
>>>> ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a Makefile
>>>> bogaraszasabol rajossz hogy hogy is van ez pontosan.
>>>>
>>>> Szoval a workflow az itten a kovetkezo:
>>>> - A *.c fileokbol *.o (object) fileokat forditunk (arm-none-eabi-gcc)
>>>> - Az egeszet összelinkeljük (arm-none-eabi-ld)
>>>> - Ez a tipusu mikrokontroller a 0x000000004-es címen található
>>>> utasításra
>>>> ugrik először, ott kezdi a program futtatását
>>>> - Ez a tipusu mikrokontroller olyan hogy a 0x000000000 tartomanyra a
>>>> 0x08000000 tartományt kepezi le alapertelmezesben
>>>> - A linkelésnél így garantálnunk kell hogy legyen "valami" ami a
>>>> 0x080000004-es címen ami értelmes _és_ kesobb meghivja a main()
>> fuggvenyt
>>>> -
>>>> merthogy minden C program a main()-nal "kezdodik". Na, ezt a feladatot
>>>> latja el
>>>> a crt.c - plusz még pár egyebet is, de azzal nem igazan kell
>>>> foglalatoskodni,
>>>> azzal majd kesobb ;)
>>>> - A linkeles utan ki kell szednunk a keletkezo *.elf filebol a
>>>> "lenyeget". Ezt
>>>> csinalja a arm-none-eabi-objcopy, es ez hozza létre a *.hex-et.
>>>> - Ezt a *.hex-et pedig az openocd-vel beleirhatjuk a mikrokontroller
>>>> flash-ébe
>>>>
>>>> Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo
>> sorrendben.
>>>> Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es az
>>>> openocd
>>>> konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani neki
>>>> hogy
>>>> milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target
>>>> mikrokontroller (azaz a programozonak tudnia kell hogy a programozasi
>>>> protokollon keresztul hogyan eri el a mikrokontroller memoriajat es ez
>>>> fugghet
>>>> a mikrokontroller konkret tipusától - bar egy csaladon belül ez elegge
>>>> egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem azt hogy
>>>> stm32f072c6t6). Plusz meg vannak egyeb finomsagok is, pl a flash-t ugye
>>>> minden
>>>> iras előtt torolni kell es csak utana írhat sz bele, stbstb. Ezeket
>>>> mind-mind
>>>> megcsinalja a Makefile-be elrejtett parancs- valami.
>>>>
>>>> Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra ugyanigy
>>>> megy,
>>>> csak itt ez a "bare metal" modszer ezt is szuksegesse teszi hogy
>> lepesrol
>>>> lepesre menj(unk) vegig a lepeseken.
>>>>
>>>> Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott
>>>> hasznaltatok
>>>> valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy hogy?
>>>>
>>>> A.
>>>>
>>>> On Tue, 17 Nov 2020, Gábor Kreinicker wrote:
>>>>
>>>>> Kedves Pál András!
```

```
>>>>>
>>>>> Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent
>>>>> telepítettem. Ami még nem sikerült, az az openocd életre bírása.
>> Csatolok
>>>>> egy képet, és bízom benne, hogy tud segíteni.
>>>>> Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az
>> st-link
>>>>> csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a
>>>>> feltöltésig
>>>>> nem jutottam el.
>>>>>
>>>>> Köszönettel: K. Gábor
>>>>> [image: Screenshot from 2020-11-17 19-53-48.jpg]
>>>>>
>>>>>
>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 16., H,
>>>>> 18:35):
>>>>>
>>>>>> Szia Gábor!
>>>>>>
>>>>>>> Ah, ertem, ez így akkor elsore tenyleg nem egyszerű. Viszont ennél
>>>>>>> specifikusabban kene tudni hogy hol is akadtál el. Tehat:
>>>>>>> - telepites ment? Minden fent van?
>>>>>>> - az arm-es toolchain az ezen csomagokbol all:
>>>>>>> * gcc-arm-none-eabi
>>>>>>> * binutils-arm-none-eabi
>>>>>>> * libnewlib-arm-none-eabi,
>>>>>>> * libstdc++-arm-none-eabi-newlib.
>>>>>>> - ami meg hasznos csomag es kelhet:
>>>>>>> * build-essential (ebben van helyi C fordito is + a `make`
>>>>> program)
>>>>>>> * srecord
>>>>>>> - openocd az csak siman ez:
>>>>>>> * openocd
>>>>>>>
>>>>>>> Mindegyiket siman `apt-get install ...` modon tudod telepiteni! Ezek
>>>>>>> sikerultek? Ha ezek mind fent vannak, akkor az altalam mellekelt
>> forrast
>>>>>>> mar le
>>>>>>> tudod fordítani es fel kell tudnod tölteni a kis board-ra. Ami meg
>>>>> kelhet
>>>>>>> a
>>>>>>> forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom
>>>>> mellekelve.
>>>>>>> En
>>>>>>> konkretan sima rendszergazdakent az /usr/local/arm-none-eabi/include
>>>>>>> alkovnytar
>>>>>>> alatt tárolok ezeket, de barhova felteheted! Akkor a mellekelt
>>>>>>> teszprogramban
>>>>>>> levő Makefile-ban az elejen (kb 12-15ik sorban) levő
>>>>>>>
>>>>>>> ARM=/usr/local/arm-none-eabi
>>>>>>>
>>>>>>> sort modosítsd oda ahova betetted! Lehet siman nálad is, a /home-od
>>>>> alatt!
>>>>>>> De
>>>>>>> ugyis panaszkodik a fordito hogyha nem találja ;) Szoval legjobb
>>>>> emlekeim
>>>>>>> szerint ennyi kell, es ami meg hiányzik az ezeknek a csomagoknak a
>>>>>>> fuggosege -
>>>>>>> szoval jo kell hogy legyen, legjobb emlekeim szerint.
>>>>>>>
>>>>>>> Az FPGA az sajnos tenyleg egy fokkal bonyolultabb. Elobb nezzuk meg
>> ezt
>>>>> az
>>>>>>> ARM + STM32 témát, es akkor utana. Valojaban azokat sem nehez
```

```
>> feltenni,
>>>> de
>>>>> kellhet par olyan development header (pl libboost-csomag) amiket
>>>>> feltetelezz
>>>>> hogy fent van de alapból szinte biztos hogy nincs fent nálad. Nalam
>>>>> konkrétan
>>>>> ezek vannak fent:
>>>>>
>>>>> libboost-atomic1.62-dev:amd64          install
>>>>> libboost-atomic1.62.0:amd64            install
>>>>> libboost-chrono1.62-dev:amd64          install
>>>>> libboost-chrono1.62.0:amd64            install
>>>>> libboost-date-time1.62-dev:amd64        install
>>>>> libboost-date-time1.62.0:amd64          install
>>>>> libboost-dev:amd64                      install
>>>>> libboost-filesystem-dev:amd64           install
>>>>> libboost-filesystem1.62-dev:amd64       install
>>>>> libboost-filesystem1.62.0:amd64         install
>>>>> libboost-iostreams-dev:amd64           install
>>>>> libboost-iostreams1.62-dev:amd64        install
>>>>> libboost-iostreams1.62.0:amd64          install
>>>>> libboost-program-options-dev:amd64      install
>>>>> libboost-program-options1.62-dev:amd64  install
>>>>> libboost-program-options1.62.0:amd64    install
>>>>> libboost-python-dev                    install
>>>>> libboost-python1.62-dev                 install
>>>>> libboost-python1.62.0                   install
>>>>> libboost-regex1.62-dev:amd64            install
>>>>> libboost-regex1.62.0:amd64              install
>>>>> libboost-serialization1.62-dev:amd64    install
>>>>> libboost-serialization1.62.0:amd64      install
>>>>> libboost-system1.62-dev:amd64           install
>>>>> libboost-system1.62.0:amd64             install
>>>>> libboost-thread-dev:amd64               install
>>>>> libboost-thread1.62-dev:amd64           install
>>>>> libboost-thread1.62.0:amd64             install
>>>>> libboost1.62-dev:amd64                  install
>>>>>
>>>>> szóval ha azt mondod hogy
>>>>>
>>>>> # apt-get --no-install-recommends install libboost-atomic-dev
>>>>> libboost-chrono-dev libboost-date-time-dev libboost-dev
>>>>> libboost-filesystem-dev libboost-iostreams-dev
>>>> libboost-program-options-dev
>>>>> libboost-python-dev libboost-regex-dev libboost-serialization-dev
>>>>> libboost-system-dev libboost-thread-dev
>>>>>
>>>>> (igen, ez így egy jó hosszú parancssor :)), akkor mindent is feltesz
>> ami
>>>>> kell a
>>>>> libboosthoz (es a sima lib-eket is felteszi ami kell a *-dev
>>>> csomagokhoz)
>>>>>
>>>>> Ami meg szinte biztos hogy kell az a: flex bison libtool. De tényleg
>> ez
>>>>> kicsit
>>>>> macerásabb, szóval jobban kellene tudnom hogy hol akad el, es mi az
>> ami
>>>>> hiányzik. Hányas verzióju milyen linuxot is tettél fel pontosan?
>>>>>
>>>>> A.
>>>>>
>>>>> On Mon, 16 Nov 2020, Gábor Kreinicker wrote:
>>>>>
>>>>>> Kedves Pál András!
>>>>>>
```

>>>>>> Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a  
>>>> küldött  
>>>>>> dolgokat olvasgattam. Sajnos sem az openocd-t, sem az icestorm  
>>>>>> toolchain-t  
>>>>>> nem sikerült "életre keltennem". Elvileg a yosys-0.9, nextpnr-ice40 és  
>>>>>> icestorm-utils csomagokat is telepítettem a megadott módon. Tud  
>> esetleg  
>>>>>> küldeni valamilyen segítséget, hogyan tudnám programozni ezeketül a  
>>>>>> lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék segítséget  
>>>>>> kérni,  
>>>>>> telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam  
>>>>>> kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom  
>>>> elérni.  
>>>>>>  
>>>>>> Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a  
>>>>>> programozásukig sajnos még nem jutottam el.  
>>>>>>  
>>>>>> Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai  
>>>>>> eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis  
>> Egyetem  
>>>>>> TDK  
>>>>>> mellett készülhetünk az országos TDK-ra. Mivel már kész van az  
>> eszköz,  
>>>> és  
>>>>>> csak bemutatni kell, így nem vesz el jelentős időt.  
>>>>>>  
>>>>>> Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy  
>>>> két  
>>>>>> szabadsági fokból álló állványon levő parabola-antenna fogja  
>>>>>> "bescannelni"  
>>>>>> az eget, és a vett jelerősséget fogja egy skála szerint osztályozni,  
>>>> majd  
>>>>>> ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni.  
>> Fő  
>>>>>> célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell  
>>>>>> érzékelnünk.  
>>>>>> Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő  
>>>> érzékelővel.  
>>>>>> Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg  
>>>>>> valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb  
>>>>>> lenne)  
>>>>>> csinálni?  
>>>>>>  
>>>>>> Köszönettel: K. Gábor  
>>>>>>  
>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo,  
>>>>>> 16:02):  
>>>>>>  
>>>>>>> Szia Gábor!  
>>>>>>>  
>>>>>>> Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper  
>>>> lesz,  
>>>>>>> azon  
>>>>>>> nagyjából mindent is meg lehet csinálni amit debian-on, es hogyha  
>>>>>>> standalone  
>>>>>>> es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!  
>>>> Headless  
>>>>>>> üzemben mar nem használnék egyatalan ubuntu-t, de az mas kerdes ;)  
>>>>>>>  
>>>>>>> Linux alá viszont van teljesen free & open souce Lattice FPGA  
>>>> toolchain:  
>>>>>>> <http://www.clifford.at/icestorm/>, ezt erdemes lehet feltelepiteni.  
>> Mi  
>>>>>>> is  
>>>>>>> hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icestorm-utils), ezekkel a  
>>>>>>> csomagokkal kozvetlenul is tudod linux alatt mindenfele

>>>> licenc-problema  
>>>>>> meg  
>>>>>> nyűg nélkül használni a barmit! Cserebe a forras(kod)bol valo  
>>>> telepites  
>>>>>> nem  
>>>>>> annyira egyszerű, elég sok a függőség, es nehezen derül ki hogy mi  
>> az.  
>>>>>> De  
>>>>>> ebben  
>>>>>> viszont a windozos cuccokkal ellentétben tudok segíteni :)  
>> Ugyanakkor  
>>>>>> természetesen en is neztem a linuxos icecube2-t, es az is mukodik  
>>>> persze,  
>>>>>> szoval  
>>>>>> hajra azzal is. Mondjuk az nekem kicsit nehezkés, a "bare metal"  
>> meg a  
>>>>>> klasszikus "unix filozofia" dolgok utan nekem ezeket az IDE-eket elég  
>>>>>> nehéz  
>>>>>> megszoknom ./ De az mas kérdés ;)  
>>>>>>  
>>>>>> A.  
>>>>>>  
>>>>>> On Sat, 7 Nov 2020, Gábor Kreinicker wrote:  
>>>>>>  
>>>>>> Kedves Pál András!  
>>>>>>  
>>>>>> Köszönök mindent, ma hozzá is állok a dolgok alapos  
>>>> tanulmányozásának.  
>>>>>> Egy közeli ismerősem ajánlásából elkezdtek föltelepíteni az  
>> ubuntu-t,  
>>>>>> mivel  
>>>>>> a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem  
>> volt  
>>>>>> zökkenőmentes a telepítés. Először a pendrive amin keresztül  
>>>> telepíteni  
>>>>>> akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt  
>> föl.  
>>>>>> Majd  
>>>>>> miután sikerült, a Grub nem települt így nem tudtam elindítani a  
>>>>>> rendszert.  
>>>>>> Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és  
>>>>>> egyebek.  
>>>>>> Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált  
>>>>>> adataimat  
>>>>>> el  
>>>>>> kell fogadják, de az is hamarosan meglesz.  
>>>>>> A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött  
>>>>>> pdf-eket  
>>>>>> át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.  
>>>>>> Szerencsére  
>>>>>> az is teljesen témába vág, hiszen C programozás lesz.  
>>>>>> Nagyon le voltam nyugózva a találkozókör és remélem minden rendben  
>>>>>> lesz!  
>>>>>>  
>>>>>> Köszönettel:  
>>>>>> K. Gábor  
>>>>>>  
>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6., P,  
>>>>>> 9:44):  
>>>>>>  
>>>>>>> Szia Gábor!  
>>>>>>>  
>>>>>>> Persze, kuldom akkor az infokat:  
>>>>>>> - mellekelve az evaluation board doksija, ebben benne kell  
>> legyen  
>>>>>>> minden

```
>>>>>>>> referencia ami hasznos lehet - bar ugye a mi boardunk az
>> masmilyen.
>>>>>>>> - a fejlesztokeszlet az ez:
>> https://www.latticesemi.com/iCEcube2,
>>>>>> van
>>>>>>>> minden
>>>>>>>> oprendszer ala (linux, windoz).
>>>>>>>> - az FPGA az ez: iCE40HX8K-BG121, ezt kell kivalasztanod majd.
>>>>>>>> - mellekelem a tegnap tesztelt kis verillog-kodot is, abban
>>>>>>>> megtalalalod a
>>>>>>>> PCF-et: ezt a PCF-et megeszi az icecube2 is! (a verillog-ot meg
>>>>>>>> nyilván)
>>>>>>>>
>>>>>>>> Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:
>>>>>>>> - mellekelem a tegnapi peldaprogramot (UART: kisbetu ->
>> nagybetu)
>>>>>>>> - a toolchain (ami a *.c fileokbol megcsinalja a *.bin es/vagy
>>>> *.hex
>>>>>>>> image-t,
>>>>>>>> amit aztan fel kell tolteni) az `arm-none-eabi` néven fut. Ennek
>>>> tobb
>>>>>>>> komponense van:
>>>>>>>> * A C/C++/asm fordito maga: ez a gcc-arm-none-eabi
>>>>>>>> * Eszkozok amikkel a leforditott fileokat lehet
>> manipulalni
>>>>>> (pl
>>>>>>>> a
>>>>>>>> *.hex-et eloallitani a *.o/*.elf-ből):
>>>>>>>> binutils-arm-none-eabi
>>>>>>>> * Es az elore leforditott konyvtaraok, hogy a jól
>> megszokott
>>>>>>>> C-s
>>>>>>>> (stdio: printf, stb) meg C++-os (STL es tarsai) dolgokat
>>>>>>>> tudd
>>>>>>>> hasznalni: libnewlib-arm-none-eabi,
>>>>>>>> libstdc++-arm-none-eabi-newlib.
>>>>>>>>
>>>>>>>> Ez kb ugyanilyen néven fut windoz alatt is, egy gyors keresessel
>> pl
>>>>>>>> ezt
>>>>>>>> talaltam:
>>>>>>>>
>>>>>>>>
>>>>>>>>
>>>>>>>>
>>>>>>>> https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi
>>>>>>>> .
>>>>>>>>
>>>>>>>> A lenyeg hogy ez a "gcc-arm-none-eabi for windows" dollogra
>> keress
>>>>>>>> rá!
>>>>>>>>
>>>>>>>> A programozo az ST-Link V2 (-1), ezt lehet OpenOCD-vel
>> progrmaozni.
>>>>>>>> Az
>>>>>>>> OpenOCD
>>>>>>>> van linux ala is, de van windozos prortja is:
>>>>>>>> https://gnutoolchains.com/arm-eabi/openocd/. Ettol fuggetlenul az
>>>>>>>> ST
>>>>>>>> saját
>>>>>>>> cuccait is lehet hasznalni valoszinuleg:
>>>>>>>> https://www.st.com/en/development-tools/stsw-link009.html, de az
>>>>>>>> OpenOCD
>>>>>>>> annyira jo hogy windozt hasznalo kollegaktol is azt hallom hogy
>>>>>>>> inkább
```

>>>>>>> az  
>>>>>>>> OpenOCD-t preferáljak az STM32-es MCU-khoz.  
>>>>>>>>  
>>>>>>>> A.  
>>>>>>>>  
>>>>>>>> On Fri, 6 Nov 2020, Gábor Kreinicker wrote:  
>>>>>>>>  
>>>>>>>>> Kedves Pál András!  
>>>>>>>>>  
>>>>>>>>> Említett nekem segédleteket a lattice boardhoz. Továbbá  
>> említette,  
>>>>> hogy  
>>>>>>>> az  
>>>>>>>>> fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is  
>>>>> szeretnék  
>>>>>>>>> kérni segítséget.  
>>>>>>>>>  
>>>>>>>>> Köszönettel: K. Gábor  
>>>>>>>>>  
>>>>>>>>> Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont:  
>>>>> 2020.  
>>>>>>>>> nov.  
>>>>>>>>> 4., Sze, 17:28):  
>>>>>>>>>  
>>>>>>>>> Rendben!  
>>>>>>>>>  
>>>>>>>>> Szerintem én is megebédelek és indulok  
>>>>>>>>>  
>>>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4.,  
>> Sze  
>>>>>>>>> 17:09):  
>>>>>>>>>  
>>>>>>>>>> Szia Gábor!  
>>>>>>>>>> Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, es akkor  
>>>>> utana  
>>>>>>>>>> indulok  
>>>>>>>>>> fel. Szoval fel 2 korultól mar leszek itt. Ha errefele jarsz  
>>>> akkor  
>>>>>>>>> hivj  
>>>>>>>>>> fel:  
>>>>>>>>>> (20)3962151, es akkor kimegyek a kapuhoz. Merthogy mi itten egy  
>>>>> masik  
>>>>>>>>>> epulet  
>>>>>>>>>> masik szekciojaban vagyunk.  
>>>>>>>>>> udv, A.  
>>>>>>>>>>  
>>>>>>>>>> On Wed, 4 Nov 2020, Gábor Kreinicker wrote:  
>>>>>>>>>>  
>>>>>>>>>>> Tisztelt Pál András!  
>>>>>>>>>>>>  
>>>>>>>>>>>>> A holnap délután részemről szuper!  
>>>>>>>>>>>>> Mikor tájtra számítsak?  
>>>>>>>>>>>>>  
>>>>>>>>>>>>> Üdv: Gábor  
>>>>>>>>>>>>>  
>>>>>>>>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4.,  
>>>> Sze  
>>>>>>>>>>>>>> 16:45):  
>>>>>>>>>>>>>>  
>>>>>>>>>>>>>>> Szia Gábor!  
>>>>>>>>>>>>>>>>  
>>>>>>>>>>>>>>>>> No, tul vagy kozben a mindenen? Illetve a het hetralevo resze  
>>>>> akkor  
>>>>>>>>>>>>>>>> mikor  
>>>>>>>>>>>>>>>>> lenne  
>>>>>>>>>>>>>>>>>> jo? En holnap delutan tervezek feljonni a normafára elso



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]











**Andras Pal** <apal@szofi.net>

2020. november 21. 11:36

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Remek! Jaj de jó!

Nade igen, a soros portos eleres nem annyira magatol ertetodo, es igen, lehet hogy a putty az mas újsor-karakter konvenciot hasznal, raadasul nem is echo-zza vissza vsz a terminalra a karaktereket (igy valojaban amit begepelsz azt nem latod mert nem a kepernyodre irja hanem kikuldi a soros porton, es csak azt latod amit visszakuld: es az valoban mar a nagybetűs változat mert ugye azt kuldi vissza a board!).

Az ujsor-karakterek minden oprendszerben masok, raadasul a harom legelterjedtebben (linux, mac/osx, windoz) is mind kulonbozoker (rendre 0A, 0D, 0D+0A), lasd: <https://en.wikipedia.org/wiki/Newline#Representation>, es jo kerdes hogy a putty-nal mi az alapertelmezett... ez is bekavarhat. (( Ennek meg regi tortente van es a klasszikus mechanikus irogepek koraig nyulik vissza, mert ott meg nem volt a klasszikus "ujisor", hanem csak a "kocsi vissza" meg a "soremelés" - hiszen mikor irogeppel irtal es kezdtel ujsort akkor a papirt is hajtanoi kellett ("feed"-elni a hengert), plusz a hengert (carriage) magat visszakuendeni az alaphelyzetebe. Ezert nehany rendszer megorizte ezt a kettosseget, mint a DOS, ill azon alapulo rendszerek, mig a unix-variansok csak egy karaktert hasznaltak a kettobol, es... hat, vagy ezt, vagy azt. ))

Szoval: használhatod a `socat` nevű programot is, de megmondom szintén azt nem tudom fejből hogy azzal hogyan megy a soros vonal összelövése. Viszont: a mellékelt programot szoktam én használni (hazi fejlesztés) amit csak simán `make`-val le tudsz fordítani. És akkor azzal így:

```
$ ./serial -c /dev/ttyACM1 -b 115200
```

es akkor mehet is a dolog es majd ctrl+c-vel megszakítod amikor "keszen vagy". A minap fent a hegyen is ezt használtuk, szóval ez visszaecho-zza azt amit beírtál és az újsor-karaktereket is önkonzisztensen kezeli (vagyis egyáltalán nem kezeli külön, emiatt nem különbözteti meg a különböző típusú entereket vagy soremeléseket, de nem is alakítja át, mint ahogy ezek szerint a putty teszi).

Probáld ki!

A.

On Sat, 21 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András! (legutóbb is elrontottam a sorrendet)

Van fejlemény! A Makefile-ban rossz helyek voltak megjelölve forrásnak (középső kép, MAIN\_LIBS sorok 32-34). Így már föltöltötte a kódot ami a csatolt kép bal oldalán látható.

A putty nevű programmal értem el a boardot (jobb oldali kép). Ugye a kód az arról szól, hogy ha nyomok egy gombot, akkor világít az LD2 led, ha pedig megnyomok egy újabb gombot, akkor pedig elalszik. Ez tökéletesen működik. Azonban putty-n keresztül valamiért csak nagybetűket tudok írni, valamint enter nyomás esetén a sor elejére ugrik. Elvileg el kéne tolnia a betűket az abc-ben, de ez nem igazán történik meg. Ennek lehet oka a putty? Emlékszem, hogy ön nem ezt használta, de nem emlékszem, mivel tudott kommunikálni vele.

Köszönöm a segítséget!  
K. Gábor

[image: image.png]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 20., P, 19:59):

Szia Gábor!

No, van barmi fejlemény? :) Pontosán milyen linuxot is hasznalsz? Ugy jobban

ra tudunk keresni az adott library (\*.a file) pontos helyére.

Ez tenyleg elegge melyviz, szóval absz ne ijedj meg tenyleg ha valami nem

megy kapasbol es tenyleg szolj ha elakadsz valahol!

A.

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Sajnos nem ez a gond, már telepítve van.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 19., Cs 21:33):

Tedd fel a libnewlib-arm-none-eabi nevű csomagot!

```
apt-get install libnewlib-arm-none-eabi
```

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Kedves András Pál!

Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül megoldanom, mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a boardra. Amit leírt a korábbiakban, mindent többször átfutottam és ellenőriztem, de kifog rajtam egyelőre a Linux.

El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a "st-info --probe" sort a terminálba. A headereket sikeresen bemásoltam a megfelelő helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a következő sorokat kapom a "make upload" esetén:

```
arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or
directory
make: *** [Makefile:54: main.elf] Error 1
```

Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt, ezért elnézést!

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 17., K, 20:36):

A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a board-ra, nem a main.c-t eteted meg kozvetlenul. Probald ki a `make upload` parancsot, az ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a Makefile bogaraszasabol rajossz hogy hogy is van ez pontosan.

Szoval a workflow az itten a kovetkezo:

- A \*.c fileokbol \*.o (object) fileokat forditunk (arm-none-eabi-gcc)
- Az egeszet összelinkeljük (arm-none-eabi-ld)
- Ez a tipusu mikrokontroller a 0x00000004-es címen található utasításra ugrik először, ott kezdi a program futtatását
- Ez a tipusu mikrokontroller olyan hogy a 0x00000000 tartományra a 0x08000000 tartományt kepezi le alapertelmezesben
- A linkelésnél így garantálnunk kell hogy legyen "valami" ami a 0x08000004-es címen ami értelmes \_és\_ kesobb meghivja a main() fuggvenyt
- merthogy minden C program a main()-nal "kezdodik". Na, ezt a feladatot latja el a crt.c - plusz még pár egyebet is, de azzal nem igazan kell foglalatostkodni, azzal majd kesobb ;)
- A linkeles utan ki kell szednunk a keletkezo \*.elf filebol a "lenyeget". Ezt csinalja a arm-none-eabi-objcopy, es ez hozza létre a \*.hex-et.
- Ezt a \*.hex-et pedig az openocd-vel beleirhatjuk a mikrokontroller flash-ébe

Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo sorrendben.

Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es az  
openocd  
konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani

neki

hogyan  
milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target  
mikrokontroller (azaz a programozónak tudnia kell hogy a programozási  
protokollon keresztül hogyan eri el a mikrokontroller memoriajat es ez  
fugghet  
a mikrokontroller konkret tipusától - bar egy csaladon belül ez elegge  
egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem azt hogy  
stm32f072cibt6). Plusz meg vannak egyeb finomsagok is, pl a flash-t

ugye

minden  
íras előtt torolni kell es csak utána írhat sz bele, stbstb. Ezeket  
mind-mind  
megcsinálja a Makefile-be elrejtett parancs- valami.

Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra ugyanigy  
megy,  
csak itt ez a "bare metal" modszer ezt is szuksegesse teszi hogy

lepesrol

lepesre menj(unk) vegig a lepeseken.

Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott  
hasznaltatok  
valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy hogy?

A.

On Tue, 17 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent  
telepítettem. Ami még nem sikerült, az az openocd életre bírása.

Csatolok

egy képet, és bízom benne, hogy tud segíteni.  
Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az

st-link

csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a  
feltöltésig  
nem jutottam el.

Köszönettel: K. Gábor  
[image: Screenshot from 2020-11-17 19-53-48.jpg]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 16., H,  
18:35):

Szia Gábor!

Ah, ertem, ez így akkor elsore tenyleg nem egyszerű. Viszont ennel  
specifikusabban kene tudni hogy hol is akadtál el. Tehat:

- telepites ment? Minden fent van?
- az arm-es toolchain az ezen csomagokbol all:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi
  - \* libnewlib-arm-none-eabi,
  - \* libstdc++-arm-none-eabi-newlib.
- ami meg hasznos csomag es kellhet:
  - \* build-essential (ebben van helyi C fordito is + a `make`

program)

- \* srecord
- openocd az csak siman ez:



\* openocd

Mindegyiket siman `apt-get install ...` modon tudod telepíteni! Ezek sikerültek? Ha ezek mind fent vannak, akkor az általam mellekelt

forrast

mar le  
tudod fordítani és fel kell tudnod tölteni a kis board-ra. Ami meg

kellhet

a  
forgatáshoz azok az STM32-specifikus headerek. Ezeket elküldöm  
mellekelve.

En  
konkretan sima rendszergazdaként az /usr/local/arm-none-eabi/include  
alkovnytar  
alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt  
teszprogramban  
levő Makefile-ban az elejen (kb 12-15ik sorban) levő

ARM=/usr/local/arm-none-eabi

sort modosítsd oda ahova betetted! Lehet siman nálad is, a /home-od  
alatt!

De  
ugyis panaszkodik a fordító hogyha nem találja ;) Szóval legjobb  
emlekeim

szerint ennyi kell, és ami meg hiányzik az ezeknek a csomagoknak a  
fuggósege -  
szóval jó kell hogy legyen, legjobb emlekeim szerint.

Az FPGA az sajnos tenyleg egy fokkal bonyolultabb. Elobb nezzük meg

ezt

az

ARM + STM32 témát, és akkor utána. Valójában azokat sem nehéz  
feltenni,  
de

kellhet pár olyan development header (pl libboost-csomag) amiket  
feltetelez  
hogy fent van de alapból szinte biztos hogy nincs fent nálad. Nalam  
konkretan  
ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-filesystem-dev:amd64	install
libboost-filesystem1.62-dev:amd64	install
libboost-filesystem1.62.0:amd64	install
libboost-iostreams-dev:amd64	install
libboost-iostreams1.62-dev:amd64	install
libboost-iostreams1.62.0:amd64	install
libboost-program-options-dev:amd64	install
libboost-program-options1.62-dev:amd64	install
libboost-program-options1.62.0:amd64	install
libboost-python-dev	install
libboost-python1.62-dev	install
libboost-python1.62.0	install
libboost-regex1.62-dev:amd64	install
libboost-regex1.62.0:amd64	install
libboost-serialization1.62-dev:amd64	install
libboost-serialization1.62.0:amd64	install
libboost-system1.62-dev:amd64	install
libboost-system1.62.0:amd64	install

```
libboost-thread-dev:amd64          install
libboost-thread1.62-dev:amd64      install
libboost-thread1.62.0:amd64        install
libboost1.62-dev:amd64             install
```

szoval ha azt monddod hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev
libboost-chrono-dev libboost-date-time-dev libboost-dev
libboost-filesystem-dev libboost-iostreams-dev
libboost-program-options-dev
libboost-python-dev libboost-regex-dev libboost-serialization-dev
libboost-system-dev libboost-thread-dev
```

(igen, ez így egy jó hosszú parancssor :)), akkor mindent is feltesz

ami

kell a  
libboosthoz (es a sima lib-eket is felteszi ami kell a \*-dev  
csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De tenyleg

ez

kicsit  
macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi az

ami

hiányzik. Hanyas verzioju milyen linuxot is tettél fel pontosan?

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

küldött

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a  
dolgozat olvasgattam. Sajnos sem az openocd-t, sem az icestorm  
toolchain-t  
nem sikerült "életre keltenem". Elvileg a yosys-0.9, nextpnr-ice40

és

esetleg

icestorm-utils csomagokat is telepítettem a megadott módon. Tud  
küldeni valamilyen segítséget, hogyan tudnám programozni ezekentől  
a  
lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék

segítséget

kérni,  
telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam  
kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom  
elérni.

de a

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal),  
programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai  
eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis

Egyetem

TDK

mellett készülhetünk az országos TDK-ra. Mivel már kész van az

eszköz,

és

csak bemutatni kell, így nem vesz el jelentős időt.

Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet.

Egy

két

szabadsági fokból álló állványon levő parabola-antenna fogja  
 "bescannelni"  
 az eget, és a vett jelerősséget fogja egy skála szerint  
 osztályozni,  
 majd  
 ezeket a parabola helyzetéből egy képen megfelelő helyre  
 illeszteni.  
 Fő  
 célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell  
 érzékelnünk.  
 Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő  
 érzékelővel.  
 Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg  
 valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb  
 lenne)  
 csinálni?  
 Köszönettel: K. Gábor  
 Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo,  
 16:02):  
 Szia Gábor!  
 Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper  
 lesz,  
 azon  
 nagyjából mindent is meg lehet csinálni amit debian-on, es hogyha  
 standalone  
 es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!  
 Headless  
 üzemben mar nem használnék egyatalan ubuntu-t, de az mas kerdes ;)  
 Linux alá viszont van teljesen free & open souce Lattice FPGA  
 toolchain:  
<http://www.clifford.at/icestorm/>, ezt erdemes lehet  
 feltelepíteni.  
 Mi  
 is  
 ezekkel a  
 csomagokkal kozvetlenul is tudod linux alatt mindenfele  
 licenc-problema  
 meg  
 nyűg nélkül hasznalni a barmit! Cserebe a forras(kod)bol valo  
 telepites  
 nem  
 annyira egyszerű, eleg sok a függőség, es nehezen derül ki hogy mi  
 az.  
 De  
 ebben  
 viszont a windozos cuccokkal ellentétben tudok segiteni :)  
 Ugyanakkor  
 termesztesen en is neztem a linuxos icecube2-t, es az is mukodik  
 persze,  
 szoval  
 hajra azzal is. Mondjuk az nekem kicsit nehezkes, a "bare metal"  
 meg a  
 klasszikus "unix filozofia" dolgok utan nekem ezeket az IDE-eket  
 eleg  
 nehez  
 megszoknom :/ De az mas kerdes ;)  
 A.  
 On Sat, 7 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos tanulmányozásának.

Egy közeli ismerősöm ajánlásából elkezdtuk föltelepíteni az ubuntu-t, mivel a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem volt telepíteni zökkenőmentes a telepítés. Először a pendrive amin keresztül akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt föl.

Majd miután sikerült, a Grub nem települt így nem tudtam elindítani a rendszert. Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és egyebek. Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált adataimat el kell fogadják, de az is hamarosan meglesz. A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött pdf-eket át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni. Szerencsére az is teljesen témába vág, hiszen C programozás lesz. Nagyon le voltam nyugózva a találkozókör és remélem minden

rendben lesz!

Köszönettel:  
K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6., P, 9:44):

Szia Gábor!

Persze, kuldom akkor az infokat:  
- mellekelve az evaluation board doksija, ebben benne kell

legyen minden referencia ami hasznos lehet - bar ugye a mi boardunk az masmilyen.  
- a fejlesztokeszlet az ez:

<https://www.latticesemi.com/iCEcube2>,

van minden oprendszer ala (linux, windoz).  
- az FPGA az ez: iCE40HX8K-BG121, ezt kell kivlasztanod majd.  
- mellekelem a tegnap tesztelt kis verillog-kodot is, abban

megtalalad a PCF-et: ezt a PCF-et megeszi az icecube2 is! (a verillog-ot meg nyilván)

Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:  
- mellekelem a tegnapi peldaprogramot (UART: kisbetu ->

nagybetu)  
- a toolchain (ami a \*.c fileokbol megcsinalja a \*.bin es/vagy \*.hex

image-t, amit aztan fel kell tolteni) az `arm-none-eabi` néven fut. Ennek tobb

komponense van:

\* A C/C++/asm fordító maga: ez a gcc-arm-none-eabi  
 \* Eszközök amikkel a lefordított fileokat lehet manipulálni (pl a \*.hex-et eloallítani a \*.o/\*.elf-ből):  
 binutils-arm-none-eabi  
 \* Es az előre lefordított könyvtárak, hogy a jól megszokott C-s (stdio: printf, stb) meg C++-os (STL és társai) dolgokat tudd használni: libnewlib-arm-none-eabi, libstdc++-arm-none-eabi-newlib.  
 Ez kb ugyanilyen néven fut windoz alatt is, egy gyors kereséssel pl ezt találtam:

<https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>

A lényeg hogy ez a "gcc-arm-none-eabi for windows" dollogra keress rá!  
 A programozó az ST-Link V2 (-1), ezt lehet OpenOCD-vel programozni.  
 Az OpenOCD van linux alá is, de van windozos prortja is:  
<https://gnutoolchains.com/arm-eabi/openocd/>. Ettől függetlenül az ST saját cuccait is lehet használni valószínűleg:  
<https://www.st.com/en/development-tools/stsw-link009.html>, de az OpenOCD annyira jó hogy windozt használó kollegáktól is azt hallom hogy inkább az OpenOCD-t preferálják az STM32-es MCU-khoz.  
 A.  
 On Fri, 6 Nov 2020, Gábor Kreinicker wrote:  
 Kedves Pál András!  
 Említett nekem segédleteket a lattice boardhoz. Továbbá említette, hogy az fpga chip nem stimmel a gyárral, hogy azzal kapcsolatban is szeretnék kérni segítséget.  
 Köszönettel: K. Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta

(időpont: 2020. nov. 4., Sze, 17:28):

Rendben!

Szerintem én is megebédelek és indulok

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 17:09):

Szia Gábor!

Megebédelek odalent, kb olyan fel 1-1-ig bezarolag, es akkor utana indulok fel. Szoval fel 2 korultól mar leszek itt. Ha errefele jarsz akkor hivj fel: (20)3962151, es akkor kimegyek a kapuhoz. Merthogy mi itten egy masik epulet masik szekciojaban vagyunk. udv, A.

On Wed, 4 Nov 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

A holnap délután részemről szuper! Mikor tájtra számítsak?

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 4., Sze 16:45):

Szia Gábor!

No, tul vagy kozben a mindenen? Illetve a het hetralevo resze akkor mikor lenne jo? En holnap delutan tervezek feljonni a normafára elso kozelitesben.

A.

On Fri, 30 Oct 2020, Gábor Kreinicker wrote:

Tisztelt Pál András!

Tud nekem ajánlani egy Windowsos verilog szimulátort gyakorlás céljából?

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta

(időpont: 2020. okt. 29., Cs 9:01):

Rendben, az nekem is most előnyös lesz.  
TDK-t ma beadjuk, onnantól lényegesen egyszerűbb lesz

minden, a hetedik hét pedig a negyedéves hajrá, annak is vége lesz a jövő hét elején.

Ma megyek Normafára a demonstrátori dolgokat aláírni, így ha minden igaz, az is meglesz.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. okt. 28., Sze, 18:44):

Oke! Akkor viszont lehet hogy diskuráljunk megiscsak

jovo heten, mert közben engem is nagyon bekapott az ekszij es/vagy a daralo a műhold kapcsan.

Most mar



serial.tgz

9K

**Gábor Kreinicker** <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)>  
Címzett: Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)>

2020. november 21. 12:15

Ááá, igen, ezt láttam fönt is, működik, köszönöm!

Akkor igazából kezdhetek játszani. Egy útmutatást tud adni, mit érdemes gyakorolni? Vagy tud adni feladatot?

Köszönöm még egyszer!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 21., Szo, 11:36):  
Szia Gábor!

Remek! Jaj de jó!

Nade igen, a soros portos eleres nem annyira magatol ertetodo, es igen, lehet hogy a putty az mas újsor-karakter konvenciot hasznal, raadasul nem is echo-zza vissza vsz a terminalra a karaktereket (igy valojaban amit begepelsz azt nem latod mert nem a kepernyodre írja hanem kikuldi a soros porton, es csak azt latod amit visszakuld: es az valoban mar a nagybetűs változat mert ugye azt kuldi vissza a board!).

Az újsor-karakterek minden oprendszerben masok, raadasul a három legelterjedtebben (linux, mac/osx, windoz) is mind kulonbozoker (rendre 0A, 0D, 0D+0A), lasd: <https://en.wikipedia.org/wiki/Newline#Representation>, es jo

kerdes hogy a puttynál mi az alapertelmezett... ez is bekavarhat. (( Ennek meg regi tortente van es a klasszikus mechanikus irogepek koraig nyulik vissza, mert ott még nem volt a klasszikus "újsor", hanem csak a "kocsi vissza" meg a "soremelés" - hiszen mikor irogeppel irtal es kezdtel ujsort akkor a papirt is hajtanod kellett ("feed"-elni a hengert), plusz a hengert (carriage) magát visszaküldeni az alaphelyzetebe. Ezert nehany rendszer megorizte ezt a kettosseget, mint a DOS, ill azon alapulo rendszerek, mig a unix-variantsok csak egy karaktert hasznaltak a kettobol, es... hat, vagy ezt, vagy azt. ))

Szoval: hasznalhatod a `socat` nevu programot is, de megmondom oszinten azt nem tudom fejbol hogy azzal hogyan megy a soros vonal osszelövése. Viszont: a mellekelt programot szoktam en hasznalni (hazi fejlesztés) amit csak siman `make`-val le tudsz fordítani. Es akkor azzal imigyen:

```
$ ./serial -c /dev/ttyACM1 -b 115200
```

es akkor mehet is a dolog es majd ctrl+c-vel megszakítod amikor "keszen vagy". A minap fent a hegyen is ezt hasznaltuk, szoval ez visszaecho-zza azt amit beirtal es az ujsor-karaktereket is önkonzisztensen kezeli (vagyis egyatalan nem kezeli külön, emiatt nem különbözteti meg a különböző típusú entereket vagy soremelesekét, de nem is alakítja át, mint ahogy ezek szerint a putty teszi).

Probald ki!

A.

On Sat, 21 Nov 2020, Gábor Kreinicker wrote:

```
> Kedves Pál András! (legutóbb is elrontottam a sorrendet)
>
> Van fejlemény! A Makefile-ban rossz helyek voltak megjelölve forrásnak
> (középső kép, MAIN_LIBS sorok 32-34). Így már föltöltötte a kódot ami a
> csatolt kép bal oldalán látható.
>
> A putty nevű programmal értem el a boardot (jobb oldali kép). Ugye a kód az
> arról szól, hogy ha nyomok egy gombot, akkor világít az LD2 led, ha pedig
> megnyomok egy újabb gombot, akkor pedig elalszik. Ez tökéletesen működik.
> Azonban putty-n keresztül valamiért csak nagybetűket tudok írni, valamint
> enter nyomás esetén a sor elejére ugrik. Elvileg el kéne tolnia a betűket
> az abc-ben, de ez nem igazán történik meg. Ennek lehet oka a putty?
> Emlékszem, hogy ön nem ezt használta, de nem emlékszem, mivel tudott
> kommunikálni vele.
>
> Köszönöm a segítséget!
> K. Gábor
>
> [image: image.png]
>
>
> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 20., P, 19:59):
>
>> Szia Gábor!
>> No, van barmi fejlemeny? :) Pontosan milyen linuxot is hasznalsz? Ugy
>> jobban
>> ra tudunk keresni az adott library (*.a file) pontos helyére.
>> Ez tenyleg elegge melyviz, szoval absz ne ijedj meg tenyleg ha valami
>> nem
>> megy kapasbol es tenyleg szolj ha elakadsz valahol!
>> A.
>>
>> On Thu, 19 Nov 2020, Gábor Kreinicker wrote:
>>
>>> Sajnos nem ez a gond, már telepítve van.
>>>
>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 19., Cs
>> 21:33):
```



```
>>>
>>>> Tedd fel a libnewlib-arm-none-eabi nevű csomagot!
>>>>
>>>> apt-get install libnewlib-arm-none-eabi
>>>>
>>>> On Thu, 19 Nov 2020, Gábor Kreinicker wrote:
>>>>
>>>>> Kedves András Pál!
>>>>>
>>>>> Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül
>>>> megoldanom,
>>>>> mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a
>>>> boardra. Amit leírt a korábbiakban, mindent többször átfutottam és
>>>>> ellenőriztem, de kifog rajtam egyelőre a Linux.
>>>>> El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a "st-info
>>>>> --probe" sort a terminálba. A headereket sikeresen bemásoltam a
>> megfelelő
>>>>> helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a következő
>>>>> sorokat kapom a "make upload" esetén:
>>>>>
>>>>> arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
>>>>> stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
>>>>> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
>>>>> arm-none-eabi-ld: cannot find
>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or
>>>> directory
>>>>> arm-none-eabi-ld: cannot find
>>>>> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or
>>>> directory
>>>>> arm-none-eabi-ld: cannot find
>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or
>>>> directory
>>>>> make: *** [Makefile:54: main.elf] Error 1
>>>>>
>>>>> Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt,
>>>> ezért
>>>>> elnézést!
>>>>>
>>>>> Köszönettel: K. Gábor
>>>>>
>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 17., K,
>>>> 20:36):
>>>>>
>>>>>> A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a
>> board-ra,
>>>>>> nem a
>>>>>> main.c-t eteted meg kozvetlenul. Probald ki a `make upload` parancsot,
>>>> az
>>>>>> az
>>>>>> ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a
>> Makefile
>>>>>> bogaraszasabol rajossz hogy hogy is van ez pontosan.
>>>>>>
>>>>>> Szoval a workflow az itten a kovetkezo:
>>>>>> - A *.c fileokbol *.o (object) fileokat forditunk
>> (arm-none-eabi-gcc)
>>>>>> - Az egeszet osszelinkeljuk (arm-none-eabi-ld)
>>>>>> - Ez a tipusu mikrokontroller a 0x000000004-es címen található
>>>>>> utasításra
>>>>>> ugrik először, ott kezdi a program futtatását
>>>>>> - Ez a tipusu mikrokontroller olyan hogy a 0x000000000 tartomanyra a
>>>>>> 0x08000000 tartományt kepezi le alapertelmezesben
>>>>>> - A linkelésnél így garantálnunk kell hogy legyen "valami" ami a
>>>>>> 0x080000004-es címen ami értelmes _és_ kesobb meghivja a main()
>>>> fuggvenyt
```

```
>>>>>> -
>>>>>> merthogy minden C program a main()-nal "kezdodik". Na, ezt a feladatot
>>>>>> latja el
>>>>>> a crt.c - plusz még pár egyebet is, de azzal nem igazan kell
>>>>>> foglalatostkodni,
>>>>>> azzal majd kesobb ;)
>>>>>> - A linkeles utan ki kell szednunk a keletkezo *.elf filebol a
>>>>>> "lenyeget". Ezt
>>>>>> csinalja a arm-none-eabi-objcopy, es ez hozza létre a *.hex-et.
>>>>>> - Ezt a *.hex-et pedig az openocd-vel beleirhatjuk a mikrokontroller
>>>>>> flash-ebe
>>>>>>
>>>>>> Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo
>>>>>> sorrendben.
>>>>>> Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es az
>>>>>> openocd
>>>>>> konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani
>> neki
>>>>>> hogy
>>>>>> milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target
>>>>>> mikrokontroller (azaz a programozónak tudnia kell hogy a programozasi
>>>>>> protokollon keresztul hogyan eri el a mikrokontroller memoriajat es ez
>>>>>> fugghet
>>>>>> a mikrokontroller konkret tipusától - bar egy csaladon belül ez elegge
>>>>>> egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem azt hogy
>>>>>> stm32f072cibt6). Plusz meg vannak egyeb finomsagok is, pl a flash-t
>> ugye
>>>>>> minden
>>>>>> iras előtt torolni kell es csak utána írhat sz bele, stbstb. Ezeket
>>>>>> mind-mind
>>>>>> megcsinalja a Makefile-be elrejtett parancs- valami.
>>>>>>
>>>>>> Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra ugyanigy
>>>>>> megy,
>>>>>> csak itt ez a "bare metal" modszert is szuksegesse teszi hogy
>>>>>> lepesrol
>>>>>> lepesre menj(unk) vegig a lepeseken.
>>>>>>
>>>>>> Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott
>>>>>> hasznaltatok
>>>>>> valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy hogy?
>>>>>>
>>>>>> A.
>>>>>>
>>>>>> On Tue, 17 Nov 2020, Gábor Kreinicker wrote:
>>>>>>
>>>>>> Kedves Pál András!
>>>>>>
>>>>>> Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent
>>>>>> telepítettem. Ami még nem sikerült, az az openocd életre bírása.
>>>>>> Csatolok
>>>>>> egy képet, és bízom benne, hogy tud segíteni.
>>>>>> Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az
>>>>>> st-link
>>>>>> csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a
>>>>>> feltöltésig
>>>>>> nem jutottam el.
>>>>>>
>>>>>> Köszönettel: K. Gábor
>>>>>> [image: Screenshot from 2020-11-17 19-53-48.jpg]
>>>>>>
>>>>>>
>>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 16., H,
>>>>>> 18:35):
>>>>>>
>>>>>>> Szia Gábor!
```

```
>>>>>>>
>>>>>>> Ah, ertem, ez igy akkor elsore tenyleg nem egyszeru. Viszont ennél
>>>>>>> specifikusabban kene tudni hogy hol is akadtál el. Tehat:
>>>>>>> - telepites ment? Minden fent van?
>>>>>>> - az arm-es toolchain az ezen csomagokbol all:
>>>>>>>   * gcc-arm-none-eabi
>>>>>>>   * binutils-arm-none-eabi
>>>>>>>   * libnewlib-arm-none-eabi,
>>>>>>>   * libstdc++-arm-none-eabi-newlib.
>>>>>>> - ami meg hasznos csomag es kellhet:
>>>>>>>   * build-essential (ebben van helyi C fordito is + a `make`
>>>>>>> program)
>>>>>>>   * srecord
>>>>>>> - openocd az csak siman ez:
>>>>>>>   * openocd
>>>>>>>
>>>>>>> Mindegyiket siman `apt-get install ...` modon tudod telepiteni! Ezek
>>>>>>> sikerultek? Ha ezek mind fent vannak, akkor az altalam mellekelt
>>>> forrast
>>>>>>> mar le
>>>>>>> tudod fordítani es fel kell tudnod tölteni a kis board-ra. Ami meg
>>>>>>> kellhet
>>>>>>> a
>>>>>>> forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom
>>>>>>> mellekelve.
>>>>>>> En
>>>>>>> konkretan sima rendszergazdakent az /usr/local/arm-none-eabi/include
>>>>>>> alkovnytar
>>>>>>> alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt
>>>>>>> teszprogramban
>>>>>>> levő Makefile-ban az elejen (kb 12-15ik sorban) levő
>>>>>>>
>>>>>>> ARM=/usr/local/arm-none-eabi
>>>>>>>
>>>>>>> sort modosítsd oda ahova betetted! Lehet siman nálad is, a /home-od
>>>>>>> alatt!
>>>>>>> De
>>>>>>> ugyis panaszkodik a fordito hogyha nem találja ;) Szoval legjobb
>>>>>>> emlekeim
>>>>>>> szerint ennyi kell, es ami meg hiányzik az ezeknek a csomagoknak a
>>>>>>> fuggosege -
>>>>>>> szoval jo kell hogy legyen, legjobb emlekeim szerint.
>>>>>>>
>>>>>>> Az FPGA az sajnos tenyleg egy fokkal bonyolultabb. Elobb nezzuk meg
>>>> ezt
>>>>>>> az
>>>>>>> ARM + STM32 témát, es akkor utana. Valojaban azokat sem nehez
>>>> feltenni,
>>>>>>> de
>>>>>>> kellhet par olyan development header (pl libboost-csomag) amiket
>>>>>>> feltetelez
>>>>>>> hogy fent van de alapbol szinte biztos hogy nincs fent nálad. Nalam
>>>>>>> konkretan
>>>>>>> ezek vannak fent:
>>>>>>>
>>>>>>> libboost-atomic1.62-dev:amd64          install
>>>>>>> libboost-atomic1.62.0:amd64             install
>>>>>>> libboost-chrono1.62-dev:amd64           install
>>>>>>> libboost-chrono1.62.0:amd64             install
>>>>>>> libboost-date-time1.62-dev:amd64        install
>>>>>>> libboost-date-time1.62.0:amd64          install
>>>>>>> libboost-dev:amd64                      install
>>>>>>> libboost-filesystem-dev:amd64           install
>>>>>>> libboost-filesystem1.62-dev:amd64       install
>>>>>>> libboost-filesystem1.62.0:amd64         install
>>>>>>> libboost-iostreams-dev:amd64            install
```

```
>>>>>>> libboost-iostreams1.62-dev:amd64      install
>>>>>>> libboost-iostreams1.62.0:amd64        install
>>>>>>> libboost-program-options-dev:amd64      install
>>>>>>> libboost-program-options1.62-dev:amd64  install
>>>>>>> libboost-program-options1.62.0:amd64    install
>>>>>>> libboost-python-dev                    install
>>>>>>> libboost-python1.62-dev                 install
>>>>>>> libboost-python1.62.0                   install
>>>>>>> libboost-regex1.62-dev:amd64            install
>>>>>>> libboost-regex1.62.0:amd64              install
>>>>>>> libboost-serialization1.62-dev:amd64    install
>>>>>>> libboost-serialization1.62.0:amd64      install
>>>>>>> libboost-system1.62-dev:amd64           install
>>>>>>> libboost-system1.62.0:amd64             install
>>>>>>> libboost-thread-dev:amd64               install
>>>>>>> libboost-thread1.62-dev:amd64           install
>>>>>>> libboost-thread1.62.0:amd64             install
>>>>>>> libboost1.62-dev:amd64                  install
>>>>>>>
>>>>>>> szoval ha azt mondod hogy
>>>>>>>
>>>>>>> # apt-get --no-install-recommends install libboost-atomic-dev
>>>>>>> libboost-chrono-dev libboost-date-time-dev libboost-dev
>>>>>>> libboost-filesystem-dev libboost-iostreams-dev
>>>>>>> libboost-program-options-dev
>>>>>>> libboost-python-dev libboost-regex-dev libboost-serialization-dev
>>>>>>> libboost-system-dev libboost-thread-dev
>>>>>>>
>>>>>>> (igen, ez így egy jó hosszú parancssor :)), akkor mindent is feltesz
>>>> ami
>>>>>>> kell a
>>>>>>> libboosthoz (es a sima lib-eket is felteszi ami kell a *-dev
>>>>>>> csomagokhoz)
>>>>>>>
>>>>>>> Ami meg szinte biztos hogy kell az a: flex bison libtool. De tenyleg
>>>> ez
>>>>>>> kicsit
>>>>>>> macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi az
>>>> ami
>>>>>>> hiányzik. Hanyas verzioju milyen linuxot is tettél fel pontosan?
>>>>>>>
>>>>>>> A.
>>>>>>>
>>>>>>> On Mon, 16 Nov 2020, Gábor Kreinicker wrote:
>>>>>>>
>>>>>>> Kedves Pál András!
>>>>>>>
>>>>>>> Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a
>>>>>>> küldött
>>>>>>> dolgokat olvastam. Sajnos sem az openocd-t, sem az icestorm
>>>>>>> toolchain-t
>>>>>>> nem sikerült "életre keltenem". Elvileg a yosys-0.9, nextpnr-ice40
>> és
>>>>>>> icestorm-utils csomagokat is telepítettem a megadott módon. Tud
>>>> esetleg
>>>>>>> küldeni valamilyen segítséget, hogyan tudnám programozni ezeketül
>> a
>>>>>>> lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék
>> segítséget
>>>>>>> kérni,
>>>>>>> telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam
>>>>>>> kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom
>>>>>>> elérni.
>>>>>>>
>>>>>>> Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal),
>> de a
```

>>>>>>> programozásukig sajnos még nem jutottam el.  
>>>>>>>  
>>>>>>> Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai  
>>>>>>> eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis  
>>>> Egyetem  
>>>>>>> TDK  
>>>>>>> mellett készülhetünk az országos TDK-ra. Mivel már kész van az  
>>>> eszköz,  
>>>>>>> és  
>>>>>>> csak bemutatni kell, így nem vesz el jelentős időt.  
>>>>>>>  
>>>>>>> Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet.  
>> Egy  
>>>>>>> két  
>>>>>>> szabadsági fokból álló állványon levő parabola-antenna fogja  
>>>>>>> "bescannelni"  
>>>>>>> az eget, és a vett jelerősséget fogja egy skála szerint  
>> osztályozni,  
>>>>>>> majd  
>>>>>>> ezeket a parabola helyzetéből egy képen megfelelő helyre  
>> illeszteni.  
>>>> Fő  
>>>>>>> célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell  
>>>>>>> érzékelnünk.  
>>>>>>> Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő  
>>>>>>> érzékelővel.  
>>>>>>> Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg  
>>>>>>> valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb  
>>>>>>> lenne)  
>>>>>>> csinálni?  
>>>>>>>  
>>>>>>> Köszönettel: K. Gábor  
>>>>>>>  
>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo,  
>>>>>>> 16:02):  
>>>>>>>  
>>>>>>> Szia Gábor!  
>>>>>>>  
>>>>>>>> Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper  
>>>>>>> lesz,  
>>>>>>>> azon  
>>>>>>>> nagyjából mindent is meg lehet csinálni amit debian-on, es hogyha  
>>>>>>>> standalone  
>>>>>>>> es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!  
>>>>>>> Headless  
>>>>>>>> üzemben mar nem használnék egyatalan ubuntut, de az mas kerdes ;)  
>>>>>>>>  
>>>>>>>> Linux alá viszont van teljesen free & open souce Lattice FPGA  
>>>>>>> toolchain:  
>>>>>>>> <http://www.clifford.at/icestorm/>, ezt erdemes lehet  
>> feltelepíteni.  
>>>> Mi  
>>>>>>> is  
>>>>>>>> hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icestorm-utils),  
>> ezekkel a  
>>>>>>>> csomagokkal kozvetlenul is tudod linux alatt mindenfele  
>>>>>>> licenc-problema  
>>>>>>>> meg  
>>>>>>>> nyűg nélkül használni a barmit! Cserebe a forras(kod)bol valo  
>>>>>>> telepites  
>>>>>>>> nem  
>>>>>>>> annyira egyszerű, eleg sok a függőség, es nehezen derül ki hogy mi  
>>>> az.  
>>>>>>> De  
>>>>>>>> ebben  
>>>>>>>> viszont a windozos cuccokkal ellentétben tudok segíteni :)

```

>>>> Ugyanakkor
>>>>>>>> természetesen én is néztem a linuxos icecube2-t, és az is működik
>>>>>>>> persze,
>>>>>>>> szóval
>>>>>>>> hajra azzal is. Mondjuk az nekem kicsit nehezkés, a "bare metal"
>>>> meg a
>>>>>>>> klasszikus "unix filozofia" dolgok után nekem ezeket az IDE-eket
>> elég
>>>>>>>> nehéz
>>>>>>>> megszoknom :/ De az más kérdés ;)
>>>>>>>>
>>>>>>>> A.
>>>>>>>>
>>>>>>>> On Sat, 7 Nov 2020, Gábor Kreinicker wrote:
>>>>>>>>
>>>>>>>> Kedves Pál András!
>>>>>>>>
>>>>>>>> Köszönök mindent, ma hozzá is állok a dolgok alapos
>>>>>>>> tanulmányozásának.
>>>>>>>> Egy közeli ismerősöm ajánlásából elkezdtek föltelepíteni az
>>>> ubuntu-t,
>>>>>>>> mivel
>>>>>>>> a kollégiumban ha problémám van azonnal tud segíteni. Sajnos nem
>>>> volt
>>>>>>>> zökkenőmentes a telepítés. Először a pendrive amin keresztül
>>>>>>>> telepíteni
>>>>>>>> akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt
>>>> föl.
>>>>>>>> Majd
>>>>>>>> miután sikerült, a Grub nem települt így nem tudtam elindítani a
>>>>>>>> rendszert.
>>>>>>>> Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és
>>>>>>>> egyebek.
>>>>>>>> Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált
>>>>>>>> adataimat
>>>>>>>> el
>>>>>>>> kell fogadják, de az is hamarosan meglesz.
>>>>>>>> A hétvégén szerintem gyakorlatban ennyit tudok haladni, a küldött
>>>>>>>> pdf-eket
>>>>>>>> át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.
>>>>>>>> Szerencsére
>>>>>>>> az is teljesen témába vág, hiszen C programozás lesz.
>>>>>>>> Nagyon le voltam nyűgözve a találkozókör és remélem minden
>> rendben
>>>>>>>> lesz!
>>>>>>>>
>>>>>>>> Köszönettel:
>>>>>>>> K. Gábor
>>>>>>>>
>>>>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 6., P,
>>>>>>>> 9:44):
>>>>>>>>
>>>>>>>>>>>> Szia Gábor!
>>>>>>>>>>>>
>>>>>>>>>>>> Persze, küldöm akkor az infokat:
>>>>>>>>>>>> - mellekelve az evaluation board doksija, ebben benne kell
>>>> legyen
>>>>>>>>>>>> minden
>>>>>>>>>>>> referencia ami hasznos lehet - bár ugye a mi boardunk az
>>>> masmilyen.
>>>>>>>>>>>> - a fejlesztőkészlet az ez:
>>>> https://www.latticesemi.com/iCEcube2,
>>>>>>>>>>>> van
>>>>>>>>>>>> minden
>>>>>>>>>>>> operációs rendszer alá (linux, windows).
>>>>>>>>>>>> - az FPGA az ez: iCE40HX8K-BG121, ezt kell kiválasztanod majd.

```

```
>>>>>>>>>> - mellekelem a tegnap tesztelt kis verilog-kodot is, abban
>>>>>>>>>> megtalalaod a
>>>>>>>>>> PCF-et: ezt a PCF-et megeszi az icecube2 is! (a verilog-ot meg
>>>>>>>>>> nyilván)
>>>>>>>>>>
>>>>>>>>>> Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:
>>>>>>>>>> - mellekelem a tegnapi peldaprogramot (UART: kisbetu ->
>>>> nagybetu)
>>>>>>>>>> - a toolchain (ami a *.c fileokbol megcsinalja a *.bin es/vagy
>>>>>> *.hex
>>>>>>>>>> image-t,
>>>>>>>>>> amit aztan fel kell tolteni) az `arm-none-eabi` néven fut. Ennek
>>>>>>>>>> tobb
>>>>>>>>>> komponense van:
>>>>>>>>>> * A C/C++/asm fordito maga: ez a gcc-arm-none-eabi
>>>>>>>>>> * Eszkozok amikkel a leforditott fileokat lehet
>>>> manipulalni
>>>>>>>> (pl
>>>>>>>>>> a
>>>>>>>>>> *.hex-et eloallitani a *.o/*.elf-ből):
>>>>>>>>>> binutils-arm-none-eabi
>>>>>>>>>> * Es az elore leforditott konyvtaraok, hogy a jól
>>>> megszokott
>>>>>>>>>> C-s
>>>>>>>>>> (stdio: printf, stb) meg C++-os (STL es tarsai)
>> dolgokat
>>>>>>>>>> tudd
>>>>>>>>>> hasznalni: libnewlib-arm-none-eabi,
>>>>>>>>>> libstdc++-arm-none-eabi-newlib.
>>>>>>>>>>
>>>>>>>>>> Ez kb ugyanilyen néven fut windoz alatt is, egy gyors keresessel
>>>> pl
>>>>>>>>>> ezt
>>>>>>>>>> talaltam:
>>>>>>>>>>
>>>>>>>>>>
>>>>>>>>>>
>>>>>>>>>>
>> https://marketplace.visualstudio.com/items?itemName=metacode-eu.windows-arm-none-eabi
>>>>>>>>>> .
>>>>>>>>>>
>>>>>>>>>> A lenyeg hogy ez a "gcc-arm-none-eabi for windows" dolologra
>>>> keress
>>>>>>>>>> rá!
>>>>>>>>>>
>>>>>>>>>> A programozo az ST-Link V2 (-1), ezt lehet OpenOCD-vel
>>>> progmaozni.
>>>>>>>>>> Az
>>>>>>>>>> OpenOCD
>>>>>>>>>> van linux ala is, de van windozos prortja is:
>>>>>>>>>> https://gnutoolchains.com/arm-eabi/openocd/. Ettol fuggetlenul
>> az
>>>>>>>>>> ST
>>>>>>>>>> saját
>>>>>>>>>> cuccait is lehet hasznalni valoszinuleg:
>>>>>>>>>> https://www.st.com/en/development-tools/stsw-link009.html, de
>> az
>>>>>>>>>> OpenOCD
>>>>>>>>>> annyira jo hogy windozt hasznalo kollegaktol is azt hallom hogy
>>>>>>>>>> inkabb
>>>>>>>>>> az
>>>>>>>>>> OpenOCD-t preferaljak az STM32-es MCU-khoz.
>>>>>>>>>>
>>>>>>>>>> A.
```

[illegible]



[illegible]



[illegible]





[illegible]



















Akkor igazából kezdhettek játszani. Egy útmutatást tud adni, mit érdemes gyakorolni? Vagy tud adni feladatot?

Köszönöm még egyszer!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 21., Szo, 11:36):

Szia Gábor!

Remek! Jaj de jó!

Nade igen, a soros portos eleres nem annyira magától értetődő, és igen, lehet hogy a putty az más újsor-karakter konvenciót használ, ráadásul nem is echo-zza vissza vsz a terminalra a karaktereket (így valóban amit begépelsz azt nem látod mert nem a képernyőre írja hanem kiküldi a soros porton, és csak azt látod amit visszaküld: és az valóban már a nagybetűs változat mert ugye azt küldi vissza a board!).

Az újsor-karakterek minden operációs rendszerben mások, ráadásul a három legelterjedtebben (linux, mac/osx, windows) is mind különbözőek (rendre OA, OD, OD+OA), lásd: <https://en.wikipedia.org/wiki/Newline#Representation>, és jó kérdés hogy a putty-nál mi az alapértelmezett... ez is bekavarhat. (( Ennek meg régi története van és a klasszikus mechanikus írógépek koráig nyúlik vissza, mert ott még nem volt a klasszikus "újsor", hanem csak a "kocsi vissza" meg a "soremelés" - hiszen mikor írógéppel írtál és kezdtél újsort akkor a papírt is hajtani kellett ("feed"-elni a hengert), plusz a hengert (carriage) magát visszaküldeni az alaphelyzetébe. Ezért néhány rendszer megőrizte ezt a kettőseget, mint a DOS, ill azon alapuló rendszerek, míg a unix-variantok csak egy karaktert használtak a kettőből, és... hat, vagy ezt, vagy azt. ))

Szóval: használható a `socat` nevű programot is, de megmondom őszintén azt nem tudom fejből hogy azzal hogyan megy a soros vonal összelövése. Viszont: a mellékelt programot szoktam én használni (házi fejlesztés) amit csak simán `make`-val le tudsz fordítani. És akkor azzal imígyen:

```
$ ./serial -c /dev/ttyACM1 -b 115200
```

és akkor mehet is a dolog és majd ctrl+c-vel megszakítod amikor "keszen vagy". A minap fent a hégyen is ezt használtuk, szóval ez visszaecho-zza azt amit beírtál és az újsor-karaktereket is önkonzisztensen kezeli (vagyis egyáltalán nem kezeli külön, emiatt nem különbözteti meg a különböző típusú entereket vagy soremeléseket, de nem is alakítja át, mint ahogy ezekszerint a putty teszi).

Probáld ki!

A.

On Sat, 21 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András! (legutóbb is elrontottam a sorrendet)

Van fejlemény! A Makefile-ban rossz helyek voltak megjelölve forrásnak (középső kép, MAIN\_LIBS sorok 32-34). Így már föltöltötte a kódot ami a csatolt kép bal oldalán látható.

A putty nevű programmal értem el a boardot (jobb oldali kép). Ugye a kód az

arról szól, hogy ha nyomok egy gombot, akkor világít az LD2 led, ha pedig megnyomok egy újabb gombot, akkor pedig elalszik. Ez tökéletesen működik. Azonban putty-n keresztül valamiért csak nagybetűket tudok írni, valamint enter nyomás esetén a sor elejére ugrik. Elvileg el kéne tolnia a betűket az abc-ben, de ez nem igazán történik meg. Ennek lehet oka a putty? Emlékszem, hogy ön nem ezt használta, de nem emlékszem, mivel tudott kommunikálni vele.

Köszönöm a segítséget!  
K. Gábor

[image: image.png]

Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 20., P, 19:59):

Szia Gábor!

No, van barmi fejlemeny? :) Pontosan milyen linuxot is hasznalsz? Ugy jobban

ra tudunk keresni az adott library (\*.a file) pontos helyére.

Ez tenyleg elegge melyviz, szoval absz ne ijedj meg tenyleg ha valami nem

megy kapasbol es tenyleg szolj ha elakadsz valahol!

A.

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Sajnos nem ez a gond, már telepítve van.

Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 19., Cs 21:33):

Tedd fel a libnewlib-arm-none-eabi nevű csomagot!

apt-get install libnewlib-arm-none-eabi

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Kedves András Pál!

Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül megoldanom,

mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a boardra. Amit leírt a korábbiakban, mindent többször átfutottam és ellenőriztem, de kifog rajtam egyelőre a Linux.

El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a

"st-info

--probe" sort a terminálba. A headereket sikeresen bemásoltam a megfelelő

következő

helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a

sorokat kapom a "make upload" esetén:

```
arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
```



```

arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or
directory
make: *** [Makefile:54: main.elf] Error 1

```

Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt,  
ezért  
elnézést!

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 17., K,  
20:36):

```

A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a
board-ra,
nem a
main.c-t eteted meg kozvetlenul. Probald ki a `make upload`
parancsot,
az
az
ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a
Makefile
bogaraszasabol rajossz hogy hogy is van ez pontosan.

Szoval a workflow az itten a kovetkezo:
- A *.c fileokbol *.o (object) fileokat forditunk
(arm-none-eabi-gcc)
- Az egeszet osszelinekujuk (arm-none-eabi-ld)
- Ez a tipusu mikrokontroller a 0x000000004-es címen található
utasításra
ugrik először, ott kezdi a program futtatását
- Ez a tipusu mikrokontroller olyan hogy a 0x000000000
tartományra a
0x08000000 tartományt kepezi le alapertelmezesben
- A linkelésnél így garantálnunk kell hogy legyen "valami" ami a
0x080000004-es címen ami értelmes _és_ kesobb meghivja a main()
fuggvenyt
-
merthogy minden C program a main()-nal "kezdodik". Na, ezt a
feladatot
latja el
a crt.c - plusz még pár egyebet is, de azzal nem igazan kell
foglalatostkodni,
azzal majd kesobb ;)
- A linkeles utan ki kell szednunk a keletkezo *.elf filebol a
"lenyeget". Ezt
csinalja a arm-none-eabi-objcopy, es ez hozza létre a *.hex-et.
- Ezt a *.hex-et pedig az openocd-vel beleirhatjuk a
mikrokontroller
flash-ébe

Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo
sorrendben.
Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es az
openocd
konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani
neki
hogy
milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target

```

mikrokontroller (azaz a programozónak tudnia kell hogy a programozási protokollon keresztül hogyan éri el a mikrokontroller memóriáját és ez függhet a mikrokontroller konkrét típusától - bár egy családon belül ez elegendő, ezért elég csak azt mondani hogy "stm32f0x", és nem azt hogy stm32f072cibt6). Plusz meg vannak egyéb finomságok is, pl a flash-t úgye minden írás előtt törölni kell és csak utána írhatod bele, stb. Ezeket mind-mind megcsinálja a Makefile-be elrejtett parancs-*valami*.

Egyébként ez a fenti sema nagyjából minden beagyazott cuccra ugyanígy megy, csak itt ez a "bare metal" módszer ezt is szükségessé teszi hogy lépésről lépésre menj(unk) végig a lépéseken.

Az arduino-s projektnél is vsz ugyanígy volt, csak gondolom ott használtak valami fejlesztőkörnyezetet ami mindent is megcsinált, nem? Vagy hogy?

A.

On Tue, 17 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent telepítettem. Ami még nem sikerült, az az openocd életre bírása.

Csatolok

egy képet, és bízom benne, hogy tud segíteni.

Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az

st-link

csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a feltöltésig nem jutottam el.

Köszönettel: K. Gábor

[image: Screenshot from 2020-11-17 19-53-48.jpg]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 16., H, 18:35):

Szia Gábor!

Ah, értem, ez így akkor elsore tényleg nem egyszerű. Viszont ennél specifikusabban kéne tudni hogy hol is akadtál el. Tehát:

- telepítés ment? Minden fent van?
- az arm-es toolchain az ezen csomagokból áll:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi
  - \* libnewlib-arm-none-eabi,
  - \* libstdc++-arm-none-eabi-newlib.
- ami meg hasznos csomag és kellhet:
  - \* build-essential (ebben van helyi C fordító is + a `make` program)
  - \* srecord
- openocd az csak simán ez:

\* openocd

Mindegyiket siman `apt-get install ...` modon tudod telepíteni!

Ezek

sikerültek? Ha ezek mind fent vannak, akkor az általam mellekelt

forrast

mar le

tudod fordítani és fel kell tudnod tölteni a kis board-ra. Ami meg

kellhet

a

forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldok

mellekelve.

En

konkretan sima rendszergazdakent az

/usr/local/arm-none-eabi/include

alkovnytar

alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt

teszprogramban

levő Makefile-ban az elejen (kb 12-15ik sorban) levő

ARM=/usr/local/arm-none-eabi

sort modosítsd oda ahova betetted! Lehet siman nálad is, a

/home-od

alatt!

De

ugyis panaszkodik a fordító hogyha nem találja ;) Szóval legjobb

emlekeim

szerint ennyi kell, és ami meg hiányzik az ezeknek a csomagoknak a

fuggosege -

szóval jó kell hogy legyen, legjobb emlekeim szerint.

Az FPGA az sajnos tenyleg egy fokkal bonyolultabb. Előbb nezzük

meg

ezt

az

ARM + STM32 témát, és akkor utána. Valójában azokat sem nehéz

feltenni,

de

kellhet pár olyan development header (pl libboost-csomag) amiket

feltetelez

hogy fent van de alapból szinte biztos hogy nincs fent nálad.

Nálam

konkretan

ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-filesystem-dev:amd64	install
libboost-filesystem1.62-dev:amd64	install
libboost-filesystem1.62.0:amd64	install
libboost-iostreams-dev:amd64	install
libboost-iostreams1.62-dev:amd64	install
libboost-iostreams1.62.0:amd64	install
libboost-program-options-dev:amd64	install
libboost-program-options1.62-dev:amd64	install
libboost-program-options1.62.0:amd64	install
libboost-python-dev	install
libboost-python1.62-dev	install
libboost-python1.62.0	install
libboost-regex1.62-dev:amd64	install

```
libboost-regex1.62.0:amd64      install
libboost-serialization1.62-dev:amd64  install
libboost-serialization1.62.0:amd64    install
libboost-system1.62-dev:amd64        install
libboost-system1.62.0:amd64          install
libboost-thread-dev:amd64            install
libboost-thread1.62-dev:amd64        install
libboost-thread1.62.0:amd64          install
libboost1.62-dev:amd64              install
```

szoval ha azt mondod hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev
libboost-chrono-dev libboost-date-time-dev libboost-dev
libboost-filesystem-dev libboost-iostreams-dev
libboost-program-options-dev
libboost-python-dev libboost-regex-dev libboost-serialization-dev
libboost-system-dev libboost-thread-dev
```

(igen, ez így egy jó hosszú parancssor :)), akkor mindent is

feltesz

ami

kell a  
libboosthoz (es a sima lib-eket is felteszi ami kell a \*-dev  
csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De

tenyleg

ez

kicsit  
macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi

az

ami

hiányzik. Hanyas verzioju milyen linuxot is tettél fel pontosan?

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a  
küldött

dolgokat olvasgattam. Sajnos sem az openocd-t, sem az icestorm  
toolchain-t  
nem sikerült "éltre keltenem". Elvileg a yosys-0.9,

nextpnr-ice40

és

icestorm-utils csomagokat is telepítettem a megadott módon. Tud

esetleg

küldeni valamilyen segítséget, hogyan tudnám programozni

ezekentúl

a

lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék

segítséget

kérni,  
telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam  
kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom  
elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal),

de a

programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai  
eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis

Egyetem  
TDK  
mellett készülhetünk az országos TDK-ra. Mivel már kész van az  
eszköz,  
és  
csak bemutatni kell, így nem vesz el jelentős időt.  
Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet.  
Egy  
két  
szabadsági fokból álló állványon levő parabola-antenna fogja  
"bescannelni"  
az eget, és a vett jelerősséget fogja egy skála szerint  
osztályozni,  
majd  
ezeket a parabola helyzetéből egy képen megfelelő helyre  
illeszteni.  
Fő  
célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell  
érzékelnünk.  
Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő  
érzékelővel.  
Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e  
esetleg  
valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami  
jobb  
lenne)  
csinálni?  
Köszönettel: K. Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7.,  
Szo,  
16:02):  
Szia Gábor!  
Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor  
szuper  
lesz,  
azon  
nagyjából mindent is meg lehet csinálni amit debian-on, és  
hogyhá  
standalone  
es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!  
Headless  
üzemben már nem használnék egyáltalán ubuntu-t, de az más kérdés  
;) )  
Linux alá viszont van teljesen free & open source Lattice FPGA  
toolchain:  
<http://www.clifford.at/icestorm/>, ezt érdemes lehet  
feltelepíteni.  
Mi  
is  
hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icestorm-utils),  
ezekkel a  
csomagokkal közvetlenül is tudod linux alatt mindenféle  
licenc-problema  
meg  
nyűg nélkül használni a barmit! Cserebe a forrás(kod)bol való  
telepítés  
nem  
annyira egyszerű, elég sok a függőség, és nehezen derül ki hogy  
mi  
az.

De  
ebben  
viszont a windozos cuccokkal ellentétben tudok segíteni :)

Ugyanakkor  
természetesen én is néztem a linuxos icecube2-t, és az is működik

persze,  
szóval  
hajra azzal is. Mondjuk az nekem kicsit nehezkés, a "bare metal"

meg a  
klasszikus "unix filozofia" dolgok után nekem ezeket az IDE-eket

eleg  
nehez  
megszoknom :/ De az más kérdés ;)

A.

On Sat, 7 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönök mindent, ma hozzá is állok a dolgok alapos  
tanulmányozásának.

Egy közeli ismerősöm ajánlásából elkezdtuk föltelepíteni az

ubuntut,  
mivel  
a kollégiumban ha problémám van azonnal tud segíteni. Sajnos

nem  
volt  
zökkenőmentes a telepítés. Először a pendrive amin keresztül  
telepíteni  
akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt

föl.

Majd  
miután sikerült, a Grub nem települt így nem tudtam elindítani

a  
rendszer.

Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és  
egyebek.

Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált  
adataimat  
el  
kell fogadják, de az is hamarosan meglesz.

A hétvégén szerintem gyakorlatban ennyit tudok haladni, a

küldött

pdf-eket  
át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.

Szerencsére  
az is teljesen témába vág, hiszen C programozás lesz.

Nagyon le voltam nyugózva a találkozókör és remélem minden

rendben

lesz!

Köszönettel:  
K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 6.,

P,

9:44):

Szia Gábor!

Persze, küldöm akkor az infokat:  
- mellekelve az evaluation board doksija, ebben benne kell

legyen  
minden  
referencia ami hasznos lehet - bár ugye a mi boardunk az

masmilyen.

- a fejlesztokeszlet az ez:

<https://www.latticesemi.com/iCEcube2>,

van

minden

oprendszer ala (linux, windoz).

- az FPGA az ez: iCE40HX8K-BG121, ezt kell kiválasztanod

majd.

- mellekelem a tegnap tesztelt kis verilog-kodot is, abban

megtalalod a

PCF-et: ezt a PCF-et meg teszi az icecube2 is! (a verilog-ot meg

nyilván)

Illetve az ARM-es dolgokkal kapcsolatban is akkor pár info:

- mellekelem a tegnapi példaprogramot (UART: kisbetű ->

nagybetű)

- a toolchain (ami a \*.c fileokból megcsinálja a \*.bin

es/vagy

\*.hex

image-t,

amit aztán fel kell tölteni) az `arm-none-eabi` néven fut.

Ennek

több

komponense van:

\* A C/C++/asm fordító maga: ez a gcc-arm-none-eabi

\* Eszközök amikkel a lefordított fileokat lehet

manipulálni

(pl

a

\*.hex-et előállítani a \*.o/\*.elf-ből):

binutils-arm-none-eabi

\* Es az előre lefordított könyvtárak, hogy a jól

megszokott

C-s

(stdio: printf, stb) meg C++-os (STL és társai)

dolgokat

tud

használni: libnewlib-arm-none-eabi,

libstdc++-arm-none-eabi-newlib.

Ez kb ugyanilyen néven fut windoz alatt is, egy gyors

kereséssel

pl

ezt

találtam:

<https://marketplace.visualstudio.com/items?itemName=metalcode-eu.windows-arm-none-eabi>

keress

rá!

A lényeg hogy ez a "gcc-arm-none-eabi for windows" dollogra

programozni.

Az

A programozó az ST-Link V2 (-1), ezt lehet OpenOCD-vel

OpenOCD

van linux alá is, de van windozos prótyja is:

<https://gnutoolchains.com/arm-eabi/openocd/>. Ettől

függetlenül

az  
ST  
sajat  
cuccait is lehet hasznalni valoszinuleg:  
<https://www.st.com/en/development-tools/stsw-link009.html>, de  
az  
OpenOCD  
annyira jo hogy windozt hasznalo kollegaktol is azt hallom  
hogyan  
inkabb  
az  
OpenOCD-t preferaljak az STM32-es MCU-khoz.

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. november 28. 21:21

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Szuper, ennek örülök!

Elnézést, hogy nem írtam a fejleményekről, ennek az az oka, hogy nem volt jelentős fejlemény.

Miután sikerült feltölteni a legutóbb a kódot, azt próbáltam megérteni, meg próbáltam variálni a visszakapott adatokat.

Elkezdtem utána olvasni a fopencookie() dolgoknak, nagyon kevés és nagyon bonyolult dolgot találtam, nem igazán jutottam előrébb.

A nucleo board-nak a datasheet-jét megkerestem és megnéztem, hogy hogyan van az stm a többi dologgal összekötve.

Az UART dolgoknak is utánajártam, hogy ne legyen légbőlkapott az alkalmazása.

Sajnos az a helyzet, hogy még a kapott kódot sem látom át rendesen, így nehezen tudok nekifogni. Ugye mi most C-t tanulunk, de a kód felépítését leszámítva kevés igazán ismerősdolgot találok, aminek az oka gondolom az, hogy board specifikusan programozunk. Így sajnos a 0. feladat is kifog rajtam. Szuper lenne, ha tudna ajánlani egy olvasmányt az stm32 programozásának alapjaiból.

Köszönettel: K. Gábor

Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 28., Szo, 18:03):

No, bármi fejlemény?

Mi kozben befejeztuk a muholdat, szoval lassan majd megnyilik az ablak a kovetkezo projektek felé!

A.

On Sat, 21 Nov 2020, Gábor Kreinicker wrote:

> Ááá, igen, ezt láttam fönt is, működik, köszönöm!

>

> Akkor igazából kezdhetek játszani. Egy útmutatást tud adni, mit érdemes

> gyakorolni? Vagy tud adni feladatot?

>

> Köszönöm mégegyszer!

>

> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 21., Szo, 11:36):

>

>> Szia Gábor!

>>

>> Remek! Jaj de jó!

>>

>> Nade igen, a soros portos eleres nem annyira magatol ertetodo, es igen,

>> lehet

>> hogy a putty az mas újsor-karakter konvenciot hasznal, raadasul nem is

>> echo-zza

>> vissza vsz a terminalra a karaktereket (igy valojaban amit begepelsz azt

>> nem



```
>> latod mert nem a kepernyodre irja hanem kikuldi a soros porton, es csak
>> azt
>> latod amit visszakuld: es az valoban mar a nagybetűs változat mert ugye
>> azt
>> kuldi vissza a board!).
>>
>> Az ujsor-karakterek minden oprendszerben masok, raadasul a harom
>> legelterjedtebben (linux, mac/osx, windoz) is mind kulonbozoker (rendre
>> 0A, 0D,
>> 0D+0A), lasd: https://en.wikipedia.org/wiki/Newline#Representation, es jo
>> kerdes hogy a puttynál mi az alapertelmezett... ez is bekavarhat. (( Ennek
>> meg
>> regi tortente van es a klasszikus mechanikus irogepek koraig nyulik
>> vissza,
>> mert ott még nem volt a klasszikus "újsor", hanem csak a "kocsi vissza"
>> meg a
>> "soremelés" - hiszen mikor irogeppel irtal es kezdtel ujsort akkor a
>> papirt is
>> hajtandod kellett ("feed"-elni a hengert), plusz a hengert (carriage) magát
>> visszaküldeni az alaphelyzetebe. Ezert nehany rendszer megorizte ezt a
>> kettosseget, mint a DOS, ill azon alapulo rendszerek, mig a unix-variansok
>> csak
>> egy karaktert hasznaltak a kettobol, es... hat, vagy ezt, vagy azt. ))
>>
>> Szoval: hasznalhatod a `socat` nevu programot is, de megmondom oszinten
>> azt nem
>> tudom fejbol hogy azzal hogyan megy a soros vonal összelövése. Viszont: a
>> mellekelt programot szoktam en hasznalni (hazi fejlesztés) amit csak siman
>> `make`-val le tudsz fordítani. Es akkor azzal imigyen:
>>
>> $ ./serial -c /dev/ttyACM1 -b 115200
>>
>> es akkor mehet is a dolog es majd ctrl+c-vel megszakitod amikor "keszen
>> vagy".
>> A minap fent a hegyen is ezt hasznaltuk, szoval ez visszaecho-zza azt amit
>> beirtal es az ujsor-karaktereket is önkonzisztensen kezeli (vagyis
>> egyatalan
>> nem kezeli kulon, emiatt nem kulonbozteti meg a kulonbozo tipusu entereket
>> vagy
>> soremeléseket, de nem is alakítja at, mint ahogy ezekszerint a putty
>> teszi).
>>
>> Probald ki!
>>
>> A.
>>
>> On Sat, 21 Nov 2020, Gábor Kreinicker wrote:
>>
>>> Kedves Pál András! (legutóbb is elrontottam a sorrendet)
>>>
>>> Van fejlemény! A Makefile-ban rossz helyek voltak megjelölve forrásnak
>>> (középső kép, MAIN_LIBS sorok 32-34). Így már föltöltötte a kódot ami a
>>> csatolt kép bal oldalán látható.
>>>
>>> A putty nevű programmal értem el a boardot (jobb oldali kép). Ugye a kód
>> az
>>> arról szól, hogy ha nyomok egy gombot, akkor világít az LD2 led, ha pedig
>>> megnyomok egy újabb gombot, akkor pedig elalszik. Ez tökéletesen működik.
>>> Azonban putty-n keresztül valamiért csak nagybetűket tudok írni, valamint
>>> enter nyomás esetén a sor elejére ugrik. Elvileg el kéne tolnia a betűket
>>> az abc-ben, de ez nem igazán történik meg. Ennek lehet oka a putty?
>>> Emlékszem, hogy ön nem ezt használta, de nem emlékszem, mivel tudott
>>> kommunikálni vele.
>>>
>>> Köszönöm a segítséget!
>>> K. Gábor
```

```
>>>
>>> [image: image.png]
>>>
>>>
>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 20., P,
>> 19:59):
>>>
>>>> Szia Gábor!
>>>> No, van barmi fejlemeny? :) Pontosan milyen linuxot is hasznalsz? Ugy
>>>> jobban
>>>> ra tudunk keresni az adott library (*.a file) pontos helyére.
>>>> Ez tenyleg elegge melyviz, szoval absz ne ijedj meg tenyleg ha valami
>>>> nem
>>>> megy kapasbol es tenyleg szolj ha elakadsz valahol!
>>>> A.
>>>>
>>>> On Thu, 19 Nov 2020, Gábor Kreinicker wrote:
>>>>
>>>>> Sajnos nem ez a gond, már telepítve van.
>>>>>
>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 19., Cs
>>>>> 21:33):
>>>>>
>>>>>> Tedd fel a libnewlib-arm-none-eabi nevű csomagot!
>>>>>>
>>>>>> apt-get install libnewlib-arm-none-eabi
>>>>>>
>>>>>> On Thu, 19 Nov 2020, Gábor Kreinicker wrote:
>>>>>>
>>>>>>> Kedves András Pál!
>>>>>>>
>>>>>>> Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül
>>>>>>> megoldanom,
>>>>>>> mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni semmit a
>>>>>>> boardra. Amit leírt a korábbiakban, mindent többször átfutottam és
>>>>>>> ellenőriztem, de kifog rajtam egyelőre a Linux.
>>>>>>> El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a
>> "st-info
>>>>>>> --probe" sort a terminálba. A headereket sikeresen bemásoltam a
>>>> megfelelő
>>>>>>> helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a
>> következő
>>>>>>> sorokat kapom a "make upload" esetén:
>>>>>>>
>>>>>>>> arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
>>>>>>>> stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
>>>>>>>> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
>>>>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
>>>>>>>> arm-none-eabi-ld: cannot find
>>>>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or
>>>>>>>> directory
>>>>>>>> arm-none-eabi-ld: cannot find
>>>>>>>> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or
>>>>>>>> directory
>>>>>>>> arm-none-eabi-ld: cannot find
>>>>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or
>>>>>>>> directory
>>>>>>>> make: *** [Makefile:54: main.elf] Error 1
>>>>>>>
>>>>>>>> Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt,
>>>>>>>> ezért
>>>>>>>> elnézést!
>>>>>>>>
>>>>>>>> Köszönettel: K. Gábor
>>>>>>>>
>>>>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 17., K,
```

```
>>>>>> 20:36):
>>>>>>
>>>>>>> A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a
>>>> board-ra,
>>>>>>> nem a
>>>>>>> main.c-t eteted meg kozvetlenul. Probald ki a `make upload`
>> parancsot,
>>>>>> az
>>>>>>> az
>>>>>>> ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a
>>>> Makefile
>>>>>>> bogaraszasabol rajossz hogy hogy is van ez pontosan.
>>>>>>>
>>>>>>> Szoval a workflow az itten a kovetkezo:
>>>>>>> - A *.c fileokbol *.o (object) fileokat forditunk
>>>> (arm-none-eabi-gcc)
>>>>>>> - Az egeszet osszelinkeljuk (arm-none-eabi-ld)
>>>>>>> - Ez a tipusu mikrokontroller a 0x000000004-es címen található
>>>>>>> utasításra
>>>>>>> ugrik először, ott kezdi a program futtatását
>>>>>>> - Ez a tipusu mikrokontroller olyan hogy a 0x000000000
>> tartomanyra a
>>>>>>> 0x08000000 tartományt kepezi le alapertelmezesben
>>>>>>> - A linkelésnél így garantálnunk kell hogy legyen "valami" ami a
>>>>>>> 0x080000004-es címen ami értelmes _és_ kesobb meghivja a main()
>>>>>> fuggvenyt
>>>>>>> -
>>>>>>> merthogy minden C program a main()-nal "kezdodik". Na, ezt a
>> feladatot
>>>>>>> latja el
>>>>>>> a crt.c - plusz még pár egyebet is, de azzal nem igazan kell
>>>>>>> foglalatostkodni,
>>>>>>> azzal majd kesobb ;)
>>>>>>> - A linkeles utan ki kell szednunk a keletkezo *.elf filebol a
>>>>>>> "lenyeget". Ezt
>>>>>>> csinalja a arm-none-eabi-objcopy, es ez hozza létre a *.hex-et.
>>>>>>> - Ezt a *.hex-et pedig az openocd-vel beleirhatjuk a
>> mikrokontroller
>>>>>>> flash-ébe
>>>>>>>
>>>>>>> Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo
>>>>>>> sorrendben.
>>>>>>> Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es az
>>>>>>> openocd
>>>>>>> konfiguracioja sem annyira egyszeru, mert egyreszt meg kell mondani
>>>> neki
>>>>>>> hogy
>>>>>>> milyen tipusu programozót hasznalsz, masreszt milyen tipusu a target
>>>>>>> mikrokontroller (azaz a programozonak tudnia kell hogy a
>> programozasi
>>>>>>> protokollon keresztul hogyan eri el a mikrokontroller memoriajat es
>> ez
>>>>>>> fugghet
>>>>>>> a mikrokontroller konkret tipusától - bar egy csaladon belül ez
>> elegge
>>>>>>> egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem azt
>> hogy
>>>>>>> stm32f072cibt6). Plusz meg vannak egyeb finomsagok is, pl a flash-t
>>>> ugye
>>>>>>> minden
>>>>>>> iras előtt torolni kell es csak utana irhatsz bele, stbstb. Ezeket
>>>>>>> mind-mind
>>>>>>> megcsinalja a Makefile-be elrejtett parancs-valami.
>>>>>>>
>>>>>>> Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra
>> ugyanigy
```

>>>>>>> megy,  
>>>>>>> csak itt ez a "bare metal" módszer ezt is szuksegesse teszi hogy  
>>>>>>> lepesrol  
>>>>>>> lepesre menj(unk) vegig a lepeseken.  
>>>>>>>  
>>>>>>> Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott  
>>>>>>> hasznaltatok  
>>>>>>> valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy  
>> hogy?  
>>>>>>>  
>>>>>>> A.  
>>>>>>>  
>>>>>>> On Tue, 17 Nov 2020, Gábor Kreinicker wrote:  
>>>>>>>  
>>>>>>> Kedves Pál András!  
>>>>>>>  
>>>>>>>> Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent  
>>>>>>>> telepítettem. Ami még nem sikerült, az az openocd életre bírása.  
>>>>>>>> Csatolok  
>>>>>>>> egy képet, és bízom benne, hogy tud segíteni.  
>>>>>>>> Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az  
>>>>>>>> st-link  
>>>>>>>> csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a  
>>>>>>>> feltöltésig  
>>>>>>>> nem jutottam el.  
>>>>>>>>  
>>>>>>>> Köszönettel: K. Gábor  
>>>>>>>> [image: Screenshot from 2020-11-17 19-53-48.jpg]  
>>>>>>>>  
>>>>>>>>  
>>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 16., H,  
>>>>>>>> 18:35):  
>>>>>>>>  
>>>>>>>> Szia Gábor!  
>>>>>>>>  
>>>>>>>>> Ah, ertem, ez így akkor elsore tenyleg nem egyszerű. Viszont ennel  
>>>>>>>>> specifikusabban kene tudni hogy hol is akadtál el. Tehat:  
>>>>>>>>> - telepites ment? Minden fent van?  
>>>>>>>>> - az arm-es toolchain az ezen csomagokbol all:  
>>>>>>>>> \* gcc-arm-none-eabi  
>>>>>>>>> \* binutils-arm-none-eabi  
>>>>>>>>> \* libnewlib-arm-none-eabi,  
>>>>>>>>> \* libstdc++-arm-none-eabi-newlib.  
>>>>>>>>> - ami meg hasznos csomag es kellhet:  
>>>>>>>>> \* build-essential (ebben van helyi C fordito is + a `make`  
>>>>>>>>> program)  
>>>>>>>>> \* srecord  
>>>>>>>>> - openocd az csak siman ez:  
>>>>>>>>> \* openocd  
>>>>>>>>>  
>>>>>>>>> Mindegyiket siman `apt-get install ...` modon tudod telepiteni!  
>> Ezek  
>>>>>>>>> sikerultek? Ha ezek mind fent vannak, akkor az altalam mellekelt  
>>>>>>>>> forrast  
>>>>>>>>> mar le  
>>>>>>>>> tudod fordítani es fel kell tudnod tölteni a kis board-ra. Ami meg  
>>>>>>>>> kellhet  
>>>>>>>>> a  
>>>>>>>>>> forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom  
>>>>>>>>>> mellekelve.  
>>>>>>>>>> En  
>>>>>>>>>> konkretan sima rendszergazdakent az  
>> /usr/local/arm-none-eabi/include  
>>>>>>>>>> alkovnytar  
>>>>>>>>>> alatt tárolok ezeket, de barhova felteheted! Akkor a mellekelt  
>>>>>>>>>> teszprogramban

```
>>>>>>>> levő Makefile-ban az elejen (kb 12-15ik sorban) levő
>>>>>>>>
>>>>>>>> ARM=/usr/local/arm-none-eabi
>>>>>>>>
>>>>>>>> sort modosítsd oda ahova betetted! Lehet simán nálad is, a
>> /home-od
>>>>>>>> alatt!
>>>>>>>> De
>>>>>>>> ugyis panaszkodik a fordító hogyha nem találja ;) Szóval legjobb
>>>>>>>> emlékeim
>>>>>>>> szerint ennyi kell, és ami meg hiányzik az ezeknek a csomagoknak a
>>>>>>>> függősege -
>>>>>>>> szóval jó kell hogy legyen, legjobb emlékeim szerint.
>>>>>>>>
>>>>>>>> Az FPGA az sajnos tényleg egy fokkal bonyolultabb. Előbb nezzük
>> meg
>>>>>>>> ezt
>>>>>>>> az
>>>>>>>> ARM + STM32 témát, és akkor utána. Valójában azokat sem nehéz
>>>>>>>> feltenni,
>>>>>>>> de
>>>>>>>> kellhet pár olyan development header (pl libboost-csomag) amiket
>>>>>>>> feltételez
>>>>>>>> hogy fent van de alából szinte biztos hogy nincs fent nálad.
>> Nálam
>>>>>>>> konkrétan
>>>>>>>> ezek vannak fent:
>>>>>>>>
>>>>>>>> libboost-atomic1.62-dev:amd64          install
>>>>>>>> libboost-atomic1.62.0:amd64              install
>>>>>>>> libboost-chrono1.62-dev:amd64             install
>>>>>>>> libboost-chrono1.62.0:amd64               install
>>>>>>>> libboost-date-time1.62-dev:amd64           install
>>>>>>>> libboost-date-time1.62.0:amd64            install
>>>>>>>> libboost-dev:amd64                        install
>>>>>>>> libboost-filesystem-dev:amd64              install
>>>>>>>> libboost-filesystem1.62-dev:amd64          install
>>>>>>>> libboost-filesystem1.62.0:amd64            install
>>>>>>>> libboost-iostreams-dev:amd64              install
>>>>>>>> libboost-iostreams1.62-dev:amd64           install
>>>>>>>> libboost-iostreams1.62.0:amd64            install
>>>>>>>> libboost-program-options-dev:amd64         install
>>>>>>>> libboost-program-options1.62-dev:amd64     install
>>>>>>>> libboost-program-options1.62.0:amd64       install
>>>>>>>> libboost-python-dev                       install
>>>>>>>> libboost-python1.62-dev                   install
>>>>>>>> libboost-python1.62.0                     install
>>>>>>>> libboost-regex1.62-dev:amd64               install
>>>>>>>> libboost-regex1.62.0:amd64                 install
>>>>>>>> libboost-serialization1.62-dev:amd64       install
>>>>>>>> libboost-serialization1.62.0:amd64         install
>>>>>>>> libboost-system1.62-dev:amd64              install
>>>>>>>> libboost-system1.62.0:amd64                install
>>>>>>>> libboost-thread-dev:amd64                  install
>>>>>>>> libboost-thread1.62-dev:amd64              install
>>>>>>>> libboost-thread1.62.0:amd64                install
>>>>>>>> libboost1.62-dev:amd64                     install
>>>>>>>>
>>>>>>>> szóval ha azt mondod hogy
>>>>>>>>
>>>>>>>> # apt-get --no-install-recommends install libboost-atomic-dev
>>>>>>>> libboost-chrono-dev libboost-date-time-dev libboost-dev
>>>>>>>> libboost-filesystem-dev libboost-iostreams-dev
>>>>>>>> libboost-program-options-dev
>>>>>>>> libboost-python-dev libboost-regex-dev libboost-serialization-dev
>>>>>>>> libboost-system-dev libboost-thread-dev
```

```
>>>>>>>>
>>>>>>>> (igen, ez így egy jó hosszú parancssor :)), akkor mindent is
>> feltesz
>>>>> ami
>>>>>>>> kell a
>>>>>>>> libboosthoz (es a sima lib-eket is felteszi ami kell a *-dev
>>>>>>>> csomagokhoz)
>>>>>>>>
>>>>>>>> Ami meg szinte biztos hogy kell az a: flex bison libtool. De
>> tenyleg
>>>>>> ez
>>>>>>>> kicsit
>>>>>>>> macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi
>> az
>>>>>> ami
>>>>>>>> hiányzik. Hanyas verzioju milyen linuxot is tettél fel pontosan?
>>>>>>>>
>>>>>>>> A.
>>>>>>>>
>>>>>>>> On Mon, 16 Nov 2020, Gábor Kreinicker wrote:
>>>>>>>>
>>>>>>>> Kedves Pál András!
>>>>>>>>
>>>>>>>> Elnézést a kései bejelentkezésért. A leírtakat telepítettem és a
>>>>>>>> küldött
>>>>>>>> dolgokat olvagattam. Sajnos sem az openocd-t, sem az icestorm
>>>>>>>> toolchain-t
>>>>>>>> nem sikerült "éleltre keltenem". Elvileg a yosys-0.9,
>> nextpnr-ice40
>>>> és
>>>>>>>> icestorm-utils csomagokat is telepítettem a megadott módon. Tud
>>>>>> esetleg
>>>>>>>> küldeni valamilyen segítséget, hogyan tudnám programozni
>> ezeken túl
>>>> a
>>>>>>>> lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék
>>>> segítséget
>>>>>>>> kérni,
>>>>>>>> telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam
>>>>>>>> kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom
>>>>>>>> elérni.
>>>>>>>>
>>>>>>>> Ami biztos, hogy a boardokat felismeri a gép (lsusb parancssal),
>>>> de a
>>>>>>>> programozásukig sajnos még nem jutottam el.
>>>>>>>>
>>>>>>>> Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai
>>>>>>>> eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis
>>>>>> Egyetem
>>>>>>>> TDK
>>>>>>>> mellett készülhetünk az országos TDK-ra. Mivel már kész van az
>>>>>> eszköz,
>>>>>>>> és
>>>>>>>> csak bemutatni kell, így nem vesz el jelentős időt.
>>>>>>>>
>>>>>>>> Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet.
>>>> Egy
>>>>>>>> két
>>>>>>>> szabadsági fokból álló állványon levő parabola-antenna fogja
>>>>>>>> "bescannelni"
>>>>>>>> az eget, és a vett jelerősséget fogja egy skála szerint
>>>> osztályozni,
>>>>>>>> majd
>>>>>>>> ezeket a parabola helyzetéből egy képen megfelelő helyre
>>>> illeszteni.
>>>>>> Fő
```

>>>>>>>>> célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell  
>>>>>>>>> érzékelnünk.  
>>>>>>>>> Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő  
>>>>>>>>> érzékelővel.  
>>>>>>>>> Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e  
>> esetleg  
>>>>>>>>> valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami  
>> jobb  
>>>>>>>>> lenne)  
>>>>>>>>> csinálni?  
>>>>>>>>>  
>>>>>>>>> Köszönettel: K. Gábor  
>>>>>>>>>  
>>>>>>>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7.,  
>> Szo,  
>>>>>>>>> 16:02):  
>>>>>>>>>  
>>>>>>>>> Szia Gábor!  
>>>>>>>>>  
>>>>>>>>> Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor  
>> szuper  
>>>>>>>>> lesz,  
>>>>>>>>> azon  
>>>>>>>>> nagyjából mindent is meg lehet csinálni amit debian-on, es  
>> hogyha  
>>>>>>>>> standalone  
>>>>>>>>> es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!  
>>>>>>>>> Headless  
>>>>>>>>> üzemben mar nem használnék egyatalan ubuntu-t, de az mas kerdes  
>> ;)  
>>>>>>>>>  
>>>>>>>>> Linux alá viszont van teljesen free & open souce Lattice FPGA  
>>>>>>>>> toolchain:  
>>>>>>>>> <http://www.clifford.at/icestorm/>, ezt erdemes lehet  
>>>> feltelepíteni.  
>>>>>>>>> Mi  
>>>>>>>>> is  
>>>>>>>>> hasznaljuk ezt (yosys-0.9, nextpnr-ice40, icestorm-utils),  
>>>> ezekkel a  
>>>>>>>>> csomagokkal kozvetlenul is tudod linux alatt mindenfele  
>>>>>>>>> licenc-problema  
>>>>>>>>> meg  
>>>>>>>>> nyűg nélkül hasznalni a barmit! Cserebe a forras(kod)bol valo  
>>>>>>>>> telepites  
>>>>>>>>> nem  
>>>>>>>>> annyira egyszerű, eleg sok a függőség, es nehezen derül ki hogy  
>> mi  
>>>>>>>>> az.  
>>>>>>>>> De  
>>>>>>>>> ebben  
>>>>>>>>> viszont a windozos cuccokkal ellentétben tudok segíteni :)  
>>>>>>>>> Ugyanakkor  
>>>>>>>>> természetesen en is neztem a linuxos icecube2-t, es az is mukodik  
>>>>>>>>> persze,  
>>>>>>>>> szoval  
>>>>>>>>> hajra azzal is. Mondjuk az nekem kicsit nehezkes, a "bare metal"  
>>>>>>>>> meg a  
>>>>>>>>> klasszikus "unix filozofia" dolgok utan nekem ezeket az IDE-ket  
>>>> eleg  
>>>>>>>>> nehez  
>>>>>>>>> megszoknom :/ De az mas kerdes ;)  
>>>>>>>>>  
>>>>>>>>> A.  
>>>>>>>>>  
>>>>>>>>> On Sat, 7 Nov 2020, Gábor Kreinicker wrote:  
>>>>>>>>>

```
>>>>>>>>>> Kedves Pál András!
>>>>>>>>>>
>>>>>>>>>> Köszönök mindent, ma hozzá is állok a dolgok alapos
>>>>>>>>>> tanulmányozásának.
>>>>>>>>>> Egy közeli ismerősöm ajánlásából elkezdtük föltelepíteni az
>>>>>>>>>> ubuntu-t,
>>>>>>>>>> mivel
>>>>>>>>>> a kollégiumban ha problémám van azonnal tud segíteni. Sajnos
>> nem
>>>>>>>>>> volt
>>>>>>>>>> zökkenőmentes a telepítés. Először a pendrive amin keresztül
>>>>>>>>>> telepíteni
>>>>>>>>>> akartam volt rossz, ami csak a 3. sikertelen kísérlet után tűnt
>>>>>>>>>> föl.
>>>>>>>>>> Majd
>>>>>>>>>> miután sikerült, a Grub nem települt így nem tudtam elindítani
>> a
>>>>>>>>>> rendszert.
>>>>>>>>>> Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás és
>>>>>>>>>> egyebek.
>>>>>>>>>> Az iCEcube2 telepítéséhez még várnom kell, mivel a regisztrált
>>>>>>>>>> adataimat
>>>>>>>>>> el
>>>>>>>>>> kell fogadják, de az is hamarosan meglesz.
>>>>>>>>>> A hétvégén szerintem gyakorlatban ennyit tudok haladni, a
>> küldött
>>>>>>>>>> pdf-eket
>>>>>>>>>> át tanulmányozom még, de igyekezek a jövő heti zh-ra készülni.
>>>>>>>>>> Szerencsére
>>>>>>>>>> az is teljesen témába vág, hiszen C programozás lesz.
>>>>>>>>>> Nagyon le voltam nyűgözve a találkozáskor és remélem minden
>>>> rendben
>>>>>>>>>> lesz!
>>>>>>>>>>
>>>>>>>>>> Köszönettel:
>>>>>>>>>> K. Gábor
>>>>>>>>>>
>>>>>>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 6.,
>> P,
>>>>>>>>>> 9:44):
>>>>>>>>>>
>>>>>>>>>> Szia Gábor!
>>>>>>>>>>
>>>>>>>>>> Persze, kuldom akkor az infokat:
>>>>>>>>>> - mellekelve az evaluation board doksija, ebben benne kell
>>>>>>>>>> legyen
>>>>>>>>>> minden
>>>>>>>>>> referencia ami hasznos lehet - bar ugye a mi boardunk az
>>>>>>>>>> masmilyen.
>>>>>>>>>> - a fejlesztokeszlet az ez:
>>>>>>>>>> https://www.latticesemi.com/iCEcube2,
>>>>>>>>>> van
>>>>>>>>>> minden
>>>>>>>>>> oprendszer ala (linux, windoz).
>>>>>>>>>> - az FPGA az ez: iCE40HX8K-BG121, ezt kell kivalasztanod
>> majd.
>>>>>>>>>> - mellekelem a tegnap tesztelt kis verillog-kodot is, abban
>>>>>>>>>> megtalalaoad a
>>>>>>>>>> PCF-et: ezt a PCF-et megeszi az icecube2 is! (a verillog-ot meg
>>>>>>>>>> nyilván)
>>>>>>>>>>
>>>>>>>>>> Illetve az ARM-es dolgokkal kapcsolatban is akkor par info:
>>>>>>>>>> - mellekelem a tegnapi peldaprogramot (UART: kisbetu ->
>>>>>>>>>> nagybetu)
>>>>>>>>>> - a toolchain (ami a *.c fileokbol megcsinalja a *.bin
>> es/vagy
```



[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

No, tehát: melyik reszet nem látod at, pontosan? Ez így összesen ~350 sornyi kód, de első körben ami reszedol érdekes az a main.c csak. A többin is végig tudunk menni természetesen, mert előbbutobb tényleg mindent érdemes jól átnézni! Ha ezeket már átnéztél akkor tényleg sima ügy lesz mindenféle beagyazott alacsony szintű programozás, kb programnyelvtől függetlenül is - szóval nyugodtan szánd, szanjuk rá az időt.

Megmondom osztinten, a fopencookie()-t meg en sem hasznaltam igy ebben a formaban, csak egy nagyon hasonlot, az fdev\_setup\_stream()-et. Ami kimondottan AVR-ekhez van, lehet hogy ezt hasznaltad mar esetleg arduino-nal, lehet hogy nem. A lenyeg hogy az fopencookie() az ennek a kicsit altalanosabb valtozata, kicsit több overhead-del. Szoval ez mar tenyleg absz "uj dolog tanulasa" lenne reszemrol is ;)



A C az ugyan egy alacsony szintű nyelv, de ahhoz már elege magas szintű hogy tkp olasymí amit írsz vagyis kérdezel hogy "stm32 programozásának alapjaiból" dolog az nem kimondhatóan C specifikus. Pl ez a fopencookie() letezik minden libc-architekturan, linuxon, vsz windozon is, kvazi része a C standardnak. Az "STM32 specifikus dolgok" azok a hardver-regiszterek elerése. A lenyeg minden ilyen programozasnal hogy ezeket a specifikus, hardver-regiszter elerese dolgokat minel hamarabb "elrejtjük" a program fő részei elöl, es ugy használjuk mint egy "kész könyvtár". Aztan utána így megyünk lepesrol lepesre.

A.

On Sat, 28 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Szuper, ennek örülök!

Elnézést, hogy nem írtam a fejleményekről, ennek az az oka, hogy nem volt jelentős fejlemény.

Miután sikerült feltölteni a legutóbb a kódot, azt próbáltam megérteni, meg próbáltam variálni a visszkapott adatokat. Elkezdtem utána olvasni a fopencookie() dolgoknak, nagyon kevés és nagyon bonyolult dolgot találtam, nem igazán jutottam előrébb. A nucleo board-nak a datasheet-jét megkerestem és megnéztem, hogy hogyan van az stm a többi dologgal összekötve. Az UART dolgoknak is utánajártam, hogy ne legyen légbőlkapott az alkalmazása.

Sajnos az a helyzet, hogy még a kapott kódot sem látom át rendesen, így nehezen tudok nekifogni. Ugye mi most C-t tanulunk, de a kód felépítését leszámítva kevés igazán ismerősdolgot találok, aminek az oka gondolom az, hogy board specifikusan programozunk. Így sajnos a 0. feladat is kifog rajtam. Szuper lenne, ha tudna ajánlani egy olvasmányt az stm32 programozásának alapjaiból.

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 28., Szo, 18:03):

No, bármi fejlemény?

Mi közben befejeztük a muholdat, szóval lassan majd megnyílik az ablak a következő projektek felé!

A.

On Sat, 21 Nov 2020, Gábor Kreinicker wrote:

Ááá, igen, ezt láttam fönt is, működik, köszönöm!

Akkor igazából kezdhettek játszani. Egy útmutatást tud adni, mit érdemes gyakorolni? Vagy tud adni feladatot?

Köszönöm még egyszer!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 21., Szo, 11:36):

Szia Gábor!

Remek! Jaj de jó!

Nade igen, a soros portos eleres nem annyira magától értetődő, és igen, lehet hogy a putty az más újsor-karakter konvenciót használ, ráadásul nem is echo-zza

vissza vsz a terminalra a karaktereket (így valójában amit begepelsz azt nem látod mert nem a képernyőre írja hanem kiküldi a soros porton, és csak azt látod amit visszaküld: és az valóban már a nagybetűs változat mert úgyeküldi vissza a board!).

Az újsor-karakterek minden operációs rendszerben mások, ráadásul a három legelterjedtebben (linux, mac/osx, windows) is mind különbözőek (rendre 0A, 0D, 0D+0A), lásd: <https://en.wikipedia.org/wiki/Newline#Representation>, és

jo

kerdes hogy a putty-nál mi az alapértelmezett... ez is bekavarhat. ((

Ennek

meg régi története van és a klasszikus mechanikus írógépek koráig nyúlik vissza, mert ott még nem volt a klasszikus "újsor", hanem csak a "kocsi vissza" meg a "soremelés" - hiszen mikor írógéppel irtal és kezdte új sort akkor a papírt is hajtani kellett ("feed"-elni a hengert), plusz a hengert (carriage)

magát

visszaküldeni az alaphelyzetébe. Ezért néhány rendszer megőrizte ezt a kettősséget, mint a DOS, ill azon alapuló rendszerek, míg a

unix-variantok

csak egy karaktert használtak a kettőből, és... hát, vagy ezt, vagy azt. ))

Szóval: használhatod a `socat` nevű programot is, de megmondom őszintén azt nem tudom fejből hogy azzal hogyan megy a soros vonal összelövése. Viszont:

a

mellekelt programot szoktam én használni (házi fejlesztés) amit csak

simán

`make`-val le tudsz fordítani. És akkor azzal imígyen:

```
$ ./serial -c /dev/ttyACM1 -b 115200
```

és akkor mehet is a dolog és majd ctrl+c-vel megszakítod amikor "készén vagy".

A minap fent a hegyen is ezt használtuk, szóval ez visszaecho-zza azt

amit

beírtal és az újsor-karaktereket is önkonzisztensen kezeli (vagyis egyáltalán nem kezeli külön, emiatt nem különbözteti meg a különböző típusú

entereket

vagy soremelesek, de nem is alakítja át, mint ahogy ezékszerint a putty teszi).

Probáld ki!

A.

On Sat, 21 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András! (legutóbb is elrontottam a sorrendet)

Van fejlemény! A Makefile-ban rossz helyek voltak megjelölve forrásnak (középső kép, MAIN\_LIBS sorok 32-34). Így már föltöltötte a kódot ami a csatolt kép bal oldalán látható.

A putty nevű programmal értem el a boardot (jobb oldali kép). Ugye a

kód

az  
 arról szól, hogy ha nyomok egy gombot, akkor világít az LD2 led, ha  
 pedig  
 megnyomok egy újabb gombot, akkor pedig elalszik. Ez tökéletesen  
 működik.  
 Azonban putty-n keresztül valamiért csak nagybetűket tudok írni,  
 valamint  
 enter nyomás esetén a sor elejére ugrik. Elvileg el kéne tolnia a  
 betűket  
 az abc-ben, de ez nem igazán történik meg. Ennek lehet oka a putty?  
 Emlékszem, hogy ön nem ezt használta, de nem emlékszem, mivel tudott  
 kommunikálni vele.

Köszönöm a segítséget!  
 K. Gábor

[image: image.png]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 20., P,  
 19:59):

Szia Gábor!  
 No, van barmi fejlemeny? :) Pontosan milyen linuxot is hasznalsz?  
 Ugy  
 jobban  
 ra tudunk keresni az adott library (\*.a file) pontos helyére.  
 Ez tenyleg elegge melyviz, szoval absz ne ijedj meg tenyleg ha

valami  
 nem  
 megy kapasbol es tenyleg szolj ha elakadsz valahol!  
 A.

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Sajnos nem ez a gond, már telepítve van.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 19., Cs  
 21:33):

Tedd fel a libnewlib-arm-none-eabi nevű csomagot!

apt-get install libnewlib-arm-none-eabi

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Kedves András Pál!

Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül  
 megoldanom,  
 mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni

semmit a  
 boardra. Amit leírt a korábbiakban, mindent többször átfutottam és  
 ellenőriztem, de kifog rajtam egyelőre a Linux.  
 El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a

"st-info  
 --probe" sort a terminálba. A headereket sikeresen bemásoltam a  
 megfelelő  
 helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a  
 következő  
 sorokat kapom a "make upload" esetén:

```
arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
```

```

arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or
directory
arm-none-eabi-ld: cannot find
/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or
directory
make: *** [Makefile:54: main.elf] Error 1

```

Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka előtt,

ezért  
elnézést!

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 17., K, 20:36):

```

    A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a
board-ra,
    nem a
    main.c-t eteted meg kozvetlenul. Probald ki a `make upload`
parancsot,
    az
    az
    ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a
Makefile
    bogaraszasabol rajossz hogy hogy is van ez pontosan.

    Szoval a workflow az itten a kovetkezo:
    - A *.c fileokbol *.o (object) fileokat forditunk
    (arm-none-eabi-gcc)
    - Az egeszet osszelinkeljük (arm-none-eabi-ld)
    - Ez a tipusu mikrokontroller a 0x000000004-es címen található
    utasításra
    ugrik először, ott kezdi a program futtatását
    - Ez a tipusu mikrokontroller olyan hogy a 0x000000000
tartomanyra a
    0x08000000 tartományt kepezi le alapertelmezesben
    - A linkelésnél így garantálnunk kell hogy legyen "valami" ami a
    0x080000004-es címen ami értelmes _és_ kesobb meghivja a main()
    fuggvenyt
    -
    merthogy minden C program a main()-nal "kezdodik". Na, ezt a
feladatot
    latja el
    a crt.c - plusz még pár egyebet is, de azzal nem igazan kell
    foglalatzkodni,
    azzal majd kesobb ;)
    - A linkeles utan ki kell szednunk a keletkezo *.elf filebol a
    "lenyeget". Ezt
    csinalja a arm-none-eabi-objcopy, es ez hozza létre a *.hex-et.
    - Ezt a *.hex-et pedig az openocd-vel beleirhatjuk a
mikrokontroller
    flash-ébe

    Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo
    sorrendben.
    Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es
az
    openocd
    konfiguracioja sem annyira egyszeru, mert egyreszt meg kell
mondani

```

neki  
 hogy  
 milyen típusu programozót hasznalsz, masreszt milyen típusu a  
 target  
 mikrokontroller (azaz a programozónak tudnia kell hogy a  
 programozasi  
 protokollon keresztül hogyan eri el a mikrokontroller memoriajat  
 es  
 ez  
 fugghet  
 a mikrokontroller konkret tipusától - bar egy csaladon belül ez  
 elegge  
 egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem azt  
 hogy  
 stm32f072cbt6). Plusz meg vannak egyeb finomsagok is, pl a flash-t  
 ugye  
 minden  
 iras előtt torolni kell es csak utana írhat sz bele, stbstb. Ezeket  
 mind-mind  
 megcsinalja a Makefile-be elrejtett parancs- valami.  
 Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra  
 ugyanigy  
 megy,  
 csak itt ez a "bare metal" modszer ezt is szuksegesse teszi hogy  
 lepesrol  
 lepesre menj(unk) vegig a lepeseken.  
 Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott  
 hasznaltatok  
 valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy  
 hogy?

A.

On Tue, 17 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent  
 telepítettem. Ami még nem sikerült, az az openocd életre bírása.

Csatolok

egy képet, és bízom benne, hogy tud segíteni.

Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az

st-link

csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a  
 feltöltésig  
 nem jutottam el.

Köszönettel: K. Gábor

[image: Screenshot from 2020-11-17 19-53-48.jpg]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 16.,

H,

18:35):

Szia Gábor!

Ah, ertem, ez így akkor elsore tenyleg nem egyszerű. Viszont

ennel

specifikusabban kene tudni hogy hol is akadtál el. Tehat:

- telepites ment? Minden fent van?
- az arm-es toolchain az ezen csomagokbol all:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi

\* libnewlib-arm-none-eabi,  
 \* libstdc++-arm-none-eabi-newlib.  
 - ami meg hasznos csomag es kellhet:  
 \* build-essential (ebben van helyi C fordito is + a

`make`  
 program)  
 \* srecord  
 - openocd az csak siman ez:  
 \* openocd

Ezek  
 Mindegyiket siman `apt-get install ...` modon tudod telepiteni!

forrast  
 sikerultek? Ha ezek mind fent vannak, akkor az altalam mellekelt  
 mar le  
 tudod fordítani es fel kell tudnod tölteni a kis board-ra. Ami

meg  
 kellhet  
 a  
 forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom  
 mellekelve.  
 En  
 konkretan sima rendszergazdakent az  
 /usr/local/arm-none-eabi/include  
 alkovnytar  
 alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt  
 teszprogramban  
 levő Makefile-ban az elejen (kb 12-15ik sorban) levő  
 ARM=/usr/local/arm-none-eabi  
 sort modosítsd oda ahova betetted! Lehet siman nálad is, a

/home-od  
 alatt!  
 De  
 ugyis panaszkodik a fordito hogyha nem találja ;) Szoval legjobb  
 emlekeim  
 szerint ennyi kell, es ami meg hianyzik az ezeknek a

csomagoknak a  
 fuggosege -  
 szoval jo kell hogy legyen, legjobb emlekeim szerint.

meg  
 Az FPGA az sajnos tenyleg egy fokkal bonyolultabb. Elobb nezzuk

ezt  
 az  
 ARM + STM32 témát, es akkor utana. Valojaban azokat sem nehez  
 feltenni,  
 de  
 kellhet par olyan development header (pl libboost-csomag) amiket  
 feltetelez  
 hogy fent van de alapbol szinte biztos hogy nincs fent nálad.

Nalam  
 konkretan  
 ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-filesystem-dev:amd64	install
libboost-filesystem1.62-dev:amd64	install
libboost-filesystem1.62.0:amd64	install

```
libboost-iostreams-dev:amd64      install
libboost-iostreams1.62-dev:amd64  install
libboost-iostreams1.62.0:amd64    install
libboost-program-options-dev:amd64 install
libboost-program-options1.62-dev:amd64 install
libboost-program-options1.62.0:amd64 install
libboost-python-dev               install
libboost-python1.62-dev           install
libboost-python1.62.0            install
libboost-regex1.62-dev:amd64      install
libboost-regex1.62.0:amd64        install
libboost-serialization1.62-dev:amd64 install
libboost-serialization1.62.0:amd64 install
libboost-system1.62-dev:amd64      install
libboost-system1.62.0:amd64        install
libboost-thread-dev:amd64          install
libboost-thread1.62-dev:amd64      install
libboost-thread1.62.0:amd64        install
libboost1.62-dev:amd64            install
```

szoval ha azt mondd hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev
libboost-chrono-dev libboost-date-time-dev libboost-dev
libboost-filesystem-dev libboost-iostreams-dev
```

```
libboost-program-options-dev
libboost-python-dev libboost-regex-dev
```

```
libboost-serialization-dev
libboost-system-dev libboost-thread-dev
```

(igen, ez így egy jó hosszú parancssor :)), akkor mindent is

feltesz

ami

kell a

libboosthoz (es a sima lib-eket is felteszi ami kell a \*-dev csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De

tenyleg

ez

kicsit

macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi

az

ami

hiányzik. Hanyas verzióju milyen linuxot is tettél fel pontosan?

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem és

a

küldött

dolgokat olvastam. Sajnos sem az openocd-t, sem az icestorm toolchain-t

nem sikerült "életre keltenem". Elvileg a yosys-0.9,

nextpnr-ice40

és

icestorm-utils csomagokat is telepítettem a megadott módon. Tud

esetleg

küldeni valamilyen segítséget, hogyan tudnám programozni

ezekentül

a

lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék

segítséget kérni,  
telepítettem (elvileg, habár az elmúlt hétben a linuxot próbáltam kiismerni, de még nem érzem át a rendszert), de sehogy nem tudom elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb paranccsal), de a programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás diagnosztikai eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis Egyetem TDK mellett készülhetünk az országos TDK-ra. Mivel már kész van az eszköz, és csak bemutatni kell, így nem vesz el jelentős időt.

Kicsit más téma, egyetemi projektként építünk egy rádiótávcsövet. Egy két szabadsági fokból álló állványon levő parabola-antenna fogja "bescannelni" az eget, és a vett jelerősséget fogja egy skála szerint osztályozni, majd ezeket a parabola helyzetéből egy képen megfelelő helyre illeszteni.

Fő célunk a Tejút "lefotózása", amihez az ~1,4 GHz-es jelet kell érzékelnünk. Ehhez egy ~1,4 GHz-es erősítőre lenne szükségünk, megfelelő érzékelővel.

Nem igen van erről sok infó, így az iránt érdeklődöm, tud-e esetleg valamilyen formában segíteni, hol lehet ilyet venni, vagy (ami jobb lenne) csinálni?

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 7., Szo, 16:02):

Szia Gábor!

Remek, koszi az update-et! Igen, az ubuntu kezdetnek akkor szuper lesz, azon nagyjabol mindent is meg lehet csinálni amit debian-on, es hogyha standalone es/vagy desktop üzemben hasznalod akkor tenyleg nem lehet gond!

Headless üzemben mar nem használnék egyatalan ubuntu-t, de az mas kerdes ;)



Linux alá viszont van teljesen free & open source Lattice FPGA  
 toolchain:  
<http://www.clifford.at/icestorm/>, ezt érdemes lehet  
 feltelepíteni.  
 Mi is  
 ezekkel a használguk ezt (yosys-0.9, nextpnr-ice40, icestorm-utils),  
 csomagokkal közvetlenül is tudod linux alatt mindenféle  
 licenc-problema  
 meg  
 nyűg nélkül használni a barmit! Cserebe a forrás(kod)bol való  
 telepítés  
 nem  
 annyira egyszerű, elég sok a függőség, és nehezen derül ki  
 hogy  
 mi  
 az.  
 De  
 ebben  
 viszont a windowsos cuccokkal ellentétben tudok segíteni :)  
 Ugyanakkor  
 természetesen én is néztem a linuxos icecube2-t, és az is  
 működik  
 persze,  
 szóval  
 hajrá azzal is. Mondjuk az nekem kicsit nehezkés, a "bare  
 metal"  
 meg a  
 klasszikus "unix filozofia" dolgok után nekem ezeket az  
 IDE-eket  
 elég  
 nehéz  
 megszoknom :/ De az más kérdés ;)  
 A.  
 On Sat, 7 Nov 2020, Gábor Kreinicker wrote:  
 Kedves Pál András!  
 Köszönök mindent, ma hozzá is állok a dolgok alapos  
 tanulmányozásának.  
 Egy közeli ismerősöm ajánlásából elkezdtek föltelepíteni az  
 ubuntu-t,  
 mivel  
 a kollégiumban ha problémám van azonnal tud segíteni. Sajnos  
 nem  
 volt  
 zökkenőmentes a telepítés. Először a pendrive amin keresztül  
 telepíteni  
 akartam volt rossz, ami csak a 3. sikertelen kísérlet után  
 tűnt  
 föl.  
 Majd  
 miután sikerült, a Grub nem települt így nem tudtam  
 elindítani  
 a  
 rendszert.  
 Ezt ma sikerült megoldani, úgyhogy kezdődhet a linux tanulás  
 és  
 egyéb.  
 Az iCEcube2 telepítéséhez még várnom kell, mivel a

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Az elmúlt héten a következő módon haladtam:

Mivel nem igazán értettem, hogy működik az egész, így felkutattam az internetet. Találtam egy egész jó cikket, ami alapján megírtam egy villogó led kódot, és a feltöltéshez szükséges programokat (crt.s, linker.ld, makefile).

Az első problémám az volt, hogy a kódok, ami alapján megírtam az enyéim, nem f072-es stm32-re készült, hanem f103c8t6-ra. Amiket észrevettem, hogy nem stimmelnek (pl. cpu), azokat átírtam.

Elvileg a "make flash" paranccsal tudom feltölteni a kódot, ami megfelelően lefordul, és talán fel is töltődik. Csak azért talán, mert egy rövid ideig vált át zöldre az st-link ledje, de a kódot nem csinálja. Ugye az Ön kódjával a "make upload" paranccsal átváltott zöldre és vissza sem váltott. Gondolom azért, mert állandóan várta, hogy USART-on keresztül küldök-e valamit, de ez is nyugtalanított.

A kód az LD2 led villogtatása for-ciklusos várakozással. A datasheet alapján az LD2 led a PA5 vagy PB13 pinen van. Már ez is gyanús, hogy nem stimmel, de sajnos ezt még nem bírtam kikövetkeztetni.

A négy kódot kép formájában csatolom, de ha kell, az egész mappa tartalmát szívesen átküldöm!

Szóval így állok egyelőre, de egyre több szabadidőm van, így mostmár haladni fogok! Remélem Ön is jól áll a korábban említett dolgokkal!

Köszönettel: K. Gábor

Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. dec. 1., K, 23:19):

Szia Gábor!

Igen, az biztos hogy ez az egész problemakör meg programozasi metodus amit hasznalunk az elsore is (meg masodikra is) elegge melyviz. Szoval tenyleg, ha barhol elakadsz, szolj! Ram meg most zudultak a muhold-szeszereleskor felgyulemlett egyeb munkalatok, de igyekszem gyorsan reagalni.

No, tehat: melyik reszet nem latod at, pontosan? Ez igy osszesen ~350 sornyi kód, de elso korben ami reszedol erdekes az a main.c csak. A tobbin is vegig tudunk menni termeszetesen, mert elobbutobb tenyleg mindent erdemes jól atlatni! Ha ezeket mar atlatod akkor tenyleg sima ügy lesz mindenfele beagyazott alacsonyszintu programozas, kb programnyelvtol fuggetlenul is - szoval nyugodtan szand, szanjuk ra az idot.

Megmondom oszinten, a fopencookie()-t meg en sem hasznaltam igy ebben a formaban, csak egy nagyon hasonlot, az fdev\_setup\_stream()-et. Ami kimondottan AVR-ekhez van, lehet hogy ezt hasznaltad mar esetleg arduino-nal, lehet hogy nem. A lenyeg hogy az fopencookie() az ennek a kicsit altalanosabb valtozata, kicsit tobb overhead-del. Szoval ez mar tenyleg absz "uj dolog tanulasa" lenne reszemrol is ;)

A C az ugyan egy alacsony szintu nyelv, de ahhoz mar elegge magas szintű hogy tkp olasymí amit irsz vagyis kerdezel hogy ``stm32 programozásának alapjaiból" dolog az nem kimondottan C specifikus. Pl ez a fopencookie() letezik minden libc-architekturan, linuxon, vsz windozon is, kvazi resze a C standardnak. Az "STM32 specifikus dolgok" azok a hardver-regiszterek elerese. A lenyeg minden ilyen programozasnal hogy ezeket a specifikus, hardver-regiszter elerese dolgokat minel hamarabb "elrejtsuk" a program fő részei elől, es ugy használjuk mint egy "kesz konyvtarat". Aztan utana igy megyunk lepesrol lepesre.

A.

On Sat, 28 Nov 2020, Gábor Kreinicker wrote:

> Kedves Pál András!  
>  
> Szuper, ennek örülök!  
>

> Elnézést, hogy nem írtam a fejleményekről, ennek az az oka, hogy nem volt  
> jelentős fejlemény.  
>  
> Miután sikerült feltölteni a legutóbb a kódot, azt próbáltam megérteni, meg  
> próbáltam variálni a visszakapott adatokat.  
> Elkezdtem utána olvasni a fopencookie() dolgoknak, nagyon kevés és nagyon  
> bonyolult dolgot találtam, nem igazán jutottam előrébb.  
> A nucleo board-nak a datasheet-jét megkerestem és megnéztem, hogy hogyan  
> van az stm a többi dologgal összekötve.  
> Az UART dolgoknak is utánajártam, hogy ne legyen légbőlkapott az  
> alkalmazása.  
>  
> Sajnos az a helyzet, hogy még a kapott kódot sem látom át rendesen, így  
> nehezen tudok nekifogni. Ugye mi most C-t tanulunk, de a kód felépítését  
> leszámítva kevés igazán ismerősdolgot találok, aminek az oka gondolom az,  
> hogy board specifikusan programozunk. Így sajnos a 0. feladat is kifog  
> rajtam. Szuper lenne, ha tudna ajánlani egy olvasmányt az stm32  
> programozásának alapjaiból.  
>  
> Köszönettel: K. Gábor  
>  
> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 28., Szo, 18:03):  
>  
>> No, bármi fejlemény?  
>>  
>> Mi közben befejeztük a muholdat, szóval lassan majd megnyílik az ablak a  
>> következő projektek felé!  
>>  
>> A.  
>>  
>> On Sat, 21 Nov 2020, Gábor Kreinicker wrote:  
>>  
>>> Ááá, igen, ezt láttam fönt is, működik, köszönöm!  
>>>  
>>> Akkor igazából kezdhetek játszani. Egy útmutatást tud adni, mit érdemes  
>>> gyakorolni? Vagy tud adni feladatot?  
>>>  
>>> Köszönöm még egyszer!  
>>>  
>>> Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 21., Szo,  
>> 11:36):  
>>>  
>>>> Szia Gábor!  
>>>>  
>>>> Remek! Jaj de jó!  
>>>>  
>>>> Nade igen, a soros portos eleres nem annyira magától értetődő, és igen,  
>>>> lehet  
>>>> hogy a putty az más újsor-karakter konvenciót használ, ráadásul nem is  
>>>> echo-zza  
>>>> vissza vsz a terminalra a karaktereket (így valójában amit begépelsz azt  
>>>> nem  
>>>> látod mert nem a képernyőre írja hanem kiküldi a soros porton, és csak  
>>>> azt  
>>>> látod amit visszaküld: és az valóban már a nagybetűs változat mert ugye  
>>>> azt  
>>>> küldi vissza a board!).  
>>>>  
>>>> Az újsor-karakterek minden operációs rendszerben mások, ráadásul a három  
>>>> legelterjedtebben (linux, mac/osx, windows) is mind különbözők (rendre  
>>>> 0A, 0D,  
>>>> 0D+0A), lásd: <https://en.wikipedia.org/wiki/Newline#Representation>, és  
>> jo  
>>>> kérdés hogy a putty-nál mi az alapértelmezett... ez is bekavarhat. ((  
>> Ennek  
>>>> meg

```
>>>> regi tortente van es a klasszikus mechanikus irogepek koraig nyulik
>>>> vissza,
>>>> mert ott még nem volt a klasszikus "újsor", hanem csak a "kocsi vissza"
>>>> meg a
>>>> "soremelés" - hiszen mikor irogeppel irtal es kezdtel ujsort akkor a
>>>> papirt is
>>>> hajtanod kellett ("feed"-elni a hengert), plusz a hengert (carriage)
>> magát
>>>> visszaküldeni az alaphelyzetebe. Ezert nehany rendszer megorizte ezt a
>>>> kettosseget, mint a DOS, ill azon alapulo rendszerek, mig a
>> unix-variantsok
>>>> csak
>>>> egy karaktert hasznaltak a kettobol, es... hat, vagy ezt, vagy azt. ))
>>>>
>>>> Szoval: hasznalhatod a `socat` nevu programot is, de megmondom oszinten
>>>> azt nem
>>>> tudom fejbol hogy azzal hogyan megy a soros vonal összelövése. Viszont:
>> a
>>>> mellekelt programot szoktam en hasznalni (hazi fejlesztés) amit csak
>> siman
>>>> `make`-val le tudsz fordítani. Es akkor azzal imigyen:
>>>>
>>>> $ ./serial -c /dev/ttyACM1 -b 115200
>>>>
>>>> es akkor mehet is a dolog es majd ctrl+c-vel megszakítod amikor "keszen
>>>> vagy".
>>>> A minap fent a hegyen is ezt hasznaltuk, szoval ez visszaecho-zza azt
>> amit
>>>> beirtal es az ujsor-karaktereket is önkonzisztensen kezeli (vagyis
>>>> egyatalan
>>>> nem kezeli külön, emiatt nem különbözteti meg a különbözö típusu
>> entereket
>>>> vagy
>>>> soremelesek, de nem is alakítja at, mint ahogy ezekszerint a putty
>>>> teszi).
>>>>
>>>> Probald ki!
>>>>
>>>> A.
>>>>
>>>> On Sat, 21 Nov 2020, Gábor Kreinicker wrote:
>>>>
>>>>> Kedves Pál András! (legutóbb is elrontottam a sorrendet)
>>>>>
>>>>> Van fejlemény! A Makefile-ban rossz helyek voltak megjelölve forrásnak
>>>>> (középső kép, MAIN_LIBS sorok 32-34). Így már föltöltötte a kódot ami a
>>>>> csatolt kép bal oldalán látható.
>>>>>
>>>>> A putty nevű programmal értem el a boardot (jobb oldali kép). Ugye a
>> kód
>>>> az
>>>>> arról szól, hogy ha nyomok egy gombot, akkor világít az LD2 led, ha
>> pedig
>>>>> megnyomok egy újabb gombot, akkor pedig elalszik. Ez tökéletesen
>> működik.
>>>>> Azonban putty-n keresztül valamiért csak nagybetűket tudok írni,
>> valamint
>>>>> enter nyomás esetén a sor elejére ugrik. Elvileg el kéne tolnia a
>> betűket
>>>>> az abc-ben, de ez nem igazán történik meg. Ennek lehet oka a putty?
>>>>> Emlékszem, hogy ön nem ezt használta, de nem emlékszem, mivel tudott
>>>>> kommunikálni vele.
>>>>>
>>>>> Köszönöm a segítséget!
>>>>> K. Gábor
>>>>>
```

```

>>>> [image: image.png]
>>>>
>>>>
>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 20., P,
>>>> 19:59):
>>>>
>>>>> Szia Gábor!
>>>>> No, van barmi fejlemeny? :) Pontosán milyen linuxot is hasznalsz?
>> Ugy
>>>>> jobban
>>>>> ra tudunk keresni az adott library (*.a file) pontos helyére.
>>>>> Ez tenyleg elegge melyviz, szoval absz ne ijedj meg tenyleg ha
>> valami
>>>>> nem
>>>>> megy kapasbol es tenyleg szolj ha elakadsz valahol!
>>>>> A.
>>>>>
>>>>> On Thu, 19 Nov 2020, Gábor Kreinicker wrote:
>>>>>
>>>>>> Sajnos nem ez a gond, már telepítve van.
>>>>>>
>>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 19., Cs
>>>>>> 21:33):
>>>>>>
>>>>>>> Tedd fel a libnewlib-arm-none-eabi nevű csomagot!
>>>>>>>
>>>>>>> apt-get install libnewlib-arm-none-eabi
>>>>>>>
>>>>>>> On Thu, 19 Nov 2020, Gábor Kreinicker wrote:
>>>>>>>
>>>>>>>> Kedves András Pál!
>>>>>>>>
>>>>>>>>> Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül
>>>>>>>>> megoldanom,
>>>>>>>>> mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni
>> semmit a
>>>>>>>>> boardra. Amit leírt a korábbiakban, mindent többször átfutottam és
>>>>>>>>> ellenőriztem, de kifog rajtam egyelőre a Linux.
>>>>>>>>> El tudom érni, hogy az st-link lede zöldre váljon, ha beírom a
>>>> "st-info
>>>>>>>>> --probe" sort a terminálba. A headereket sikeresen bemásoltam a
>>>>>> megfelelő
>>>>>>>>> helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a
>>>> következő
>>>>>>>>> sorokat kapom a "make upload" esetén:
>>>>>>>>>
>>>>>>>>>> arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o
>>>>>>>>>> stm32_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a
>>>>>>>>>> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a
>>>>>>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a
>>>>>>>>>> arm-none-eabi-ld: cannot find
>>>>>>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libc_nano.a: No such file or
>>>>>>>>>> directory
>>>>>>>>>> arm-none-eabi-ld: cannot find
>>>>>>>>>> /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file or
>>>>>>>>>> directory
>>>>>>>>>> arm-none-eabi-ld: cannot find
>>>>>>>>>> /usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or
>>>>>>>>>> directory
>>>>>>>>>> make: *** [Makefile:54: main.elf] Error 1
>>>>>>>>>>
>>>>>>>>>>> Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka
>> előtt,
>>>>>>>>>>> ezért
>>>>>>>>>>> elnézést!
>>>>>>>>>>>

```

```
>>>>>>> Köszönettel: K. Gábor
>>>>>>>
>>>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 17., K,
>>>>>>> 20:36):
>>>>>>>
>>>>>>> A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a
>>>>>>> board-ra,
>>>>>>> nem a
>>>>>>> main.c-t eteted meg kozvetlenul. Probald ki a `make upload`
>>>> parancsot,
>>>>>>> az
>>>>>>> az
>>>>>>> ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a
>>>>>>> Makefile
>>>>>>> bogaraszasabol rajossz hogy hogy is van ez pontosan.
>>>>>>>
>>>>>>> Szoval a workflow az itten a kovetkezo:
>>>>>>> - A *.c fileokbol *.o (object) fileokat forditunk
>>>>>>> (arm-none-eabi-gcc)
>>>>>>> - Az egeszet osszelinkeljük (arm-none-eabi-ld)
>>>>>>> - Ez a tipusu mikrokontroller a 0x000000004-es címen található
>>>>>>> utasításra
>>>>>>> ugrik először, ott kezdi a program futtatását
>>>>>>> - Ez a tipusu mikrokontroller olyan hogy a 0x000000000
>>>> tartomanyra a
>>>>>>> 0x08000000 tartományt kepezi le alapertelmezesben
>>>>>>> - A linkelésnél így garantálnunk kell hogy legyen "valami" ami a
>>>>>>> 0x080000004-es címen ami értelmes _és_ kesobb meghivja a main()
>>>>>>> fuggvenyt
>>>>>>> -
>>>>>>> merthogy minden C program a main()-nal "kezdodik". Na, ezt a
>>>> feladatot
>>>>>>> latja el
>>>>>>> a crt.c - plusz még pár egyebet is, de azzal nem igazan kell
>>>>>>> foglalatostkodni,
>>>>>>> azzal majd kesobb ;)
>>>>>>> - A linkeles utan ki kell szednunk a keletkezo *.elf filebol a
>>>>>>> "lenyeget". Ezt
>>>>>>> csinalja a arm-none-eabi-objcopy, es ez hozza létre a *.hex-et.
>>>>>>> - Ezt a *.hex-et pedig az openocd-vel beleirhatjuk a
>>>> mikrokontroller
>>>>>>> flash-ébe
>>>>>>>
>>>>>>> Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo
>>>>>>> sorrendben.
>>>>>>> Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es
>> az
>>>>>>> openocd
>>>>>>> konfiguracioja sem annyira egyszeru, mert egyreszt meg kell
>> mondani
>>>>>>> neki
>>>>>>> hogy
>>>>>>> milyen tipusu programozót hasznalsz, masreszt milyen tipusu a
>> target
>>>>>>> mikrokontroller (azaz a programozonak tudnia kell hogy a
>>>> programozasi
>>>>>>> protokollon keresztul hogyan eri el a mikrokontroller memoriajat
>> es
>>>> ez
>>>>>>> fugghet
>>>>>>> a mikrokontroller konkret tipusától - bar egy csaladon belül ez
>>>> elegge
>>>>>>> egyseges, ezért elég csak azt mondani hogy "stm32f0x", es nem azt
>>>> hogy
>>>>>>> stm32f072cbt6). Plusz meg vannak egyeb finomsagok is, pl a flash-t
>>>>>>> ugye
```

```
>>>>>>>> minden
>>>>>>>> iras előtt torolni kell es csak utana írhat sz bele, stbstb. Ezeket
>>>>>>>> mind-mind
>>>>>>>> megcsinalja a Makefile-be elrejtett parancs-valami.
>>>>>>>>
>>>>>>>> Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra
>>>> ugyanigy
>>>>>>>> megy,
>>>>>>>> csak itt ez a "bare metal" módszer ezt is szuksegesse teszi hogy
>>>>>>>> lepesrol
>>>>>>>> lepesre menj(unk) vegig a lepeseken.
>>>>>>>>
>>>>>>>> Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott
>>>>>>>> hasznaltatok
>>>>>>>> valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy
>>>> hogy?
>>>>>>>>
>>>>>>>> A.
>>>>>>>>
>>>>>>>> On Tue, 17 Nov 2020, Gábor Kreinicker wrote:
>>>>>>>>
>>>>>>>> Kedves Pál András!
>>>>>>>>
>>>>>>>> Köszönöm a választ. Az stm32-re specializálódva, elvileg mindent
>>>>>>>> telepítettem. Ami még nem sikerült, az az openocd életre bírása.
>>>>>>>> Csatolok
>>>>>>>> egy képet, és bízom benne, hogy tud segíteni.
>>>>>>>> Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott az
>>>>>>>> st-link
>>>>>>>> csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a
>>>>>>>> feltöltésig
>>>>>>>> nem jutottam el.
>>>>>>>>
>>>>>>>> Köszönettel: K. Gábor
>>>>>>>> [image: Screenshot from 2020-11-17 19-53-48.jpg]
>>>>>>>>
>>>>>>>>
>>>>>>>> Andras Pal <apal@szofi.net> ezt írta (időpont: 2020. nov. 16.,
>>>> H,
>>>>>>>> 18:35):
>>>>>>>>
>>>>>>>> Szia Gábor!
>>>>>>>>
>>>>>>>> Ah, ertem, ez így akkor elsore tenyleg nem egyszerű. Viszont
>>>> ennel
>>>>>>>>> specifikusabban kene tudni hogy hol is akadtál el. Tehat:
>>>>>>>>> - telepites ment? Minden fent van?
>>>>>>>>> - az arm-es toolchain az ezen csomagokbol all:
>>>>>>>>> * gcc-arm-none-eabi
>>>>>>>>> * binutils-arm-none-eabi
>>>>>>>>> * libnewlib-arm-none-eabi,
>>>>>>>>> * libstdc++-arm-none-eabi-newlib.
>>>>>>>>> - ami meg hasznos csomag es kellhet:
>>>>>>>>> * build-essential (ebben van helyi C fordito is + a
>>>> `make`
>>>>>>>>> program)
>>>>>>>>> * srecord
>>>>>>>>> - openocd az csak siman ez:
>>>>>>>>> * openocd
>>>>>>>>>
>>>>>>>>> Mindegyiket siman `apt-get install ...` modon tudod telepiteni!
>>>> Ezek
>>>>>>>>> sikerultek? Ha ezek mind fent vannak, akkor az altalam mellekelt
>>>>>>>>> forrast
>>>>>>>>> mar le
>>>>>>>>> tudod fordítani es fel kell tudnod tölteni a kis board-ra. Ami
```

```
>> meg
>>>>>>>> kellhet
>>>>>>>> a
>>>>>>>> forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom
>>>>>>>> mellekelve.
>>>>>>>> En
>>>>>>>> konkrétan sima rendszergazdaként az
>>>> /usr/local/arm-none-eabi/include
>>>>>>>> alkovnytar
>>>>>>>> alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt
>>>>>>>> teszprogramban
>>>>>>>> levő Makefile-ban az elejen (kb 12-15ik sorban) levő
>>>>>>>>
>>>>>>>> ARM=/usr/local/arm-none-eabi
>>>>>>>>
>>>>>>>> sort modosítsd oda ahova betetted! Lehet simán nálad is, a
>>>> /home-od
>>>>>>>> alatt!
>>>>>>>> De
>>>>>>>> ugyis panaszkodik a fordító hogyha nem találja ;) Szóval legjobb
>>>>>>>> emlékeim
>>>>>>>> szerint ennyi kell, és ami meg hiányzik az ezeknek a
>> csomagoknak a
>>>>>>>> függősege -
>>>>>>>> szóval jó kell hogy legyen, legjobb emlékeim szerint.
>>>>>>>>
>>>>>>>> Az FPGA az sajnos tényleg egy fokkal bonyolultabb. Előbb nezzük
>>>> meg
>>>>>>>> ezt
>>>>>>>> az
>>>>>>>> ARM + STM32 témát, és akkor utána. Valójában azokat sem nehéz
>>>>>>>> feltenni,
>>>>>>>> de
>>>>>>>> kellhet pár olyan development header (pl libboost-csomag) amiket
>>>>>>>> feltetelez
>>>>>>>> hogy fent van de alából szinte biztos hogy nincs fent nálad.
>>>> Nálam
>>>>>>>> konkrétan
>>>>>>>> ezek vannak fent:
>>>>>>>>
>>>>>>>> libboost-atomic1.62-dev:amd64          install
>>>>>>>> libboost-atomic1.62.0:amd64              install
>>>>>>>> libboost-chrono1.62-dev:amd64             install
>>>>>>>> libboost-chrono1.62.0:amd64               install
>>>>>>>> libboost-date-time1.62-dev:amd64           install
>>>>>>>> libboost-date-time1.62.0:amd64            install
>>>>>>>> libboost-dev:amd64                        install
>>>>>>>> libboost-filesystem-dev:amd64              install
>>>>>>>> libboost-filesystem1.62-dev:amd64          install
>>>>>>>> libboost-filesystem1.62.0:amd64            install
>>>>>>>> libboost-iostreams-dev:amd64               install
>>>>>>>> libboost-iostreams1.62-dev:amd64           install
>>>>>>>> libboost-iostreams1.62.0:amd64             install
>>>>>>>> libboost-program-options-dev:amd64          install
>>>>>>>> libboost-program-options1.62-dev:amd64      install
>>>>>>>> libboost-program-options1.62.0:amd64        install
>>>>>>>> libboost-python-dev                        install
>>>>>>>> libboost-python1.62-dev                    install
>>>>>>>> libboost-python1.62.0                      install
>>>>>>>> libboost-regex1.62-dev:amd64                install
>>>>>>>> libboost-regex1.62.0:amd64                 install
>>>>>>>> libboost-serialization1.62-dev:amd64        install
>>>>>>>> libboost-serialization1.62.0:amd64          install
>>>>>>>> libboost-system1.62-dev:amd64               install
>>>>>>>> libboost-system1.62.0:amd64                 install
>>>>>>>> libboost-thread-dev:amd64                  install
```



```
>>>>>>>>>> libboost-thread1.62-dev:amd64          install
>>>>>>>>>> libboost-thread1.62.0:amd64              install
>>>>>>>>>> libboost1.62-dev:amd64                      install
>>>>>>>>>>
>>>>>>>>>> szoval ha azt mondod hogy
>>>>>>>>>>
>>>>>>>>>> # apt-get --no-install-recommends install libboost-atomic-dev
>>>>>>>>>> libboost-chrono-dev libboost-date-time-dev libboost-dev
>>>>>>>>>> libboost-filesystem-dev libboost-iostreams-dev
>>>>>>>>>> libboost-program-options-dev
>>>>>>>>>> libboost-python-dev libboost-regex-dev
>> libboost-serialization-dev
>>>>>>>>>> libboost-system-dev libboost-thread-dev
>>>>>>>>>>
>>>>>>>>>> (igen, ez így egy jó hosszú parancssor :)), akkor mindent is
>>>> feltesz
>>>>>>> ami
>>>>>>>>>> kell a
>>>>>>>>>> libboosthoz (es a sima lib-eket is felteszi ami kell a *-dev
>>>>>>>>>> csomagokhoz)
>>>>>>>>>>
>>>>>>>>>> Ami meg szinte biztos hogy kell az a: flex bison libtool. De
>>>> tenyleg
>>>>>>> ez
>>>>>>>>>> kicsit
>>>>>>>>>> macerasabb, szoval jobban kellene tudnom hogy hol akad el, es mi
>>>> az
>>>>>>> ami
>>>>>>>>>> hiányzik. Hanyas verzioju milyen linuxot is tettél pontosan?
>>>>>>>>>>
>>>>>>>>>> A.
>>>>>>>>>>
>>>>>>>>>> On Mon, 16 Nov 2020, Gábor Kreinicker wrote:
>>>>>>>>>>
>>>>>>>>>> Kedves Pál András!
>>>>>>>>>>
>>>>>>>>>> Elnézést a kései bejelentkezésért. A leírtakat telepítettem és
>> a
>>>>>>>>>> küldött
>>>>>>>>>> dolgokat olvastam. Sajnos sem az openocd-t, sem az icestorm
>>>>>>>>>> toolchain-t
>>>>>>>>>> nem sikerült "életre keltenem". Elvileg a yosys-0.9,
>>>> nextpnr-ice40
>>>>>>> és
>>>>>>>>>> icestorm-utils csomagokat is telepítettem a megadott módon. Tud
>>>>>>>>>> esetleg
>>>>>>>>>> küldeni valamilyen segítséget, hogyan tudnám programozni
>>>> ezekentúl
>>>>>>> a
>>>>>>>>>> lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék
>>>>>>>>>> segítséget
>>>>>>>>>> kérni,
>>>>>>>>>> telepíttem (elvileg, habár az elmúlt hétben a linuxot
>> próbáltam
>>>>>>>>>> kiismerni, de még nem érzem át a rendszert), de sehogy nem
>> tudom
>>>>>>>>>> elérni.
>>>>>>>>>>
>>>>>>>>>> Ami biztos, hogy a boardokat felismeri a gép (lsusb
>> paranccsal),
>>>>>>>>>> de a
>>>>>>>>>> programozásukig sajnos még nem jutottam el.
>>>>>>>>>>
>>>>>>>>>> Közben ha szabad, büszkén megemlítem, hogy a járás
>> diagnosztikai
>>>>>>>>>> eszközünkkel TDK csoportunkban elsők lettünk, így a Semmelweis
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
>>>> A felsoroltak közül az I2C-t alkalmaztam  
>>>> már  
>>>> többször  
>>>> is, a  
>>>> legutóbb  
>>>> a  
>>>> járásdiagnosztikai eszköz 7db szenzorának  
>>>> olvasásakor  
>>>> jelentett  
>>>> megoldást a  
>>>> számomra. Viszont az SPI működési elve  
>>>> alapján  
>>>> gyorsabbnak  
>>>> bizonyult,  
>>>> mely  
>>>> az eszköz számára nagyon fontos lenne,  
>> így  
>>>> arra  
>>>> fogunk  
>>>> áttérni.  
>>>> Az  
>>>> I2C-vel  
>>>> ellentétben az SPI-t még nem alkalmaztam  
>>>> gyakorlatban,  
>>>> de a  
>>>> működését  
>>>> már  
>>>> tanulmányoztam többször is. A többiről  
>> csak  
>>>> hallottam.  
>>>>  
>>>> Üdvözlettel:  
>>>> Kreinicker Gábor  
>>>>  
>>>> Andras Pal <apal@szofi.net> ezt írta  
>>>> (időpont:  
>>>> 2020.  
>>>> okt.  
>>>> 12.,  
>>>> H,  
>>>> 13:14):  
>>>>  
>>>> Szia Gábor!  
>>>>  
>>>> Koszi az összefoglalót! Na, akkor nezzük  
>>>> sorban:  
>>>> - az Arduinio az határozottn jo, mert  
>> mi  
>>>> eleg  
>>>> sokat  
>>>> dolgozunk  
>>>> AVR  
>>>> alapokon.  
>>>> Sok elektronikánk kimonodttn ugyanerre  
>> az  
>>>> MCU-ra  
>>>> (ATmega328P)  
>>>> epul,  
>>>> mint  
>>>> ami  
>>>> az Arduino Uno-ban van. Illetve most  
>>>> mostanaban  
>>>> egyre  
>>>> tobbet  
>>>> hasznalun  
>>>> AVR
```

[illegible]



[illegible]

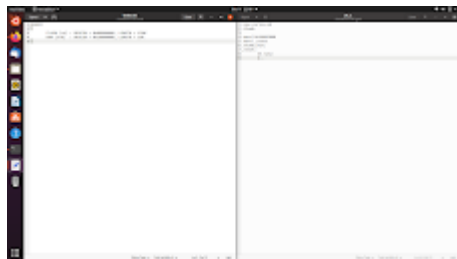
[illegible]

[illegible]

[illegible]

**2 melléklet**

**Screenshot from 2020-12-11 22-07-45.png**  
268K



**Screenshot from 2020-12-11 22-08-40.png**  
134K

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. december 12. 21:01

Szia Gábor!

No, koszi az összefoglalót! Igen, Az F0, F1, F4, ... sorozat tagjai között azért már szépszámmal akadnak különbségek, ráadásul pont az F1 ami kicsit kilóg olyan értelemben hogy az "alternate function" dolgait kicsit maskepp kell a regisztereken keresztül programozni.

At tudnad küldeni ezt amit irtal ebben a formában? A szkínshotok nem rosszak, de ilyenkor érdemes a teljes "bundle"-t is megosztanod, hatta az alapon könnyebben rajovunk a dolgok mikéntjére. F1-es sorozatból nekünk nincsenek tesztboardjaink, ezt az al-csaladot nem használtuk (még), csak F0 meg F4-eseket.

Az tok jó hogy így tenyleg alapjairól indítod el a fejlesztéseket. És meg ha elsore nehezkesen is indul, nagyon sokat lehet ezekből tanulni. En is így keztem ;) Viszont az is igaz hogy nem feltétlenül kell 0-ról kezdened, főleg ha vannak egyszerű példák amiből már ki tudsz indulni. Nezd meg ezt az stm32f072-test.tgz csomagot amit anno küldtem (elküldöm megegyszer, ld melléklet). Inkább azt javallom hogy ennek a módosításával-fejlesztéssel folytassuk az ismerkedést. Egyrészt mert ez tényleg F072-re van optimalizálva, másrészt meg ennek jobban ismerem már a lelkivilágot - noha ez is az egyik netes ilyen howto alapján készült (ami nem is kizárt hogy atfedest mutat azzal amit találtál).

Annyi lenne a keres, hogy már az elején módosítsd ebben az stm32\_rcc.c file-ban a

```
RCC->CR |= RCC_CR_HSEON;
```

sort, megpedig erre:

```
RCC->CR |= RCC_CR_HSEON|RCC_CR_HSEBYP;
```

Ez a csomag úgy működik továbbra is, rendben? Szóval ha lefordítod (`make`) majd feltöltöd (`make upload`) akkor az UART: kisbetű -> nagybetű átalakítás + LED-villogtatásnak mennie kell. Ezek akkor mennek, rendben?

Amit még szintén egy fontos megjegyzés hogy ezeknek a mikrokontrollereknek a teljes regiszterkészlete előre definiáltan elérhető a

```
#include <stm32f0xx.h>
```

header file-on keresztül, és nem kell így kezddel definiálnod, mint amit így valamelyik képernyőmentesen láttam nálad. Lehet csinálni persze úgy is, csak az úgy elég macerás ;) Mások már megcsináltak ezt előttünk. A lényeg a lényeg hogy mikor fordítod akkor ezeket a fileokat (stm32f0xx.h és társai) egy adott könyvtárban érdemes tárolni és a Makefile-ban a CFLAGS =... sorokban a -I kapcsoló után megadhatod. En konkrétan a /usr/local/arm-none-eabi/include/ könyvtárba pakolom őket, de persze lehet a saját home-on belül is bárhol!

Szóval próbáld meg ebből a csomagból kiindulni, mert amiket mutattam az meg a "bare metal" programozásnál is

alacsonyabb szintu. Ami persze nem baj, viszont az altalam is kuldott peldaban a teljes libc benne van mar (a newlib-eken keresztul) es azokat erdemes is kihasznalni - hiszen a C programnyelv legalapvetobb rendszerfuggetlen fuggvenyei mar igy elerhetőek lesznek!

Ha gondolod valamikor zoom-olhatnánk is egyet a temaban, hogyha valahol elakadsz akkor az interaktivabb kommunikacio az gyorsabb/hatekonyabb. Most sem a virus-helyzet, sem nekunk a muhold ezt nem konnyiti/konnyitette meg (sajnos), de azert probaljunk a korulmenyekhez kepest optimalizalni!

A.

On Fri, 11 Dec 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Az elmúlt héten a következő módon haladtam:

Mivel nem igazán értettem, hogy működik az egész, így felkutattam az internetet. Találtam egy egész jó cikket, ami alapján megírtam egy villogó led kódot, és a feltöltéshez szükséges programokat (crt.s, linker.ld, makefile).

Az első problémám az volt, hogy a kódok, ami alapján megírtam az enyéim, nem f072-es stm32-re készült, hanem f103c8t6-ra. Amiket észrevettem, hogy nem stimmelnek (pl. cpu), azokat átírtam.

Elvileg a "make flash" paranccsal tudom feltölteni a kódot, ami megfelelően lefordul, és talán fel is töltődik. Csak azért talán, mert egy rövid ideig vált át zöldre az st-link ledje, de a kódot nem csinálja. Ugye az Ön kódjával a "make upload" paranccsal átváltott zöldre és vissza sem váltott. Gondolom azért, mert állandóan várta, hogy USART-on keresztül küldök-e valamit, de ez is nyugtalanított.

A kód az LD2 led villogtatása for-ciklusos várakozással. A datasheet alapján az LD2 led a PA5 vagy PB13 pinen van. Már ez is gyanús, hogy nem stimmel, de sajnos ezt még nem bírtam kikövetkeztetni.

A négy kódot kép formájában csatolom, de ha kell, az egész mappa tartalmát szívesen átküldöm!

Szóval így állok egyelőre, de egyre több szabadidőm van, így mostmár haladni fogok! Remélem Ön is jól áll a korábban említett dolgokkal!

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. dec. 1., K, 23:19):

Szia Gábor!

Igen, az biztos hogy ez az egész problemakör meg programozási metódus amit használunk az elsőre is (meg másodikra is) elege melyviz. Szóval tenyleg, ha barhol elakadsz, szólj! Ram meg most zúdultak a muhold-szeszeresleskor felgyulemlett egyeb munkalatok, de igyekszem gyorsan reagálni.

No, tehát: melyik részt nem látod át, pontosan? Ez így összesen ~350 sornyi kód, de első körben ami reszedol érdekes az a main.c csak. A többin is végig tudunk menni természetesen, mert előbbutobb tenyleg mindent érdemes jól átlatni! Ha ezeket már átlatod akkor tenyleg sima ügy lesz mindenféle beagyazott alacsony szintű programozás, kb programnyelvtől függetlenül is - szóval nyugodtan szánd, szanjuk rá az időt.

Megmondom őszintén, a fopencookie()-t meg én sem használtam így ebben a formában, csak egy nagyon hasonlót, az fdev\_setup\_stream()-et. Ami kimondottan

AVR-ekhez van, lehet hogy ezt hasznaltad mar esetleg arduino-nal, lehet hogy nem. A lenyeg hogy az fopencookie() az ennek a kicsit altalanosabb változata, kicsit több overhead-del. Szoval ez mar tenyleg absz "uj dolog tanulasa" lenne  
reszemrol is ;)

A C az ugyan egy alacsony szintu nyelv, de ahhoz mar elegge magas szintű hogy tkp olasymi amit irsz vagyis kerdezel hogy ``stm32 programozásának alapjaiból" dolog az nem kimondoottan C specifikus. Pl ez a fopencookie() letezik minden libc-architekturan, linuxon, vsz windozon is, kvazi resze a C standardnak. Az "STM32 specifikus dolgok" azok a hardver-regiszterek elerese. A lenyeg minden ilyen programozasnal hogy ezeket a specifikus, hardver-regiszter elerese dolgokat minel hamarabb "elrejsuk" a program fő részei elől, es ugy használjuk mint egy "kesz konyvtarat". Aztan utana igy megyunk lepesrol lepesre.

A.

On Sat, 28 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Szuper, ennek örülök!

Elnézést, hogy nem írtam a fejleményekről, ennek az az oka, hogy nem volt jelentős fejlemény.

Miután sikerült feltölteni a legutóbb a kódot, azt próbáltam megérteni, meg próbáltam variálni a visszakapott adatokat. Elkezdtem utána olvasni a fopencookie() dolgoknak, nagyon kevés és nagyon bonyolult dolgot találtam, nem igazán jutottam előrébb. A nucleo board-nak a datasheet-jét megkerestem és megnéztem, hogy hogyan van az stm a többi dologgal összekötve. Az UART dolgoknak is utánajártam, hogy ne legyen légbőlkapott az alkalmazása.

Sajnos az a helyzet, hogy még a kapott kódot sem látom át rendesen, így nehezen tudok nekifogni. Ugye mi most C-t tanulunk, de a kód felépítését leszámítva kevés igazán ismerősdolgot találok, aminek az oka gondolom az, hogy board specifikusan programozunk. Így sajnos a 0. feladat is kifog rajtam. Szuper lenne, ha tudna ajánlani egy olvasmányt az stm32 programozásának alapjaiból.

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 28., Szo, 18:03):

No, bármi fejlemény?

Mi kozben befejeztuk a muholdat, szoval lassan majd megnyilik az ablak a kovetkezo projektek felé!

A.

On Sat, 21 Nov 2020, Gábor Kreinicker wrote:

Ááá, igen, ezt láttam fönt is, működik, köszönöm!

Akkor igazából kezdhettek játszani. Egy útmutatást tud adni, mit érdemes gyakorolni? Vagy tud adni feladatot?

Köszönöm mégegyszer!

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 21., Szo, 11:36):

Szia Gábor!

Remek! Jaj de jó!

Nade igen, a soros portos eleres nem annyira magától értetődő, és

igen,

lehet  
hogy a putty az más újsor-karakter konvenciót használ, ráadásul nem is  
echo-zza  
vissza vsz a terminalra a karaktereket (így valójában amit begépsz

azt

nem  
látod mert nem a képernyőre írja hanem kiküldi a soros porton, és

csak

azt  
látod amit visszaküld: és az valóban már a nagybetűs változat mert

ugye

azt  
küldi vissza a board!).

Az újsor-karakterek minden operációs rendszerben mások, ráadásul a három  
legelterjedtebben (linux, mac/osx, windows) is mind különbözőek

(rendre

0A, 0D,  
0D+0A), lásd: <https://en.wikipedia.org/wiki/Newline#Representation>,

es

jo  
kérdés hogy a putty-nál mi az alapértelmezett... ez is bekavarhat. ((  
Ennek

meg  
regi története van és a klasszikus mechanikus írógépek koráig nyúlik  
vissza,  
mert ott még nem volt a klasszikus "újsor", hanem csak a "kocsi

vissza"

meg a  
"soremelés" - hiszen mikor írógéppel írtál és kezdtél újsort akkor a  
papírt is  
hajtani kellett ("feed"-elni a hengert), plusz a hengert (carriage)

magát

visszaküldeni az alaphelyzetébe. Ezért néhány rendszer megőrizte ezt a  
kettőseget, mint a DOS, ill azon alapuló rendszerek, míg a

unix-variantok

csak  
egy karaktert használtak a kettőből, és... hat, vagy ezt, vagy azt. ))

Szóval: használhatod a `socat` nevű programot is, de megmondom

oszinten

azt nem  
tudom fejből hogy azzal hogyan megy a soros vonal összelövése.

Viszont:

a

mellekelt programot szoktam én használni (házi fejlesztés) amit csak

simán

`make`-val le tudsz fordítani. És akkor azzal így:

```
$ ./serial -c /dev/ttyACM1 -b 115200
```



es akkor mehet is a dolog es majd ctrl+c-vel megszakitod amikor  
 "keszen  
 vagy".  
 A minap fent a hegyen is ezt hasznaltuk, szoval ez visszaecho-zza azt  
 amit  
 beirtal es az ujsor-karaktereket is onkonzisztensen kezeli (vagyis  
 egyatalan  
 nem kezeli kulon, emiatt nem kulonbozteti meg a kulonbozo tipusu  
 entereket  
 vagy  
 soremeleseket, de nem is alakitja at, mint ahogy ezekszerint a putty  
 teszi).

Probald ki!

A.

On Sat, 21 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András! (legutobb is elrontottam a sorrendet)

Van fejlemeny! A Makefile-ban rossz helyek voltak megjelölve  
 forrásnak  
 ami a  
 (középső kép, MAIN\_LIBS sorok 32-34). Így már föltöltötte a kódot  
 csatolt kép bal oldalán látható.

A putty nevű programmal értem el a boardot (jobb oldali kép). Ugye a  
 kód  
 az  
 arról szól, hogy ha nyomok egy gombot, akkor világít az LD2 led, ha  
 pedig  
 megnyomok egy újabb gombot, akkor pedig elalszik. Ez tökéletesen  
 működik.  
 Azonban putty-n keresztül valamiért csak nagybetűket tudok írni,  
 valamint  
 enter nyomás esetén a sor elejére ugrik. Elvileg el kéne tolnia a  
 betűket  
 az abc-ben, de ez nem igazán történik meg. Ennek lehet oka a putty?  
 Emlékszem, hogy ön nem ezt használta, de nem emlékszem, mivel tudott  
 kommunikálni vele.

Köszönöm a segítséget!  
 K. Gábor

[image: image.png]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 20., P,  
 19:59):

Szia Gábor!

No, van barmi fejlemeny? :) Pontosan milyen linuxot is hasznalsz?

Ugy  
 jobban  
 ra tudunk keresni az adott library (\*.a file) pontos helyére.  
 Ez tenyleg elegge melyviz, szoval absz ne ijedj meg tenyleg ha  
 valami  
 nem  
 megy kapasbol es tenyleg szolj ha elakadsz valahol!

A.

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Sajnos nem ez a gond, már telepítve van.

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 19., Cs 21:33):

Tedd fel a libnewlib-arm-none-eabi nevű csomagot!

apt-get install libnewlib-arm-none-eabi

On Thu, 19 Nov 2020, Gábor Kreinicker wrote:

Kedves András Pál!

Tegnap és ma is próbáltam megoldani a feltöltést, hátha sikerül megoldanom,

semmit a mert már kezdem nagyon szégyellni, hogy nem tudok feltölteni

és boardra. Amit leírt a korábbiakban, mindent többször átfutottam

ellenőriztem, de kifog rajtam egyelőre a Linux.

EI tudom érni, hogy az st-link lede zöldre váljon, ha beírom a

"st-info

--probe" sort a terminálba. A headereket sikeresen bemásoltam a

megfelelő

helyre és átírtam a "Makefile"-ban amit írt. Ennek ellenére a

következő

sorokat kapom a "make upload" esetén:

arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o

main.o

stm32\_rcc.o /usr/lib/arm-none-eabi/newlib/armv6-m/libc\_nano.a

/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a

/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a

arm-none-eabi-ld: cannot find

/usr/lib/arm-none-eabi/newlib/armv6-m/libc\_nano.a: No such file

or

directory

arm-none-eabi-ld: cannot find

/usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a: No such file

or

directory

arm-none-eabi-ld: cannot find

/usr/lib/arm-none-eabi/newlib/armv6-m/libnosys.a: No such file or

directory

make: \*\*\* [Makefile:54: main.elf] Error 1

Komolyan nagyon szégyellem, hogy itt bénázok a tartalmi munka

előtt,

ezért

elnézést!

Köszönettel: K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 17.,

K,

20:36):

board-ra, A! nem! Az openocd az a binaris image-t (main.hex) tölti fel a

nem a

main.c-t eteted meg kozvetlenul. Probald ki a `make upload`

parancsot,

az

az

ami feltölt mindent, es akkor a kepernyore kiirt dolgokbol es a

Makefile

bogaraszasabol rajossz hogy hogy is van ez pontosan.

Szoval a workflow az itten a kovetkezo:  
- A \*.c fileokbol \*.o (object) fileokat forditunk  
(arm-none-eabi-gcc)  
- Az egeszet osszelinkeljuk (arm-none-eabi-ld)  
- Ez a tipusu mikrokontroller a 0x000000004-es címen található  
utasításra  
ugrik először, ott kezdi a program futtatását  
- Ez a tipusu mikrokontroller olyan hogy a 0x000000000  
tartományra a  
0x08000000 tartományt kepezi le alapertelmezesben  
- A linkelésnél így garantálnunk kell hogy legyen "valami"  
ami a  
0x080000004-es címen ami értelmes \_és\_ kesobb meghivja a main()  
fuggvenyt  
-  
merthogy minden C program a main()-nal "kezdodik". Na, ezt a  
feladatot  
latja el  
a crt.c - plusz még pár egyebet is, de azzal nem igazan kell  
foglalatoskodni,  
azzal majd kesobb ;)  
- A linkeles utan ki kell szednunk a keletkezo \*.elf filebol a  
"lenyeget". Ezt  
csinalja a arm-none-eabi-objcopy, es ez hozza létre a \*.hex-et.  
- Ezt a \*.hex-et pedig az openocd-vel beleirhatjuk a  
mikrokontroller  
flash-ébe  
  
Ha megnezed, a Makefile az ebbol mindent megcsinal, megfelelo  
sorrendben.  
Kicsit korulmenyesnek/hosszunak tunik, de ez egy ilyen jatek. Es  
az  
openocd  
konfiguracioja sem annyira egyszeru, mert egyreszt meg kell  
mondani  
neki  
hogy  
milyen tipusu programozót hasznalsz, masreszt milyen tipusu a  
target  
mikrokontroller (azaz a programozonak tudnia kell hogy a  
programozasi  
protokollon keresztul hogyan eri el a mikrokontroller memoriajat  
es  
ez  
fugghet  
a mikrokontroller konkret tipusától - bar egy csaladon belül ez  
elegge  
egyseges, ezért eleg csak azt mondani hogy "stm32f0x", es nem  
azt  
hogy  
stm32f072cbt6). Plusz meg vannak egyeb finomsagok is, pl a  
flash-t  
ugye  
minden  
iras előtt torolni kell es csak utana irhatsz bele, stbstb.  
Ezeket  
mind-mind  
megcsinalja a Makefile-be elrejtett parancs-valami.  
  
Egyebkent ez a fenti sema nagyjabol minden beagyazott cuccra  
ugyanigy  
megy,  
csak itt ez a "bare metal" modszert is szuksegesse teszi hogy  
lepesrol  
lepesre menj(unk) vegig a lepeseken.

Az arduino-s projektnel is vsz ugyanigy volt, csak gondolom ott hasznaltatok valami fejlesztokornyezetet ami mindent is megcsinalt, nem? Vagy

hogya?

A.

On Tue, 17 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm a választ. Az stm32-re specializálódva, elvileg

mindent

bírása.

telepítettem. Ami még nem sikerült, az az openocd életre

Csatolok

egy képet, és bízom benne, hogy tud segíteni.

Ami különös, hogy egyszer nem figyeltem oda és zöldre váltott

az

st-link

csatlakozás ledje. Ennek még utána járok, de az biztos, hogy a feltöltésig nem jutottam el.

Köszönettel: K. Gábor

[image: Screenshot from 2020-11-17 19-53-48.jpg]

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. nov. 16.,

H,

18:35):

Szia Gábor!

Ah, ertem, ez így akkor elsore tenyleg nem egyszerű. Viszont

ennel

specifikusabban kene tudni hogy hol is akadtál el. Tehat:

- telepites ment? Minden fent van?
- az arm-es toolchain az ezen csomagokbol all:
  - \* gcc-arm-none-eabi
  - \* binutils-arm-none-eabi
  - \* libnewlib-arm-none-eabi,
  - \* libstdc++-arm-none-eabi-newlib.
- ami meg hasznos csomag es kellhet:
  - \* build-essential (ebben van helyi C fordito is + a

`make`

program)

- \* srecord
- openocd az csak siman ez:
  - \* openocd

Mindegyiket siman `apt-get install ...` modon tudod

telepiteni!

Ezek

sikerultek? Ha ezek mind fent vannak, akkor az altalam

mellekelt

forrast

mar le

tudod forditani es fel kell tudnod tölteni a kis board-ra. Ami

meg

kellhet

a

forgatashoz azok az STM32-specifikus headerek. Ezeket elkuldom mellekelve.

En

konkretan sima rendszergazdakent az

/usr/local/arm-none-eabi/include

alkovnytar  
alatt tárolom ezeket, de barhova felteheted! Akkor a mellekelt  
teszprogramban  
levő Makefile-ban az elejen (kb 12-15ik sorban) levő

ARM=/usr/local/arm-none-eabi

sort modosítsd oda ahova betetted! Lehet simán nálad is, a

/home-od

alatt!

De  
ugyis panaszkodik a fordító hogyha nem találja ;) Szóval

legjobb

emlekeim

szerint ennyi kell, és ami meg hiányzik az ezeknek a

csomagoknak a

fuggósege -  
szóval jó kell hogy legyen, legjobb emlekeim szerint.

Az FPGA az sajnos tényleg egy fokkal bonyolultabb. Előbb

nezzük

meg

ezt

az

ARM + STM32 témát, és akkor utána. Valójában azokat sem nehéz

feltenni,  
de

kellhet pár olyan development header (pl libboost-csomag)

amiket

feltételez

hogy fent van de alapból szinte biztos hogy nincs fent nálad.

Nálam

konkrétan  
ezek vannak fent:

libboost-atomic1.62-dev:amd64	install
libboost-atomic1.62.0:amd64	install
libboost-chrono1.62-dev:amd64	install
libboost-chrono1.62.0:amd64	install
libboost-date-time1.62-dev:amd64	install
libboost-date-time1.62.0:amd64	install
libboost-dev:amd64	install
libboost-file-system-dev:amd64	install
libboost-file-system1.62-dev:amd64	install
libboost-file-system1.62.0:amd64	install
libboost-iostreams-dev:amd64	install
libboost-iostreams1.62-dev:amd64	install
libboost-iostreams1.62.0:amd64	install
libboost-program-options-dev:amd64	install
libboost-program-options1.62-dev:amd64	install
libboost-program-options1.62.0:amd64	install
libboost-python-dev	install
libboost-python1.62-dev	install
libboost-python1.62.0	install
libboost-regex1.62-dev:amd64	install
libboost-regex1.62.0:amd64	install
libboost-serialization1.62-dev:amd64	install
libboost-serialization1.62.0:amd64	install
libboost-system1.62-dev:amd64	install
libboost-system1.62.0:amd64	install
libboost-thread-dev:amd64	install
libboost-thread1.62-dev:amd64	install
libboost-thread1.62.0:amd64	install
libboost1.62-dev:amd64	install

szoval ha azt mondod hogy

```
# apt-get --no-install-recommends install libboost-atomic-dev
libboost-chrono-dev libboost-date-time-dev libboost-dev
libboost-filesystem-dev libboost-iostreams-dev
libboost-program-options-dev
libboost-python-dev libboost-regex-dev
libboost-serialization-dev
libboost-system-dev libboost-thread-dev
```

(igen, ez így egy jó hosszú parancssor :)), akkor mindent is

feltesz

ami

kell a  
libboosthoz (es a sima lib-eket is felteszi ami kell a \*-dev  
csomagokhoz)

Ami meg szinte biztos hogy kell az a: flex bison libtool. De

tenyleg

ez

kicsit  
macerasabb, szoval jobban kellene tudnom hogy hol akad el, es

mi

az

ami

pontosan?

hiányzik. Hányas verzióju milyen linuxot is tettél fel

A.

On Mon, 16 Nov 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a kései bejelentkezésért. A leírtakat telepítettem

és

a

küldött

icestorm

dolgokat olvastam. Sajnos sem az openocd-t, sem az

toolchain-t

nextpnr-ice40

nem sikerült "életre keltenem". Elvileg a yosys-0.9,

és

icestorm-utils csomagokat is telepítettem a megadott módon.

Tud

esetleg

küldeni valamilyen segítséget, hogyan tudnám programozni

ezekentúl

a

lattice board-ot? Ugyanígy az stm32 boardhoz is szeretnék

segítséget

kérni,

telepítettem (elvileg, habár az elmúlt hétben a linuxot

próbáltam

kiismerni, de még nem érzem át a rendszert), de sehogy nem

tudom

elérni.

Ami biztos, hogy a boardokat felismeri a gép (lsusb

paranccsal),

de a

programozásukig sajnos még nem jutottam el.

Közben ha szabad, büszkén megemlítem, hogy a járás

diagnosztikai

eszközünkkel TDK csoportunkban elsők lettünk, így a  
 Semmelweis Egyetem TDK  
 mellett készülhetünk az országos TDK-ra. Mivel már kész van  
 az eszköz,

 **stm32f072-test.tgz**  
 4K

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
 Címzett: Andras Pal <apal@szofi.net>

2020. december 13. 23:04

Kedves Pál András!

Köszönöm a segítséget, a zoom-os beszélgetés pedig nagyszerű lenne! Ha lehet, akkor 21-e után, addig végzem a húzóssabb vizsgákkal.

Természetesen a bevált programokat nem lecserélni szerettem volna, hanem megérteni. Ez alakul, de még jó lenne olvasni róla. Esetleg el tudná küldeni, ami alapján készítette? Csak jobb lenne F0 szériát vizsgálni, csak sajnos magamtól csak másmylet találtam.

Továbbá a megfelelő sort átírva működik továbbra is a kis-nagy betű átalakítás, viszont nem tiszta, ez min változtatott!

Habár lehet nem aktuális már, de akkor csatolom az általam talált és átalakítgatott feltöltő kódokat és a villogás kódot!

Ennek az üzenetnek az elküldése közben akadtak gondjaim, így ha bármi rendellenességet tapasztal az üzenetben, az emiatt lesz.

Köszönettel: K. Gábor

 **Sajat\_feltoltes\_villogas.zip**  
 2K

**Andras Pal** <apal@szofi.net>  
 Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. december 13. 23:34

No, igen, ez így akkor ebben a formában:

```
RCC_APB2ENR |= RCC_IOPCEN;
GPIOC_CRH  &= 0xFF0FFFFFFF;
GPIOC_CRH  |= 0x00200000;
while(1)
{ GPIOC_ODR |= GPIOC13;
  for (int i = 0; i < 500000; i++); // k  sleltet  s hossz  s for-ciklussal
  GPIOC_ODR &= ~GPIOC13;
  for (int i = 0; i < 500000; i++);
}
```

meg a fentebbi #define-okkal tenyleg teljesen j  , de probald meg akkor hasznalni azt a #include <stm32f0xx.h> format is. Abb  n is majdnem így, de nem egzaktul ugyanigy hivjak a regisztereket. Azaz a fenti program kb így fog kinezni:

```
RCC->AHBENR |= RCC_AHBENR_GPIOCEN;
GPIOC->MODER &= ~(3<<(2*13));
GPIOC->MODER |= (1<<(2*13));
while ( 1 )
{   GPIOC->ODR |= (1<<13);
    for (int i = 0; i < 500000; i++);
}
```

```

        GPIOC->ODR &= ~(1<<13);
        for (int i = 0; i < 500000; i++);
    }

```

illette nezd meg azt a board-ot amit adtam mert ott szerintem nem a PC13-ason van a led hanem a PA5-ösön. Azaz imigye kene villognia:

```

RCC->AHBENR |= RCC_AHBENR_GPIOAEN;
GPIOA->MODER &= ~(3<<(2*5));
GPIOA->MODER |= (1<<(2*5));
while ( 1 )
{
    GPIOA->ODR |= (1<<5);
    for (int i = 0; i < 500000; i++);
    GPIOA->ODR &= ~(1<<5);
    for (int i = 0; i < 500000; i++);
}

```

(azaz GPIOC helyett GPIOA es 13 helyett 5 van a megfelelo utasitasokban, de egyebkent ugyanaz - kiprobálható hasonló logikával az általad definiált változatokat is!)

Es itt ebben az <stm32f0xx.h> header file-ban ugy vannak definialva ezek a regiszter-keszletek hogy egy adott periferia-típus egy-egy példányhoz ugyanaz az objektum van rendelve. Itt az "objektum" egy regiszterkészlet, pl a GPIOx-eknél ebben van a MODER meg az ODR is. De ennek több példány van: GPIOA, GPIOB, GPIOC, ... Es így a C pointerekkel es strukturakkal, azaz a -> operátorral tudod az egyes példányokon (GPIOA, GPIOC) belüli egyes regisztereket ugyanugy elérni.

Ez elég hasznos akkor hogyha egy mikrokontrolleren belül több ugyanolyan, teljesen identikus periferia is van ;) Az viszont igaz, hogy az AVR-eknél például teljesen az általad is definiált módon használják a regisztereket. Szóval ígyis meg úgyis teljesen jó, de ha a "gyári" header fileokat akarjuk használni akkor itt az ARM-es mikrokontrollerekkel ez a struktúrás - pointeres (GPIOx->WHATEVERREG) jellegű megközelítés a megszokott.

A varakozos ciklusoknál egyebkent erdemes az "int" deklaracioja elé egy `volatile` kulcsszót is tennünk:

```
for ( volatile int i=0; i<500000; i++ );
```

Ugye ha jobban megnezed, akkor a sima számlálós ciklus valójában semmit nem csinál. Marmint túl azon hogy várakozna. Namost a C fordító (arm-none-eabi-gcc) számára viszont ez a sima számlálós ciklus semmilyen állapotváltozást nem okozhat a programban, így ha a fordító okos, akkor azt kioptimalizálja. A volatile kulcsszó azt implikálja a fordító számára hogy az "i" változót ebben az esetben (kisebb-nagyobb esellyel, de) más is befolyásolhatja. Így tenylegesen kénytelen lefuttatni 500ezerszer a ciklust, ha törik, ha szakad, mert csak úgy garantálható a program "helyessége". Itt a "helyességet" csak azért teszem idezőjelbe mert a "helyes" viselkedés az az hogy az i++ növekszik. Ezen kívül más nem befolyásolhatja - de a fordító ezt persze nem tud(hat)ja ilyenkor, nem tételezi ezt fel, es ezt hangsúlyozzuk a volatile kulcsszóval.

Ugye a volatile-nak akkor van jelentősége ha valami más tenylegesen is befolyásolja, ezért is kell nekünk is a periferia-változatokat volatile-ként deklarálni (hiszen ott a konkrét hardver \_tenyleg\_ megváltoztatja ezeket a változatokat). Itt a ciklus-keszeltetésnél nyilván nincs ennek jelentősége ebben az értelemben, de más értelemben viszont van...

Tenyleg, azt akartam is kérdezni, hogy a vizsgaidőszak az most hogy alakul? Mennyire lesz sűrű most ez, hogy látod?

A.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. december 14. 19:41

Kedves Pál András!

Köszönöm a segítséget!

Valóban, a led másik kimeneten volt, ha jól emlékszem le is írtam a gyanumat.

Kipróbáltam, tényleg szükség van a "volatile"-ra, nem gondoltam, hogy ilyen optimalizáció van jelen. Ez szerintem mostmár teljesen világos, ennek megfelelően haladok tovább!



Jövőhét keddig lesz két vizsgám, majd Január 4-én egy harmadik. Ha minden igaz utána egy hónapig teljes mértékben tudok ezzel foglalkozni.

Lehet legutóbb lemaradt, el tudná küldeni a linket, amivel megcsinálta feltöltő programokat?

Köszönettel: K. Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2020. december 17. 17:30

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Huh, bocsanat, erre elfelejtettem válaszolni... abban az stm32f072-test.tgz file-ban van egy Makefile, es abban van egy `make upload` target. Az hívja meg az `openocd`-t, es ugy van beallitva hogy ezt a Nucleo-testboardot fel tudja programozni. Vagy mire gondolsz?

A.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. december 17. 20:30

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Ugye a felröltéshez több lépcsőben le kell fordítani a .c fájlt. Azoknak a fájloknak érdekel a forrása (ami alapján elkészült), amelyekkel eljutunk a .c fájlól a feltöltésig.

Köszönettel: K. Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2020. december 17. 21:04

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Pontosan mit ertes azalatt hogy "azoknak erdekel a forrasa, amelyekkel eljutunk a feltoltesig"? Itten a folyamat relative osszetett es több lépcsős. Mindegyik amivel mi dolgozunk az nyílt forraskodu, tehat le tudod vadaszni a forrasat, es mindegyik része a linuxnak (debian, ubuntu). Csak kerdes hogy mennyire van kedved atragni egy több tízmillio soros, bootstrappinggel is fordított C fordító rendszert vagy ilyesmit ;)

De ettől függetlenül azért érdemes ismerni a munkafolyamatokat magukat. Nagyjából ez úgy a \*.c => (fordítás) \*.o => (linkelés) => \*.elf => (objektumok kimásolása) => \*.hex a sorrend, ezek a file-ok keletkeznek le ott abban az stm32f072-test könyvtárban is amikor beírod hogy `make`.

Es a legvegen itten a \*.hex az a memória-tartalom amit már a mikrokontroller mint olyan az megészik. Az `openocd` teszi bele a `make upload` kiadásakor.

Hogy ez miért ennyire bonyolult vagyis miért áll ebből a 3 fő lépésből az jó kérdés, leginkább történeti oka van. Es az egészben a fordítás a legeslegbonyolultabb, az már közvetlenül gépi kódot állít elő amit a mikrokontroller (vagy barmi CPU) megészik. A linkelés meg az objektum-kimásolás (`objcopy`) már sokkal-sokkal mechanikusabb es gyorsabb es akár ezeknek a forraskodja már meg is érthető magunkfajta földi halandóknak is.

A fordítás során (tehat a \*.o fileokban) már gépi kód van, viszont az összes hivatkozás még a levegőben lóg. Azaz egy változó vagy memóriaterület címe még nincs beírva a konkrét gépi kódba, hanem helyette csak 0-k vannak (vagy barmi egyéb placeholder érték) merthogy leginkább azért mert ezek konkrét helye még nem ismert a fordítási időben. A gépi kód ezen fordítás során kihagyott memória-területeire a linkelés során kerülnek be a konkrét memóriacímek, amikor a teljes futtatandó program összeáll.

Ezen felül persze vannak még finom részletek, konkrétan a memória felosztása, azon belül hogyan helyezkedik el a tárgykód (ami fut), az adatoknak mi az eloszlása, illetve milyen memória-modellekből eredő következmények vannak. Itt ilyesmire gondolj például hogy lehet olyan memóriaterület amit csak olvasol és nem módosítasz (leggyyszerűbb példa erre a karakteres-szöveges stringek), lehet olyan memóriaterület amit rendszeren írsz és

olvasol (a legtöbb mezei változó az ilyen), lehet olyan memóriaterület amit fenn kell tartanod és a fordító kell hogy garantálja azt hogy abban kezdetben 0-k legyenek, lehet hogy memóriaterület ahova dinamikusan foglalhatsz memóriát (es felszabadíthatsz, ha már nem kell), es lehet olyan memóriaterület is, amit írhatasz-olvashatsz, de nem 0-k a kezdeti értékei hanem valami más. Na, es ezeket a kulonbozo memoria-teruleteket is "el kell rendezgetned" okosan & ügyesen, mégmielőtt elkészül az a \*.hex file amit bele kell töltened a mikrokontrollerbe. A crt0.c vagy crt.s fileok is, illetve a \*.ld linker szkript fileok azok amik ugye benne vannak ebben a forrásfában (már az stm32f072-test könyvtárban is). Es ezért mondom/javallom hogy ezekkel (a konkrét main.ld-vel mert a crt0.c-vel) most első körben ne foglalkozzunk mert ezek "csak" a C nyelv standardjairól illetve a mikrokontroller memóriatérképéről gondoskodnak. Ha jobban érdekel, ezt is át tudjuk majd beszélgetni, szívesen elmondom -- főleg azért is mert noha míg most egy efféle ARM mikrokontrollernél nem igazán kell annyira ismerni ezeket, de pl bootloader-eknél vagy FPGA-ba tett processzorokhoz tarsított memoriák "felszabdalasana" már elege kell tudnunk ezeket is ismerni. Csak ezt nehezebb emailben, inkább majd élőben ha a vizsgák után össze tudunk futni ;)

A.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. december 21. 21:19

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Elnézést, csak mintha említette volna, hogy egy netes "tutorial" alapján készítette, erre kérdeztem rá, ezek szerint kicsit szerencsétlenül. A leírtak nagyon szuperek, teljesen világos, köszönöm!

Ma este haladtam a dolgokkal, az usarton küldött karaktereket toltam el az abc-ben, meg karaktereket helyettesítettem, meg ilyen kis dolgok, mígnem akadt egy problémám. Írtam egy kódot, ami a következő: usarton írok kis karaktereket, amiket visszaküld nagy karakterre változtatva (ugye a megszokott kód). Annyit változtattam, hogy visszaküldéskor karakterenként villant egyet. A visszaküldésből csak az első karakter érkezik meg, de amint egynél több karaktert írok be, semmit nem küld vissza. Villogni meg egyáltalán nem villog. Csatolom a kódot!

Ma végeztem a keményebb heteimmel, így 2020 végéig teljes energiámmal tudok ezzel foglalkozni, beleértve a beszélgetést is.

Köszönettel: K. Gábor

[Az idézett szöveg el van rejtve]

---

 **main.c**  
2K

---

**Andras Pal** <apal@szofi.net>

2020. december 21. 21:49

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

A main.c-hez:

tegyél be egy ilyet az USART konfiguráció részhez:

```
USART2->CR3 |= USART_CR3_OVRDIS;
```

ezt még az USART\_CR1\_UE elé ird be.

ez \_reszlegesen\_ megoldja a problémát, tehát valamit fogsz látni, azonban itten a helyzet egy kicsit trükkösebb. Az van, ugyanis, ha megnezed ezt a kódot, hogy emez történik: varakozol, varakozol, egyszercsak jön egy karakter. Akkor "beindul" a belső if() magja: beolvasod a karaktert, átalakítod nagybetűvé, majd \_vársz valamennyit\_ (közben villogtatsz), majd kiírod a karaktert. Namost, mi történik akkor ha itt az alahúzott rész alatt, azaz amikor a villogtatás miatt jön be egy másik, egy újabb karakter? Esetleg még egy? Hat, az bizony elveszik. Merthogy nem csinál vele senki semmit.

Ugye ezek a mikrokontrolleres perifériák alpból nagyon-nagyon primitívek: szólnak ha jön egy karakter, azt akkor úgy mihamarabb illik kiolvasni, mert ha nem olvasod ki időben és jön közben a következő karakter akkor abból bizony baj lehet. Namost, ez a "baj" sokféleképp jelentkezhet. Eme mikrokontroller eme perifériája (es általában az

összes UART receiver periferia) ilyenkor atmegy "overrun" (tulfutás, tulcsordulas) állapotba. Azaz itten valamit tennie kell a hardvernek hogy legalabb lassuk hogy valami van. A legegyszerubb ezt egy "overrun event" vagy "overrun error"-kent jelezni, de ez az STM32F0xx család ezenfelul ilyenkor még le is tiltja a receiver-t. Ezert latod azt hogy csak egy karakter erkezik, mert a masodik akkor csattan be mikor az első után varakozol. ez a USART\_CR3\_OVRDIS megakadalyozza hogy overrun esetben letiltson a receiver, viszont a karakterek ugyanugy el fognak veszni. Mas mikrokontrollerek (AVR, pl) is jelzik ha overrun van, de ott az az alapertelmezett hogy csak egy overrun-bit formajaban latod hogy legalabb 1 karakter elveszett.

Lesznek erre is megoldasok, sőt, igazabol a beágyazott programozásnak pont ez az egyik leglenyegesebb momentuma hogy memoria- és erőforrás korlátok mellett hogyan is tud kezelni ezeket a túlfutásokat. Ugye sok periféria az olyan hogy az adat az jön, gondolkodás és megállás nélkül (az összes aszinkron protokollal, azaz az UART, a CAN, az Ethernet, Wi-Fi ilyen, de a szinkron protokollok slave-oldalai, azaz SPI slave, I2C slave is ilyen, szoval valojaban eleg sok). Es ezt le kell tudni kezelni.

Szoval ezen gondolkozz el hogy hogyan lehetne algoritmizálni azt amit szeretnél csinálni. Illetve pontosabban foglamazva: eloszor is definitáld azt hogy hogy mit is szeretnél. Ugyanis az van hogy ha be is állítasz valami UART-sebesseget (ha jól latom, itt 115200 baud-ot allitasz be), akkor a karakterek atlagosan siman jöhetnek 0.1 milliszekundumonként(!) is. Szoval mit is jelent akkor az hogy mondjuk "0.2 masodpercre villanjon fel egy led" probléma? Azaz itt már a pontos feladatnak a jó leírása/megfogalmazása is előttünk van, hiszen az adat az így jön, ha kéri, ha nem: megpedig siman jöhet masodpercenként 10ezer karakter is! Azaz probaljuk meg ezt eloszor így fejben megfogalmazni: mit is szeretnenk pontosan? Kisbetu -> nagybetu, es villogas, de hogyan?

Illetve még egy ötlet a fejlesztéshez ami nekem nagyon bevált: csinálj egy

```
void msleep(int millisec)
```

```
{  
...  
}
```

fuggvenyt, ami azt csinálja hogy a (...) -ben jelzett, milliszekundumban értett időpontig varakozik. Azaz ha azt monddod hogy

```
while ( 1 )  
{  
    led_on();  
    msleep(500);  
    led_off();  
    msleep(500);  
}
```

akkor ez pontosan 1 masodpercenként villogtassa a led-et.

Kicsit maskepp is megfogalmazhatjuk: mennyit is kell irni a for (volatile int i=0; i<...; i++ ); ciklusban a ... helyére hogy a/ 1 millisec-ig 2/ x millisec-ig varakozzon a processzor?

Egy ilyen fuggveny nagyon hasznos debuggolashoz-fejlesztéshez. A műholdas elektronikák is ezzel kezdik a boot-olas utani életüket :)

A.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. december 22. 13:16

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Látom a problémát, igen. Azt hittem, hogy a kód karakterenként megnézi, átalakítja és visszaküldi a szavakat. Így viszont problémásabb, megoldást még nem találtam rá, de rajta vagyok.

Közben megírtam a késleltető (alvas) függvényt, meg írtam egy led függvényt is, így átláthatóbbá és egyszerűbbé téve a kódot. Átalakítottam továbbá úgy, hogy ha nem tudja átalakítani a beírt karaktert (tehát nem 'a' és 'z' közt van), akkor villant 4-et. Ami furcsaság, hogy nem emel sort a "serial" program. Emellett az enternél is bejelez érthető okokból, de ezt még nem sikerült kiküszöbölni. Szintén csatolom a kódot!

Ma még beleásom magam a még ki-nem-tisztult részekbe, és utána már ilyen kisebb gondjaim talán nem lesznek.

Köszönettel: K. Gábor

[Az idézett szöveg el van rejtve]

---

 **main.c**  
2K

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. december 22. 17:54

Egy gyors kérdés: a 741656 hogy jött ki?

A.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. december 22. 17:58

Több hosszat mértem, és visszaszámoltam

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. december 22. 18:05

Hogyan mérted a hosszat? Tudsz csinálni olyat is, például, hogy alszik a program valameddig (mondjuk 1000000 ciklusig), és utána küldesz egy karaktert (simán USART2->TDR='x'; módon). ha a `serial` kimenetét hexában + timestampmel ellátva íratod ki (`serial /dev/ttyACM0 -b 115200 -x -t`) akkor kiírja az időbélyeget elég pontosan (mikrosec pontossággal).

Es akkor azzal be tudod kalibrálni:

```
while ( 1 )  
{  
    USART2->TDR='x';  
    for ( volatile int i=0; i<1000000;i++ );  
};
```

Ugye az egy fontos dolog ennél hogy egy ilyen ciklus az \_valamennyi\_ órajelig tart, viszont ez a valamennyi az mindig egy egész szám. Szóval egy másodperc alatt 32000000 órajel megy le, és így a mért időpontból plusz az időkulonbsegek merésenek az arányosításából ki tudod számolni hogy egy ciklus-futás mennyi órajelig is tart. Ha mondjuk 27.999324 vagy 25.0023 jön ki így (a mérési hibák miatt) akkor lehet sejteni hogy jó úton jársz :)

Probald ki ezeket!

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. december 22. 19:20

Na igen, az időbélyegzőt nem sikerült megjelenítenem, úgyhogy fogtam egy stoppert, és egy percen keresztül számoltam, mennyit villant 500000 esetén, majd innen visszaszámoltam. Persze ez sokkal pontatlanabb, mint az időbélyegzős megoldás, de akkor megcsinálom úgy is, és kiderül, mennyire kapok más eredményt.

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. december 22. 19:59

Jaja, igen, valami ilyesmi! Probald ki a `serial`-t, ezzel az -x és -t kapcsolóval egyszerre (a többi az ugyanaz, a /dev/... és a baud rate megadása), és akkor tenyleg pontosan le tudod mérni az időt! Fogadni nem kell karaktereket, elég csak simán kiküldeni a TDR regiszter irásával.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. december 23. 19:02

Kedves Pál András!

Megcsináltam, valóban más érték jött ki. Új érték: 1999568. Az biztos, hogy ha serial-on vizsgálom időbélyegzővel, akkor ez az érték jó, de led villogtatáskor jelentőset torzít (nyilván emiatt kaptam közel 2.5-szeres értéket). Ilyen sok ideig kapcsolja be a ledet?

Viszont a következő napokban nem szerettem volna zavarni ilyen "problémákkal", így Kellemes Karácsonyi Ünnepeket, és utána folytatjuk!

Üdv: K. Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2020. december 23. 19:44

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Hm... pontosan mire gondolsz hogy "Ilyen sok ideig kapcsolja be a ledet?" Itt nem keveredik össze a két program? Ugyertem, az UART-os meg a led-villogtatos? Nem azért tart valami tovább mert közben az RXNE-re is vár?

A ledvillogtatásnak tényleg ez a legegyszerűbb és kvázi az egyetlen precíz módja ilyen busy wait jellegű varakozással:

```
while ( 1 )
{
    led_on();
    msleep(500);
    led_off();
    msleep(500);
}
```

csinálhatsz egyebkent ilyet is:

```
led_toggle()
{
    GPIOA->ODR ^= (1<<5);
}
```

és akkor így meg egyszerűbb:

```
while ( 1 )
{
    led_toogle();
    msleep(500);
}
```

de ha ez keveredik pl uart-varakozással akkor már ottan lehetnek érdekes dolgok...

On Wed, 23 Dec 2020, Gábor Kreinicker wrote:

Kedves Pál András!

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2020. december 28. 15:22

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Elnézést a késői válaszáért. Na igen, volt keveredés, végül azonban kaptam egy 727479 értéket. Szerintem ez már pontos lesz.

Igyekeztem továbbá minden részletét tudatosan használni, a kiindulási kódnak (kis-nagy betűs átalakítás), így még játszottam vele egy kicsit.

Neki szerettem volna fogni egy olyan függvény írásának, ami hasonlóan mint a "printf" működik (ha érti, mire gondolok) a standard c-ben. Hamar rájöttem, hogy az eddigi karakterenkénti átalakító kód nem igazán jó alap, így böngészttem, hátha találok egy nekem megfelelő kiindulási alapot. Sajnos még nem találtam, de már látom, hogy nem lesz egyszerű. Sok megoldást találtam, ami HAL alapú, így azoknak sok hasznát nem vettem.

Emellett találtam elég jó cikkeket UART-os kommunikációra két arduino között, így lehet megpróbálkozom egy Nucleo - Arduino kommunikációval.

Remélem rendben telt a Karácsony!

Üdvözzel:

K. Gábor

[Az idézett szöveg el van rejtve]

**Andras Pal** <apal@szofi.net>

2020. december 30. 0:35

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Igen, ez a 727479 sem tunik annyira rossznak: 32000000/727479 az 43.9875... szóval egészen közel van a 44-hez. De lehet hogy megmmernem többször is, hogy tenyleg legyen mondjuk egy ~tucatnyi szám es akkor jobban latod hogy tenyleg 44 körül szór

Egyebkent messze nem kizart hogy fugg az optimalizalastol. A gyakorlatban sok optimalizacio lehet, es a volatile változatokat elegge unhatekonyan (hatekonytalanul, szóval nem annyira hatekonyan ) optimalizalja a fordito pont azert mert minden elemi muveletnel fel kell teteleznie hogy valami megváltoztatja. Szóval nem irreális a 44 sem, bar elsore en csak olyan 20-30 kozotti szamot mondtam volna. A gyakorlatban ilyen busy wait jellegu varakozasra pont ezen bizonytalansagok kikuszobolesere (azaz hogy milyen fordito hogy optimalizal) talaltak ki az assembly betéteket. Ezeket altalaban valaki megirja es akkor lehet hasznalni, es nem igazan kell vacakolni. En pl egy ilyet hasznalok:

```
static inline void msleep(int count)
{
    while ( 0<count )
    {
        _delay_loop(F_CPU/8000);
        count--;
    }
}
```

Ahol az F\_CPU az 320000000 definialhato nálad:

```
#define F_CPU      320000000
```

Es a \_delay\_loop() meg így néz ki:

```
static inline void _delay_loop(uint32_t loops)
{
    __asm__ ("syntax unified");

    __asm__ volatile
    (
        "mov    r3,%[loops]    \n\t" /* load the initial counter    */
        "1:      \n\t"
        "nop      \n\t"
        "subs   r3, #1        \n\t"
        "bne    1b            \n\t"
        : /* empty output list */
        : [loops] "r" (loops) /* input to the asm routine */
        : "r3", "cc" /* clobber list */
    );
}
```

Hogy miért ilyen, azon végigmehetünk majd, de a fő assembly rutin az viszonylag egyértelmű:

```
        mov    r3, counter
loop_start:  nop
        subs   r3, 1
        bne    loop_start
```

szóval beteszi az r3 regiszterbe a counter értéket, es egy ciklusban egy utasítás erejéig nem csinál semmit, a masikban meg csökkenti az r3 értéket eggyel. es ha az eredmény nem azonosan egyenlő nullával (Branch if Not Equal /to zero/), akkor kezd elölrol a ciklust. Adott processzortípus meg architektura mellett ez mindig egy jól

meghatározott ideig tart. Hazi feladat lehet kimérni hogy ez meddig tart ;) Viszont ha megnezed, itt is ott van a volatile deklaráció - nem véletlenül!

Nade. Azt nem értem hogy ``ami hasonlóan mint a "printf" működik", itt mire gondoltal? Miert, itten mi lenne a megoldando feladat? Altalában az a celszerű a C-nél hogy a fő ANSI standard könyvtárat kihasználjuk. Az ANSI89-es (igen, 1989-es) szabványnak tartott könyvtár az abszolút alapnak tekinthető, az elfut egy kávéfőzőn is. De már a 99-es is lehet jó alap, bár mondjuk ott inkább a nyelvi elemek miatt érdemes kiterjeszteni a dolgokat, nem is a standard (libc) könyvtár "tudása" miatt.

Ettől értendő dolgozt, pl saját printf()-szerű implementációt persze lehet csinálni, de általában beagyazott rendszereken is csak akkor szoktak ha egyébként a libc-féle implementáció nem férne bele. Vagy annyira lassú az adott célfeladatra hogy nem. Namost ha azt mondom hogy az FPGA-ra írt saját (AVR) processzorhoz társított 2k-nyi programmemóriába is befér egy printf() plusz az összes kommunikációs csatlakozás, akkor azért láthatod hogy ez már azért elege alap - és elege jól összerakott ahhoz hogy érdemes használni ha van.

Nade egy arduino <-> nucleo uart kommunikáció valóban jól hangzik, ezzel is tudsz kísérletezni! Hajra ;) Miket próbalsz ki? Itt már barmi, ami ad életjelet az fasza lehet, nem biztos hogy érdemes túrlogozni az elején - de aztán később lehet bonyolítani.

Koszi, minden rendben volt a karácsonyi időszakkal, remelem feléd is! Főleg hogy így már mindjárt újev is van!

Jaj, és megegy ünnepretono kérdés... mikor is vegzel kb a vizsgákkal?

A.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2020. december 31. 11:37

Kedves Pál András"

A szórásvizsgálatot elvégeztem, így kb. maradt ez az érték.

Megnézve az ön várakozás függvényét, érthető a működése, azonban nem értem, hogy ez miért megbízhatóbb. Nagyon rosszul fogalmaztam, soros porton szerettem volna kiírni dolgokat. (Valamint a printf függvény kiíratását hogy lehet megjeleníteni? Mert ugye egy visual studioval szemben itt nem igazán van lehetőség konzolra kiírni.) Ezeket leszámítva még foglalkoztam a dolgokkal, de sajnos nem sikerült még előre haladnom.

Utolsó vizsgám Január 4-én lesz, onnantól (ha jól sikerül a vizsga) teljes energiámmal tudok majd foglalkozni bármilyen projekttel.

Üdv: Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2020. december 31. 12:53

Szia Gábor!

Ez a "miért megbízhatóbb az a \_delay\_loop() így assembly-ben megírva" mint az egyszerű for ciklus az több szempontból is érdekes. Egyrészt az igaz, hogy amennyiben tudod hogy mennyi ideig fut le (azaz hogy egy ciklusmag hany orajelig tart, mint itt a 44) addig pontosan ugyanannyira lesz megbízható. A gond az három helyen jelentkezhet:

- egyrészt az hogy most ez pont 44 oraciklus az kvazi attól függ hogy milyen fordító melyik verziójának milyen optimalizációjával fordítod. Ha bármelyik is változik akkor ez is változhat
- másrészt meg az hogy változhat hogy bizonyos utasítások (meg assembly szinten is) mennyi ideig tartanak, hany oraciklusig. függ a program memória elérésétől (latency, wait cycle), az instruction pipeline-tól. így ilyenertelembe egy assembly betét sem feltétlen megbízhatóbb de nagyságrendekkel kisebb az esélye ~3-4 utasításnal és ~8 orajelnel hogy belekeveredj egy ilyenbe mint ~20 utasításnal és ~40 orajelnel.
- a harmadik dolog amiért ez az asm-jelleg a preferált az a sebesség: egyszerűen ez gyorsabb, és emiatt precízebb időzítéseket lehet vele elérni ;)

Oke, nezek akkor majd egy egyszerű példát amivel tudsz sima printf()-et használni majd a nucleo-n is ezen a virtuális soros porton. Az a baj hogy pont most nincs nálam (és meg kb egy hétig nem lesz) egy ilyen board, de valahogy kikísérletezem hogy jó legyen ;]

A

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. január 11. 19:47

Kedves Pál András!

Először is elnézést, hogy el voltam tűnve, sajnos alulpontozták az utolsó vizsgám, így ma újra elmentem javítani, ami már megfelelően sikerült, ezzel lezárva a félévemet.

Persze közben igyekeztem haladni. Több délután keresgéltem printf() jellegű UART-on kiírató függvény után, azonban sok bare metal meg még több barkács függvényektől túltelített kódokat leszámítva nem igazán találtam semmi számomra használhatót, úgyhogy nekifogtam magam.

A kiindulási kódban (szeretnék így hivatkozni rá a későbbiekben) karaktereket változtattunk meg és írtunk ki UART-on, ezt vettem alapul. Írtam egy függvényt, amely egy sztringet karakterekre bont, majd karakterenként ír ki UART-on. Még nagyon kezdetleges a megoldás, de már hatalmas sikerélmény volt. (csatolom a kódot)

Innentől mostmár szabadabban fogok mozogni, hogy lesz egy "ismerős" függvény :D

Ha van javaslata a továbbhaladásról, vagy a függvényem finomításáról, akkor szívesen csinálom.

Üdvözlettel: Gábor

[Az idézett szöveg el van rejtve]

---

 **main.c**  
2K

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. január 12. 20:12



Kedves Pál András!

Közben feltűnt, hogy itt a 10 szekundumos várakozás jóval kevesebb a valóságban. Hogy lehet kevesebb?

Kerestem a függvényem hibáit, így csökkentettem a várakozások hosszán (főleg az 50 milliszekundum) valamint jobb átláthatóság érdekében az:

```
if (USART2->ISR & USART_ISR_RXNE)
```

sort is átvittem a függvénybe.

Mivel csak sztringet tudok kiírni, így számokat a "sprintf()" függvénnyel előtte sztringgé kell alakítsam. Próbálkoztam olyan megoldással, amely során két paraméteres a függvény (pl: kiirat("pelda kifejezes", s); ), amelynek az első paramétere lenne a kiírandó dolog, a második paramétere pedig a kifejezés típusa (pl: s --> sztring). Ekkor sajnos abba a problémába ütköztem, hogy a függvény első sorában meg kell adnom a paraméter típusát, ami nekem pont nem jó. Találtam olyasmi megoldást, hogy:

```
void kiirat (void* kifejezes, char tipus) {
    if (tipus == 'i') {
        char* kif
        sprintf(kif, "%i", kifejezes);
        [...] //innenről olyan, mint korábban küldtem
    }
    [...]
}
```

Sajnos ezzel a megoldással semmit nem csinált, így elakadtam.

Holnap reggel ezzel folytatom!

Üdv: Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2021. január 15. 22:41

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Huh, ne haragudj a kesei válaszaért - a hét elején még benne voltam egy-két dologban, de utána viszont jól elfelejtettem :/

No, egyrészt hogy haladjunk és tudj úgy tesztelni ilyen környezetben "mintha" mezei pc-s vagy ilyesmi környezetben vagyunk, csináltam gyorsan egy ilyen fopencookie() alapú printf() függvényt. Probáld ki! Ha megnezed, ez is simán azt az USART-ot (USART2-t, ami a Nucleo board-on is van kötve az USB-soros atalkitorra) használja, de így ezek a beállítások, a fopencookie(), az usart\_write() meg a setlinebuf() az leveszi a terhet és akkor a printf() mint olyan már úgy működik ahogy a nagykönyvben (sima sztenderd C-ben) működnie kell - úgy hogy az alapértelmezett stdout lesz az az U(S)ART amin a Nucleo board az USB-n keresztül tud egyszerűen kommunikálni.

Az fontos, hogy ez így ebben a formában csak irni tud (azaz csak a Nucleo tud adatot küldeni a gép felé), az stdin az ebben a formában nincs implementálva. Persze lehetne, meg majd nemsokára meg is fogjuk tudni csinálni - de ahhoz kell még egy pár egyéb dolog is (pl megszakításkezelés). Ezek sem bonyolultak.

A másik meg hogyha tulvagy a vizsgakon akkor majd diskuráljunk személyesen is valamikor, mert úgy rengeteget tudnánk haladni. A jövő hét első felén vsz Piskésen lesznek, ott is közbejött pár munka, de a hét második felétől már jó lesz! Most merrefele vagy így a vizsgák után? Itt Budapesten?

A.

[Az idézett szöveg el van rejtve]



**stm32f072-test.tgz**

4K

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2021. január 17. 11:07

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

Köszönöm szépen, nagy segítség!

Ránézésre az én UART-os printf() függvényemnél precízebb, de ettől sajnos nem jutottam tovább:

Sajnos miután átírtam a makefile-ban a szükséges flogokat, és definiáltam az F\_CPU-t valamint a BCD\_VERSION\_xx sorokat a main.c-ben, feltöltés közben a következő hibaüzeneteket kaptam (ha nem itt jelenik meg a kép, akkor a képre gondoltam):

```
kreinicker@kreinicker:~/Desktop/csfk/STM32/uart_print_PA$ make upload
arm-none-eabi-gcc -fno-common -mcpu=cortex-m0 -march=armv6-m -mthumb -masm-syntax-unified -DSTM32F072
-Wall -Wno-pointer-sign -O3 -I/usr/lib/arm-none-eabi/include -c main.c
main.c: In function 'main':
main.c:60:2: error: unknown type name 'cookie_io_functions_t'
  60 |   cookie_io_functions_t cookie_uart;
      |   ^
main.c:72:13: error: request for member 'read' in something not a structure or union
  72 |   cookie_uart.read=NULL;
      |             ^
main.c:73:13: error: request for member 'write' in something not a structure or union
  73 |   cookie_uart.write=usart_write;
      |             ^
main.c:74:13: error: request for member 'seek' in something not a structure or union
  74 |   cookie_uart.seek=NULL;
      |             ^
main.c:75:13: error: request for member 'close' in something not a structure or union
  75 |   cookie_uart.close=NULL;
      |             ^
main.c:77:11: warning: implicit declaration of function 'fopencookie' [-Wimplicit-function-declaration]
  77 |   stdout = fopencookie(USART2,"w",cookie_uart);
      |   ^
main.c:77:9: warning: assignment to '__FILE *' {aka 'struct __sFILE *'} from 'int' makes pointer from integer without a cast [-Wint-conversion]
  77 |   stdout = fopencookie(USART2,"w",cookie_uart);
      |   ^
make: *** [Makefile:51: main.o] Error 1
```

Utánajárás után a övetkező sort írtam be a kód elejére:

```
#define _GNU_SOURCE
```

Ezt követően a kód látszólag megfelelően lefordult és feltöltődött a board-ra, de serial-on semmi nem jelenik meg. Ebben szeretnék segítséget kérni.

A személyes találkozó részemről bármikor megoldható, de ha megoldható, legyen hétköznapi, + jövőhéten csütörtökön nem jó, de más egyelőre nincs betervezve. A kollégiumból novemberben ki lettünk költöztetve covid miatt, így azóta Balástyán (ithon) vagyok, de szívesen felvonatozom, nem tart sokból.

Üdv: Gábor

[Az idézett szöveg el van rejtve]

---

Andras Pal <apal@szofi.net>

2021. január 17. 11:52

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Szia Gábor!

Huha, na, akkor ez a személyes konzultációs logisztika valóban nem lesz egyszerű... most mondhatnám hogy zoom vagy ilyesmi, de pont erre ami nekünk kellene meg az is elegeg hatékonytalan tud lenni :/ Namajd ezen akkor agyaljuk. Balástya sincs annyira közel, bar ketsegtelen hogy a közvetlen vonat az segít ;) De ugye most mondhatnám hogy uljunk akkor be valahova a Nyugati környeken, de a virus-helyzet miatt ez sem opcio. Es onnan a normaafa az még jó messze van :/ Szoval ize.

Őőő, igen, annyi lehet itten a gond hogy mig az a nucleo-board az 8MHz-s orajellel rendelkezik, mi/en elegeg sok 12MHz-s orajelet használunk. Ez az stm32\_rcc-ben okozhat bonyodalmat, merthogy ottan egy fix PLL-szorozó van beallitva (amit atkuldtem abban konkretan RCC\_CFGR\_PLLMUL6 van, tehat a 8MHz-t meg 6x-ozza 48MHz-ra), es ugye az UART frekvencia is az orajeltol fugg. Szoval ezt nezd meg hogy jo legyen:

- az alap orajel nalad (a nucleo-nal 8MHz)
- nezd meg hogy azt mivel szorzod (PLL4 v. PLLMUL6 van-e ottan)
- es az USART2->BRR-t ennek megfeleloen allitsd be.

Azaz:

- vagy PLLMUL4 legyen es 278-as BRR;
- vagy PLLMUL6 legyen es 417-es BRR.

(persze lehet más is, csak a lenyeg hogy jól kiejtsd az orajel ;))

Az F\_CPU-t is ennek megfelelően állítsd be! Itt egy picit érdekesebb ez a dolog hogy hogysmint érdemes definiálni az F\_CPU-t, de most egyelőre ezzel nem lőnek nyulat hiszen az msleep()-en kívül más nem használja. Ugye a "logisztikai nehezítés" vagyis "paradigma-kerdes" itten az az hogy honnan is jön az F\_CPU. Egy nagyon egyszerű mikrokontrollernél (mint pl az AVR-ek/Arduino-k) van \_a\_ kulso orajel, annak van egy frekvenciája, es azzal ketyeg a CPU is. Itt viszont van ugyan egy kulso orajel (8MHz, konkrétan, ennél a Nucleo-sorozatnál) de azt at tudjuk állítani szoftversen a PLL segítségével... szóval ezt a "kulso" F\_CPU definíciót összhangba kell hozni a "belso" dolgokkal (magával a forráskóddal). Ez egyelőre nincs így meg, de majd a későbbiekben ezt is megcsináljuk kulturáltabbra!

A.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. január 17. 14:58

Kedves Pál András!

Valóban nem volt összeegyeztetve a PLL és a BRR, de sajnos ettől még ugyanazt csinálja (tehát, hogy semmit). Sajnos meg nem sok mindent tudok ezekről az fopencookie dolgokról, így lehet butaságot kérdezek, de a while(1) részen belül szereplő printf() függvényt hogy hozza összefüggésbe a kód at UART-os kiíratással? Nekem úgy tűnik, mintha használnánk a függvényt, de nincs ami megjelenítené.

Továbbá ha serial-on küldök szöveget, akkor néha visszaküldi azt, amit begépelek, ez is furcsa (pláne, hogy tényleg nem mindig csinálja, de nem is kéne neki csinálni)

Tényleg kicsit nehézkes az utazás, de a haladás érdekében nem okoz gondot!

Üdv: Gábor

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. január 22. 11:38

Kedves Pál András!

A korábbi napok folyamán próbáltam rájönni, mi lehet a gond. Amire rájöttem, hogy az előző üzenetemben leírt "printf()-es észrevételem" teljesen butaság. Próbáltam összeegyeztetni mind az F\_CPU - PLL - BRR dolgokat, mely sikerült, de továbbra sincs semmi jele a működésnek.

Találtam több, hasonló megoldást, de azok hasonlóan semmit sem csináltak. Eszembe jutott, hátha csak a "serial" program nem kedveli, így Putty-n keresztül megpróbáltam, de ez sem lehet probléma. Mivel a villogás sincs jelen, így biztosan nem a megjelenítéssel van a gond, hanem a függvény ill. az fopencookie körül, melyet még nem látok át teljesen, így sajnos elakadtam.

Ennek ellenére igyekezni fogok mihamarabb megoldást találni és/vagy haladni!

Már alig várom, hogy lássam sorszámozva kiírva az "xyz"-t :D

Üdv: Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2021. január 22. 17:29

Szia Gábor!

Hu, ne haragudj de egy hulye/gyors kérdés: azt a forrást amit az elobb küldtem (meg pont egy hete, stm32f072-test.tgz néven) az úgy 1:1-ben mennyire tudod feltenni és használni és az mennyire ad életjelet? Csak mert azt

egy egzaktul ugyanolyan nucleo-boardon teszteltem ki amit adtam, szoval annak mennie kell.

Persze utana saját modositasokat tudsz csinálni meg fejleszteni meg ilyesmi, de van valahol valami szűk keresztmetszet amit így nehéz belokalizálni. Ha van egy mukodo valtozatod es van egy nem mukodo valtozatod, akkor utana elkezdheted "közelíteni" a két forrásfát, es akkor elobbutobb csakcsak meglesz hogy mi az elakadás!

A.

[Az idézett szöveg el van rejtve]

Gábor Kreinicker <kreinickergabor@gmail.com>2021. január 22. 19:41

Címzett: Andras Pal <apal@szofi.net>

Kedves Pál András!

1:1-ben ezt a hibaüzenetet kapom:

```
kreinicker@kreinicker:~/Desktop/csfk/STM32/fopencookie_PAS make upload
arm-none-eabi-gcc -fno-common -mcpu=cortex-m0 -march=armv6-m -mthumb -masn-syntax-unified -DSTM32F072 -Wall -Wno-pointer-sign -O3 -I/usr/lib/arm-none-eabi/include -c crt0.c
arm-none-eabi-gcc -fno-common -mcpu=cortex-m0 -march=armv6-m -mthumb -masn-syntax-unified -DSTM32F072 -Wall -Wno-pointer-sign -O3 -I/usr/lib/arm-none-eabi/include -c main.c
main.c: In function 'usart_write':
main.c:19:17: error: 'F_CPU' undeclared (first use in this function)
   19 |     { _delay_loop(F_CPU/8000L);
     |               ^~~~~~
main.c:19:17: note: each undeclared identifier is reported only once for each function it appears in
main.c: In function 'usart_write':
main.c:63:5: warning: this 'while' clause does not guard... [-Wmisleading-indentation]
   63 |     { while ( ! (USART->ISR & USART_ISR_TXE) );
     |     ^~~~~~
main.c:64:2: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the 'while'
   64 |     USART->TDR=buf;
     |     ^~~~~~
main.c: In function 'main':
main.c:75:2: error: unknown type name 'cookie_io_functions_t'
   75 |     cookie_io_functions_t cookie_uart;
     |     ^~~~~~~~~~~~~~~~~~~~~
main.c:106:13: error: request for member 'read' in something not a structure or union
  106 |     cookie_uart.read=NULL;
     |             ^
main.c:107:13: error: request for member 'write' in something not a structure or union
  107 |     cookie_uart.write=usart_write;
     |             ^
main.c:108:13: error: request for member 'seek' in something not a structure or union
  108 |     cookie_uart.seek=NULL;
     |             ^
main.c:109:13: error: request for member 'close' in something not a structure or union
  109 |     cookie_uart.close=NULL;
     |             ^
main.c:111:9: warning: implicit declaration of function 'fopencookie' [-Wimplicit-function-declaration]
  111 |     stdout=fopencookie(USART2,"w",cookie_uart);
     |     ^~~~~~
main.c:111:8: warning: assignment to '___FILE *__s' (aka 'struct __sFILE *') from 'int' makes pointer from integer without a cast [-Wint-conversion]
  111 |     stdout=fopencookie(USART2,"w",cookie_uart);
     |     ^~~~~~
make: *** [Makefile:51: main.o] Error 1
```

Pontosabban nem 1:1-ben, mert a makefile-ben át kell javítsam azokat a dolgokat, melyeket még a legelején is át kellett javítsam. Azért is különös, hogy Önnél működött, mert az F\_CPU nem volt definiálva (ezt a terminál jelezte is, ahogy a képen látszik). Ezt ha javítom, akkor a többi hibaüzenet továbbra is marad. Ahogy korábban le is írtam, azok megszűnnek, ha a kód elejére beírom a:

```
#define _GNU_SOURCE
```

sort. Ekkor a program feltöltődik a board-ra, de semmilyen életjelet nem ad. Azért, hogy látszódjon, ezt is befotóztam:

```
kreinicker@kreinicker:~/Desktop/csfk/STM32/fopencookie_PAS make upload
arm-none-eabi-gcc -fno-common -mcpu=cortex-m0 -march=armv6-m -mthumb -masn-syntax-unified -DSTM32F072 -Wall -Wno-pointer-sign -O3 -I/usr/lib/arm-none-eabi/include -c crt0.c
arm-none-eabi-gcc -fno-common -mcpu=cortex-m0 -march=armv6-m -mthumb -masn-syntax-unified -DSTM32F072 -Wall -Wno-pointer-sign -O3 -I/usr/lib/arm-none-eabi/include -c main.c
main.c: In function 'usart_write':
main.c:66:5: warning: this 'while' clause does not guard... [-Wmisleading-indentation]
   66 |     { while ( ! (USART->ISR & USART_ISR_TXE) );
     |     ^~~~~~
main.c:67:2: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the 'while'
   67 |     USART->TDR=buf;
     |     ^~~~~~
arm-none-eabi-gcc -fno-common -mcpu=cortex-m0 -march=armv6-m -mthumb -masn-syntax-unified -DSTM32F072 -Wall -Wno-pointer-sign -O3 -I/usr/lib/arm-none-eabi/include -c stm32_f072.c
arm-none-eabi-gcc -fno-common -mcpu=cortex-m0 -march=armv6-m -mthumb -masn-syntax-unified -DSTM32F072 -Wall -Wno-pointer-sign -O3 -I/usr/lib/arm-none-eabi/include -c stm32_f072.o
arm-none-eabi-objcopy -O binary main.elf main.hex
srec_cat main.hex -I -crop 0x00000000 0xFFFFFFFF -o main.hex -I
openocd -f interface/stlink-v2-if.cfg -c "transport select hla_swd" -f target/stm32f072.cfg -c "adapter_khz 100" -c "reset_config srst_only srst_nogate" -c init -c "reset halt" -c "stm32f072 unlock 0" -c "
reset halt" -c "flash write_image erase main.hex 0x0" -c "reset" -c "exit" || exit 0
Open On-chip Debugger 0.10.0
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html
hla_swd
Info: The selected transport took over low-level target control. The results might differ compared to plain JTAG/SWD
adapter speed: 1000 khz
adapter_nsrst_delay: 100
none separate
adapter speed: 100 khz
srst_only separate srst_nogate srst_open_drain connect_deassert_srst
Info: clock speed 100 khz
Info: STLINK V2 JTAG V29 API V2 SWIM V18 VID 0x0403 PID 0x374B
Info: using stlink apb v2
Info: Target voltage: 3.253061
Info: stm32f072.cpu: hardware has 4 breakpoints, 2 watchpoints
Info: Unable to match requested speed 1000 khz, using 950 khz
Info: Unable to match requested speed 1000 khz, using 950 khz
adapter speed: 950 khz
target halted due to debug-request, current mode: Thread
xPSR: 0x10000000 pc: 0x00000010 msp: 0x20004000
Info: device ID = 0x20010448
Info: flash size = 128kbytes
target halted due to breakpoint, current mode: Thread
xPSR: 0x01000000 pc: 0x00000010 msp: 0x20004000
stm32x unlocked.
INFO: a reset or power cycle is required for the new settings to take effect.
Info: Unable to match requested speed 1000 khz, using 950 khz
Info: Unable to match requested speed 1000 khz, using 950 khz
adapter speed: 950 khz
target halted due to debug-request, current mode: Thread
xPSR: 0x10000000 pc: 0x00000010 msp: 0x20004000
auto erase enabled
target halted due to breakpoint, current mode: Thread
xPSR: 0x01000000 pc: 0x00000010 msp: 0x20004000
wrote 14336 bytes from file main.hex in 0.648942s (21.574 KiB/s)
Info: Unable to match requested speed 1000 khz, using 950 khz
Info: Unable to match requested speed 1000 khz, using 950 khz
adapter speed: 950 khz
```

Persze amikor én próbáltam "megjavítani" akkor a warning és a note is megszűnt csupán azzal, hogy formailag számomra átláthatóbbá tettem.  
Valójában életjelet ad, habár nem egészen rendeltetésszerűt, pontosabban ha küldök karaktereket serial-on, akkor (szerintem) véletlenszerűen ugyanazt visszaküldi:

```
kreinicker@kreinicker:~/Desktop/csfk/STM32/serial$ ./serial -c /dev/ttyACM0 -b 1
15200

egy
egy
ketto
harom
negy
ot
ot
hat
het
nyolc
nyolc
kilenc
tiz
```

Ekkor egyessével írtam a számokat. Természetesen a baud-rate és a PLLMUL stimmelnek. Valóban egyszerű lett volna a megoldás, ha fejlesztés közben veszik el, de sajnos nem ez a helyzet.

Remélem jól látható és érthető a probléma és hamar megoldást találunk!

Üdv: Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2021. január 23. 17:06

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Hm... igen, úgy látom hogy az F\_CPU-t is definálni kell, meg a \_GNU\_SOURCE-t is, valóban. A \_GNU\_SOURCE az sima ügy, de az F\_CPU-nak értéket is kell adni. Ennek viszont az értéket meg kell adni. A BRR=417-es beállítás akkor lesz jó ha a RCC\_CFGR\_PLLMUL6-t hasznalod az alap 8MHz-s orajel 6x-ozasara es így az F\_CPU-t 48000000-nak definíalod.

A legegyszerűbb ezt úgy hogy hozzáírod a gcc parancssorához:

```
-D_GNU_SOURCE -DF_CPU=48000000
```

Persze egy

```
#define F_CPU 48000000
```

is teljesen jó!

Hm... a visszaecho-zás az viszont furcsa. Mindenesetre nezd meg ezeket!

On Fri, 22 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

1:1-ben ezt a hibaüzenetet kapom:

[image: image.png]

Pontosabban nem 1:1-ben, mert a makefile-ban át kell javítsam azokat a dolgokat, melyeket még a legelején is át kellett javítsam.  
Azért is különös, hogy Önnél működött, mert az F\_CPU nem volt definiálva (ezt a terminál jelezte is, ahogy a képen látszik). Ezt ha javítom, akkor a többi hibaüzenet továbbra is marad. Ahogy korábban le is írtam, azok megszűnnek, ha a kód elejére beírom a:

```
#define _GNU_SOURCE
```

sort. Ekkor a program feltöltődik a board-ra, de semmilyen életjelet nem ad. Azért, hogy látszódjon, ezt is befotóztam:

[image: image.png]

Persze amikor én próbáltam "megjavítani" akkor a warning és a note is megszűnt csupán azzal, hogy formailag számomra átláthatóbbá tettem. Valójában életjelet ad, habár nem egészen rendeltetésszerűt, pontosabban ha küldök karaktereket serial-on, akkor (szerintem) véletlenszerűen ugyanazt visszaküldi:

[image: image.png]

Ekkor egyessével írtam a számokat. Természetesen a baud-rate és a PLLMUL stimmelnek. Valóban egyszerű lett volna a megoldás, ha fejlesztés közben veszik el, de sajnos nem ez a helyzet.

Remélem jól látható és érthető a probléma és hamar megoldást találunk!

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2021. jan. 22., P, 17:29):

Szia Gábor!

Hu, ne haragudj de egy hulye/gyors kérdés: azt a forrasfat amit az elobb kuldtam (meg pont egy hete, stm32f072-test.tgz néven) az ugy 1:1-ben mennyier tudod feltenni es hasznalni es az mennyire ad életjelet? Csak mert azt egy egzaktul ugyanolyan nucleo-boardon teszteltem ki amit adtam, szoval annak mennie kell.

Persze utana saját módosításokat tudsz csinálni meg fejleszteni meg ilyesmi, de van valahol valami szűk keresztmetszet amit így nehéz belokalizálni. Ha van egy mukodo változatod es van egy nem mukodo változatod, akkor utana elkezdheted "közelíteni" a két forrásfát, es akkor elobbutobb csakcsak meglesz hogy mi az elakadás!

A.

On Fri, 22 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

A korábbi napok folyamán próbáltam rájönni, mi lehet a gond. Amire rájöttem, hogy az előző üzenetemben leírt "printf()-es észrevételem" teljesen butaság. Próbáltam összeegyeztetni mind az F\_CPU - PLL - BRR dolgokat, mely sikerült, de továbbra sincs semmi jele a működésnek.

Találtam több, hasonló megoldást, de azok hasonlóan semmit sem csináltak. Eszembe jutott, hátha csak a "serial" program nem kedveli, így Putty-n keresztül megpróbáltam, de ez sem lehet probléma. Mivel a villogás sincs jelen, így biztosan nem a megjelenítéssel van a gond, hanem a függvény ill. az fopencookie körül, melyet még nem látok át teljesen, így sajnos elakadtam.

Ennek ellenére igyekezni fogok mihamarabb megoldást találni és/vagy haladni!

Már alig várom, hogy lássam sorszámozva kiírva az "xyz"-t :D

Üdv: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2021. jan. 17., V, 14:58):

Kedves Pál András!

Valóban nem volt összeegyeztetve a PLL és a BRR, de sajnos ettől még ugyanazt csinálja (tehát, hogy semmit)  
Sajnos meg nem sok mindent tudok ezekről az fopencookie dolgokról, így lehet butaságot kérdezek, de a while(1) részen belül szereplő printf() függvényt hogy hozza összefüggésbe a kód at UART-os kiíratással? Nekem

úgy tűnik, mintha használnánk a függvényt, de nincs ami megjelenítené.

Továbbá ha serial-on küldök szöveget, akkor néha visszaküldi azt, amit begépelek, ez is furcsa (pláne, hogy tényleg nem mindig csinálja, de nem is kéne neki csinálni)

Tényleg kicsit nehézkes az utazás, de a haladás érdekében nem okoz gondot!

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2021. jan. 17., V, 11:52):

Szia Gábor!

Huha, na, akkor ez a személyes konzultacios logisztika valoban nem lesz egyszerű... most mondhatnam hogy zoom vagy ilyesmi, de pont erre ami nekunk kellene meg az is elegge hatékonytalan tud lenni ./ Namajd ezen akkor agyaljuk.  
Balástya sincs annyira kozel, bar ketsegtelen hogy a közvetlen vonat az segít  
;) De ugye most mondhatnam hogy uljunk akkor be valahova a Nyugati környeken,  
de a virus-helyzet miatt ez sem opcio. Es onnan a normafa az még jó messze van  
:/ Szoval ize.

Őőő, igen, annyi lehet itten a gond hogy mig az a nucleo-board az 8MHz-s

orajellel rendelkezik, mi/en eleg sok 12MHz-s orajelet hasznalunk. Ez az

stm32\_rcc-ben okozhat bonyodalmat, merthogy ottan egy fix PLL-szorozó van

beallitva (amit atkuldtem abban konkretan RCC\_CFGR\_PLLMUL6 van, tehat a 8MHz-t meg 6x-ozza 48MHz-ra), es ugye az UART frekvencia is az orajeltol fugg.  
Szoval

ezt nezd meg hogy jo legyen:

- az alap orajel nalad (a nucleo-nal 8MHz)
- nezd meg hogy azt mivel szorzod (PLLMUL4 v. PLLMUL6 van-e ottan)
- es az USART2->BRR-t ennek megfeleloen allitsd be.

Azaz:

- vagy PLLMUL4 legyen es 278-as BRR;
- vagy PLLMUL6 legyen es 417-es BRR.

(persze lehet más is, csak a lenyeg hogy jól kiejön az orajel ;))

Az F\_CPU-t is ennek megfeleloen allitsd be! Itt egy picit erdekesebb ez a



dolog  
 hogy hogysmint erdemes definialni az F\_CPU-t, de most egyelore ezzel

nem

lőnek  
 nyulat hiszen az msleep()-en kívül más nem használja. Ugye a

"logisztikai  
 nehezites" vagyis "paradigma-kerdes" itten az az hogy honnan is jön az  
 F\_CPU.  
 Egy nagyon egyszeru mikrokontrollernel (mint pl az AVR-ek/Arduino-k)

van

\_a\_  
 kulso orajel, annak van egy frekvenciaja, es azzal ketyeg a CPU is. Itt  
 viszont  
 van ugyan egy kulso orajel (8MHz, konkretan, ennél a Nucleo-sorozatnál)  
 de azt  
 at tudjuk allitani szoftversen a PLL segitsegevel... szoval ezt a

"kulso"

F\_CPU  
 definiciot osszhangba kell hozni a "belso" dolgokkal (magaval a  
 forraskoddal).  
 Ez egyelore nincs így meg, de majd a kesobbiekben ezt is megcsinaljuk  
 kulturaltabbra!

A.

On Sun, 17 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm szépen, nagy segítség!  
 Ránézésre az én UART-os printf() függvényemnél precízebb, de ettől  
 sajnos  
 nem jutottam tovább:

Sajnos miután átírtam a makefile-ban a szükséges folgokat, és  
 definiáltam  
 az F\_CPU-t valamint a BCD\_VERSION\_xx sorokat a main.c-ben, feltöltés  
 közben  
 a következő hibaüzeneteket kaptam (ha nem itt jelenik meg a kép,  
 akkor a  
 képre gondoltam):

[image: image.png]

Utánajárás után a övetkező sort írtam be a kód elejére:

```
#define _GNU_SOURCE
```

Ezt követően a kód látszólag megfelelően lefordult és feltöltődött a  
 board-ra, de serial-on semmi nem jelenik meg. Ebben szeretnék  
 segítséget  
 kérni.

A személyes találkozó részemről bármikor megoldható, de ha megoldható,  
 legyen hétköznap, + jövőhéten csütörtökön nem jó, de más egyelőre

nincs

betervezve. A kollégiumból novemberben ki lettünk költöztetve covid  
 miatt,

így azóta Balástyán (ittthon) vagyok, de szívesen felvonatozom, nem

tart

sokból.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2021. jan. 15., P,



22:41):

Szia Gábor!

Huh, ne haragudj a kesei válaszaért - a hét elején még benne voltam egy-két dologban, de utána viszont jól elfelejtettem :/

No, egyrészt hogy haladjunk és tudj úgy tesztelni ilyen

környezetben

"mintha"

mezei PC-s vagy ilyesmi környezetben vagyunk, csináltam gyorsan egy

ilyen

fopencookie() alapú printf() függvényt. Probáld ki! Ha megnezed, ez

is

simán

azt az USART-ot (USART2-t, ami a Nucleo board-on van kötve az

USB-soros

átalakító) használja, de így ezek a beállítások, a fopencookie(),

az

usart\_write() meg a setlinebuf() az leveszi a terhet és akkor a

printf()

mint

olyan már úgy működik ahogy a nagykönyvben (sima sztenderd C-ben)

működni

kell

- úgy hogy az alapértelmezett stdout lesz az az U(S)ART amin a Nucleo board az

USB-n keresztül tud egyszerűen kommunikálni.

Az fontos, hogy ez így ebben a formában csak írni tud (azaz csak a

Nucleo tud

adatot küldeni a gép felé), az stdin az ebben a formában nincs

implementálva.

Persze lehetne, meg majd nemsokára meg is fogjuk tudni csinálni - de

ahhoz

kell

még egy pár egyéb dolog is (pl megszakításkezelés). Ezek sem

bonyolultak.

A másik meg hogyha talvagy a vizsgakon akkor majd diskuráljunk

személyesen is

valamikor már, mert úgy rengeteget tudnánk haladni. A jövő hét első

félén

vsz

Pizkesen leszek, ott is közbejött pár munka, de a hét második

felelő

már

jó

lesz! Most merre fele vagy így a vizsgák után? Itt Budapesten?

A.

On Tue, 12 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

Közben feltűnt, hogy itt a 10 szekundumos várakozás jóval kevesebb a valószínűségben. Hogy lehet kevesebb?

Kerestem a függvényem hibáit, így csökkentettem a várakozások

hosszát

(főleg az 50 milliszekundum) valamint jobb átláthatóság

érdekében

az:

```
if (USART2->ISR & USART_ISR_RXNE)
```

sort is átvittem a függvénybe.

Mivel csak sztringet tudok kiírni, így számokat a "sprintf()"

függvénnyel  
előtte sztringgé kell alakítsam. Próbálkoztam olyan megoldással,  
amely  
s);  
,  
amelynek az első paramétere lenne a kiírandó dolog, a második  
paramétere  
pedig a kifejezés típusa (pl: s --> sztring). Ekkor sajnos abba a  
problémába ütköztem, hogy a függvény első sorában meg kell adnom a  
paraméter típusát, ami nekem pont nem jó. Találtam olyasmi

megoldást,  
hogy:

```
void kiirat (void* kifejezes, char tipus) {
    if (tipus == 'i') {
        char* kif
        sprintf(kif, "%i", kifejezes);
        [...] //innenől olyan, mint
    }
    [...]
}
```

korábban  
küldtem

Sajnos ezzel a megoldással semmit nem csinált, így elakadtam.

Holnap reggel ezzel folytatom!  
Üdv: Gábor

2021. Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont:

jan.  
11., H, 19:47):

Kedves Pál András!

Először is elnézést, hogy el voltam tűnve, sajnos alulpontozták az  
utolsó  
vizsgám, így ma újra elmentem javítani, ami már megfelelően

sikerült,  
ezzel  
lezárva a félévemet.

Persze közben igyekeztem haladni. Több délután keresgéltem printf()  
jellegű UART-on kiírató függvény után, azonban sok bare metal meg  
még több  
barkács függvényektől túltelített kódokat leszámítva nem igazán

találtam  
semmi számomra használhatót, úgyhogy nekifogtam magam.

A kiindulási kódban (szeretnék így hivatkozni rá a későbbiekben)  
karaktereket változtattunk meg és írtunk ki UART-on, ezt vettem

alapul.  
Írtam egy függvényt, amely egy sztringet karakterekre bont, majd  
karakterenként ír ki UART-on. Még nagyon kezdetleges a megoldás, de  
már  
hatalmas sikerélmény volt. (csatolom a kódot)

Innentől mostmár szabadabban fogok mozogni, hogy lesz egy "ismerős"  
függvény :D

finomításáról,  
Ha van javaslata a továbbhaladásról, vagy a függvényem  
akkor szívesen csinálom.

Üdvözléssel: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. dec. 31., Cs, 12:54):

Szia Gábor!

Ez a "miért megbízhatóbb az a `_delay_loop()` így assembly-ben

megírva"

mint az egyszerű for ciklus az több szempontból is érdekes. Egyrészt az

igaz,

hogyan amennyiben tudod hogy mennyi ideig fut le (azaz hogy egy ciklusmag

hany

orajelig tart, mint itt a 44) addig pontosan ugyanannyira lesz megbízható. A

gond az három helyen jelentkezhet:

- egyrészt az hogy most ez pont 44 oraciklus az kvazi attól függ

hogyan

milyen fordító melyik verziójának milyen optimalizációjával fordított. Ha bármelyik is változik akkor ez is változhat  
- másrészt meg az hogy változhat hogy bizonyos utasítások (meg

assembly

szinten is) mennyi ideig tartanak, hány oraciklusig. függ a

program

memória eleresetől (latency, wait cycle), az instruction pipeline-tól. így ilyenértelenben egy assembly betét sem feltétlen megbízhatóbb de nagyságrendekkel kisebb az esélye ~3-4 utasításnál és ~8 orajelnél

hogyan

belekeveredj egy ilyenbe mint ~20 utasításnál és ~40 orajelnél.  
- a harmadik dolog amiért ez az asm-jelleg a preferált az a

sebesség:

egyszerűen ez gyorsabb, és emiatt precízebb időzítéseket lehet

vele

elérni ;)

Oke, nezek akkor majd egy egyszerű példát amivel tudsz sima

printf()-et

használni majd a nucleo-n is ezen a virtualis soros porton. Az a

baj

hogyan

pont most nincs nálam (és meg kb egy hétig nem lesz) egy ilyen board,

de

valahogy kikísérletezem hogy jó legyen ;]

A

On Thu, 31 Dec 2020, Gábor Kreinicker wrote:

Kedves Pál András"

A szórásvizsgálatot elvégeztem, így kb. maradt ez az érték.  
Megnézve az ön várakozás függvényét, érthető a működése, azonban

nem

értem,  
hogyan ez miért megbízhatóbb.  
Nagyon rosszul fogalmaztam, soros porton szerettem volna kiírni dolgokat.  
(Valamint a printf függvény kiírását hogy lehet megjeleníteni?

Mert  
ugye  
egy visual studioval szemben itt nem igazán van lehetőség  
konzolra  
kiírni.)  
Ezeket leszámítva még foglalkoztam a dolgokkal, de sajnos nem  
sikerült  
még  
előre haladnom.  
Utolsó vizsgám Január 4-én lesz, onnantól (ha jól sikerül a  
vizsga)  
teljes  
energiámmal tudok majd foglalkozni bármilyen projekttel.  
Üdv: Gábor  
Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. dec. 30.,  
Sze,  
0:35):  
Szia Gábor!  
Igen, ez a 727479 sem tunik annyira rossznak: 32000000/727479 az  
43.9875...  
szoval egészen közel van a 44-hez. De lehet hogy megmmernem  
többször  
is,  
hogymint  
tenyleg legyen mondjuk egy ~tucatnyi szám es akkor jobban latod  
hogymint  
tenyleg 44  
korul szór  
Egyebkent messze nem kizart hogy fugg az optimalizalastol. A  
gyakorlatban  
sok  
optimalizacio lehet, es a volatile valtozokat elegge  
unhatekonyan  
(hatekonytalanul, szoval nem annyira hatekonyan ) optimalizalja  
a  
fordito  
pont  
azert mert minden elemi muveletnel fel kell teteleznie hogy  
valami  
megvaltoztatja. Szoval nem irrealis a 44 sem, bar elsore en csak  
olyan  
20-30  
kozotti szamot mondtam volna. A gyakorlatban ilyen busy wait  
jellegu  
varakozasra pont ezen bizonytalansagok kikuszobolesere (azaz  
hogymint  
milyen  
fordito hogy optimalizal) talaltak ki az assembly betéteket.  
Ezeket  
altalaban  
valaki megirja es akkor lehet hasznalni, es nem igazan kell  
vacakolni.  
En  
pl  
egy ilyet hasznalok:  

```
static inline void msleep(int count)
{
    while ( 0<count )
    {    _delay_loop(F_CPU/8000);
```

```
count--;  
}  
}
```

Ahol az F\_CPU az 320000000 definialható nálad:

```
#define F_CPU      320000000
```

Es a \_delay\_loop() meg így néz ki:

```
static inline void _delay_loop(uint32_t loops)  
{  
    __asm__ (".syntax unified");  
  
    __asm__ volatile  
    ( "mov    r3,%[loops]    \n\t" /* load the initial
```

counter

```
*/  
    "1:      \n\t"  
    "nop     \n\t"  
    "subs    r3, #1      \n\t"  
    "bne     1b          \n\t"  
    :        /* empty output list  
*/  
    : [loops] "r" (loops) /* input to the asm
```

routine

```
*/  
    : "r3", "cc" /* clobber list  
*/  
    );  
}
```

Hogy miért ilyen, azon végigmehetünk majd, de a fő assembly

rutin

az

viszonylag  
egyetelmű:

```
        mov    r3, counter  
loop_start: nop  
        subs   r3, 1  
        bne    loop_start
```

ciklusban

szóval beteszi az r3 regiszterbe a counter értéket, és egy

egy

utasítás erejéig nem csinál semmit, a másikban meg csökkenti az

r3

értéket

eggyel. és ha az eredmény nem azonosan egyenlő nullával (Branch

if

Not

Equal  
/to zero/), akkor kezdi elölrol a ciklust. Adott processzortípus

meg

architektúra mellett ez mindig egy jól meghatározott ideig tart.

Hazi

feladat  
lehet kimérni hogy ez meddig tart ;) Viszont ha megnezed, itt is

ott

van a

volatile deklaráció - nem véletlenül!

Nade. Azt nem értem hogy ``ami hasonlóan mint a "printf"

működik",  
itt

mire

gondoltal? Miert, itten mi lenne a megoldando feladat? Altalaban  
az a  
celszerű  
a C-nél hogy a fő ANSI standard könyvtárat kihasználjuk. Az  
ANSI89-es  
(igen,  
1989-es) szabvanynek tartott konyvtar az abszolut alapnak  
tekintheto,  
az  
elfut  
egy kavefőzőn is. De mar a 99-es is lehet jo alap, bar mondjuk  
ott  
inkabb  
a  
nyelvi elemek miatt erdemes kiterjeszteni a dolgokat, nem is a  
standard  
(libc)  
könyvtár "tudása" miatt.  
Ettol értető dolgoakt, pl saját printf()-szerű implementaciot  
persze  
lehet  
csinalni, de altalaban beagyazott rendszereken is csak akkor  
szoktak  
ha  
egyebkent a libc-féle implementáció nem férne bele. Vagy annyira  
lassú  
az  
adott  
célfeladatra hogy nem. Namost ha azt mondom hogy az FPGA-ra irt  
saját  
(AVR) processzorhoz tarsitott 2k-nyi programmemoriaba is belefér  
egy  
printf()  
plusz az osszes kommunikacios cubehör, akkor azert lathatod hogy  
ez  
mar  
azert  
elegge alap - es elegge jól összerakott ahhoz hogy erdemes  
hasznalni  
ha  
van.  
Nade egy arduino <-> nucleo uart kommunikáció valóban jól  
hangzik,  
ezzel  
is  
tudsz kiserletezni! Hajra ;) Miket probalsz ki? Itt mar barmi,  
ami ad  
életjelet  
az fasza lehet, nem biztos hogy erdemes tulragozni az elejen -  
de  
aztan  
kesobb  
lehet bonyolitani.  
Koszi, minden rendben volt a karacsonyi idoszakkal, remelem  
feléd  
is!  
Foleg  
hogy igy mar mindjart ujev is van!  
vizsgakkal?  
Jaj, es megegy ünnepronto kerdes... mikor is vegzel kb a  
A.

On Mon, 28 Dec 2020, Gábor Kreinicker wrote:

Kedves Pál András!

Elnézést a késői válaszáért. Na igen, volt keveredés, végül

azonban

kaptam

egy 727479 értéket. Szerintem ez már pontos lesz.  
Igyekeztem továbbá minden részletét tudatosan használni, a

kiindulási

kódnak (kis-nagy betűs átalakítás), így még játszottam vele egy  
kicsit.

Neki szerettem volna fogni egy olyan függvény írásának, ami

hasonlóan

mint

a "printf" működik (ha érti, mire gondolok) a standard c-ben.

Hamar

rájöttem, hogy az eddigi karakterenkénti átalakító kód nem

igazán

jó

alap,

így böngésztem, hátha találok egy nekem megfelelő kiindulási

alapot.

Sajnos

még nem találtam, de már látom, hogy nem lesz egyszerű. Sok

megoldást

találtam, ami HAL alapú, így azoknak sok hasznát nem vettem.

Emellett találtam elég jó cikkeket UART-os kommunikációra két

arduino

között, így lehet megpróbálkozom egy Nucleo - Arduino  
kommunikációval.

Remélem rendben telt a Karácsony!

Üdvözlettel:

K. Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. dec. 23.,

Sze,

19:44):

Hm... pontosan mire gondolsz hogy "Ilyen sok ideig kapcsolja

be

a

ledet?"

Itt

nem keveredik össze a két program? Ugyertem, az UART-os meg a  
led-villogtatos?

Nem azért tart valami tovább mert közben az RXNE-re is vár?

A ledvillogtatásnak tenyleg ez a legegyszerűbb és kvazi az

egyetlen

precíz

modja ilyen busy wait jellegű varakozásnal:

```
while ( 1 )
{
    led_on();
    msleep(500);
    led_off();
    msleep(500);
}
```

csinalhatsz egyebkent ilyet is:





**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. január 28. 11:28

Kedves Pál András!

Az elmúlt napokban próbáltam megoldást találni, de nem sikerült. Ilyenkor a legbosszantóbb a vírushelyzet, valószínűleg Normafán seperc alatt megoldottuk volna.  
Viszont nagyon jó lenne tovább haladni, van javaslata, mivel érdemes folytatni? Persze mellette továbbra is keresni fogom a hibát az fopencookie()-s kódban.

Köszönettel: Gábor  
[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2021. január 29. 14:31

Szia Gábor!

Huh, hat, tenyleg sajnalom. Nade. Fel tudnad tenni a gépedre a zoom nevű csodaprogramot? Telefonon/tableten is van. Mi mostanában azzal szoktunk ilyen videokonferenciazasokat csinálni, meg azon a kepernyomegosztás is, és akkor próbáljuk meg azzal első körben.

Van valami hírad/infód közben azzal kapcsolatban hogy visszatérhet-e a kollegiumi menetrend hogy itten is tudjuk majd dolgoztatni kényelmesen? Az tenyleg nagyon-nagyon jó lenne, ilyen kezzelfogható dolgoknál meg főleg.

A

On Thu, 28 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

Az elmúlt napokban próbáltam megoldást találni, de nem sikerült. Ilyenkor a legbosszantóbb a vírushelyzet, valószínűleg Normafán seperc alatt megoldottuk volna.  
Viszont nagyon jó lenne tovább haladni, van javaslata, mivel érdemes folytatni? Persze mellette továbbra is keresni fogom a hibát az fopencookie()-s kódban.

Köszönettel: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2021. jan. 24., V, 15:59):

Kedves Pál András!

Igen, pontatlanul fogalmaztam, az F\_CPU-t értékkel (48000000) definiáltam. Sajnos nem ez volt a gond.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2021. jan. 23., Szo 17:06):

Hm... igen, úgy látom hogy az F\_CPU-t is definiálni kell, meg a \_GNU\_SOURCE-t is, valóban. A \_GNU\_SOURCE az sima ügy, de az F\_CPU-nak értéket is kell adni.  
Ennek viszont az értéket meg kell adni. A BRR=417-es beállítás akkor lesz jó ha a RCC\_CFGR\_PLLMUL6-t hasznalod az alap 8MHz-s orajel 6x-osasára és így az F\_CPU-t 48000000-nak definiálsz.

A legegyszerűbb ezt úgy hogy hozzáírod a gcc parancssorához:

```
-D_GNU_SOURCE -DF_CPU=48000000
```

Persze egy

```
#define      F_CPU  48000000
```

is teljesen jó!

Hm... a visszaecho-zás az viszont furcsa. Mindenesetre nezd meg ezeket!

On Fri, 22 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

1:1-ben ezt a hibaüzenetet kapom:

[image: image.png]

Pontosabban nem 1:1-ben, mert a makefile-ban át kell javítsam azokat a dolgokat, melyeket még a legelején is át kellett javítsam. Azért is különös, hogy Önnél működött, mert az F\_CPU nem volt definiálva (ezt a terminál jelezte is, ahogy a képen látszik). Ezt ha javítom,

akkor a

többi hibaüzenet továbbra is marad. Ahogy korábban le is írtam, azok megszűnnek, ha a kód elejére beírom a:

```
#define _GNU_SOURCE
```

sort. Ekkor a program feltöltődik a board-ra, de semmilyen életjelet nem ad. Azért, hogy látszódjon, ezt is befotóztam:

[image: image.png]

Persze amikor én próbáltam "megjavítani" akkor a warning és a note is megszűnt csupán azzal, hogy formailag számomra átláthatóbbá tettem. Valójában életjelet ad, habár nem egészen rendeltetésszerűt,

pontosabban ha

küldök karaktereket serial-on, akkor (szerintem) véletlenszerűen

ugyanazt

visszaküldi:

[image: image.png]

Ekkor egyessével írtam a számokat. Természetesen a baud-rate és a PLLMUL stimmelnek. Valóban egyszerű lett volna a megoldás, ha fejlesztés közben veszik el, de sajnos nem ez a helyzet.

Remélem jól látható és érthető a probléma és hamar megoldást találunk!

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2021. jan. 22., P, 17:29):

Szia Gábor!

Hu, ne haragudj de egy hulye/gyors kérdés: azt a forrasfat amit az elobb

kuldtam (meg pont egy hete, stm32f072-test.tgz néven) az ugy 1:1-ben mennyier

tudod feltenni es hasznalni es az mennyire ad életjelet? Csak mert azt egy

egzaktnul ugyanolyan nucleo-boardon teszteltem ki amit adtam, szoval annak

mennie kell.

Persze utana saját modositásokat tudsz csinálni meg fejleszteni meg ilyesmi, de van valahol valami szűk keresztmetszet amit így nehéz belokalizálni. Ha

van egy  
mukodo valtozatod es van egy nem mukodo valtozatod, akkor utana  
elkezdheted  
"közéltetni" a két forrásfát, es akkor elobbutobb csakcsak meglesz

hogya mi

az  
elakadás!

A.

On Fri, 22 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

A korábbi napok folyamán próbáltam rájönni, mi lehet a gond. Amire  
rájöttem, hogy az előző üzenetemben leírt "printf()-es észrevételem"  
teljesen butaság.

Próbáltam összeegyeztetni mind az F\_CPU - PLL - BRR dolgokat, mely  
sikerült, de továbbra sincs semmi jele a működésnek.

Találtam több, hasonló megoldást, de azok hasonlóan semmit sem  
csináltak.

Eszembe jutott, hátha csak a "serial" program nem kedveli, így Putty-n  
keresztül megpróbáltam, de ez sem lehet probléma.

Mivel a villogás sincs jelen, így biztosan nem a megjelenítéssel van a  
gond, hanem a függvény ill. az fopencookie körül, melyet még nem

látok át

teljesen, így sajnos elakadtam.

Ennek ellenére igyekezni fogok mihamarabb megoldást találni és/vagy  
haladni!

Már alig várom, hogy lássam sorszámozva kiírva az "xyz"-t :D

Üdv: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2021.  
jan.

17., V, 14:58):

Kedves Pál András!

Valóban nem volt összeegyeztetve a PLL és a BRR, de sajnos ettől még  
ugyanazt csinálja (tehát, hogy semmit)

Sajnos meg nem sok mindent tudok ezekről az fopencookie dolgokról,

így

lehet butaságot kérdezek, de a while(1) részen belül szereplő

printf()

függvényt hogy hozza összefüggésbe a kód at UART-os kiíratással?

Nekem

úgy

tűnik, mintha használnánk a függvényt, de nincs ami megjelenítené.

Továbbá ha serial-on küldök szöveget, akkor néha visszaküldi azt,

amit

begépelek, ez is furcsa (pláne, hogy tényleg nem mindig csinálja, de

nem is

kéne neki csinálni)

Tényleg kicsit nehézkes az utazás, de a haladás érdekében nem okoz  
gondot!

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2021. jan. 17., V,  
11:52):

Szia Gábor!

Huha, na, akkor ez a személyes konzultációs logisztika valóban nem

lesz

egyszerű... most mondhatnám hogy zoom vagy ilyesmi, de pont erre ami nekünk

kellene meg az is elege hatékonytalan tud lenni :/ Namajd ezen

akkor

agyaljuk.

Balástya sincs annyira közel, bár kétségtelen hogy a közvetlen

vonat az

segít

; ) De úgyé most mondhatnám hogy uljunk akkor be valahova a Nyugati környéken,

de a vírus-helyzet miatt ez sem opció. És onnan a normafa az még jó messze van

:/ Szóval ize.

Őőő, igen, annyi lehet itten a gond hogy míg az a nucleo-board az

8MHz-s

orajellel rendelkezik, mi/en elég sok 12MHz-s orajelet használunk.

Ez

az

stm32\_rcc-ben okozhat bonyodalmat, merthogy ottan egy fix

PLL-szorozó

van

beállítva (amit atküdtem abban konkrétan RCC\_CFGR\_PLLMUL6 van,

tehát a

8MHz-t

még 6x-ozza 48MHz-ra), és úgyé az UART frekvencia is az orajeltól

fugg.

Szóval

ezt nezd meg hogy jó legyen:

- az alap orajel nálad (a nucleo-nál 8MHz)
- nezd meg hogy azt mivel szorozod (PLLMUL4 v. PLLMUL6 van-e ottan)
- és az USART2->BRR-t ennek megfelelően állítsd be.

Azaz:

- vagy PLLMUL4 legyen és 278-as BRR;
- vagy PLLMUL6 legyen és 417-es BRR.

(persze lehet más is, csak a lényeg hogy jól kijöjjön az orajel ;))

Az F\_CPU-t is ennek megfelelően állítsd be! Itt egy picit érdekesebb

ez a

dolog

hogy hogysmint érdemes definiálni az F\_CPU-t, de most egyelőre ezzel

nem

lőnek

nyulat hiszen az msleep()-en kívül más nem használja. Úgyé a

"logisztikai

nehezítés" vagyis "paradigma-kérdés" itten az az hogy honnan is jön

az

F\_CPU.

Egy nagyon egyszerű mikrokontrollernél (mint pl az AVR-ek/Arduino-k)

van

\_a\_

kulso orajel, annak van egy frekvenciaja, és azzal ketyeg a CPU is.

Itt

viszont

van ugyan egy kulso orajel (8MHz, konkrétan, ennél a

Nucleo-sorozatnál)

de azt

at tudjuk állítani szoftversen a PLL segítségével... szóval ezt a

"kulso"

F\_CPU

definíciót összhangba kell hozni a "belso" dolgokkal (magával a forraskoddal).  
Ez egyelőre nincs így meg, de majd a későbbiekben ezt is megcsináljuk kulturaltabbra!

A.

On Sun, 17 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

Köszönöm szépen, nagy segítség!  
Ránézésre az én UART-os printf() függvényemnél precízebb, de ettől sajnos nem jutottam tovább:

Sajnos miután átírtam a makefile-ban a szükséges dolgokat, és definiáltam az F\_CPU-t valamint a BCD\_VERSION\_xx sorokat a main.c-ben, közben a következő hibaüzeneteket kaptam (ha nem itt jelenik meg a kép, akkor a képre gondoltam):

[image: image.png]

Utánajárás után a övetkező sort írtam be a kód elejére:

```
#define      _GNU_SOURCE
```

Ezt követően a kód látszólag megfelelően lefordult és feltöltődött a board-ra, de serial-on semmi nem jelenik meg. Ebben szeretnék segítséget kérni.

A személyes találkozó részemről bármikor megoldható, de ha megoldható, nincs tart sokból.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2021. jan. 15., P, 22:41):

Szia Gábor!  
Huh, ne haragudj a kesei valaszert - a het elejen még benne voltam egy-ket dologban, de utana viszont jól elfelejtettem :/  
No, egyreszt hogy haladjunk es tudj úgy tesztelni ilyen környezetben "mintha" mezei pc-s vagy ilyesmi környezetben vagyunk, csináltam gyorsan egy ilyen fopencookie() alapu printf() fuggvenyt. Probald ki! Ha megnezed,

ez is siman azt az USART-ot (USART2-t, ami a Nucleo board-on ra van kotve az usb-soros atalakitora) használja, de így ezek a beállítások, a fopencookie(), az usart\_write() meg a setlinebuf() az leveszi a terhet és akkor a printf() mint olyan már úgy működik ahogy a nagykönyvben (sima sztenderd C-ben) működnie kell - úgy hogy az alapértelmezett stdout lesz az az U(S)ART amin a Nucleo board az USB-n keresztül tud egyszerűen kommunikálni. Az fontos, hogy ez így ebben a formában csak írni tud (azaz csak a Nucleo tud adatot küldeni a gép fele), az stdin az ebben a formában nincs implementálva. Persze lehetne, meg majd nemsokára meg is fogjuk tudni csinálni - de ahhoz kell még egy pár egyéb dolog is (pl megszakításkezeles). Ezek sem bonyolultak. A másik meg hogyha tulvagy a vizsgakon akkor majd diskuraljunk személyesen is valamikor már, mert úgy rengeteget tudnánk haladni. A jövő hét elso felen vsz Pizkesen leszek, ott is kozbejott par munka, de a het masodik feletol mar jo lesz! Most merrefele vagy így a vizsgak utan? Itt Budapesten? A.

On Tue, 12 Jan 2021, Gábor Kreinicker wrote:

Kedves Pál András!

Közben feltűnt, hogy itt a 10 szekundumos várakozás jóval kevesebb a valóságban. Hogy lehet kevesebb?

Kerestem a függvényem hibáit, így csökkentettem a várakozások hosszán (főlösleges az 50 milliszekundum) valamint jobb átláthatóság érdekében az:

```
if (USART2->ISR & USART_ISR_RXNE)
```

sort is átvittem a függvénybe.

Mivel csak sztringet tudok kiírni, így számokat a "sprintf()" függvénnyel előtte sztringgé kell alakítsam. Próbálkoztam olyan megoldással, amely során két paraméteres a függvény (pl: kiirat("pelda kifejezes",

s);  
),  
amelynek az első paramétere lenne a kiírandó dolog, a második paramétere pedig a kifejezés típusa (pl: s --> sztring). Ekkor sajnos abba a problémába ütköztem, hogy a függvény első sorában meg kell adnom

a paraméter típusát, ami nekem pont nem jó. Találtam olyasmi megoldást, hogy:

```
void kiirat (void* kifejezes, char tipus) {
    if (tipus == 'i') {
        char* kif
        sprintf(kif, "%i",
kifejezes);
        [...] //innenről olyan, mint
korábban
küldtem
    }
    [...]
}
```

Sajnos ezzel a megoldással semmit nem csinált, így elakadtam.

Holnap reggel ezzel folytatom!  
Üdv: Gábor

Gábor Kreinicker <[kreinickergabor@gmail.com](mailto:kreinickergabor@gmail.com)> ezt írta (időpont: 2021.

jan.  
11., H, 19:47):

Kedves Pál András!

Először is elnézést, hogy el voltam tűnve, sajnos alulpontozták

az utolsó  
vizsgám, így ma újra elmentem javítani, ami már megfelelően  
sikerkült,  
ezzel  
lezárva a félévemet.

Persze közben igyekeztem haladni. Több délután keresgéltem  
printf()  
jellegű UART-on kiírató függvény után, azonban sok bare metal  
meg

még több  
barkács függvényektől túltelített kódokat leszámítva nem igazán  
találtam  
semmi számomra használhatót, úgyhogy nekifogtam magam.

A kiindulási kódban (szeretnék így hivatkozni rá a későbbiekben)  
karaktereket változtattunk meg és írtunk ki UART-on, ezt vettem

alapul.  
Írtam egy függvényt, amely egy sztringet karakterekre bont, majd  
karakterenként ír ki UART-on. Még nagyon kezdetleges a

megoldás, de  
már hatalmas sikerélmény volt. (csatolom a kódot)

"ismerős"  
Innentől mostmár szabadabban fogok mozogni, hogy lesz egy  
függvény :D

finomításáról, Ha van javaslata a továbbhaladásról, vagy a függvényem akkor szívesen csinálom.

Üdvözlettel: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. dec. 31., Cs, 12:54):

Szia Gábor!

Ez a "miért megbízhatóbb az a `_delay_loop()` így assembly-ben megírva" mint az egyszerű for ciklus az több szempontból is érdekes. Egyrészt az igaz, hogy amennyiben tudod hogy mennyi ideig fut le (azaz hogy egy ciklusmag hany orajelig tart, mint itt a 44) addig pontosan ugyanannyira lesz megbízható. A gond az három helyen jelentkezhet:

- egyrészt az hogy most ez pont 44 oraciklus az kvazi attól függ hogy milyen fordító melyik verziójának milyen optimalizációjával fordítod.

Ha bármelyik is változik akkor ez is változhat

- másrészt meg az hogy változhat hogy bizonyos utasítások (meg assembly szinten is) mennyi ideig tartanak, hany oraciklusig. függ a program memoria eleresetől (latency, wait cycle), az instruction pipeline-tól.

így ilyenértelenben egy assembly betét sem feltétlen megbízhatóbb de nagyságrendekkel kisebb az esélye ~3-4 utasításnal és ~8 orajelnel hogy belekeveredj egy ilyenbe mint ~20 utasításnal és ~40 orajelnel.

- a harmadik dolog amiért ez az asm-jelleg a preferált az a sebesség: egyszerűen ez gyorsabb, és emiatt precízebb időzítéseket lehet vele elérni ;)

Oke, nezek akkor majd egy egyszerű példát amivel tudsz sima `printf()`-et használni majd a nucleo-n is ezen a virtualis soros porton. Az a baj hogy pont most nincs nálam (és meg kb egy hétig nem lesz) egy ilyen board, de valahogy kikísérletezem hogy jó legyen ;]

A



On Thu, 31 Dec 2020, Gábor Kreinicker wrote:

Kedves Pál András"

A szórásvizsgálatot elvégeztem, így kb. maradt ez az érték.  
Megnézve az ön várakozás függvényét, érthető a működése,

azonban

nem

értem,  
hogy ez miért megbízhatóbb.  
Nagyon rosszul fogalmaztam, soros porton szerettem volna

kiíratni

dolgokat.  
(Valamint a printf függvény kiíratását hogy lehet

megjeleníteni?

Mert

ugye  
egy visual studioval szemben itt nem igazán van lehetőség

konzolra

kiírni.)  
Ezeket leszámítva még foglalkoztam a dolgokkal, de sajnos nem

sikerült

még  
előre haladnom.

vizsga)

teljes  
energiámmal tudok majd foglalkozni bármilyen projekttel.

Üdv: Gábor

Andras Pal <[apal@szofi.net](mailto:apal@szofi.net)> ezt írta (időpont: 2020. dec.

30.,

Sze,

0:35):

Szia Gábor!

Igen, ez a 727479 sem tunik annyira rossznak:

32000000/727479 az

43.9875...  
szoval egészen közel van a 44-hez. De lehet hogy megmmernem

többszor

is,  
hogy  
tenyleg legyen mondjuk egy ~tucatnyi szám es akkor jobban

latod

hogy

tenyleg 44  
korul szór

Egyebkent messze nem kizart hogy fugg az optimalizalastol. A  
gyakorlatban  
sok  
optimalizacio lehet, es a volatile valtozokat elegge

unhatekonyan

(hatekonytalanul, szoval nem annyira hatekonyan )

optimalizálja

a

fordito  
pont  
azert mert minden elemi muveletnel fel kell teteleznie hogy  
megvaltoztatja. Szoval nem irreális a 44 sem, bar elsore en

valami

csak

olyan

20-30

közötti számot mondtam volna. A gyakorlatban ilyen busy wait

jellegu

varakozásra pont ezen bizonytalanságok kikuszobolesere (azaz

hogya

milyen

fordító hogy optimalizál) találtak ki az assembly betéteket.

Ezeket

általában

valaki megírja és akkor lehet használni, és nem igazán kell

vacakolni.

En

pl

egy ilyen használok:

```
static inline void msleep(int count)
{
    while ( 0<count )
    {    _delay_loop(F_CPU/8000);
        count--;
    }
}
```

Ahol az F\_CPU az 320000000 definíálható nálad:

#define F\_CPU 320000000

Es a \_delay\_loop() meg így néz ki:

```
static inline void _delay_loop(uint32_t loops)
{
    __asm__ (".syntax unified");

    __asm__ volatile
    (    "mov    r3,%[loops]    \n\t" /* load the initial
```

counter

\*/

```
    "1:        \n\t"
    "nop        \n\t"
    "subs    r3, #1    \n\t"
    "bne     1b        \n\t"
    :                /* empty output list
```

\*/

: [loops] "r" (loops) /\* input to the asm

routine

\*/

: "r3", "cc" /\* clobber list

\*/

);

}

Hogy miért ilyen, azon végigmehetünk majd, de a fő assembly

rutin

az

viszonylag

Gábor Kreinicker &lt;kreinickergabor@gmail.com&gt;

2021. január 29. 14:39

Címzett: Andras Pal &lt;apal@szofi.net&gt;

Kedves Pál András!

Persze, a Zoom már linuxon is fön van, így azzal gond nem lesz. Viszont leghamarabb holnap délután leszek gépközelben (ilyen disznóvágásos dolgok miatt).

Kollégium jelentkezés volt, de az biztos, hogy leghamarabb márciusban tudunk beköltözni, ez a vírushelyzet alakulásától függ. Valóban jó lenne már projektbe bekapcsolódni.



**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. február 3. 22:20

Kedves Pál András!

Nekem a holnap 9 tökéletes! Majd ha küld egy zoom linket, akkor ahhoz csatlakozom!

Köszönettel: Gábor  
[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2021. február 3. 22:23

Oke! Esetleg kicsit korábban mennyire lenne jo?  
[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. február 3. 22:26

Amennyire szükséges, nehogy miattam késsen el a hegyről ;)  
[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2021. február 3. 22:28

Irok akkor reggel meg kuldok zoom-linket ;)  
[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. február 3. 22:29

Rendben, köszönöm!  
[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2021. február 4. 8:53

827 899 9626 - már megnyitottam!  
[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2021. február 4. 9:30

`arm-none-eabi-ld -Tmain.ld -nostartfiles -o main.elf crt0.o main.o stm32_rcc.o /usr/lib/arm-none-eabi/newlib/  
armv6-m/libc_nano.a /usr/lib/gcc/arm-none-eabi/5.4.1/armv6-m/libgcc.a /usr/lib/arm-none-eabi/newlib/armv6-  
m/libnosys.a`

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>  
Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

2021. február 4. 9:33

`/usr/lib/arm-none-eabi/newlib/thumb/v6-m/nofp/libc_nano.a  
/usr/lib/arm-none-eabi/newlib/thumb/v6-m/nofp/libnosys.a  
/usr/lib/gcc/arm-none-eabi/9.2.1/thumb/v6-m/nofp/libgcc.a`  
[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. február 4. 17:40

Kedves Pál András!

Köszönöm még egyszer a segítséget!

Csináltam egy saját led "toggle" függvényt (ami terjedelmében hosszabb, igazából sok haszna nem volt, de működik), majd utána megcsináltam a Fibonacci-sorozat kiíratást. Ezután egy prímszám felsoroló programot is írtam, nem sokkal bonyolultabb a Fibonacci-sorozatostól. Egyelőre az idő kiíratásával meggyűlt a bajom, de ez szerintem csak a bénázásom miatt van.

Ami feltűnt, hogy a printf() csak akkor működik, ha van benne "\n" (új sor kezdése). Ha ez nincs, akkor semmit sem fog csinálni. Ennek mi lehet az oka?

Üdv: Gábor

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2021. február 4. 18:46

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

Ha csinalsz a printf() utan egy fflush(stdout); hivast az segit?

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2021. február 4. 19:34

Címzett: Andras Pal <apal@szofi.net>

Ó igen, az volt a gond, köszönöm!

[Az idézett szöveg el van rejtve]

---

**Andras Pal** <apal@szofi.net>

2021. február 5. 10:31

Címzett: Gábor Kreinicker <kreinickergabor@gmail.com>

No, akkor hurrá :)

Szoval akkor itt az ideje az időzítőknak/óráknak/szamlaloknak a hasznalataval megismerkednünk. Ugye a busy wait-es varakozas (lasd msleep()) az nem rossz, de ugye kozben nem tudunk csinalni semmit. Az időzítőknak/óráknak/szamlaloknak meglesz az az elonye hogy tudunk ilyen busy wait-es modon is varakozni, de ettől függetlenül az "abszolút" időnek a múlását is könnyen tudjuk kezelni.

Az ilyen/effele mikrokontrollerekben mint ez az STM32 (meg hasonloan az AVR-eknél, stb) ezek ugye a timer-ek. Ezek alapjaraton nagyon sokmindent tudunkak (legnkabb a PWM, [https://en.wikipedia.org/wiki/Pulse-width\\_modulation](https://en.wikipedia.org/wiki/Pulse-width_modulation) miatt), így neha túl bonyolultnak tunik elsore (foleg az "advanced" vagy "general-purpose" timerek), de a legtöbb alapfunkcio azoknál is gyorsan elérhető.

Ugye ezek olyan periferiak, amiket az UART-hoz hasonloan be kell kapcsolni, de nem kommunikálnak kulso vezetekekkel (mint az Rx/Tx vonal). Hanem "csak" a processzor környezetében tevékenykednek, és számol(gat)nak. Persze lehetnek olyan timer-ek amik használnak kulso orajelet is (pl egy kulso oszcillator vagy barmi más jelét számolják), ezek ilyenertelembe mar fizikailag is (igaz picit, de) kezdenek hasonlítani az UART-okhoz meg ezekhez a kommunikacios periferiakhoz. Nade ilyennel első körben nem foglalkozunk.

A legegyszerubb timer az az ami számol, és semmi más nem csinál. Van egy regisztered valahol (x bit hosszú, x=8, 16, 24, 32, barmi), aminek az értéke minden orajel hatására eggyel nő, és akkor egyszer csak tulcsordul és kezdi elölrol. Ez annyira alapszintu hogy ilyen "csak úgy" onmagaban nincs is. 3 dologgal szoktak ezt kiegészíteni:

- nem egy orajelre ugrik egyet a szamlalo, hanem P darab orajelre, ezt hivjak úgy hogy pre-scaler;
- nem akkor csordul túl amikor a bit-ertek telitodik, hanem akkor amikor mi azt felprogramozzuk. pl egy x=16 bites szamlalo alapbol 0 és 65535 kozott számol, de lehet csinalni olyat hogy mondjuk 1000-nel forduljon (0, 1, 2, ... 999, 0, 1, 2, ...). Ezt az értéket itt az STM-nél "auto reload"-nak hivjak, ezért ARR (auto-reload register) lesz a neve. Sajnos ez kicsit felrevezeto, mert auto reload-nak általában a lefele számlálót (999, 998, ... 1, 0, 999, 998, ...) szoktak hívni, dehat ez van.
- amikor tulcsordulas van, akkor kapj egy esemenyt, azaz egy egy bites információt, hogy "hé, tulcsordultam".

Ez a három együtt alkalmazva már nagyon-nagyon kenyelmes lesz. Ilyet tud a TIM6 nevű szamlaloja is ennek az STM32-nek (lasd: [https://www.st.com/resource/en/reference\\_manual/dm00031936-stm32f0x1stm32f0x2stm32f0x8-advanced-armbased-32bit-mcus-stmicroelectronics.pdf](https://www.st.com/resource/en/reference_manual/dm00031936-stm32f0x1stm32f0x2stm32f0x8-advanced-armbased-32bit-mcus-stmicroelectronics.pdf), 547. oldal).

Eloszor bekapcsolod:

```
RCC->APB1ENR |= RCC_APB1ENR_TIM6EN;
```

majd beallithatod a prescaler meg a tulcsordulasos regisztert:

```
TIM6->PSC = 48-1;
TIM6->ARR = 1000-1;
```

Majd hajrá, indulhat a számlálás:

```
TIM6->CR1 |= TIM_CR1_CEN;
```

A számláló értéke (TIM6->CNT) így 0 és 999 között teker, úgy hogy 48-cal osztja le a bemenő órajelet. Hogy miért 47-et és 999-et kell megadni itten azt majd később megnezzük (előljárobán annyit hogy elektronikusan így jóval egyszerűbb lesz a felepítése egy ilyen számlalónak, de az "auto reload" eredeti, lefele számlálós jelentéséből is már láthatod mi lehet a háttérben).

Lenyeg hogy mind a PSC, mind az ARR az 16 bites (és így a CNT is), szóval nagyobb értékeket is megadhatasz ezeknél.

Ha pedig tulcsordulas van akkor a TIM6->SR regiszternek a TIM\_SR\_UIF bitje lesz 1:

```
if ( TIM6->SR & TIM_SR_UIF )
{   printf("hé, tulcsordultam!\n");
}
```

Persze ha észlelsz tulcsordulást (mert ugye ez egy óra, tehát előbbutobb mindenkepp túl fog csordulni) akkor ki kell nulláznod ezt a bitet: TIM6->SR &= ~TIM\_SR\_UIF; és akkor tudod érzekelni a tulcsordulást \_ujra\_. Azaz:

```
if ( TIM6->SR & TIM_SR_UIF )
{   printf("hé, tulcsordultam!\n");
    TIM6->SR &= ~TIM_SR_UIF;
}
```

Szóval ha 2x csordult túl az ciki, azt már nem tudid így megkülönböztetni (azaz ha egyszer 0->1-re ment ez a bit, akkor a következő tulcsordulasnál nem fog például 1->0-ra menni, hanem marad 1).

Ez mit jelent nálunk? 48MHz-s az óra, tehát a CNT számláló pont 1 mikroszekundumonként ugrik egyet. És <1000-ig számol, tehát pont 1 milliszekundumonként van tulcsordulas. Ez meg mindig lehet hogy kicsit gyors lesz bizonyos játékokhoz, de mint láthatod, mindket regisztert (PSC, ARR) 16-bitesre tudod állítani, azaz a 48MHz-s órajeleden alapuló számláló akár 89.48 másodpercenként(!) is tulcsordulhat. Ami meg már egészen makroszkopikus idő ;]

Na, és innen már indulhatnak a feladatok ;) Példa: csinálj egy olyan led-villogtatót ami pontosan 1 másodpercenként villog (0.5 sec on, 0.5 sec off). s nem vegytisztán busy wait, hanem timer alapú!

Aztan jöhet a többi.

[Az idézett szöveg el van rejtve]

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>  
Címzett: Andras Pal <apal@szofi.net>

2021. február 5. 15:55

Kedves Pál András!

Köszönöm, a részletes leírást! Ennek megfelelően a következő módon haladtam:

Átolvastam a dolgokat, majd miután előkészítettem és elindítottam a számlálást, a while(1) ciklusba a következő kódrészletet alkalmaztam:

```
int TSZ = 0, i = 0;

RCC->APB1ENR |= RCC_APB1ENR_TIM6EN;
TIM6->PSC = 48 - 1;
TIM6->ARR = 1000 - 1;
TIM6->CR1 |= TIM_CR1_CEN;

while (1) {
```

```
    if (TIM6->SR & TIM_SR_UIF) {
        TIM6->SR &= ~TIM_SR_UIF;
        TSZ++;
        if (TSZ == 500) {
            printf("[%d] tele\n", i);
            led(2); // led toggle
            i++;
            TSZ = 0;
        }
    }
}
```

Habár látszólag működik (időbélyegzővel serial-on látszólag stabil másodpercenként villan föl), de ez nem megfelelő megoldás, mivel a TSZ++ számítása is idő, így valószínű, hogy lesz drift. (TSZ mint túlcsordulások száma, semmi '89 előtti időkre utalás)

Erre azért volt szükség, mert ahogy próbálkoztam, nem tudtam összehozni fél sec-et az ARR módosításával. Miután az előbb említett megoldást megcsináltam, megpróbáltam a PSC-t is, ami alapján a következő értékek beállításával sikerült a led villogtatása egy sec-es periódussal:

```
RCC->APB1ENR |= RCC_APB1ENR_TIM6EN;
TIM6->PSC = 480 - 1;
TIM6->ARR = 50000 - 1;
TIM6->CR1 |= TIM_CR1_CEN;

while (1) {
    if (TIM6->SR & TIM_SR_UIF) {
        printf("[%i] tele\n", j);
        TIM6->SR &= ~TIM_SR_UIF;
        j++;
        led(2);
    }
}
```

Amit észrevettem, hogy pár perc alatt azért ezzel a módszerrel is létrejön egy ~0,001 sec-es drift.

A második kódot csatolom, ha netán lenne rá szükség.

Üdv: Gábor

[Az idézett szöveg el van rejtve]

---

 **main.c**  
3K

---

**Gábor Kreinicker** <kreinickergabor@gmail.com>

2021. február 5. 16:02

Címzett: Andras Pal <apal@szofi.net>

Helyesbítés: Az első megoldás lehet ettől még, hogy nem megfelelő, de a TSZ++ biztos nem zavar be. Raktam egy msleep(100);-at bele, és azzal is tökéletesen működött

[Az idézett szöveg el van rejtve]