

Na tom by nebolo nič zlé, ale v skupine zameranej na:

- Embedded systémy

elektroniky (digitálna analógová) "divných" otázok považšine študentov, vlastné skúsenosti pri analýze chýb v prinesených "neposlušných" komerčných produktoch a nakoniec aj príspevky z FB (Facebook) skupiny "Embedded Systems", ktorá názvom jasne definuje jej zameranie.

V poslednom čase sa v (nielen) tejto FB skupine objavil jeden "nešvar" - začali sa tam vo veľkej miere objavovať príspevky obsahujúce fotku, schému
 BIAS je očakávaná elementárna znalosť problematiky a tak by príspevky mali byť vecné a konštruktívne. Detto aj otázky.

A tu je problém.

V príspevkoch ohľadom **duino, autori v 99% prípadov nemajú tušenia, ako tá z ich pohľadu geniálna vec, vôbec funguje. O elektronike a o programovaní nehovoriac.

Takto v skupine, kde by sa malo riešiť časovanie DDRAM, okakanie UART-u mikrokontroléra na 9-bitový režim aj keď ho natívne nepodporuje a podobných vychytávkach, objavuje hromada príspevkov, ktoré už len pri čítaní vzbudzujú pocity podobné ranajšej kombinácii hrušiek s mliekom.

Vrcholom je fotka komerčného produktu, ktorého srdcom je táto "Big product in the universe" doska a otázka typu: neviete prečo mi po čase prestane komunikovať s teplotným snímačom?

V princípe to nie je problém len tejto konkrétnej skupiny, ale ako tak pozerám, všetkých podobne zameraných skupín. A to nielen na FB.

Čo ma vlastne na **duino "žerie"?

Najskôr letmý exkurz do roku 2005, kedy bola doska arduino uvoľnená do sveta. Áno, myšlienka vznikla skôr (r.2000) ako diplomový projekt, ale teraz sa zameriam už na produkt samotný.

Prvá doska Arduino vznikla ako **pomoc pre študentov** s chýbajúcimi znalosťami v oblasti elektroniky a programovania mikrokontrolérov.

Od vtedy sa stala najobľúbenejším **nástrojom pre prototypovanie** elektroniky, ktorý využívajú technici a dokonca aj veľké spoločnosti (fakty zo stránky arduino.cc).

"pomoc pre študentov"

Autori takto chceli pomôcť niekomu, kto napríklad chce v rámci svojej študentskej/domácej práce, vniesť určitý stupeň riadenia, pre svoj výrobok.

Laicky povedané - mám pár motorov a LEDiek, viem ich zapracovať do môjho výrobku, ale neviem ako ich budem ovládať. Použijem Arduino!

Potiaľto paráda, všetko ide ako má. Motorčeky motorčekujú, LEDky ledkujú.... študentská práca hotová, všetci spokojný.

Diablici

Pokiaľ práca bola napríklad o "algoritme" automatu na plnenie fliaš" - potom je to v poriadku.

Problém nastáva, ak práca bola o "návrhu a výrobe" automatu na plnenie fliaš".

Ak má byť niečo komerčné alebo určené do priemyslu - t.j. optimálnejšie (lacnejšie/blbovzdornejšie/rýchlejšie...) nemôžeme použiť na riadenie niečo, čo netušíme ako vo vnútri funguje aj keď to stojí pár centov a rovnako v riadiacej funkcii!

Ak chcem niečo vyrobiť pre priemyselné využitie alebo pre komerčný predaj, nemôžem predsa začať napríklad vyhadzovať sériové rezistory k LEDkám, minimalizovať blokovacie kapacity bez analýzy spínacích prechodov a ich vplyvu na napájanie a to všetko v domnení, že zariadenie má vydržať v reálnej prevádzke xx hodín denne.

Program (firmwér) je kapitola sama o sebe.

Je super použiť na projekt knižnicu I2C a do jednoduchého kódu dopísať:

char Value = Wire.read(); a očakávať že čítanie prebehne a Value bude vždy naplnená.

I2C snímač môže predsa (aj vplyvom zlého blokovania napájania) v najlepšom prípade poslať zlú hodnotu, v tom horšom prestať komunikovať.

A vieme ako sa s týmto vysporiada dodávaná knižničná rutina Wire.read?

Už je len krôčik k zastaveniu programu. A celej linky na plnenie fliaš.

A to sú práve **diablici**, ktorí komplikujú vývoj/optimalizáciu celého zariadenia - náhodné (niekedy aj očakávané) vplyvy, vstupujúce do celého reťazca riadiaceho procesu za účelom jeho destabilizácie.

Na toto bohužiaľ, **duino už nie je stavané, resp. celá jeho filozofia s týmito vplyvmi akosi nepočítala (ani nemala dôvod).

Nechcem týmto povedať, že je to nemožné ale skôr to, že dizajnéri nie sú nútení to robiť - veď na stole im to funguje.

A sme opäť v škole

Ak pri prácach podobného zamerania použijú túto dosku ako riadiacu časť, nevenujú pozornosť "diablikom". Prostredie arduina ich k tomu nijak nenúti.

Jednoducho zoberú dosku a naklikajú aplikačný flow.

Nie je to celkom chyba dosky, alebo celého ekosystému - je to skôr chyba prístupu k nemu.

Pokiaľ by dizajnér namiesto **duino použil samostatný mikrokontrolér na vlastnej doske s potrebným okolím, a programovanie by prebiehalo na úrovni registrov, určite by mal viac možností si uvedomiť aj možné chybové stavy a ošetriť ich.

"nástrojom pre prototypovanie"

Je tenká hranica medzi prototypom a finálnym produktom. Resp. kedy sa stáva prototyp produktom?

Ak by sa dal univerzálne graficky na časovej osi popísať vývojový cyklus zariadenia bolo by to pekné, ale hraničí to s utópiou.

A so zaznačením bodu prechodu z prototypu na finál je to podobné.

Nielenže do vývojového procesu vstupuje množstvo premenných a neznámych, ktoré sa doťahujú počas vývoja, celkovo vývoj je časovo (čítaj finančne) ohraničený. Nehovoriac o nikdy nekončiacich požiadavkách na zariadenie zo strany zadávateľa. A nikdy nekončiacich úpravách FW (optimalizácia, odstraňovanie skrytých chýb,...) - viz. Murphyho zákony.

Vo väčšine prípadov sa takto z prototypu stane finálny produkt v čase, keď zariadenie už funguje, určitá úroveň optimalizácie a ošetrovanie chybových stavov prebehlo, ale hlavne keď už došiel čas na vývoj (z praxe vieme, že času nikdy nie je dosť).

Pre výklad slova prototyp mám interne zafixované niečo, čo mi pomáha otestovať funkcionality časti celku, algoritmus celku, prípadne niečo, na čom vyvíjam aplikáciu zatiaľ čo je vo výrobe jedna z finálnych verzií dosky s plošnými spojmami.

V praxi to môže znamenať, že vezmem hocikáký výrobcom dodávaný dev. kit (s MCU ktoré bude aj vo finále), pripojím perifériu ktorú chcem otestovať a programovaním FW si ju takto otestujem. Daný FW následne slúži ako kostra programu pre finál.

Áno, je možné aj tuto použiť **duino, no ktorú časť kódu recyklujem do finálu? Rozpítvám **duinovské knižnice? ... veľa šťastia.

A neuveriteľné sa stáva skutočnosťou - mnoho prototypov sa tak stáva finálnym produktom aj s použitým **duinom!

Fast win!

... pokiaľ sa neobjavia "diablici"

Ešte raz - čo ma vlastne na **duine "žerie"?

Vyššie som sa snažil napísať vlastne dve veci:

- povrchné znalosti dizajnéra v elektronike a v programovaní a absentujúca nutnosť štúdia
- nevhodné použitie **duino

Nechcem týmto povedať, že **duino je zlé. To nie. Ale je zle využívané (samozrejme nie globálne).

Viem si predstaviť túto dosku použiť v procese výučby napríklad na demonštráciu algoritmu funkcie svetelnej križovatky (ozaj, učí sa to ešte?) alebo výťahu.

Viem si ju predstaviť ako riadiacu dosku domáceho kútila pri výrobe monitora teploty a vlhkosti kompostu na záhrade, alebo ako skúška blikajúcich očíek z LEDiek pre bábijku v rukách otecka ktorý sa snaží vštepiť základy elektrotechniky hravou formou svojim deťom.

Ale tu by som asi skončil.

Všetky ďalšie fázy vývoja vyššie uvedených príkladov, by mali už byť postavené na samostatnom MCU na (kludne aj) prototypovej doske alebo vlastnom plošnom spoji na to vytvorenom.

Zariadenia určené do priemyslu, alebo komerčné, by určite mali mať vlastný plošný spoj s vlastným mikrokontrolérom.

Preto som za oddelovanie **duino aplikácii, otázok a tutoriálov od embedded systémov a elektroniky.

V princípe tieto oblasti súvisia, ale nie z pohľadu užívateľa.

A nakoniec - ak treba použiť **duino, použite radšej eval. kit alebo dev. kit od výrobcu mikrokontrolérov. Nie sú zasa tak drahé a pri práci s nimi sa naučíte omnoho, mnoho viac.

Teraz ma môžete ukameňovať.

**duino - schválne nekonkretizujem typ dosky/prostredia, pretože už ich je dostupných ako v iindii bohov.

p.s. pár príkladov komunikácie s/medzi **duinistami tak ako si ich pamätám z dávnejšieho čítania:

A: nevieš prečo mi prestane blikať LEDka počas vysielania dlhšieho textu do počítača?

B: Myslíš komunikáciu do PC cez UART?

A: No cez USB/TTL prevodník do toho okna v ArduinoIDE

B: Ah, to bude UART. Pracuje v interrupt režime?

A: V čom? ... *nasleduje fragment kódu.*

B: No, funkcia sa neukončí pokiaľ sa neodvysielala celý reťazec. Ďalšia funkcia teda blikne s LEDkou (zmení jej stav), až keď sa predchádzajúca ukončí. Zrejme posielal dlhší reťazec a to dlhšie trvá. A tak to na LEDke vidno.

A: jj, posielam dlhší text "Teplota snímača 1: 25,3 °C", prerobil som to na "T1:25,3" a už to nerobí. Paráda.

B: robí ale nie je to viditeľné, mal by si to posilať v režime prerušenia a....

A: Je to OK, konečne mi nemešká ani zastavenie motora. (skočil mi do rozpísaného textu)

Q: Pri nahrávaní do arduina mi to hlási, že je málo pamäte. čo stým?

A: Až pri nahrávaní?

Q: No jasné, už pri tom kontrolovaní, šak sa rozumieme...

A: máš dlhý kód, nevojde to zvolenej dosky.

Q: aha, tak mám zvoliť inú dosku?

A: ak máš inú skús.

Q: mám iba Arduino leonardo

A: tak skráť kód.

Q: Ako?

.. nasledovali odborné rady na skrátenie kódu od minimalizácie počtu premenných až po vytvorení nových funkcií pre časti kódu ktoré sa opakovali.

A: kašlem na to, objednáam si inú dosku

..atď

Tento príspevok prezentuje chvíľkové nekontrolovateľné výlevy autora. Berte ho s rezervou.