

Userspace I/O Documentatie

Joris van Rooij <jorrizza@jrrzz.net>
Fontys Hogescholen, Lectoraat Architectuur van Embedded Systemen

4 februari 2010

Inhoudsopgave

1	Inleiding	3
2	I²C	4
2.1	Werking	4
2.2	Drivers	4
2.2.1	PCA9532	4
2.2.2	PCA8575	5
3	Hardware	6
3.1	Platform	6
3.2	Test I/O	6
4	Linux Target Image Builder	8
4.1	Root File System	8
4.2	Eigen Software	8

1 Inleiding

Naar aanleiding van een onderzoeksaanvraag van Robopharma is er een traject gestart waarin getracht wordt om aan deze onderzoeksaanvraag invulling te geven. De industriële automatisering die wordt aangeleverd door Robopharma draait grotendeels op PLC's. In dit onderzoek wordt bekeken of er een vervanging voor deze PLC's door middel van Embedded Linux gemaakt kan worden, met alle voordelen van dien.

Een oplossing voor deze uitdaging kan het gebruik maken van I²C zijn. In dit document wordt deze mogelijke invulling van de gevraagde functionaliteit verder uitgewerkt tot een werkende Proof of Concept.

2 I²C

Om I/O gemakkelijk toegankelijk te maken voor C-programmeurs zal er een abstractie van deze I/O in userspace beschikbaar gemaakt moeten worden. Dit levert verschillende voordelen op. Ten eerste is het veiliger, aangezien de kernel veel van de mogelijke fouten in userspace voor je afvangt. Ook debugging is gemakkelijker, en goed vergelijkbaar met het debuggen dat gebeurt tijdens het ontwikkelen van desktopapplicaties. I²C is een ideale oplossing voor het naar userspace brengen van I/O.

Linux levert I²C-functionaliteit in userspace. De userspace I²C-API is bijna identiek aan de kernelspace-API, waardoor geschreven code zowel in user- als in kernelspace gebruikt kan worden. Wat Linux levert is volledige controle over de aanwezige I²C master nodes die aanwezig zijn, en een abstractie van de I²C- en SMBus-protocollen.

2.1 Werking

I²C is een seriële bus ontwikkeld door Philips, nu onder het beheer van NXP. Het is een redelijk eenvoudige bus waarbij een master node kan communiceren met verschillende slave nodes aan dezelfde bus. Door het juiste adres op de bus te zetten wordt communicatie geïnitieerd met de juiste slave node. Daarna kan door middel van het schrijven of lezen naar registers in deze slave nodes de aangesloten hardware bediend worden.

Ieder type slave node heeft een uniek adres. Vaak kan per peripheral een bepaalde adres-suffix ingesteld worden om meerdere chips van hetzelfde type toe te laten op dezelfde bus. De unieke adressen zitten in een 7-bits range en worden beheerd door NXP.

De bus zelf levert alleen voldoende voeding voor de peripheralchips. Voor de hardware die door deze chips aangestuurd wordt moet een aparte voeding geregeld worden.

2.2 Drivers

De communicatie over de I²C-bus mag dan wel gestandaardiseerd zijn, de werking van de peripherals is dat zeker niet. Daarom zal voor elk type peripheral een driver geschreven worden die de registers abstraheert. Daarbovenop kan vervolgens weer de hardware-representatielaag geschreven worden. Er zijn in deze PoC twee drivers geschreven.

2.2.1 PCA9532

[Documentatie](#)

Deze chip zit standaard op de LPC3250 en biedt dus een ideaal testdoel voor de userspace I²C-functionaliteit. Hij levert twee hoofdfuncties. Aan de ene kant is het een LED-dimmer, waarbij door middel van een instelbare PWM LED's aangestuurd kunnen worden. Ook kan de chip gebruikt worden als I/O expander, waarbij de

uitgaande lijnen simpelweg als I/O gebruikt worden. Aangezien enkele uitgangen van deze chip al verbonden zijn met genummerde LED's op het bordje, is er een driver geschreven die alleen deze LED's abstraheert.

De registers en I/O pennen komen 1-op-1 overeen. Daarom is deze driver niets meer dan een headerfile die definieert welke LED bij welk register hoort. Dit is gecombineerd met wat voorbeeldcode en verpakt in het bijgevoegd [looplicht-project](#).

Dit project werkt echter niet met de standaard-installatie die door LTIB wordt gebouwd (later in het document). Deze software werkt alleen wanneer de PCA9532-driver in de kernel uitgeschakeld wordt. De kernel laat namelijk een userspace driver niet toe voor devices waarvoor deze al een driver actief heeft.

2.2.2 PCA8575

Documentatie

Deze chip is de simpelere versie van de PCA9532, waarbij alleen een 16-bit I/O-expander geleverd wordt. Daarnaast is een interrupt-pen toegevoegd die hoog wordt op het moment dat de status van de I/O-pennen verandert. Deze peripheral vormt de basis voor de I/O-kant van deze PoC.

Hardwarematig kunnen alle 16 pennen van de PCA8575 gebruikt worden voor I/O. Er wordt geen gebruik gemaakt van register-functie mapping. De eerste twee registers zijn gelijk aan de 16-bits waarde van de pennen. Hierdoor kan door middel van het SMBus-protocol in één keer in 16-bit gelezen of geschreven worden.

Het adres van deze chip kan veranderd worden in de laatste drie bits, waardoor er dus effectief acht op dezelfde bus terecht kunnen.

De driver voor deze chip is lichtelijk gecompliceerder, omdat de I/O in industriële toepassingen in de praktijk onbelast verloopt. Dit komt er aan de softwarekant op neer dat de boolean-waardes omgekeerd worden. Deze chip heeft hiernaast ook de eigenaardige eigenschap om 16-bit SMBus write commando's om te draaien, waardoor de laatste byte in het eerste register geschreven wordt en vice versa. Ook hiervoor dient gecompenseerd te worden in de driver. Al met al is de driver voor deze expander nog steeds eenvoudig. De driver, inclusief een voorbeeldproject, is terug te vinden in het bijgevoegde [ioexp-project](#).

3 Hardware

3.1 Platform

Om I²C onder Embedded Linux te gebruiken zijn er eigenlijk maar twee eisen echt van belang. Ten eerste zal de hardware ondersteund moeten worden door de Linux kernel. Het is verstandig een platform te gebruiken dat al geleverd wordt met een werkende Linux-installatie om overbodig werk te voorkomen. Ten tweede is de aanwezigheid van I²C natuurlijk een vereiste. Hoe meer I²C-connectiviteit er vanuit de chip gerealiseerd kan worden, hoe beter. Randvoorwaarden maken de keuze van het bord compleet.

In dit geval zijn er geen randvoorwaarden, aangezien het om een PoC gaat. Daarom wordt een standaardplatform dat al bekend is binnen het lectoraat gebruikt voor het fabriceren van deze PoC.

De *Embedded Artists LPC3250* is een development-board gebouwd rond NXP's *ARM926EJ-S LPC3250* platform. Het levert naast I²C ook de nodige GPIO en peripherals. Een paar hiervan worden gebruikt om in een later stadium het ontwikkelen van software te vereenvoudigen.

3.2 Test I/O

In plaats van het potentieel gevaarlijke ontwikkelen bovenop de te automatiseren hardware is gekozen voor een hardware-stub. Om I/O te simuleren is er een klein bordje ontwikkeld met drie knoppen en vier LED's. Deze hangen rechtstreeks aan de I/O-pinnen van een standaard 50-polige connector. Dezelfde connector, met dezelfde verdeling, als de connector op het 16-bits industrieel I/O-bord.

Deze 50-polige connector wordt aangesloten op een PCA8575, waardoor de 16-bits I/O expander rechtstreeks de I/O afhandelt die afkomstig is van de aangesloten hardware. I²C zelf loopt via een RJ-45 connector naar de LPC3250. Aan de andere kant van deze constructie is een tweede RJ-45 connector geplaatst, waardoor effectief acht PCA8575's (128 bits I/O) in een daisy chain door middel van UTP aangesloten kunnen worden. De PCA8575 en de connectoren zijn samen op een railbevestiging geplaatst, zodat deze oplossing conform industriestandaarden geplaatst kan worden.

Uiteindelijk zal de LPC3250 SoC van het Embedded Artists basisbord afgehaald kunnen worden, en op een geminimaliseerd bord van de juiste connectoren voorzien moeten worden. Op dit moment moeten de I²C-lijnen nog op het developmentbord geprikt worden. De connector hiervoor zit in het midden van het bord.

<i>Kleur/Functie</i>	<i>Pen</i>
Blauw/VDD+	2
Geel/Interrupt	-
Groen/SDA	25
Rood/GND	1
Zwart/SCL	26
Wit/GND	3

Zoals te zien is, is de interrupt-pen nog niet aangesloten. Voor deze PoC stond I/O centraal, interrupts kunnen later geïmplementeerd worden. Deze kan binnen

Linux op meerdere manieren naar user-space geleid worden. Er kan door middel van polling een soft interrupt gegenereerd worden. Ook kan door middel van een UIO kernelspace module een driver gemaakt worden die de interrupt omzet naar een blocking *select(2)* call. De keuze hiertussen zal voornamelijk van reactietijd afhangen. UIO is per definitie sneller dan polling.

4 Linux Target Image Builder

LTIB is de bouwomgeving die door Embedded Artists wordt meegeleverd. Dit document zal niet heel LTIB beschrijven, maar enkel de stappen die nodig zijn om hiermee de werkende PoC te genereren. Voor alle documentatie is de [website](#) de beste plaats om te gaan zoeken.

4.1 Root File System

De initiële setup is simpelweg het uitvoeren van `./ltib` na het downloaden ervan. Mocht tijdens het lezen van dit document LTIB niet meer bestaan, is er een [mirror beschikbaar](#) van de door Fontys gebruikte versie. Kies tijdens de setup voor de LPC3250, en alles gaat vanzelf. Dit resulteert in een root file system dat óf omgezet kan worden tot een image óf gebruikt kan worden als NFS export voor het booten vanaf het netwerk. Ook de kernel wordt gebouwd, en is beschikbaar.

4.2 Eigen Software

LTIB werkt intern met RPM, en er zullen dus spec-files geschreven moeten worden voor ieder project dat gecompileerd wordt. De spec-files zijn meestal hetzelfde voor ieder project, dus alleen de projectnaam hoeft aangepast te worden. Deze files leven in `ltib/dist/lfs-5.1/` en hebben een directory per project. De [spec-files voor de twee voorbeeldprojecten](#) zijn aangeleverd.

De source zelf moet in tarball-vorm (zoals de aangeleverde projecten) in `ltib/rpm/SOURCES` staan. Tijdens het ontwikkelen is het wellicht gemakkelijk om in een makefile automatisch dit bestand bij iedere build te laten verversen. Het bouwen en installeren van het project gaat vervolgens op de volgende manier:

```
$ cd ltib
$ rm -rf rpm/BUILD/ioexp-0.1
$ ./ltib -p ioexp.spec -m prep
$ ./ltib -p ioexp.spec -m deploy
```

Op deze manier worden oude buildfiles verwijderd, wordt het programma gebouwd en vervolgens geïnstalleerd op het root file system.

Deze methode eist alleen van de Makefile in het project dat het bouwt zonder argument. Mocht er een configure-script aan worden toegevoegd, zal ook de spec-file aangepast moeten worden. In dit geval kan de vers gebouwde software op het target systeem gestart worden met het commando `ioexp`.