

Project ARS

(adding to the proposal of Google's 'project Ara',
covering separate areas)

rev. 1.0

June 09 2014

ARS Technologies
www.arstech.com

to contact us, submit notes, comments, feedback

<http://arstech.com/install/cms-contact/COREseo.html>
info@arstech.com

Table of content

1. Introduction
2. Mechanical
3. Electrical
4. System software - added components
5. System software - implementation for proposed interfaces
6. Call for action
7. References

1. Introduction

The current 'project ARS' proposal was specifically created in response to Google's 'project Ara'.

Both projects enables devices like smartphones to be able to connect with external hardware.

What are the major differences in the proposals:

- while 'project ara' targets specifically smartphones, 'project ars' embraces any type of device - smatrphones, tablets, and any other types of devices
- mechanically the connectors for connecting external hardware in 'project ara' are located on the back of a device, while the connectors in 'project ars' are located at the edges
- 'project ara' uses custom 10pin connector, 'project ars' uses standard / mass produced connectors
- on a system level 'project ara' uses LVDS level signals from MIPI M-PHY and UniPro protocols, while 'project ars' uses interfaces available on the SOC and wires them directly to connectors
- broadly defined and unclear system software in 'project ara', versus deployed and working software, allowing upgrade-ability of existing and older systems in 'project ars'

2. Mechanical

A device maker (smartphone, tablet, etc.) did invest millions of dollars to design their device(s).

From the point of view of a device maker designing of a new device or redesigning of an existing device - to compete in the the same market as the existing devices is most likely a non-starter.

On the other hand small changes to the existing designs will be much easier to implement and achieve.

The existing designs of smartphones and tablets place i/o connectors on the edges of these devices.

The proposal calls for placing of additional connectors on the edges of the devices as well.



****pic.1 - tablet + phone + connectors*

In such case there will be minimal design modifications -

- modifying of the pc board to add connectors and wiring
- modifying of the enclosure - making openings for connectors

3. Electrical

A user may want to connect to a device:

- variety of sensors - temperature, acceleration, magnetic field, gyro ... etc.; most of these are based on I2C interface, and some on SPI interface;
- chips based on a parallel bus, which include address lines, data lines (8/16bit), and control lines;
- standard PCI Express and equivalent based peripheral cards, including - ExpressCard, Compact PCI Express, PXI Express, etc.
- general I/O interface providing digital inputs and outputs

Regarding powering, we propose using a standard and regulated voltage level -> +3.3v .

The following are the one by one look at the initially proposed interfaces.

3.1 Adding I2C interface

I propose placing a 'mini usb a-b' connector or 'micro usb a-b' connector' like for ex -
mouser.com part number: 538-47491-0001



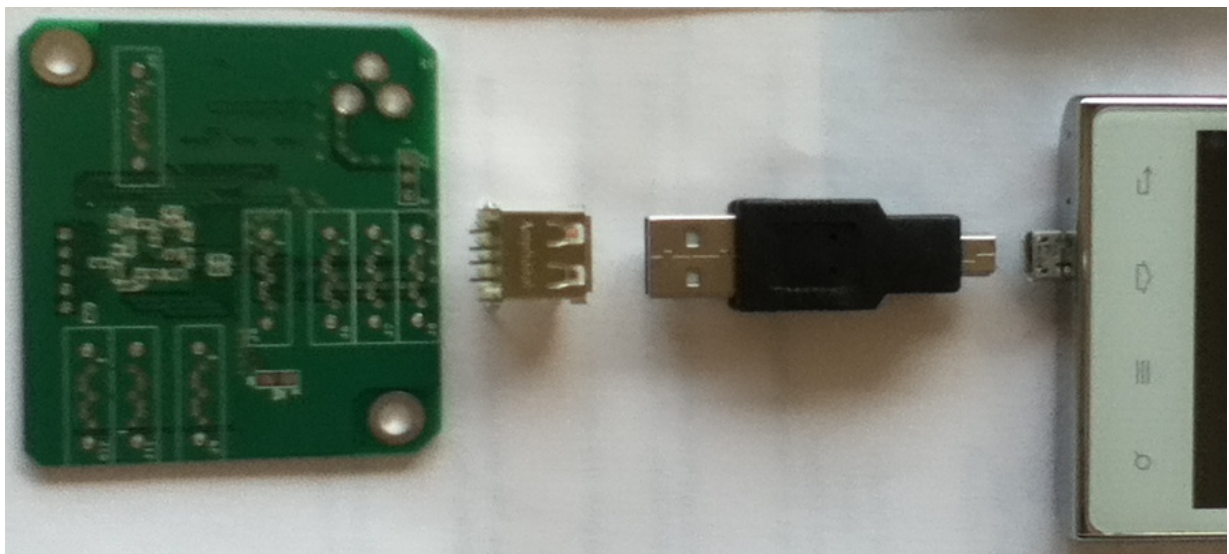
****pic.2 - phone with usb micro a-b connector*

The proposed pin assignment -

1. +3.3v
2. gnd
3. SCL - i2c clock
4. SDA - i2c data

To implement the proposal a device maker has to modify the pc board - add wiring, and modify the enclosure. Most, if not all of ARM SOC include on-chip i2c controller.

A user can utilize a device enable in the above way by using for example a gender changer which then goes to a pc board.



****pic.3 - possible implementation with: usb con., gender changer , pc board*

3.2 Adding PCI Express interface

PCI Express interface is available only on some SOCs. Most likely such chips will be placed mostly on tablets.

One example of an ARM SOC which includes the PCI Express interface - NVIDIA Tegra 3 HD Mobile Processors: T30 Series, AP30 Series .

We propose placing a 'hdmi' connector, like for ex. -
mouser.com part number: 649-10029449-001RLF



****pic.4 - tablet with hdmi connector*

The proposed pin assignment -

1. PERST# - pci express reset #
2. GND
3. 3.3v
4. CLKN - clock pair negative
5. GND
6. CLKP - clock pair positive
7. TXN - transmit pair negative
8. GND
9. TXP - transmit pair positive
10. RXN - receive pair negative
11. GND
12. RXP - receive pair positive
13. 3.3v
14. 3.3v
15. 3.3v
16. 3.3v
- 17.
- 18.
- 19.

A user can utilize a device enabled in the above way by using a cable to connect to PCI cards:



****pic.5 - a tablet + HDMI con., a cable , **xprs-pci-x3** + 3 PCI cards*

The above image shows real products. For details please check - ch.7 References .

Another example of utilizing a device enabled in the above way by using a cable to connect to PCI Express -x16 card:



****pic.6 - a tablet + HDMI con., a cable , **xprs-px-x16** + PCI Express -x16 card*

The above image shows real products. For details please check - ch.7 References .

There are links provided showing way of connecting - an ExpressCard card, and a Compact PCI Express, PXI Express card.

3.2 Adding GPIO interface

By adding a connector with GPIO pins the user can handle the following cases:

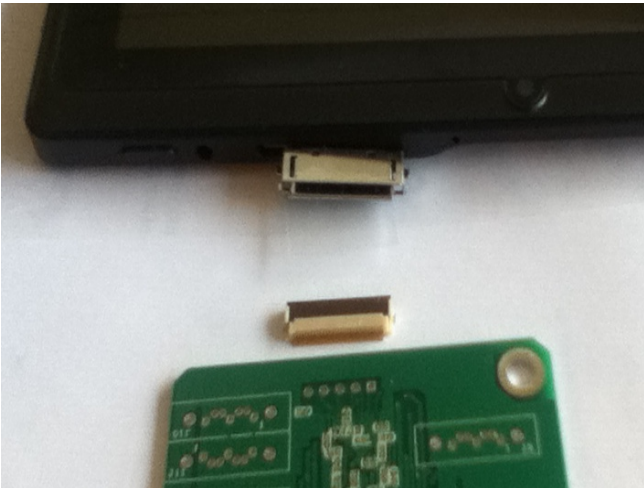
- chips based on a parallel bus, which include address lines, data lines (8/16bit), and control lines;
- general I/O interface providing digital inputs and outputs

We propose selecting a 30pin connector and placing it on a device like for ex. -

mouser.com part number: 656-dd1b030va1

The proposed pin assignment -

1. GND
2. 3.3v
3. GPIO0
- ...
28. GPIO25
29. 3.3v
30. GND



****pic.7,8 - 30pin connectors, and possible implementations through a cable connector or directly mounting to a pc board*

A user can utilize a device enable in the above way by using a cable or directly a pc board.

4. System software - added components

The chapter makes the assumption that the device will run Google's Android operating system.

To assist with adding a new hardware we are providing 2 components:

- software engine
- linux driver binary builder kit - ldbbk

4.1. Software engine

After a device is manufactured rarely is the kernel and operating system updated. In most cases the devices carry the same kernel/os during the lifetime of the device.

Adding external hardware or upgrading under Google Android operating system is very difficult.

An app which wants to use the new hardware connected can not directly access it.

The way Google has been responding is to add a specific API in the next OS release. This leaves existing devices not covered. In addition it may take months, or years until the needed API is present on the majority of devices.

We did add a 'software engine' component which:

- runs independently of the Android OS level
- communicates with system and kernel levels
- takes requests from multiple apps and returns replies back to apps
- implements new capabilities not present on current OS level

Important pre-requisites for a device to be able to is to have:

- root access privileges
- and, information on the SOC, and additional components

One example of a 'software engine' + android app is -> ArsRegRw .
<https://play.google.com/store/apps/details?id=com.arstech.arsregrw>

The 'software engine' gives this app the extremely powerful capability to read/write individual registers in the whole memory space of an ARM SOC.

4.2. linux driver binary builder kit - ldbbk

Android runs on top of Linux kernel which is not friendly at all to loadable modules / drivers.

A driver built on one kernel release can not load on another kernel, or even on the same kernel if there is some difference in the system hardware.

To overcome this we have 'ldbbk' component which allows a pre-built driver binary to load and run on a wide variety of kernels and system hardware.

For details please refer to -

http://arstech.com/install/cms-display/ste_ld-bbk.html

5. System software - implementation for proposed interfaces

5.1 I2C interface

Current kernels include 'i2c' driver. The 'software engine' component has to open and communicate with the standard driver, and then service requests from apps.

If the device manufacturer 'forgot' to include i2c driver in the kernel, a binary module has to be built. This binary will be adjusted by 'ldbbk' then loaded and used by the 'software engine' component.

5.2 PCI Express interface

A PCI Express type device has normally a driver. A binary of the driver has to be built. This binary will be adjusted by 'ldbbk' and then loaded and used by the OS.

The 'software engine' component may not be needed if an app does not touches directly the PCI Express hardware.

PCI Express is normally a feature of x86 based systems and a new option on ARM based systems. A system software has to be ported dealing with - enumeration, hot plug, assigning resources.

5.3 GPIO interface

The GPIO interface may be included in the 'software engine' component and serviced directly from user level.

One of the tutorial examples included in ArsRegRw android app -> <https://play.google.com/store/apps/details?id=com.arstech.arsregrw> is dealing with the GPIO pins of a Texas Instruments ARM SOC and a Samsung ARM SOC.

6. Call for action

***** in attn. to phone/tablet device manufacturers**

Small changes in the existing designs - adding connectors, wiring, modifying enclosure will open the devices to be able to add new hardware.

If work starts right away, new - hardware enabled devices can be available in the time span of weeks or months.

***** in attn. to SOC chip makers**

Documentation of the memory map, registers, devices, interfaces in a SOC is essential to a hardware and system software developer.

SOC user's manual, technical reference manual, application notes have to be made available for free (!!!).

We are grateful to the chip makers like Samsung, TI, NVIDIA, etc. who have already done this.

We call to all other chip makers to do the same.

***** in attn. to Google, and device makers**

Root access (#) is available by default on all desktop/notebook OS-es (Windows/Linux/Mac OS x), but is disabled by default on all phone/tablet OS-es.

A hardware and system software developer has to go to the hacker community to enable a device to have root access, in order to be able to develop and test.

The user of a hardware enabled device has to do the same - go to the hacker community to enable a device to have root access.

We call to Google, and device makers to provide root access on all hardware enabled devices.

7. References

- ArsRegRw

<https://play.google.com/store/apps/details?id=com.arstech.arsregrw>

- google store app

<http://arstech.com/install/ecom-prodshow/arsregrw.html> - product page

<http://arstech.com/install/cms-display/arsregrw.html> - details on use

- LDBBK

<http://arstech.com/install/ecom-prodshow/ld-bbk.html> - product page

http://arstech.com/install/cms-display/ste_ld-bbk.html - details on use

- PCI Express based interfacing products

http://www.arstech.com/install/cms-display/ste_use_xprs_px_x1.html - for a PCI Express X1 card

http://www.arstech.com/install/cms-display/ste_use_xprs_px_x16.html - for a PCI Express X1 - X16 card

http://www.arstech.com/install/cms-display/ste_use_xprs_pci_x1.html - for one PCI card

http://www.arstech.com/install/cms-display/ste_use_xprs_pci_x3.html - for up to 3 PCI cards

http://www.arstech.com/install/cms-display/ste_use_xprs_ec.html - for an ExpressCard card

http://www.arstech.com/install/cms-display/ste_use_xprs_cpcie.html - for a Compact PCI Express / PXI Express card

To view connectors:

- go to 'mouser.com'

- enter part number

- click on 'data sheet'

- connectors for I2C interface

mouser.com

part number	description	manufacturer
538-47491-0001	usb micro b mid.mnt	Molex
538-47346-0001	usb micro b bot.mnt	
538-47589-0001	usb micro ab bot.mnt	
538-47590-0001	usb micro ab top.mnt	

- connectors for PCI Express interface

mouser.com

part number	description	manufacturer
649-10029449-001rlf	hdmi recept.smt	fci
806-kdmix-ns1-ns-b	hdmi rec.type a smt	kycon
806-kdmix-ns2-ns-b	hdmi rec.type a th/ra	
154-10428-e	hdmi rec. r/a smt	kobiconn
154-10405-e	hdmi rec. r/a smt	
154-10412-e	hdmi rec. r/a smt	

- connectors for GPIO interface

mouser.com

part number	description	manufacturer
656-dd1b030va1	30pin 0.5mm con. plug	JAE Electronics
656-dd1b030ha1r500	30pin 0.5mm con. plug	
656-dd1r030ha1r1300	30pin 0.5mm con. receptacle	
656-dd1p030ma1	30pin 0.5mm con. cable plug	

- to contact us, submit notes, comments, feedback

<http://arstech.com/install/cms-contact/COREseo.html>
info@arstech.com