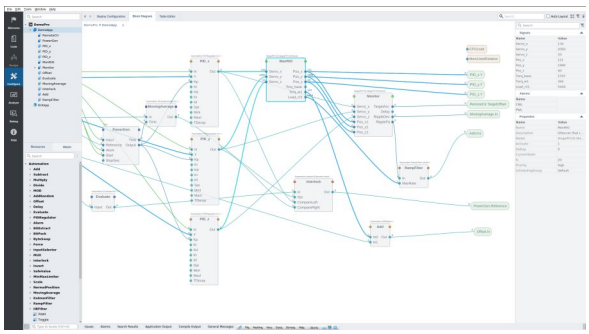
Cross platform embedded system development

With CDP Studio you have a single development tool for all your customer specific embedded systems

ARM based embedded devices Re-use code across platforms Distributed systems High level development tools Project specific solutions Reduced development cost

The cost cutting impact of having a single development platform covering a wide range of hardware is significant. With embedded devices running Linux, tools, functional libraries, people skillset, etc.. is shared across systems.

While the low-level components are developed in C++, the actual system design does not require C++ skills. With a full graphical IDE, a system is built by "drag and drop" and parametrisation of ready-made modules.



With CDP Studio you can build a distributed control system from a combination of small embedded Linux devices, automation controllers, industrial computers, and even Windows desktop systems. One integrated system developed on the same platform.

The following will focus on the embedded end of this scenario to give some inspiration on how to use CDP Studio on small Linux powered devices and in embedded hardware environments, based on the current standard toolchains that ships with CDP Studio.

Embedded solutions

With the term "embedded Linux" we normally think of standalone appliances running a tuned, stripped down, Linux system. Embedded devices are designed to do a specific task, some also have real-time requirements. These devices are locked down with a given functionality, ranging from simple electronic toys, to marine navigational systems. The simpler products are delivered "as is", i.e. no ways to update the software or ways to interact beyond the operational user interface, while industrial systems tend to be both configurable and upgradeable.

For customer specific solutions, this is an opportunity to deliver tailor made solutions, even for low volume devices. This is where CDP Studio as a development platform provide the tools to design, configure and maintain systems, making such customisation a sensible business case.

EMBEDDED DEVICE

For standalone devices, there is just one run-time application doing its tasks, running on top of a trimmed down Linux OS. The end-user will not relate to CDP Studio as such, it is only used for development and maintenance of the software application. Examples here would be devices for data acquisition or dedicated controllers, probably configured by the end user via a simple web interface.

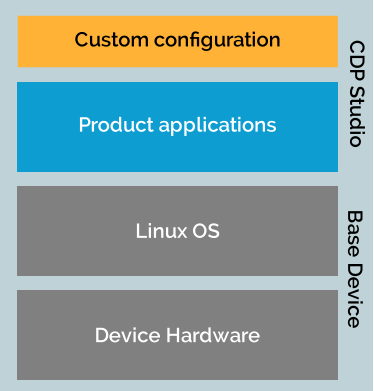
CDPStudio

Customised embedded systems ...made easy.

CUSTOMISATION

Your product may be delivered in a standard version, but you also have the freedom to deliver custom variants. Modifying and enhancing products to fit specific customers or projects, will in many cases just be configuration changes, i.e. not require any changes in the underlying C++ code. Such changes may even be implemented by the project people, not involving R&D.

Add value to a product by offering customisation of the executable code



Taking this a step further, you could provide the CDP Studio development tool including your function library to system integrators or system manufacturers. Then your product is not just open for third party applications, but you provide a very accessible development environment for your customers to add their special knowledge or specific market segment functionality. Your product has added value!

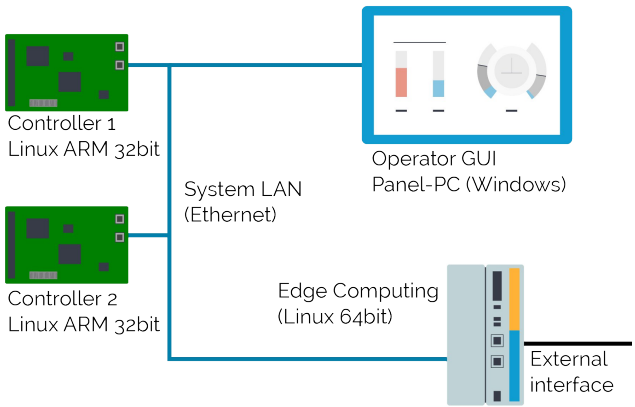
FOCUS ON THE COOL STUFF

Do you know how much time is wasted working on the development environment and basic functionality? CDP Studio will do most of this for you, which means the return on investment is extremely good. Unless your developers work for free....

DISTRIBUTED SYSTEMS

Even though embedded Linux products tend to be standalone devices, most will be attached to a network. CDP Studio has distributed system design built into the native application framework, i.e. a system is several applications working together. The applications may run on a single computer or distributed between controllers/computers on an Ethernet LAN segment. As CDP applications are abstracted from HW and OS, you may build a hybrid system solution of several low-level Linux controllers, a Windows Operator GUI, and a Linux IPC for the heavy signal processing. CDP Studio let you put functions where it makes most sense for system ruggedness, controller performance, combination with other software, storage capacity, etc...

System builders and integrators will then be able to focus on the solution, using a single development environment even if the hardware involved is coming from multiple vendors. The system as such has then increased hardware independence!



CDPStudio

Hardware

When it comes to available hardware, CDP Studio currently has two ARM toolkits relevant for embedded devices:

- A generic ARM7 toolkit for Debian based systems.
- A dedicated ARM6 toolkit for Raspberry Pi based hardware, Raspbian.

There is also a range of compact SBCs (Single Board Computers) running Linux (or Windows) on x86 architecture which could be seen as embedded devices. Here we just use the generic Linux or Windows toolkits that comes with CDP Studio.

If we look at relevant ARM based hardware suitable for CDP Studio projects, the following examples gives a better picture of the possibilities. In general, if it runs a Debian derivative, in most cases your CDP applications will work, using the appropriate toolkit. Be aware that this is also linked to the CPU platform.

RASPBERRY PI AND DERIVATIVES

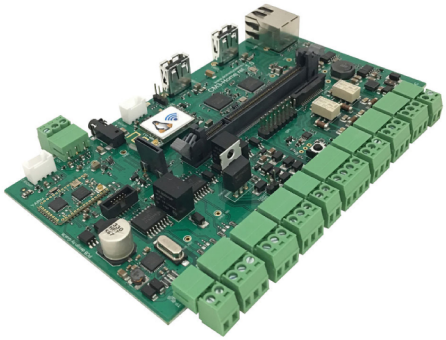
A new trend is also what you could call Raspberry Pi derivatives, the Raspberry Pi is a fantastic device, but not really suited for the industrial environment. The use of the Raspberry CM3 compute module let HW manufacturers build devices that are suitable for industrial use, and still be compatible with the Raspberry Pi space.

The Revolution Pi is an industrialised system complete with analogue and digital I/O etc.. This boils down to a capable industrial Linux controller at a reasonable price. CDP Studio comes with an I/O server for Revolution Pi, so building a controller with bus attached I/O works straight out of the box. (revolution.kunbus.com/)

BB-400 is a similar device, and with dual Ethernet and Wifi as well as 8x configurable digital I/O, this may have multiple roles in an industrial control network or industrial IoT systems.

(brainboxes.com/industrial-edge-controller)

There are also several IoT gateways with an interesting range of interfaces, rugged design and 4G/LTE connectivity. The Axotec gateways like IGX-560 and IPX-860 are good examples. Especially the IPX-860 which is an IP67 rugged device suitable for most unpleasant environments. (axotec.de/en/)



Another example geared more towards home control is the CM3-Home by Guiott (manufactured by Acme systems). The board fits in a DIN-rail enclosure and has a range of useful interfaces and proper 12-24VDC power, but no monitor connector, being a true headless embedded controller. (acmesystems.it/CM3-HOME)

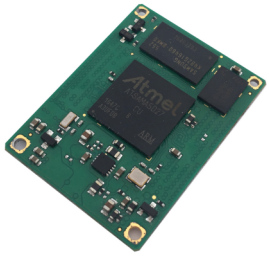
USE RASPBERRY PI FOR PROTOTYPING

As CDP Studio is HW independent, you may develop prototypes and demos on low cost HW like the RPi. Moving CDP applications to other devices, even with different toolcains is a drag'n drop exercise.

ARM BAREBONE HARDWARE

If you are going deeper embedded, then there are multiple embedded ecosystems and development kits using the popular ARM architecture. CDP Studio will need a system that comes with a Debian derivative, either generic or HW specific builds using e.g. Yocto.

The BeagleBoard.org® ecosystem, an open-source hardware community, have a range of boards coming from multiple manufacturers, which ship with Debian Linux . As the BeagleBone® is open-source hardware, integrating this into your electronics is easily doable. Variants of the BeagleBone@ may even be used directly in projects. (beagleboard.org/)



The normal approach would be to use a SoM (System on Module) including development kits or evaluation boards. An example here is the RoadRunner SoM from Acme Systems. With its single core Cortex A4 processor at 500MHz, the module is not very powerful, but still comes with a trimmed down Debian Linux. A neat package for small devices. (acmesystems.it/roadrunner)

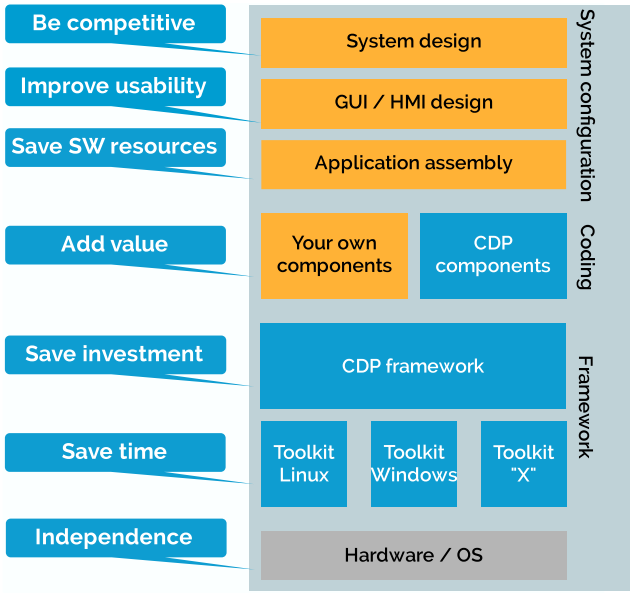
INDUSTRIAL COMPUTERS

There are several similar systems on the market, and with CDP Studio as a single development tool you will cover the full range from a tiny Cortex device to a powerful server workhorse. There is also the more powerful x86 based devices for specific markets, like the Marine computer below. (recab.com

Toolchains, framework, and IDE

As hardware is increasingly more capable at reduced cost, size, and power consumption, application execution speed is not the only parameter. This is especially true for customer or project specific solutions; the development time of a system is the only place to save significant project cost.

CDP Studio is a complete development environment, implying that the maintenance of the overall tool is managed for you. CDP Studio is not just the IDE, but comes with a development framework geared towards control systems and real-time computing, a set of standard functions and protocols, and finally toolkits for several hardware platforms. If you need a product specific toolkit, we can build it as a separate add-on to the standard CDP Studio. CDP Studio may be seen as a layered "stack" of functionality as illustrated in the figure.



The value of software tools is measured in saved developer hours; the hours to build the environment and tool-chains, as well as maintaining these, are non-productive direct cost. A development platform with relevant protocols and HW integrations in place, saves even more hours. CDP Studio let you focus on your core competence and add real value to the end-user.

CDP Technologies AS
Hundsværgata 8, 6008 Ålesund
Tel +47 990 80 900 · info@cdptech.com
cdpstudio.com