Open source virtual prototyping for faster hardware and software co-design

Guillaume Delbergue
guillaume.delbergue@hiventive.com

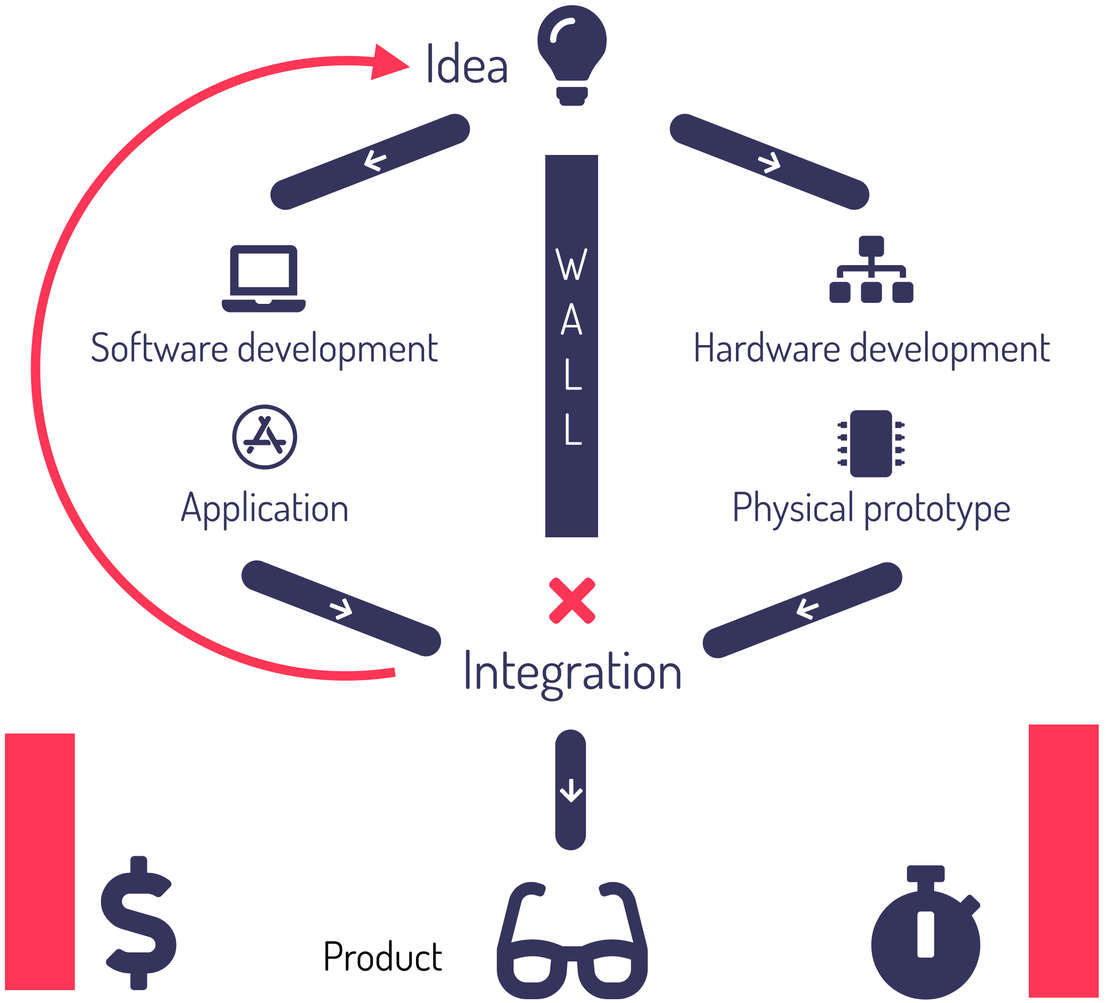
10 minutes you said ?

A virtual prototype is a software application simulating the hardware behaviour. It provides a ready-to-execute environment for your next platform. Virtual prototypes aim to solve various use cases.

It enables software development and hardware/- software testing to begin before the real hardware is available and can also be used for later use when the hardware is available. It is an alternative to prototyping on real boards. It expands the software developers productivity allowing them to develop on it as their development platform, months before the real hardware prototype.

Electronics Products Design Flow

How you design your product now



How you will design you product tomorrow

Virtual prototyping technology

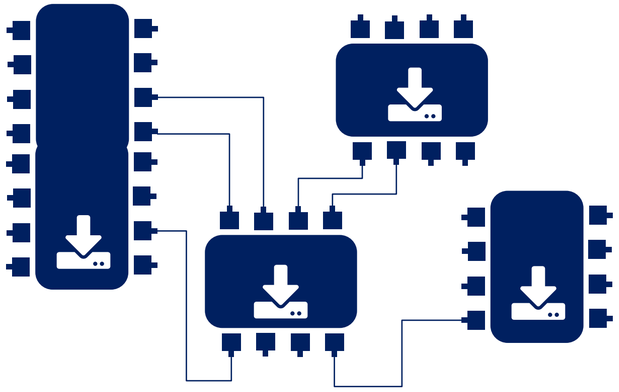
Provides platforms virtually assembled

A virtual prototype is a **software application simulating the hardware behaviour**. It provides a ready-to-execute environment for your next platform.

That simulates complete hardware

Fast virtual prototype allows HW/SW co-simulation

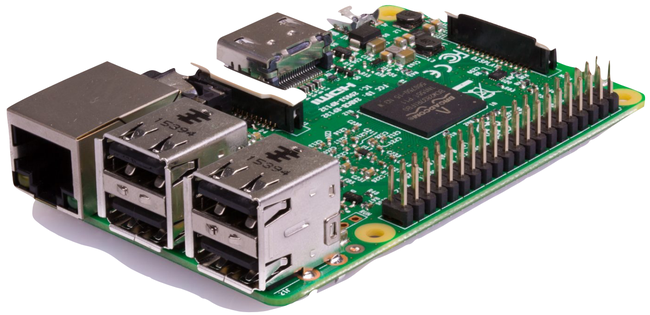
Using modelling standards from the industry



Virtual component assembly to simulate a platform

How does it looks like ?

RUN YOUR ORIGINAL (REAL) BAREMETAL BINARY, OS . . .



Entire SoC

A model

- 4 x Cortex A53
- RAM
- PL011 (UART)
- BCM2836 CONTROL
- BCM2835 GPIO
- ...

What does a model look like ?

Modelization using HVRegister, an open-source register framework

Datasheet

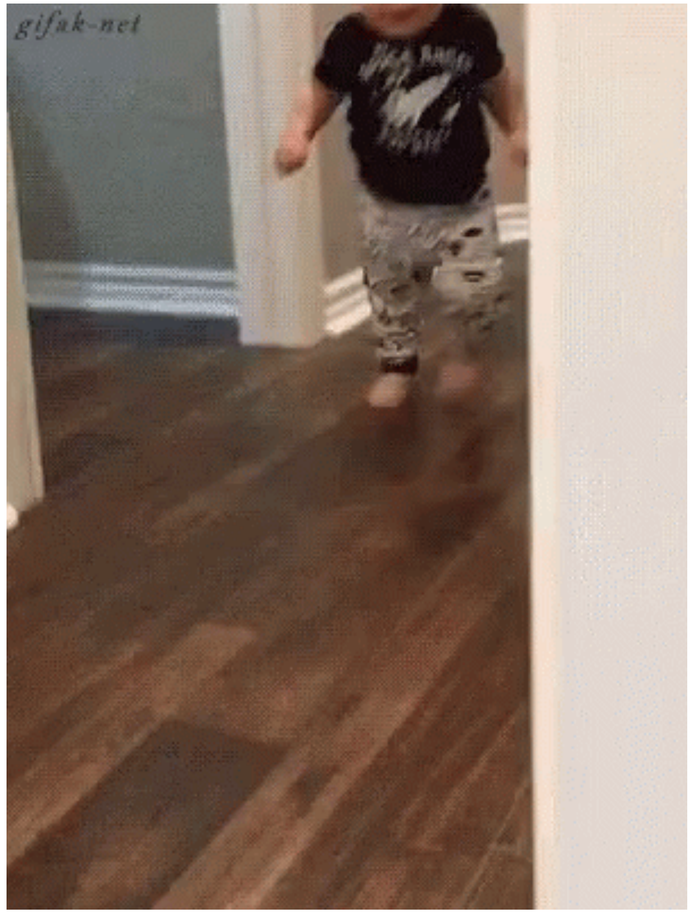
Describe the architecture and behaviour :
Programmer view

```
template<unsigned int BUSWIDTH> PL011<BUSWIDTH>::PL011(::hv::module::ModuleName
name_) : ::hv::reg::RegModule<BUSWIDTH>(name_, 4), […]
{
    this->addRegister(0x000, UARTDR);
    this->addRegister(0x004, UARTRSR_UARTECR);
    this->addRegister(0x018, UARTFR);
    […]

    UARTCR.createField("RXE", 9, 9, "Receive enable");
    UARTCR.createField("TXE", 8, 8, "Transmit enable");
    UARTCR.createField("LBE", 7, 7, "Loopback enable");
    UARTCR.createField("SIRLP", 2, 2, "SIR low-power IrDA mode");
    UARTCR.createField("SIREN", 1, 1, "SIR enable");
    UARTCR.createField("UARTEN", 0, 0, "UART enable”);
    […]
}

template<unsigned int BUSWIDTH> void
PL011<BUSWIDTH>::clearIRQ(const ::hv::reg::RegisterWriteEvent &ev) {
    UARTRIS = UARTRIS & ~ev.newValue;
    this->updateIRQ();
}
```

That’s look amazing right ? But… This is what people feel when they want to start to build a virtual prototype



Main issues

No fast growing model catalog for virtual prototype

- On request development of model
- Missing models of right level of abstraction
- Model development mainly driven by IP design & verification
- **No adapted open source offer**

Insufficient commonalization of efforts on models

- Too much components and some are too complex
- Too fast growing offer of components
- **And no share of industrial efforts?**

Virtual prototype heavy deployment

- Hard to start from scratch
- Model **interoperability doesn't mean easy reuse**
- On premise framework deployment missing
- Missing deployment solution suited for growing virtual prototype
- Missing scalable solution for simulating full system (ex.: aircraft, car…)
- DevOps needs to be adapted to solution (including HW and SW)

What's the most important with Virtual Prototypes

INTEROPERABILITY

Open Standard API is the key. With an open source approach, each member of the eco system can choose their tools.

EASY OF USE

You don't want to spend time to learn all the technology behind to run your lovely Raspberry Pi.

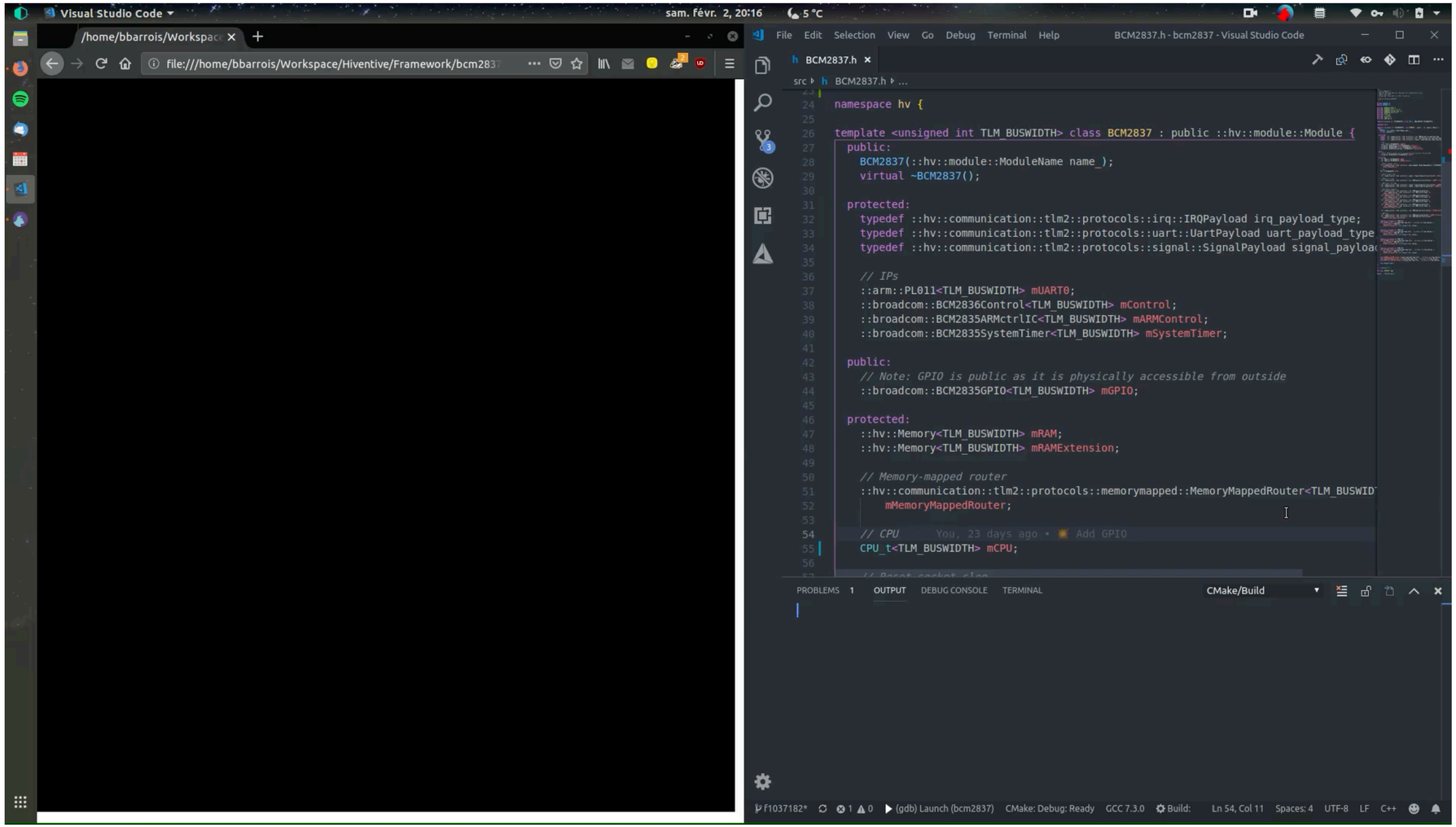
REUSABILITY

Do not reinvent the wheel.
Take advantage of community.
Improve, contribute instead of duplicate.

EASY TO BUILD

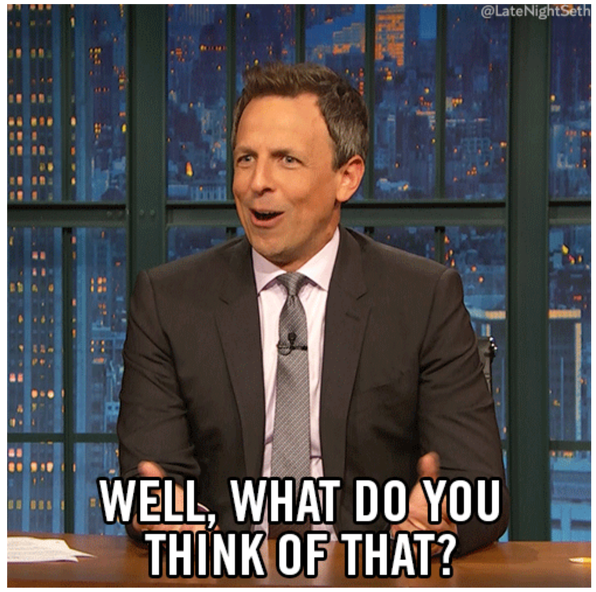
Building models should be accessible for anyone. We're even considering Python, Go, .. (higher abstraction languages to describe them)

Demo time - But no time for real



Time is over... but join the beta !

Get ready to speed up
your next design
Try our virtual prototypes
(like Raspberry Pi)



Complete release on GitHub expected in 2 months

https://github.com/hiventive