ing. Jerry J.J. Jacobs

Embedded Systems Engineer

Education

2010–2013 **Bachelor Embedded Systems Engineering**, *HAN University of Applied Sciences*, Arnhem.
Design and implementation of smart devices based around one or multiple microcontrollers, computer science, digital signal processing, control systems. and I also did a minor in embedded vision.

2006–2010 **MBO Telecommunications ICT Engineer**, *Leeuwenborgh Opleidingen*, Sittard.
Electrical engineering, telecommunications and computer networking

Experience

Feb 2019–current **IoT Cloud Architect**, *Dual Inventive*, Oisterwijk.
Responsible for the Southbound interface of the MTinfo 3000 IoT Cloud architecture which is used in the rail sector. Coupled with safety-critical and sensing devices. The Southbound interface of the cloud is the part where the IoT devices are connected, provisioned, and data routing and storage is located.

Backend technologies used:
Micro-service based architecture running under Debian Linux coupled with ZeroMQ, REST, gRPC, C++11, Golang programming language, Websockets, Redis cache, InfluxDB.

April 2015–Feb 2019 **Embedded Software Engineer**, *Dual Inventive*, Oisterwijk.
Design and implementation of IoT firmware for Cortex-M controllers in C11. And design and implementation of the MTinfo 3000 platform backend based on micro-service architecture. At first we wrote service-side software in C++11, and made the move to Golang for speed of development and improved quality.
I work in the embedded and backend team using SCRUM method with iterations of two weeks. My tasks range from designing, implementing and testing multiple firmware and software projects. I'm also responsible for keeping the Linux DTAP environment servers up and running. And configuring and updating the software with Ansible.

Embedded technologies used:
C programming, Cortex-M microprocessors (STM32), custom DI-Net RPC protocol based on JSON-RPC, CAN-bus, qualification tests automated with PHP.

Backend technologies used:
Micro-service based architecture coupled with ZeroMQ, C++11, Golang programming language, Websockets, Redis cache, InfluxDB.

2013–2015 **Embedded Software Engineer**, *Prodrive Technologies*, Son.

I developed RapidIO control-plane libraries which are used to configure the routing of the RapidIO network switches. Multiple nodes are placed in an ATCA rack interconnected with RapidIO and Ethernet. Test automation was implemented using shell scripts and C libraries from a Linux host. Nodes boot over the network using TFTP and NFS. Qualification of the software was fully automated and reports where automatically generated using a custom LaTeX package which I implemented and was adopted company wide.
Prodrive collaborated with Integrated Device Technology, Inc. (IDT) to design and develop the RapidIO networking stack in user-space. I contributed to the kernel driver which is capable to expose the interconnect to user space. And the driver has landed into the Linux kernel source tree `https://github.com/torvalds/linux/blob/master/drivers/rapidio/devices/rio_mport_cdev.c`

Technologies used:
C programming, Embedded Linux (wind river linux, buildroot), boot-loader, RapidIO and Ethernet interconnect, shell scripting for automation.

2013–2015 **Graduate Embedded Software Engineer**, *Prodrive Technologies*, Son.

I did my final thesis at Prodrive for my Bachelor degree. My thesis subject was focused on a modular embedded linux board support package (BSP) for Freescale i.MX 6 platform. Which includes a custom U-Boot boot loader with A-B dual partition scheme for seamless system updates. The BSP would be used for sensing and control products.

Technologies used:
C programming, Embedded Linux (custom distribution with busybox), boot-loader, gnu make and shell scripting.

2011–2012 **Embedded Software Engineer**, *Artron B.V.*, Arnhem.

During my Bachelor study I worked part-time for Artron to design and implement flexible firmware for interactive led walls. The panel is split into a modular grid of nodes which have a few RGB leds and an capacitive sensor per led. The nodes where connected using a RS485 half-duplex network which is controlled from an STM32 microcontroller which functioned as the bus-master.

Technologies used:
C programming (protocol, drivers and middleware), Protothreads, AVR and Cortex-M3 microcontrollers, RS485 interconnect.

2008–2010 **Technical Lab Assistant**, *Elektor International Media B.V.*, Limbricht.

During my MBO Telecommunications ICT study I was an intern and graduate at Elektor. In my spare time I also worked in the Lab. My main responsibilities where to re-implemented sent in prototype microcontroller based circuits by hobbyists and verify they're working. I even got the chance to implement my home designed 3D led cube and publish it in the magazine during my intern-ship.

Technologies used:
Altium PCB designer, C programming (firmware), AVR microcontrollers.

Languages

Dutch **Native**
English **Fluent**
German **Basic**

Computer skills

I own a wide range of computer skills which are summed up per category below, and are ordered from most experienced to less.

- **Operating systems**: Linux (Debian, Fedora, Slackware), macOS, Windows
- **Programming languages**: C, Golang, C++11, PHP, POSIX shell/bash, Python
- **Embedded systems**: build and test automation for multiple architectures, Linux kernel programming, U-boot boot loader, AVR, Cortex-M and MSP430 microcontrollers
- **Protocols and interfaces**: CAN-bus, AT-commands, Websockets, HTTP, REST, gRPC, TCP/IP
- **Databases**: Redis, MySQL, InfluxDB
- **Automation and DevOps**: CMake build system, Ansible configuration management, Jenkins continuous integration, continuous deployment for Debian based production systems
- **Miscellaneous**: Test report automation using LaTeX

Open Source

I'm an open source enthusiast and promoter since I started using Linux in 2006 personally. In my free time and professional career I created and/or maintain the following projects on Github:

- **xor-gate/debpkg**: Software packaging for Debian written in pure Golang
- **syncthing/syncthing-macos**: Syncthing bundle for macOS written in Objective-C and Swift
- **texane/stlink**: STM32 programming and debugging tools written in C

Personality

I'm a self-taught, structured and motivated person. I like to work in a team, and I can also solve complex problems on my own. I am flexible employable and driven to reach my deadlines without an 9-to-5 mentality.

Interests

I have a broad range of interest from *beer brewing* to *cycling* and *electronic music*. I love to *cook* different kinds of *foods* from kitchens all over the world. And enjoy *growing* my own vegetables in my community garden. I like to socialize and talk and I'm always happy to meet new people.