Quality assessment and assurance of embedded systems

Embedded systems must fulfill a number of indispensable requirements when using them in industrial products. In addition to general reliability and stability, a case-specific functionality must be provided, but all of this may not be readily available. Therefore, the OSADL QA Farm was established in 2010 where a wide variety of more than hundred different industrial embedded systems is monitored continuously. The resulting data are primarily observed by companies who have provided the systems, and the data are used to optimize and fine-tune the system hardware and software. Furthermore, a large part of the data is made available to the public and may be used to select the most appropriate embedded system hardware for a given use case. Nevertheless, the collected data may be applied—at least in part—as field data in a certification process for functional safety. Finally, the setup of the OSADL QA Farm may also serve to create testing scenarios for answering scientific questions on the behavior of hardware and software components of embedded systems.

The OSADL QA Farm

Realization

The OSADL QA farm consists of a number of test racks (see Figure 1) each of which provides eight slots for test tablets. The racks can be located elsewhere in the world, the only requirements are power supply and Internet access. In the beginning of 2016, more than ten racks are located at three different test centers in Germany.

- 10/100/1000 Mb/s network switch with port mirroring
- 8-channel power distribution unit with power metering
- 8-port serial-to-network adapter
- 8-way keyboard/video/mouse adapter with network access (optional)
- Server for cross development and as peer for generating network load

Figure 1: OSADL test rack
The test systems are mounted on special DIN rail tablets (see Figure 2) – one per each of the eight slots. Electrical power is provided through a 220-V socket; two RJ45 connectors are used for network access and RS232 system console interface. They are connected to the related counterparts of the rack control systems. An optional VGA graphics connector as well as a USB or PS/2 keyboard and mouse connector may be added if needed.

![Figure 2: Tablet for test systems mounting](image1)

All test racks are connected to a central processing server via VPN channels for maintaining, storing and visualizing the collected data (see Figure 3). This server is equipped with appropriate software to allow inspection of the data from all over the world using a standard web browser. Alarm thresholds are defined for many variables and categorized into “warning” and “critical” level. In case an alarm threshold is exceeded, a previously assigned contact person will be notified by an escalation system. Email, SMS, fax and voice message may be selected as means of communication. In addition, the particular variables that are in “warning” or “critical” state are highlighted in the web interface in yellow or red color, respectively.

![Figure 3: Communication among the various OSADL QA Farm components](image2)
Hardware testing

The embedded systems under test originate from two major sources, i) OSADL member companies that are interested in the stability and reliability data of their systems, and ii) systems provided by OSADL for the development of the Linux kernel. The latter are updated continuously to the latest kernel version, while the member-provided systems either may also be updated regularly or run a stable kernel. Which strategy is used depends on the arrangement made. Of particular interest is kind of a “shadow measurement” where two identical systems are monitored — one with a stable kernel to generate field data and one that is used to test whether bleeding-edge kernels run similarly well on the same hardware. This assures the companies that an update to a new kernel version is possible anytime without requiring substantial development and testing.

According to the usual deployment in industrial embedded systems, the systems are equipped with processors of the ARM, MIPS, PowerPC and x86 family, but MIPS and PowerPC are fewer in number. Some of the processors have just been launched on the market, while some others may have a 20-year old design. The latter is important because industrial systems could perfectly have such a long life cycle, and it must be assured to install even the latest Linux kernel on them should this be required. A selection of processors tested in the OSADL QA Farm is shown in Table 1. Particular emphasis was laid on the availability of a broad range of clock frequency, memory size and processor topologies whenever possible. For instance, the clock frequencies range from 133 to 4,000 MHz, and the memory size ranges from as little as 26 MByte up to 64 GByte. Beside single-core processors of former generations, multi-core processors with up to 32 cores as well as mixed multi-core systems with several sockets and nodes are used. In addition, when selecting chip sets and peripheral devices care was taken that a wide variety of different controllers of different manufacturers are under test.

<table>
<thead>
<tr>
<th>ARM</th>
<th>Altera</th>
<th>SOC FPGA Cyclone V @600 MHz, 32 bit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Broadcom</td>
<td>BCM2708 @700 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td>Freescale</td>
<td>i.MX27 @400 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i.MX35 @532 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i.MX53 @886 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i.MX6 X4 @996 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td>Marvell</td>
<td>SheevaPlug @1200 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td>Texas Instruments</td>
<td>AM3517 @600 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OMAP3525 @720 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OMAP4430 X2 @1008 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OMAP4460 X2 @1200 MHz, 32 bit</td>
</tr>
<tr>
<td></td>
<td>Xilinx</td>
<td>Zynq 666 MHz, 32 bit</td>
</tr>
</tbody>
</table>

| MIPS | ICT | Loongson 2F @800 MHz, 64 bit |

| PowerPC | Freescale | MPC 5200 @396 MHz, 32 bit |
|         |          | P2020 @1200 MHz, 32 bit |

| x86/x86_64 | AMD | K6 3D, @333 MHz, 32 bit |
|           |     | LX-800 @500 MHz, 32 bit |
|           |     | Athlon XP 2000+, 32 bit |
|           |     | Athlon 64 2800+, 64 bit |
|           |     | G-Series T56N @1400 MHz, 64 bit |
|           |     | Phenom II X6 @3200 MHz, 64 bit |
|           |     | Opteron X3 @2100 MHz, 64 bit |
|           |     | FX-8150 X8 @3600 MHz, 64 bit |

| Intel | Pentium @133 MHz, 32 bit |
|       | Atom D510 @1667 MHz, 64 bit |
|       | Atom N270 @1600 MHz, 32 bit |
|       | Atom D2700 @2133 MHz, 64 bit |
|       | Celeron M @1500 MHz, 32 bit |
|       | Pentium M @2300 MHz, 32 bit |
|       | Xeon @2000 MHz, 32 bit |
|       | Core 2 Duo @2400 MHz, 64 bit |
|       | Core 2 Quad @2400 MHz, 32 bit |
|       | Nehalem 975 @3333 MHz, 32 bit |
|       | Gulftown X990 @3467 MHz, 64 bit |
|       | Core i7-3770 @3400 MHz, 64 bit |
|       | Xeon E3-1220L V2 @2300 MHz, 64 bit |
|       | Core i7-4960X @3600 MHz, 64 bit |
|       | Core i7-5960X @3000 MHz, 64 bit |
|       | Celeron N3150 1800 MHz, 64 bit |

| VIA | C3 Samuel 2 @533 MHz, 32 bit |
|    | C7 @1000 MHz, 32 bit |
|    | Nano X2 L4050 @1400 MHz, 64 bit |
|    | QuadCore L4700 @1200 MHz, 64 bit |

Table 1: Manufacturers and architectures of systems at the OSADL QA Farm (selection)
Monitored variables

The variables measured at the OSADL QA Farm can be divided into the groups benchmark, disk, network, NFS, processes, real-time system, e-mail, sensors, time synchronization, system and virtualization. Table 2 shows an overview of the measured variables.

**Benchmarks**
- GL benchmark gltestperf
- UnixBench (multi-core)
- UnixBench (single-core)
- UnixBench 2D graphics performance

**Disk**
- Disk I/Os per device
- Disk latency per device
- Disk throughput per device
- Disk usage in percent
- Disk utilization per device
- File system mount-scheduled checks
- File system time-scheduled checks
- Filesystem usage (in bytes)
- Inode usage in percent
- IO Service time
- IOStat
- S.M.A.R.T values of every drive

**Network**
- eth0 errors
- eth0 traffic
- Firewall Throughput
- HTTP loadtime of a page
- Netstat

**NFS**
- NFS Client
- NFSv4 Client

**Processes**
- Fork rate
- Number of threads
- Processes
- Processes priority
- Vmstat

**Real-time system**
- 5-min max. timer and wakeup latency
- 5-min max. timer offsets
- 5-min max. wakeup latency
- RT Features

**Email**
- Sendmail email traffic
- Sendmail email volumes
- Sendmail queued mails

**Sensors**
- Fans
- HDD temperature
- Power consumption
- Temperatures

**Time synchronization**
- NTP kernel PLL estimated error (secs)
- NTP kernel PLL frequency (ppm + 0)
- NTP kernel PLL offset (secs)
- NTP states
- NTP timing statistics for system peer

**System**
- Available entropy
- C states
- CPU frequency
- CPU usage
- File table usage
- Individual interrupts
- Inode table usage
- Interrupts and context switches
- Kernel version
- Load average
- Logged in users
- Memory usage
- Split memory usage
- Application memory usage
- Swap in/out
- Uptime

**Virtualization**
- Virtual domain block device I/O
- Virtual domain CPU time
- Virtual domain memory usage
- Virtual domain network I/O

Table 2: Monitored variables at the OSADL QA Farm (selection)
Variables with special importance for the use of embedded systems in industry

Variables that are especially important for the use in industrial embedded systems are those that are related to the system’s response to asynchronous external and internal events (real-time capability) as well as to the temperature profile and power consumption at different load scenarios. Additionally, the particular performance characteristics of CPU, FPU and GPU must be determined and registered for comparison purposes. Last but not least, it is necessary to record the version and release numbers of the Linux kernel in order to be able to verify whether a kernel upgrade introduced a regression of one of the measured variables. Recordings of the response delay as a measure of a system’s real-time capability as well as temperature profile, fan speed, clock frequency, sleep stages and power consumption are described below. In addition, an example of the registration of the Linux kernel version is given.

Response delay (Real-time capability)

Whenever a real-time process must resume execution due to an elapsed timer, the programmed alarm time is compared to the effective time, and the difference is recorded as timer latency in a histogram. For multi-core processors the value of every single core is recorded. For processors with energy-saving mechanism the wake-up latency can depend on the current energy-saving mode as shown in Figure 4. This figure presents the latency maximum of consecutive five-minute measurement periods.

![Figure 4: Continuously recorded latency over 30 seconds](image)

After execution of the timer’s interrupt service routine, the scheduler sets the state of the waiting user space process, if any, to “runnable” which finally will result in the so-called context switch after which execution in user space is resumed. The time differences between the effective alarm and the end of the context switch are measured as well, recorded in an histogram and their maximum graphed in 5-minute intervals. The relevant total sum of timer latency and scheduler latency is stored, processed and presented in the same manner in a third histogram (Figure 5). This sum corresponds to a very large extent with the so-called preemption latency and represents an important measure of the real-time capability of a given system. The advantage of this method is that the recording can be done continuously and independently from the system – no additional test tool is
needed. Furthermore, in case of an anomaly of the system, it is possible to retrospectively analyze whether the anomaly was correlated to and probably caused by an unusual increase of the preemption latency.

Stimulated preemption latency:
Furthermore, timer interrupts with a frequency of 5 kHz are generated two times the day over five and a half hours, and the delay between the programmed and the effective wake-up time again is measured and registered – however, this time directly in user space. The frequency of 5 kHz over the named duration results in 100 million wake-up cycles per processor and core. The longest preemption latency ever measured serves as a measure of the real-time capability of the system. For the presentation of the results, a histogram with the counts of all 100 million values is used. The histogram has a linear x axis and a logarithmic y axis to visualize even very low sample sizes. This is important, because the longest ever measured wake-up latency normally occurs only a few times or even only once. This is the case in the example shown in Figure 6, where such a measurement with 100 million cycles was performed; the maximum overall latency of the system of 55 μs was registered only once. This value is the most important result of the measurement with which the real-time capability of the system is classified. In order to achieve a high statistical confidence, the individual histograms of a large number of recordings.

![Figure 5: Continuously recorded total sum of timer and wake-up latency for the preemption latency over 30 hours](image)
are combined in a joint evaluation. Then again, the longest ever measured wake-up latency is identified – this time, however, not from only a single measurement but from many thousands of consecutive measurement periods. When measuring twice a day over a total of 2,000 measurement periods, for example, the measurement time extends to a period of almost three years. The histograms are graphically placed in a row and visualized in pseudo-3D-technology in order to allow an overview of all measurement results. Again a logarithmic y axis is chosen for this image to be able to visualize every single outlier in the form of a thin needle.

The example shown in Figure 7 originates from an embedded system with a faulty network controller. This fault results in very rare exceeded wake-up latencies – namely with a frequency of about 190 outliers in 190 billion measuring cycles. Event this low error rate of about 0.001 ppm is absolutely not acceptable and means that this kind of system cannot and may never be used in an industrial control. It should particularly be pointed out that several periods of up to 20 days occurred during which no increased wake-up latency was recorded at all. This measurement example therefore presents a further important argument for the relevance of the long-term measurements performed at the OSADL QA Farm. However, such faulty embedded systems are a rare exception. In fact, most of the tested systems show a clear deterministic response behavior, i.e. even not a single outlier was detected (e.g. Figure 8).
Figure 7: Consecutive latency histograms with many outliers of a faulty embedded system

Figure 8: Deterministic response behavior of an embedded system ideally suitable for industrial controlling tasks
Temperature profile of the different system components

Modern embedded systems are equipped with a variety of sensors with which the temperature of many system components as well as power supply voltages, fan speed and recently also energy consumption is measured. It is important for the long-term stability and durability of embedded systems to recognize and minimize the temperature of the active system components. However, this measurement is also important to differentiate a – direct or indirect – thermal cause from other causes in case of a failure.

Figure 9: Temperature profile of a multi-core system over 30 hours on load and under resting conditions

Figure 10: Fan speed in parallel with the measurement in Figure 9
When measuring temperature and fan speed simultaneously it is furthermore possible to check whether a fan control circuit, if any, works properly or not. Faultless function is indicated by the fact that the fan speed rises proportionally to the temperature. In case the temperature rises despite increasing fan speed more than expected, then the dust filter probably is clogged; in case the fan speed remains steady, the controller is faulty. In each case a corresponding alarm has to be given as otherwise the system could be damaged because of overheating. Examples of such temperature profiles (processor, motherboard, etc.) are shown in the Figures 9 and 10. The correctly working fan control is easily recognized by the fact that the higher temperature of the CPU cores results in an increase of the fan speed and therefore in a reduction of the temperature of the motherboard. At the same time, however, the ventilator capacity of the fan is not sufficient to avoid any temperature rise of the CPU cores, and the temperature rises shortly above the significant warning limit of 80°C. The higher temperature periods are related to the special stress scenarios during which the system has to complete a set of defined load cycles.

**Version numbers of the Linux kernel**

Beside the particularly installed version of the Linux kernel (major number) the patch level (minor number) and the sub-level of the stable tree are recorded as well. As far as real-time kernels are concerned, the RT release is additionally registered. As an example, Figure 11 shows a 13-months registration of the Linux kernel version from linux-3.0.37-rt54 to linux-3.12.10-rt15.

![Kernel version by year](image)

Figure 11: Course of the version numbers of the Linux kernel in a 12-month period

The dates of the upgrades of the respective kernel versions can be seen clearly. Mainly, the recording is done to detect so-called regressions, e.g. unwanted deterioration of any performance parameters. If this ever is the case, in principle a cause analysis can be carried out with the help of the recorded version numbers prior to and after deterioration, and the upgrade patch can be corrected or reversed. In fact, considerable changes in the performance of single components can occur as shown in Figure 12 in which the results of the 2D-performance measurement taken twice a day of the accelerated graphics controller can be seen. It is obvious that the performance increases considerably from around May 12, which was obviously caused by userspace components. The increase in performance is completely destroyed by a kernel upgrade. A further kernel upgrade at the end of January only results in the restoration of the former performance of the text output, whereas all other tests still show a low performance. It is assumed that in this special case the version upgrades of the kernel are too far away
and the data recorded are not sufficient in order to find the reason of this particular regression. However, it is conceivable that the parallel recording of kernel version number and characteristics of the system performance can enable a complete regression monitoring of the kernel development.

Clock frequency, sleep stage and energy consumption

There is no doubt that the reduction of the energy consumption of electrical devices is an important contribution to reduce pollution and protect the environment. Therefore, manufacturers of processors and controllers for embedded systems also have been taking actions to reduce the energy consumption of embedded systems. At first, this applied to CPUs and graphic processors (GPUs), but also other components with relatively high power consumption such as memory chips and communication systems are concerned. The main principle in all these cases is to switch off unneeded system components or to run them at a lower speed when less performance is needed. As far as CPUs are concerned, so-called P states and C states were introduced; the first provides the gradual throttling of the processor clock frequency, the latter refers to so-called sleep stages. Therefore, clock frequency and sleep states of the processors as well as energy consumption of the systems are continuously recorded at the OSADL QA Farm. Figures 13 to 15 show the respective course of these correlated variables in a 30-hour recording. The higher energy consumption at higher clock frequency and while entering sleep stages is disabled can be seen clearly.
Figure 13: Clock frequency of the eight cores of a multi-core system

Figure 14: Sleep stages (all cores averaged) of the system of Figure 11

Figure 15: Power consumption of the system of Figure 11