Porting GNAT to Lego Mindstorm NXT: an Education-oriented Platform for High-Integrity Systems

Piotr Piwko
piotr.piwko@gmail.com

Matteo Bordin
bordin@adacore.com
Summary

• Introduction & Motivations

• GNATPro

• OSEK Operating System

• NXT Ada Library

• Demo application

• Future improvements
Introduction & Motivations

- **Project consisted in:**
  - Porting the GNATPro ZFP Technology to ARM 7
  - Writing an Ada binding for the nxtOSEK API
  - Developing a working prototype

- **Motivations to have a port of GNATPro to Mindstorms**
  - A low-cost platform for high-integrity system (HIS) development
  - Ideal for education: industrial need of experts in HIS
GNATPro

A complete development environment for Ada 83, 95 and 2005

GNAT Pro

GNAT Pro High Integrity Edition

Native Development

Embedded Development

Safety-Critical Development
GNATPro Zero Foot Print (ZFP)

- **GNATPro** ([http://www.adacore.com/home/gnatpro](http://www.adacore.com/home/gnatpro))
  - Compilation chain
  - Static analysis tools
  - IDE (including Eclipse plug-in)
  - Qualification material

- **Zero Foot Print (ZFP)**
  - A subset of Ada 2005
  - Requires minimal run-time support
  - Extremely low memory requirements
  - Common in HIS development
    - Aerospace (DO-178 certification), industrial automation
OSEK overview (http://www.osek-vdx.org/)

- OSEK - "Open Systems and the Corresponding Interfaces for Automotive Electronics"
- Single processor operating system
- OSEK is designed to require only a minimum of hardware resources
- OSEK Implementation Language (OIL)
- The code could be executed from ROM
- Real-time multi tasking
OSEK history

- History timeline of the OSEK standard

May 1993

OSEK has been founded as a joint project in the German automotive industry.

Initial project partners:
BMW, Bosch, DaimlerChrysler, Opel, Siemens, VW and the University of Karlsruhe as coordinator.
OSEK history

- History timeline of the OSEK standard


1994

The French car manufacturers PSA and Renault joined introducing their VDX-approach (Vehicle Distributed eXecutive)
OSEK history

- History timeline of the OSEK standard


**September 1995**

1.0 version of the OSEK Operating System
OSEK history

- History timeline of the OSEK standard


June 1997
2.0 version
OSEK history

- History timeline of the OSEK standard


January 2003

The latest 2.2.1 version of OSEK Operating System
OSEK compilation chain

- Optional OSEK Builder
- Application configuration files (OIL)
- System Generator (SG)
- Files produced by SG
- Compiler
- Linker
- Executable file

Source: OSEK Implementation Language Specification 2.5
OSEK application configuration files

- OSEK Implementation Language (OIL)
- The main OIL objects:

**TASK**

```ada
TASK My_Task
{
    AUTOSTART = FALSE;
    SCHEDULE = FULL;
    PRIORITY = 3;
    ACTIVATION = 1;
    RESOURCE = My_Res;
    EVENT = My_Event;
    STACKSIZE = 512;
};
```

**COUNTER**

```ada
COUNTER System_Counter
{
    MINCYCLE = 1;
    MAXALLOWEDVALUE = 10000;

    /* One tick is equal to 1msec */
    TICKSPERBASE = 1;
};
```

**EVENT**

```ada
EVENT My_Event
{
    MASK = AUTO;
};
```

**RESOURCE**

```ada
RESOURCE My_Res
{
    RESOURCEPROPERTY = STANDARD;
};
```
OSEK application configuration files

- **ALARM**

```
ALARM Cyclic_Alarm
{
    COUNTER = System_Counter;
    ACTION = ACTIVATETASK
    {
        TASK = My_Task;
    }
    AUTOSTART = TRUE
    {
        ALARMTIME = 1;

        /* My_Task is executed every 500msec */
        CYCLETIME = 500;
    }
};
```
nxtOSEK (http://lejos-osek.sourceforge.net/)

- **Open source firmware for LEGO MINDSTORMS NXT**
- **nxtOSEK provides:**
  - ANSI C/C++ programming environment by using GCC tool chain
  - C / C++ API for NXT Sensors, Motor, and other devices
  - TOPPERS OSEK – The open-source implementation of OSEK Real-Time Operating System that includes ARM7 specific porting part

- **Features**
  - USB
  - I2C bus
  - Floating Point
  - Direct boot from RAM (max. 64Kbytes single nxtOSEK application uploaded to RAM, no Flash write)
nxtOSEK + GNATPro ZFP

Source code which is written in Ada

Ada binding library for the nxtOSEK API

GNATPRO 6.1.1 arm-elf cross-compiler
NXT Ada Library

- Structure of the Ada binding library

```plaintext
nxtOSEK
  .OS
  .NXT
  .Types
  .Utils
  .Bluetooth
  .I2C
  .Sound
  .Display
  .Motor
  .Acceleration_Sensor
  .Gyrosopic_Sensor
  .Light_Sensor
  .Sonar_Sensor
  .Sound_Sensor
  .Touch_Sensor
```
Overall view

- Relation between Compiler, OS and API bindings

```ada
with nxtOSEK.OS;
with nxtOSEK.Display;

procedure My_Task;
pragma Export (C, My_Task, "TaskMainMyTask");

procedure My_Task is
begin
  nxtOSEK.Display.Update;
  nxtOSEK.OS.Terminate_Task;
end My_Task;

package nxtOSEK.Display is
  procedure Update;
private
  pragma Import (C, Update, "display_update");
end nxtOSEK.Display;

package nxtOSEK.OS is
  procedure Terminate_Task;
private
  pragma Import (C, Terminate_Task, "terminate_task");
end nxtOSEK.OS;

TASK MyTask
{
  PRIORITY = 1;
  ACTIVATION = 1;
  SCHEDULE = FULL;
  STACKSIZE = 512;
};

ALARM MyTask_Alarm
{
  COUNTER = SysCnt;
  ACTION = ACTIVATETASK
  {
    TASK = MyTask;
  }
  AUTOSTART = TRUE
  {
    ALARMTIME = 1;
    CYCLETIME = 500;
  }
};
```
It’s demo time! – The Remote Control Vehicle (I)

• **Vehicle**
  - Rear-wheel drive (2 motors)
  - Steering System (1 motor)
  - Ultrasonic Obstruction Detection (sonar sensor)
  - Communication System (Bluetooth – slave device)
It’s demo time! – The Remote Control Vehicle (II)

• Control Panel
  – Motion-based control
    – 1 Gyroscope (steering)
    – 2 Touch sensors (move forward/backward)
  – Communication System (Bluetooth - master device)
Inside the demo

- **Vehicle**

  - **Sonar task**
    - Cycle: 50ms
    - Critical distance: 30cm

  - **Receive task**
    - Cycle: 5ms

  - **Sonar warning flag**
    - Shared resource

  - **Steering system**
    - 1 motor

  - **Main propulsions**
    - 2 motors

  - **Bluetooth protocol**
Inside the demo

- Control panel

  - Movement control
    - 2 Touch sensors
  - Bluetooth protocol
  - Gyroscopic sensor

  - Buttons task
    - Cycle: 200ms
  - Receive task
    - Cycle: 50ms
  - Steering task
    - Cycle: 5ms

  LCD display
Future improvements

• **Support wider subsets of Ada**
  - Ravenscar Profile (language-level tasking)

• **Automated Code Generation**
  - OIL files
  - Application architecture
  - From SysML/AADL models

• **Object-oriented nxtOSEK Ada Library**
Conclusions

- **GNAT for Ada 2005**
  - An industrial-strength development environment for HIS

- **LEGO Mindstorm NXT**
  - An education-oriented platform for embedded systems
  - nxtOSEK: brings RTOS to the LEGO Mindstorm NXT

- **GNAT for NXT**
  - An ideal platform to teach HIS development
  - An answer to the industrial need for education in HIS

- **Planned release: Spring 2009!**