

Molole and ASEBA

Clean up your microcontrollers

Stéphane Magnenat
stephane at magnenat.net

Philippe Retornaz

Francesco Mondada
francesco.mondada at epfl.ch

March 17, 2010

Molole, a clean peripheral wrapper for dsPIC33

Molole: Overview

- ▶ Wrappers for all the peripherals of the dsPIC33
- ▶ Abstraction for GPIO
- ▶ Simple or sophisticated motor modules
- ▶ Centralised, callback-based mechanism for error reporting
- ▶ Clean and unified API
- ▶ Consistent naming style
- ▶ Integrated with Aseba
- ▶ 8500 LOC

Molole: Example of code: ADC

```
#include <adc/adc.h>

void adc1_callback(int channel, int value)
{
    // do something
}

int main()
{
    adc1_init_simple(adc1_callback, 1, 0x7, 31);
    adc1_start_simple_conversion(0);
    while(1); return 0;
}
```

Molole: Example of code: timer

```
#include <timer/timer.h>

void timer1_cb(int timer_id)
{
    // do something
}

int main()
{
    timer_init(TIMER_1, 400, 6) // 400 us
    timer_enable_interrupt(TIMER_1, timer1_cb, 1);
    timer_enable(TIMER_1);
    while(1); return 0;
}
```

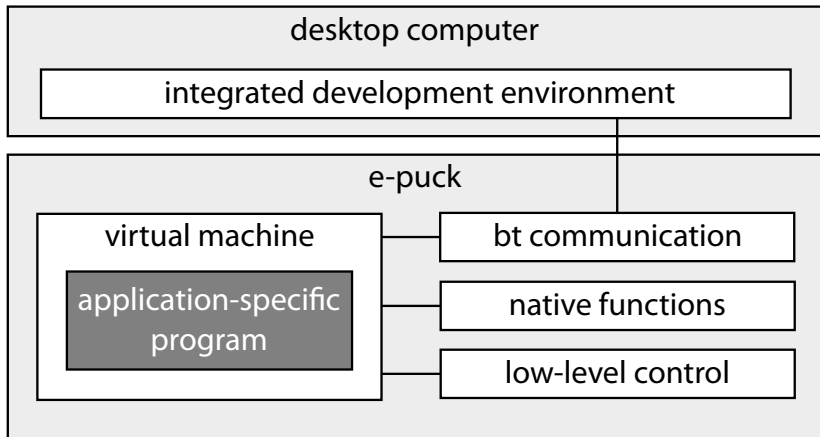
Aseba, script your robots

Aseba: Overview

An event-based framework for robots control:

- ▶ allows fast prototyping of the behaviour of the robot
- ▶ provides an integrated development editor (IDE) for script edition and debugging
- ▶ compiles scripts into bytecode through Bluetooth
- ▶ execute bytecode on microcontrollers in a virtual machine
- ▶ safe execution
- ▶ dynamic enumeration of microcontrollers state variables
- ▶ dynamic reprogramming of the microcontrollers
- ▶ asynchronous code execution upon events
- ▶ save bytecode into EEPROM for autonomous execution
- ▶ open source

Aseba: Structure



Aseba: Virtual Machine

- ▶ executes bytecode,
- ▶ stack based, 16 bits integers,
- ▶ < 1000 lines of C, including debugging logic,
- ▶ RAM: 22 bytes + user defined amount of bytecode, variable, stack, and breakpoints.
- ▶ flash: 7.5 kB flash (dsPIC30, e-puck),
- ▶ no external library requirement, excepted the implementation of bus communication.

Aseba: Language

Simple imperative scripting language, Matlab-like syntax.

- ▶ blocks of code executed upon events
- ▶ 16 bits integer variables and arrays
- ▶ common mathematical expressions and arrays access
- ▶ `if` and `when` conditionals
- ▶ `while` and `for` loops
- ▶ native functions for complex processing

Aseba: Live demo

A live demo is better than a continuous flow of bullets...

Thank you

- ▶ Thank you for your attention
- ▶ Questions are welcome