Domain-Specific Approach to Software Development for Microcontrollers

Sergey Pakharev, Boris Sedov, Alexey Syschikov, Vera Ivanova
{sergey.paharev, boris.sedov, alexey.syschikov, vera.ivanova}@guap.ru
Programming microcontrollers with visual approach

**Mindstorms**

- Support 1-2 platforms
- For education purposes
- For design small programs

**Modkit**

**TRIK studio**

20-24 April 2015  17th FRUCT Conference
Software development tools for embedded microcontrollers

• There are no convenient tools for creating a DSL (Domain Specific Language)
• Really hard to separate process of engineering and programming
Visual Development Environment (VIPE)
Visual Development Environment (VIPE)

Separation process of engineering and programming

Programmer

Expert

+ Support for DSL development

Example code snippet:

```c
int dhCalcAirlight ( DataLink *in11, DataLink *in31, DataLink *out2 )
{
    memcpy(&p, in11->Data, sizeof(int*));
    CImg<double>* brightestDarkPixels = (CImg<double>*)p;
    memcpy(&p, in31->Data, sizeof(int*));
    CImg<double>* data = (CImg<double>*)p;
    float airLight[3] = { 0.0, 0.0, 0.0 };
    **
    * brightestDarkPixels has the (x,y) coordinate
```
Development DSL for Arduino
Domain analysis

- Microcontroller
- Base peripherals
- External peripherals (with digital/analog IO)

Basis for DSL:
- Base constructions of the text language
- Functions for the most commonly used peripheral devices
Developing DSL for Arduino platform
Support of a target platform

Scheme of program in DSL Arduino

Code generator
C/C++
+ extensions
Developing DSL for Arduino platform

If, For, While + arithmetic and logical operations
Part of VPL (Visual Program Language)

Base VPL language

Library elements

Constants

DSL Arduino

20-24 April 2015  17th FRUCT Conference
Developing DSL for Arduino platform

- servo
- ultrasonic sensor
- accelerometer

28+ elements
Integration into the environment

VIPE - unified environment for creating software for microcontrollers

Compiling + uploading in the environment
Use case: automated control system
Portability

Microcontroller DSL

Schemes program

Arduino

FreeScale и STM

Intel Galileo
Results

- DSL for Arduino was designed
- Successfully transferred to the platform
- Visual approach in terms habitual to the user

Developing DSL in VIPE:
- quickly
- efficiently
- small expenses

Futures:
- The uniform environment for creation software for other microcontrollers too and developing of portability of programs
<table>
<thead>
<tr>
<th>Arduino Board</th>
<th>Shield Support</th>
<th>Interactive Tuning and monitoring</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino Due*</td>
<td>Y</td>
<td>Y</td>
<td>DAC and CAN channels not currently supported.</td>
</tr>
<tr>
<td>Arduino Uno*</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Arduino Leonardo*</td>
<td>Y</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Arduino Mega 2560*</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Arduino Mega ADK*</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Arduino Micro*</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Arduino LilyPad USB</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Arduino Esplora</td>
<td>N</td>
<td>N</td>
<td>Additional IO supported via analog multiplexer</td>
</tr>
<tr>
<td>Arduino Robot</td>
<td>N</td>
<td>N</td>
<td>Additional IO supported via analog multiplexer</td>
</tr>
<tr>
<td>Arduino Mini*</td>
<td>N</td>
<td>N</td>
<td>Mini with ATmega168 not supported</td>
</tr>
<tr>
<td>(ATmega328)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arduino Nano 3.X*</td>
<td>N</td>
<td>N</td>
<td>Nano 2.X with ATmega168 not supported</td>
</tr>
<tr>
<td>(ATmega328)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arduino Pro*</td>
<td>N</td>
<td>N</td>
<td>Pro with ATmega168 not supported</td>
</tr>
<tr>
<td>(ATmega328)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arduino Fio</td>
<td>N</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Arduino Ethernet Shield</td>
<td></td>
<td></td>
<td>See Shield Support column for compatibility</td>
</tr>
<tr>
<td>Arduino WiFi Shield</td>
<td></td>
<td></td>
<td>See Shield Support column for compatibility</td>
</tr>
</tbody>
</table>
ARM processor families compatible with Embedded Coder generated code include:

- ARM 7/9/11
- Cortex – A50
- Cortex – A
- Cortex – M
- Cortex – R

http://www.mathworks.com/hardware-support/arm.html
Using scenarios and tools

- **Step 1: Pure simulation**
  - Everything done on the PC

- **Step 2: Processor-in-the-loop (PIL)**
  - Algorithm fully executed on STM32 MCU
  - Data (input or output) exchanged between MATLAB/Simulink and STM32 MCU via UART

- **Step 3: Everything on STM32**
  - Data (input or output) obtained from STM32 MCU through its peripherals (ADC, Timers, ...) and algorithm fully executed on STM32 MCU

---
