Arduino based I/O-system for rapid prototyping of robotic systems

Alexander Entinger

LXRobotics GmbH

alexander.entinger@lxrobotics.com

May 23, 2014
Table of Content

1. Motivation
2. Implementation
3. Verification
4. Applications
Motivation

- Development of robotic systems consume an enormous amount of time
  - Hardware, Soft- and Firmware
  - Interfaces and communication protocols between numerous subsystems

- Observation: Many sensors and actors have similar interfaces
  - IR distance/temperature/light sensor → analog voltage
  - Ultrasonic distance sensor/accelerometer/gyroscope → I2C
  - DC motor, servo motor → PWM

- Intention: Develop I/O-system which aggregates the most common I/O-functions into one single reusable I/O-module
  - Reduce the time required for prototyping robotic systems
  - Stop reinventing the wheel
Implementation

- Hardware (Arduino Uno) + Software
  - Firmware for Arduino Uno
  - C++ Library libarduinoio

Highlevel control of $\mu$C-I/O-functions

```cpp
auto oD2 = io.createGpioOutputPin(arduinoio::D2, false);
bool const success = oD2->setPinValue(true));
bool const pin_value = oD2->getPinValue();
```
Capabilities

- Available via \( \mu C \)-I/O-pins:

<table>
<thead>
<tr>
<th>I/O-function</th>
<th>Number of I/O-entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Input</td>
<td>max. 6</td>
</tr>
<tr>
<td>Digital Input</td>
<td>max. 12</td>
</tr>
<tr>
<td>Digital Output</td>
<td>max. 12</td>
</tr>
<tr>
<td>Hardware PWM</td>
<td>max. 2</td>
</tr>
<tr>
<td>Software PWM</td>
<td>max. 6</td>
</tr>
<tr>
<td>Event Counter</td>
<td>max. 2</td>
</tr>
<tr>
<td>I2C-Bridge</td>
<td>max. 1</td>
</tr>
</tbody>
</table>

- Furthermore:
  - Hardware-Reset
  - 16-Bit Board-ID
  - Chip-Temperature
Verification

- Verification of individual components not constructive → Verification of complete system

- Complete system = C++ library, communication link, Arduino Firmware → **End-To-End-Verification**

- **Idea**: Configure some pins as outputs, others as inputs and interconnect them in a clever way

- **Unit Test Framework** for automatic test case execution
Verification of Analog Inputs

- Digital outputs **create** analog voltages via **R2R resistor network**
- Analog inputs **measure** those analog voltages
- Testframework **compares** measured voltage with calculated voltage

- 12 Digital outputs + 6 analog inputs $\rightarrow$ **6 R2R networks** with 2 Bit resolution

![Diagram of R2R network](image.jpg)
Applications

- **Command execution time (CED):** How long does one command take?

<table>
<thead>
<tr>
<th>Command</th>
<th>CED, [CED] = ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get ID</td>
<td>4.15532 ms</td>
</tr>
<tr>
<td>Set pin value</td>
<td>4.64086 ms</td>
</tr>
<tr>
<td>Get pin value</td>
<td>4.6447 ms</td>
</tr>
<tr>
<td>Get analog value</td>
<td>5.00152 ms</td>
</tr>
<tr>
<td>I2C Write (1 Byte)</td>
<td>4.88282 ms</td>
</tr>
<tr>
<td>I2C Read (1 Byte)</td>
<td>6.25022 ms</td>
</tr>
</tbody>
</table>

- **Conclusion:** Main purpose are applications where CED is not of utmost importance, e.g.
  - status led control, battery voltage monitoring, (servo control), etc.
Mobile Robotic Platform: Beauty Queen

- RC-Car modified for **Simultaneous Localization and Mapping** (SLAM)
- Tasks of Arduino I/O-system:
  - **control** electronic speed controller, status leds, steering and gearbox servo
  - **monitor** battery voltage
Ressources

- Sourcecode released under **Mozilla Public License 2.0 (MPL 2.0)**
  - Commercial applications possible

- https://github.com/lxrobotics/arduinoio
Questions