

Advanced Robot Control

Driver implementation on MCU

Wojciech Domski

Chair of Cybernetics and Robotics,
Wrocław University of Science and Technology

Presentation compiled for taking notes during lecture



Wrocław University
of Science and Technology



- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - Straightforward implementation
 - Logic with hardware abstraction layer
 - Multiple abstraction layers
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations
- 4 Quiz



Driver (1/1)

What is a driver?



Outline

- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - Straightforward implementation
 - Logic with hardware abstraction layer
 - Multiple abstraction layers
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations
- 4 Quiz



Each driver should contain following functions (depending on the type of a device some can be omitted):

- initialization
- deinitialization
- read/write operations
- control operations



Outline

- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - Straightforward implementation
 - Logic with hardware abstraction layer
 - Multiple abstraction layers
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations
- 4 Quiz



Types of driver implementation (1/1)

- Straightforward implementation
- Logic with hardware abstraction layer
- Multiple abstraction layers
- Driver implemented through system calls



Device representation

In order to allow for multiple presence of devices in the system the state of a single device should be separated from other devices. Therefore, the state of the device should be contained. This can be achieved through creation of a dedicated structure which holds all necessary elements including interfaces through which higher layer can communicate with the device.



Outline

- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - **Straightforward implementation**
 - Logic with hardware abstraction layer
 - Multiple abstraction layers
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations
- 4 Quiz



Straightforward implementation

This type of driver implementation characterizes with no separation of hardware dependant interfaces. Therefore, portability is very low and requires a lot of effort to move a driver to a different platform.



Outline

- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - Straightforward implementation
 - **Logic with hardware abstraction layer**
 - Multiple abstraction layers
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations
- 4 Quiz



Logic with hardware abstraction layer

Hardware interface is kept separately from logic of a driver. This kind of implementation characterizes with high portability since only the hardware layer is affected when the code is ported to a different device.



Outline

- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - Straightforward implementation
 - Logic with hardware abstraction layer
 - **Multiple abstraction layers**
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations

- 4 Quiz



Multiple abstraction layers

This type of driver implementation characterizes with more than two layers of abstraction. Usually, there are present two layers – logic layer and hardware layer.



Outline

- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - Straightforward implementation
 - Logic with hardware abstraction layer
 - Multiple abstraction layers
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations
- 4 Quiz



Driver implemented through system calls

The main idea is to assume Linux Device Driver model. Access to a device is realized through implementation of standard system calls [1].



Outline

- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - Straightforward implementation
 - Logic with hardware abstraction layer
 - Multiple abstraction layers
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations
- 4 Quiz



Techniques

One of the most important things when a driver is being implemented is a choice of the hardware decoupling method. It can be done using different techniques.



Outline

- 1 Drivers
 - Introduction
 - Architecture
- 2 Models
 - Models
 - Straightforward implementation
 - Logic with hardware abstraction layer
 - Multiple abstraction layers
 - Driver implemented through system calls
- 3 Software implementation
 - Hardware abstraction layer implementations
- 4 Quiz



Quiz (1/1)

Calculate group number as the rest from dividing the Student ID number by 4.

Example

Student ID number is 123456, thus the group is 0.

Take last 2 digits from Student ID number (56) and calculate the rest from dividing by 4 ($56 \% 4 = 0$).

Write down your name, Student ID number and group.



Literature (1/1)



J. Corbet, A. Rubini, and G. Kroah-Hartman.
Linux Device Drivers, Third Edition.
O'Reilly Media, Inc., 2005.

