

Clock Synchronisation over the Ultra Leightweight Fault Tolerant Real Time Protocol

Robert Annessi, Alexander Heinisch, Nick Mayerhofer

November 2011

Contents

- 1 Project Goal 1**
 - 1.1 Basic Idea 1
 - 1.2 Requirements 1
 - 1.2.1 ULFTRTP 1
 - 1.2.2 Clock Synchronization 2
- 2 Project Management 2**
 - 2.1 Project Idea 2
 - 2.2 Group Subject / Specialization 2
 - 2.3 Role allocation 3
 - 2.4 Project time schedule 3
 - 2.4.1 Gantt Chart 3

List of Figures

List of Tables

1 Project Goal

1.1 Basic Idea

The basic idea of our project is to evaluate different distributed clock synchronisation algorithms over an asynchronous bus communication. Therefore we are going to implement a csma/ca protocol providing basic fault tolerancy methods and an easy way to estimate message round trip time and message prioritising. Upon the protocol, detailed described in section ??, we settle the algorithms for the clock synchronisation.

Each node has divers hardware such as buttons, leds, a bulb, a lcd and some other things as its periferial. We want to display drift rates of clocks on the lcd and trigger faults in the nodes internal clock or cause bus overloads with the connected buttons.

The outcome of this project should be the experience in the field of asynchronous real time bus engineering and to be able to estimate influences of overloads and faults to applications upon them. We try to give a link between theoretical and practical aspects and how they relate.

1.2 Requirements

1.2.1 ULFTRTP

Req 1 *analyzeable*: *the protocol has to be relatively easy to analyze with respect to worst case timing.*

Req 2 *interfacing*: *the protocol design has to follow strictly interface guidelines. This means:*

- 1. lower levels of the protocol can only be accessed by higher levels through the defined layer interfaces.*
- 2. higher levels of the protocol cannot be accessed by lower levels. Data to higher layers can only be propagated using callback mechanisms.*

Req 3 *migration*: *the protocol has to be migratable in arbitrary applications with minimal effort.*

Req 4 *resource consumption*: *the protocol has to be adaptable to minimal hardware constraints.*

1.2.2 Clock Synchronization

Req 5 *analyzeable*: *the clock synchronisation has to be relatively easy to analyze.*

Req 6 *exchangeable*: *the specific clock synchronisation algorithms have to be easily interchangeable with other clock synchronisation algorithms, as we want to try out different algorithms.*

This Requirement correlates with Requirement [2](#).

2 Project Management

2.1 Project Idea

Our Claims are:

- Favorable
- easy to use (or hobbyists / tinkerers)
- low resource requirements

These ideas are leading to the following main points that our protocol has to provide:

- cost estimation
- feasibility
- cheap protocol for automation in semi-professional level
- low migration effort
- poor security measures for low cost

2.2 Group Subject / Specialization

- Specify a suitable communication protocol meeting the prerequisites of the bus properties
- Define appropriate abstraction layer to provide easy extensibility and module-oriented development
- Clock-synchronization

2.3 Role allocation

Description	Allocation
Project Manager	Nick
Technical Manager	Alex
Documentation Manager	Robert

2.4 Project time schedule

We elected eGroupware for our project management matters that provides: Time managemnt, Issue tracking, ToDo management and cost tracking.

2.4.1 Gantt Chart

from 28.10.2011 to 17.01.2012

