TIMICO Middleware and Communication Protocols for Dependable Systems
Module 4: Time Triggered Systems and Protocols

Practicalities

About: This note covers a module. A module consists of two consecutive lecture days.
Lecturer: Finn Overgaard Hansen, office E-305. foh@ase.au.dk
Revision: November 13, 2014.

Subject

The theme for this module will be an introduction to time triggered systems and protocols for hard real-time and dependable distributed systems.

Introductory background readings:

- Wikipedia: Time Triggered Protocol
- Wikipedia: FlexRay

Agenda

Day 1

- **Lecture 4.1: Introduction to Time Triggered Systems**
- **Lecture 4.2: Time triggered architecture (TTA) and TTP protocol**
- Exercise 1.
- Exercise 2.

Day 2

- **Lecture 4.3: Time Triggered Flexray Protocol**
- **Student Article Presentation:** Reading no. 3.

Details

Day 1

The lectures for this day are based on an in-depth article written by Professor Hermann Kopetz (1), who invented the Time Triggered Protocol (TTP) and the higher layer architecture called the Time Triggered Architecture (TTA).

- **Lecture 4.1: Introduction to Time Triggered Systems**
  The first lecture will give a general introduction and motivation for time triggered systems and the corresponding communication protocols.

- **Lecture 4.2: Time triggered architecture (TTA) and the TTP protocol**
  The second lecture will presents structure and principles for the TTP (Time Triggered Protocol) and the corresponding TTA (Time Triggered Architecture). TTP is currently used in high demanding distributed control systems for aircrafts and for industrial systems. Reading 2 gives a short introduction to TTP.
- Exercises: Time Triggered Systems, task 1-7.

Day 2

The FlexRay protocol is like TTP, a protocol designed for hard real-time distributed, dependable and safety critical systems and is a competing standard to TTP.

- **Lecture 4.3: Time Triggered Flexray Protocol**
  In this lecture we will study the Time-triggered FlexRay communication protocol . This lecture is based on input from the “FlexRay_ReqSpec_2, FlexRay consortium paper, April 2002 (4), which specifies a communication protocol for reliable automotive applications supplemented with other sources. The lecture will presents the structure of and the principles for the FlexRay communication protocol with focus on its differences from TTP. The FlexRay protocol have both Time-Triggered windows and additional also support for dynamic Event-triggered communication.
  FlexRay is currently a de-facto standard for high demanding applications in the automotive industry for implementing the concept called "drive-by-wire". The FlexRay protocol is specific designed to be used in the automotive industry, where it has now ended the battle, as the main part of the world’s automotive companies now has chosen this standard. The BMW Company has in the fall of 2006 produced and shipped the first car model X5, where the Flexray protocol is used (see the slide presentation).

- Student Article Presentation (3): *"A Distributed Computing Environment for Embedded Control Systems with Time-Triggered and Event-Triggered Processing"*. This paper relates both to this week and the next week course topics.

Readings

1. H. Kopetz & G. Bauer, *"The Time Triggered-Architecture"*, Proceedings of the IEEE, vol. 91, No. 1, January 2003, pp. 112-126.
   - This paper introduces the basic principles of a time-triggered architecture for real-time systems used in safety-critical applications. Read it for lecture 6.1.
2. R. Bannatyne, *"TimeTriggeredProtocol: TTP/C"*, Embedded Systems Programming, March 1999, pp. 76-86.
   - A short overview article presenting the time triggered protocol TTP/C for reliable Hard Real-Time Systems and at that time its intended use in the automotive industry. Lecture 6.2 will use material from both the Kopetz article and from the Bannatyne article.
3. Y. Itami et al., *"A Distributed Computing Environment for Embedded Control Systems with Time-Triggered and Event-Triggered Processing"*, 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 45-54, 2008.
4. R. Belschner et al., *"FlexRay Requirements Specification"*, FlexRay consortium, April 2002, page 1-52.

Slides

- Time triggered systems
- Time triggered architecture and protocol
- FlexRay protocol

Exercises

Goal: Obtain experience with design of a Time-triggered application using TTP and/or FlexRay.

Assignments:

Exercise 1. Questions regarding TTP

1.1 Explain the concept "Composability" and describe the advantages with this concepts in comparison with traditional RMI based distributed systems.

1.2 Explain how the sender of a given message is identified and how receivers of a given message are defined.

1.3 What is a babbling idiot?

1.4 How is such a malfunction avoided in TTP?

1.5 Explain the available mechanisms supported for obtaining a fault-tolerant and secure system.

1.6 Calculation exercise A:
Calculate the protocol overhead for a normal frame with 1) a user message consisting of 1 byte user data and 2) for a user message consisting of 240 bytes user data. (NB! The CRC is 24 bit).

1.7 Calculation exercise B:
Each frame on the bus must be separated by an IFG (Inter Frame Gab) for protocol services. Assume this IFG is min. 3 µS, the maximum frame is 27 µS, all frames are of equal length, all 8 nodes send a maximum frame and the transmission speed is 10 Mbit/s. A cluster consists of 1 TDMA round.
How many frames/messages can be send pr. sec?
How many user bytes can be hold in such a frame and who many bits?
What is the max. sampling rate in KHz for one of the 8 nodes?
What is the effective bandwidth for the user data in Mbit/s ?

Exercise 2. Design of a Car with Cruise Control function based on TTP.

In this car the connection between the accelerator pedal and the motor control is solely controlled by a wired TTP based solution and the same for the brake system.

The Car design consists of the following types of ECU (Electronic Control Units):

A Dashboard ECU (input: actual speed, dashboards displays the actual speed in km/h and status for the Cruise Controller: "Cruising xx km/h" or "blank" and motor status signals. Output signals: button activation of CRUISE_ON, CRUISE_OFF, RESUME, INC_SPEED, DEC_SPEED buttons).
An Accelerator Pedal ECU (continuous signal, indicating the pedal activation from 0-100%).
A Brake pedal ECU (output signal: continuous signal indicating the brake pedal activation from 0-100%).
A Clutch pedal ECU (output signal: continuous signal indicating the clutch pedal activation from 0-100%).
A Motor Control ECU (input signal: motor speed signal, output signals: oil level and temperature).

Two Front wheel ECUs: FrontWheelRight and FrontWheelLeft (input signal: continuous break signal 0-100%, output signals: speed calculated from wheel rotation and the break system status).
Two Rear wheel ECUs: RearWheelRight and RearWheelLeft (input signal: continuous break signal 0-100%, output signal: break system status).
A CarControl ECU – controlling the speed of the motor and implementing the Cruise Control function.

The CruiseControlling function is activated with the CRUISE_ON button and must maintain the actual car speed both uphill and downhill until either the brake pedal or the clutch pedal is activated or the cruise controller is turned OFF by activating the CRUISE_OFF button. If either the brake pedal or the clutch pedal was activated and released, activating the RESUME button will accelerate the car to the previous cruise speed and maintain this. Activating either INC_SPEED or DEC_SPEED while cruising will adjust the cruise speed.

2.1. Decide which of these nodes need to be duplicated to obtain a fault tolerant car system and draw a UML deployment diagram showing your system design.

2.2. Decide which signals must be transmitted to which nodes including the input and output signals for the CarControl ECU.

2.3. Estimate the number of bits required for each signal and estimates the timing requirements for each signal.

2.4. Decide if you need event channels to transmit some of the signals and if yes, reserve slots for these in the following communication layout.

2.5. Design a communication cluster consisting of a number of TDM rounds, where each round has a sending slot for each node in the system. Design the information layout in each of these TDMA rounds. Assume the communication speed is 10 Mbit/s.

Product / delivery / deadline:

Answers to the assignments are to be documented and handed-in via campusnet (see deadline on campusnet).