

# The Smart Car - a distributed controlled autonomous robot

Gernot Klingler, Alexander Kößler and Wilfried Elmenreich (Faculty Mentor)

Institut für Technische Informatik

Technische Universität

Vienna, AUSTRIA

Email: e0125519@student.tuwien.ac.at

**Abstract** — This work concerns the design and implementation of the distributed controlled system of a four wheeled robot called "Smart Car". The Smart Car's task is to navigate through a static obstacle course by perceiving its immediate environment with the help of a combination of infrared and ultrasonic distance sensors.

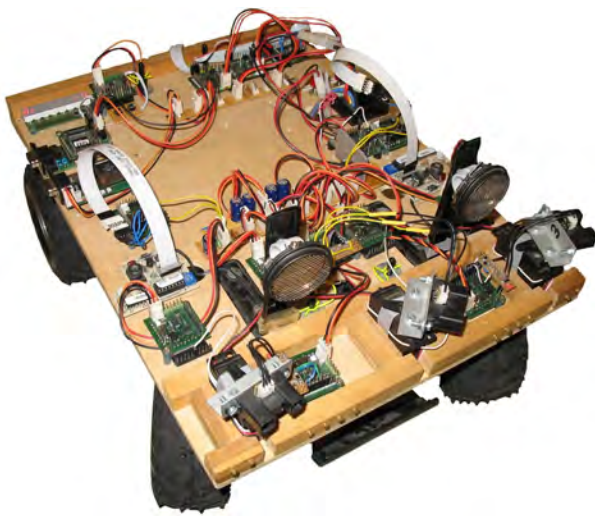


Figure 1: The Smart Car

## I. INTRODUCTION

The initial motivation for the design of the Smart Car, shown in Figure 1, was the need of a "real-live-demonstration object" for the TTP/A communication protocol. TTP/A is a Time Triggered Protocol for SAE Class A applications (TTP/A). This protocol is the latest member of the family of protocols for the *Time Triggered Architecture* (TTA) developed at the *Institut für Technische Informatik* at the Technische Universität in Vienna, Austria. TTP/A has a *Smart Transducer* interface standardized by the *Object Management Group* (OMG) [1] with the purpose of interconnecting transducer (sensor and actuator) nodes, providing hard real-time guarantees for communication. TTP/A uses the *Interface File System* (IFS) [1] which provides a unique addressing scheme for all relevant data in the *Smart Transducer* network, i.e. transducer data, configuration data, self-describing information, and internal state reports of a *Smart Transducer*. Standardized in this fashion, TTP/A represents an inter-

esting option for various sensor network applications like mobile robots. Existing projects on mobile robots typically use point to point wiring or event-triggered communication systems like CAN [2] or Ethernet [3]. The Smart Car also shows the utilization of *Sensor Fusion* [4] which is the combination of sensor data of a number of unreliable sensor measurements to a more reliable measure value. Furthermore robots always have and always will fascinate people. Especially for students this represents an extremely attractive and challenging research field.

## II. SYSTEM ARCHITECTURE OF THE ROBOT

The Smart Car's operation can be categorized into four main fields as shown in Figure 3 – software, electrical hardware, electro-mechanical hardware and mechanical hardware. The software layer consists of the application software and the TTP/A-protocol software. The application software exchanges data with the protocol software running on the same node and performs the tasks which are required for communicating in the distributed fieldbus network according to the rules specified in the protocol definition. The electrical hardware layer consists of a fieldbus network, complete with TTP/A nodes and the car's TTP/A communication bus. The smart car consists of 12 ATmega 128 nodes like the one shown in Figure 2. The *five sensor nodes* (three infrared-nodes and



Figure 2: ATmega128 Node

two ultrasonic nodes) collect the raw data from sensors, transform the data to a standardized digital representation, check and calibrate the signal, and transmit this digital signal via the TTP/A Protocol communication protocol. The *four servo nodes* control the servos on which the infrared sensors are mounted and the servo which is responsible for the steering direction. The *display node* shows debugging information. The *engine node* controls the speed of the vehicle. The *navigation node* receives

the sensor data (already calibrated by the sensor-nodes) and performs a navigation algorithm to find the best way through the obstacles and sends the calculated steering direction and speed values to the corresponding nodes.

All the nodes are arranged as a cluster, and exchange information on a serial bus with the TTP/A protocol.

The *electrical/electromechanical hardware layer* refers to the sensors, power supplies, servos, LED indicators and other components such as additional power supply busses. The *mechanical layer* consists of the main chassis of the Smart Car, which is an off-the-shelf four wheeled model car fitted with a wooden mounting board.

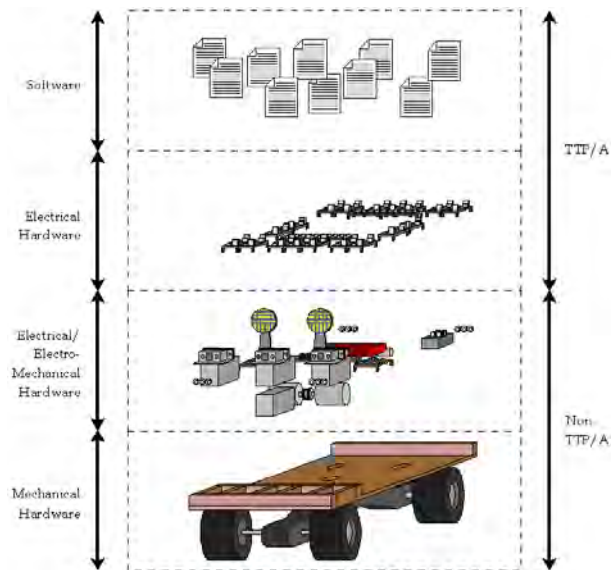


Figure 3: Hardware layers of the car

### III. SHORT DESCRIPTION OF THE NAVIGATION ALGORITHM

Before the navigation algorithm can start, the navigation-node collects the data from the sensor nodes and generates a grid with the *"robust certainty grid algorithm"* - a fault tolerant sensor fusion algorithm that was developed by Elmenreich, Schneider and Kirner [5]. The grid is seen by the car as a 17x10 array which represents a 170x100cm region around the car. The algorithm works with at least three sensors and can tolerate repeated sensor failures, with the ability to reintegrate recovered sensors. The algorithm merges the sensor data of the sensors and calculates "obstacle-likelihoods" for every element in the grid-array. When the grid-generation is completed, the grid serves as an input for the navigation algorithm. The navigation algorithm calculates - according to the grid - the direction with the least risk and sends its result to the node controlling the steering-direction and the speed of the car. If the algorithm cannot find a way, the vehicle stops.

In a future version of the navigation algorithm we plan to implement backtracking, which means that if the car

cannot find a further way, it navigates the same way back and tries to take another way through the obstacles.

### IV. RESULTS OF THE WORK WITH THE SMART CAR

The work on the Smart Car has been performed within the scope of a Bachelor's Thesis. While the mechanical hardware was available from previous projects, the work included a redesign of the transducer network, the implementation of the Smart Transducers and the setup of the *Sensor Fusion* and navigation subsystem.

The Smart Car acts as a model system for the research on designing, configuring and debugging of TTP/A networks. The car is also used for sensor data generation for the research on sensor fusion. The limited hardware resources in embedded systems sometimes led to massive problems, especially the complex dependable navigation algorithm is a very resource-hungry application. While coping with problems and pitfalls of programming embedded systems, we also learned that debugging a distributed embedded system is a challenging job.

As such, the Smart Car acts as a *Drosophila Melanogaster* for intelligent distributed embedded systems.

### ACKNOWLEDGEMENTS

The authors want to thank Angela Schöngendorfer for her constructive comments on this paper. This project was supported by the Austrian FWF project TTCAR under contract No. P1806-N04.

### REFERENCES

- [1] Object Management Group OMG. Smart transducers interface v1.0. 2003. Available at <http://doc.omg.org/formal/2003-01-01>.
- [2] G. Novak and M. Seyr. Simple path planning algorithm for two-wheeled differentially driven (2WDD) soccer robots. In *Proceedings of the Second Workshop on Intelligent Solutions for Embedded Systems (WISES'04)*, pages 91–102, Graz, Austria, June 2004.
- [3] Y.-J. Cho, J.-M. Park, J. Park, S.-R. Oh, and C. W. Lee. A control architecture to achieve manipulation task goals for a humanoid robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-98)*, pages 206–212, Piscataway, USA, May 1998.
- [4] W. Elmenreich. *Sensor Fusion in Time-Triggered Systems*. PhD thesis, Technische Universität Wien, Institut für Technische Informatik, Vienna, Austria, 2002.
- [5] W. Elmenreich, L. Schneider, and R. Kirner. A robust certainty grid algorithm for robotic vision. In *Proceedings of the 6th IEEE International Conference on Intelligent Engineering Systems (INES)*, May 2001. Available at <http://www.vmars.tuwien.ac.at>.