

Due to the fact that the mobile robotics market is still quite small when compared with general industrial automation technology, the choice of dedicated hardware and software products is relatively limited. Gernot Gebhard and Matthias Bengel from the Fraunhofer Institute for Manufacturing Engineering and Automation (IPA) provide an overview of solutions based on the application of established standard solutions such as PC Control from Beckhoff and EtherCAT as the communication system, which offer significant potential in terms of savings, performance, robustness, and more for robotics applications.

EtherCAT helps enhance mobile robots

The mobile Secur-O-bot safety robot for monitoring buildings and industrial facilities was developed by Fraunhofer IPA and is used as an integration platform. The robot, shown in figure 1, is a three-wheeled platform with a differential drive. The Secur-O-bot control software is based on the navigation system developed by Fraunhofer IPA, which has been tried and tested in a wide range of applications and environments, in some cases involving several years of continuous operation. The navigation software is supplied high-precision data via a laser scanner in order to enable autonomous robot movements. In addition to these sensors, the robot has further sensors for monitoring.

In the past, monitoring robots were only able to monitor the environment that was 'visible' from the current robot position. The new robot positions sensor probes with the aid of an additional module, in order to enable extensive monitoring of systems and buildings without fixed sensors. The automatic positioning and collection of small, autonomous sensor probes by the mobile robot unit significantly enhances the monitoring capability of each individual Secur-O-bot, while avoiding the high costs of extensive fixed installations.

Room for robot improvement

The previous mobile robots developed by Fraunhofer IPA were application-driven, i.e. the equipment to be connected was purely selected according to application aspects and not based on existing interfaces. The robot control system featured a hardware abstraction layer, since different robots incorporated a range of equipment such as drive controllers and motors. The required module was selected and configured depending on the interface and component used. The robot control system contained different driver classes, which were accessed via an integrated interface.

As a consequence, the system had to cope with significant hardware interface variety, with the result that new driver classes had to be implemented on an ongoing basis. The library of available driver classes therefore keeps growing, with a resulting increase in software maintenance effort.



Figure 1: The Secur-O-bot monitoring robot in the PC Control Application Park booth at Hanover Fair 2008

Integrated control architecture through the use of Industrial PCs, TwinCAT and EtherCAT

In order to eliminate these deficiencies, the mobile robot described above was modified, resulting in a prototype with unified software and hardware interfaces. The on-board computer of the robot was replaced with a Beckhoff C6920 control cabinet PC installed with Windows XP Embedded and TwinCAT I/O software. TwinCAT automation software permits consistent access to the connected hardware with the aid of an abstraction layer. The connected I/O channels relate to each other via a real-time driver. TwinCAT I/O enables cyclic data exchange of process variables at the task level.

The hardware layer of the computer architecture is shown in figure 2. From this perspective, the controller communicates with the corresponding EtherCAT Terminals via EtherCAT so only one fieldbus is required. The I/O station essentially consists of digital input/output ports and fieldbus terminals for other bus systems,

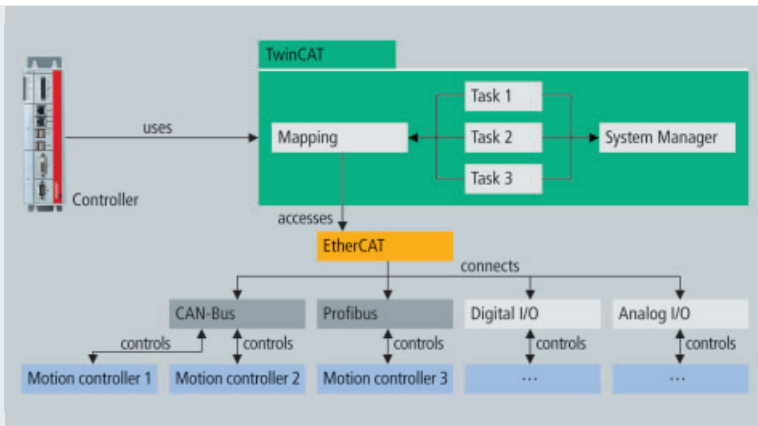


Figure 2: Hardware layer with application of TwinCAT

such as CAN bus. Other interfaces such as PROFIBUS, serial interfaces or analog inputs/outputs are also possible. The Bus Coupler encapsulates the data from other interface protocols in the EtherCAT protocol, creating an integrated view of the data. The hardware interfaces are configured via the TwinCAT System Manager. The robot controller accesses the configured process data via mapping. Controller integration simply requires the use of an automatically created header file. If the hardware is changed without a change in the type of process variables, the process is fully transparent for the controller, since the configuration steps run within TwinCAT. The controller must only be adapted if the header file is modified. Generally, this is only required if additional hardware must be used that was not available when the control software was developed. Additional implementation of hardware drivers in the controller is no longer required as a result.

The extension module for positioning sensor probes is carried "piggyback" by the mobile platform. Before the hardware conversion, the extension module was controlled by a microcontroller. The I/O terminals, which are connected to the Industrial PC (IPC) via EtherCAT, are now activated directly, so that the relatively expensive connection to the microcontroller is no longer required. The EtherCAT protocol is available right down to the field level. Handling of the I/Os via the EtherCAT Coupler frees up the PCI slot in the IPC so it can be used for other purposes.

Considerations for programming and configuration

This approach is based on TwinCAT I/O for accessing the bus system variables from the robot controller instead of using a driver for EtherCAT directly. To this end, an additional task is created in TwinCAT, which reads and writes variables cyclically. These variables map the communication with the robot controller. If the robot controller writes to these variables, they are available in TwinCAT in the next cycle and are written to the accordingly allocated bus variable (see figure 3). Read access works accordingly: TwinCAT cyclically fills the task variables, so that the robot controller can read them during the next cycle.

The highlight from the perspective of the robot controller is the utilization of these additional task variables. One aspect of the one-time configuration effort is the creation of the task, which mainly involves specifying the cycle time and priority. The other aspect is the definition of variables within this task. For each variable,

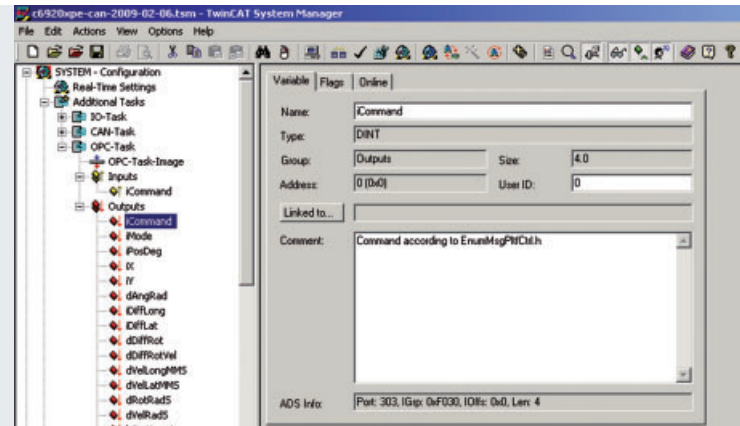


Figure 3: Configuration of the process variables in TwinCAT

the user has to specify whether it is an output or input variable and which data type it should correspond to. Within the System Manager these variables with the process variables of the devices are linked with the fieldbuses. To link the values with the robot controller, a header file written in the programming language C++ is exported from this task and is integrated in the robot controller. This header file and an additional library, which also has to be linked, are used to access the process variables.

The advantage is that the connected hardware can be allocated to the variables of the additional task via the graphical System Manager. Any hardware modification merely requires this configuration step to be repeated, without the robot controller itself being affected.

If devices such as a motor controller are connected to a fieldbus (e.g. the CAN bus as in this case), the bus hardware layer is encapsulated via EtherCAT. From a robot controller perspective only the EtherCAT device is visible. The device profiles make the controllers and the bus system downstream of EtherCAT exchangeable, without the need for significant software adaptations.

Outlook

The drive controllers for the robot's drive motors are currently connected to the higher-level software via CAN bus. The drives' control algorithms are closed via the bus in the IPC, with correspondingly stringent requirements for the bus. It should be noted that the transfer fully utilizes the bandwidth of the CAN bus right up to its limits. Particularly for complicated kinematics and associated complex control algorithms it is clear that the CAN bus is the limiting element with regard to the cycle time. In order to remedy this situation, EtherCAT will be used as the drive bus in the future.

Fraunhofer Institute IPA (Institute for Manufacturing Engineering and Automation)

www.ipa.fraunhofer.de