AS Interface
The Fieldbus for Low Level Automation

Peter Janssen
Department SI-R3, Festo AG & CO.
P.O.Box
D-73726 Esslingen
e-mail: jas@festo.com

Abstract
Consistent communications have always been striven for throughout the entire scope of any given automation solution. However, this has only been possible to a limited extent. A possibility which allows for communication with the lowest level, i.e. the sensor and actuator level, has usually been lacking.

The AS interface, recognisable by its characteristic yellow cable, places an "open bus system" for the interconnection of simple, binary sensors and actuators at the disposal of the user which allows for communications all the way down to the sensor and actuator level. The AS interface has firmly established itself within the market with further advantages such as real-time capabilities and reliable data transmission, as well as easy installation thanks to modular design and use of the penetration technique.

Through the use of an applications example, the following paper demonstrates the demands placed on the AS interface by customers, as well as data security measures.

KEYWORDS
AS interface, Low Level Automation, Fieldbus

Introduction
The use of communications – i.e. field bus systems – has brought about fundamental change in the area of automation technology. Whereas central control systems used to be the standard against which all other alternatives were judged, decentralised and decentralised intelligent sub-systems represent the current state-of-the-art in the field of automation technology. This applies to the lowest level of the
automation system as well, namely the sensor and actuator level.

According to a survey conducted by the VDMA, more than 80% of the sensors, actuators and control devices used in machinery and systems manufacturing are equipped with simple, and as a rule binary switching functions. This would indicate that a large percentage of the total time required for machine setup (approx. 80% to 90%) was spent on electrical start-up. This situation lead to the tying up of large amounts of capital. Errors or defects which were first detected at this point in time resulted in high costs and were expensive to correct. Delays and image loss ensued, not to mention possible claims for damage.

Industrial manufacturers developed a plan of action to counter these negative aspects, and the foundation for the development of the "fieldbus" was laid.

Today, approximately 100 different fieldbuses are available for a wide variety of applications. However, the most recent market surveys demonstrate a distinctive trend towards "open" bus systems such as Profibus, DeviceNet, Interbus, AS interface etc.

The AS interface was designed to simplify precisely this low level range in the field of automation right from the beginning. Typical applications include:

- Individual parts manufacturing
- Machinery manufacturing
- Conveyor technology
- Food products industry
- Automotive industry
- Assembly and handling technology
- Robot controls

The concept of the AS interface has always been that of an "open" system which compliments commonly used fieldbuses, and which is used as a "replacement for the cable harness". Its claim to fame has never involved the linking up of complex devices like controls or sub-systems, but rather has to do with:

- Simple handling, installation and assembly
- Real-time capabilities
- High level reliability
- Flexibility
- Interoperability
- Cost effective solutions

All of this taken together resulted in the following basic requirements for the AS interface:

- The AS interface is a single-master system with cyclical polling, i.e. one master per network.
- AS interface telegrams have a fixed length and are structured in a simple fashion. Four bits of useable data are always exchanged between the master and the slave.
- Cycle time for a fully expanded system (62 slaves for version 2.1) is approximately 10 ms. Querying is carried out in a deterministic fashion. After 10 ms, complete and current sensor status data are available, or an actuator update has been completed.
- The AS interface cable is a non-shielded and non-twisted 12 conductor cable which transmits data and electrical power. The use of an appropriately developed ribbon cable allows for the utilisation of new connector technologies (piercing).
- Data security is assured by means of suitable control mechanisms, even under industrial ambient conditions.

These basic requirements will be demonstrated in the following pages using a specific application as an example. A closer look will be taken at data security, as well as diagnosis and troubleshooting. Reference is made to FESTO systems where diagnosis and troubleshooting are concerned. The possibilities offered here may differ from other systems.

Integration into a Control System
The AS interface is one of the most flexible and varied interconnection methods available to the planner. The network can make use of any desired structures and components - there are over 600 available from various suppliers.

The integration of an AS interface network into a control system can be accomplished in two different ways:

- The AS interface is directly connected to the uppermost control level, i.e. it becomes an essential part of the PLC or the IPC. This allows for the creation of networks with large numbers of slaves or redundancies, or which are distributed over expansive physical areas through the use of multiple masters.
- However, an AS interface can also be connected to a higher level fieldbus, as is the case in our applications example, which then becomes, in its entirety, a slave within the fieldbus network and thus functions as a sub-system.

The first possibility supports the rapid processing of sub-tasks. The tasks at hand are broken down on-site into small, simple sub-tasks and are forwarded as
required. This type of solution offers the following advantages:

- Short programs and simple start-up, even for portions of the system
- Reduced downtime, because if individual stations fail, the system continues to function in a limited fashion
- Simple, easy assembly
- Integration of the control cabinet into the machine
- Simple expansion

A cost effective replacement for the cable harness is the main argument in this case.

Emphasis is placed upon other targeted requirements for the second alternative, such as:

- Interconnection of many AS interface networks at a single control system
- The bridging of great distances
- Decentralisation for large systems
- Ability to transfer data amongst several hierarchical levels, or amongst otherwise independent control systems

This alternative focuses on the communications aspect.

**Application - Production of Barbie Doll Legs**

**MATTEL Indonesia**

**System Description**

A Mattel production monitoring system is being created to accommodate a requirement for immediate troubleshooting within the plant. The first application program is being used for the injection moulding department in building 1. Basically, building 1 consists of 12 production lines (A through L). Each line includes approximately 10 machines. The system includes 1 controller and 1 display monitor for every 2 lines. The line supervisor will be able to monitor machine status in real-time. Status is indicated as the percentage of production output which has been scrapped, efficiency and downtime.

The system receives an input signal from the machine each time a full cycle is completed. The system receives signals from the control panel regarding the quality of the product, which are keyed in by the operator. All of the data are processed and calculated by the system. The processed data is displayed at the monitors and transferred to the Festo server located in the MIS department. Data is saved to the Festo server every 30 minutes. This allows for tracking the history of each of the lines, as well as the individual machines within the lines.

The existing Mattel server accesses Festo's server in order to forward data from each machine to its clients. All of the clients in the Mattel server are capable of displaying real-time data from the injection moulding machines.

**Why the Customer Ordered this System**

Ever since Mattel started collecting statistical data, the average scrap level for Barbie doll legs has been approximately 5%. The statistics are analysed on a day to day basis, and Mattel had been unable to reduce the scrap level because the range of moulding errors varied on a daily basis. We met with Mattel to discuss the problem and they expressed the need for a system which is capable of providing real-time data to the production line supervisor, who is responsible for 24 machines, in order to provide him with an overview which indicates how much scrap is being produced (the operator then keys in the scrap at the control panel and the percentage of scrap is calculated in comparison with overall production for the entire shift, and for the last 15 minutes). With this system, the line supervisor is able to go to any machine which is producing scrap and find out why (problems may be related to the machine, the material or the operator). Mattel is thus provided with an early warning system for the reduction of scrap.

**Why We Used the AS Interface**

We used AS interface because we needed a bus system which would provide a small number of inputs at a great distance from the master. Each machine uses 4 inputs (perfect for the AS interface), and the last machine is 200 meters away from the first. This meant we would have had to pull a large quantity of cables to the master, and we determined that the AS interface was less expensive based on cable costs alone. An additional argument is favour of the AS interface is its simple installation, its flexibility and its reliability, which has been proven in many applications.

**Customer Benefits**

After the system was installed, the scrap rate fell directly to 2% during the first week, and has now settled in at a constant 0.5%, which is lower than any other Barbie plant. The benefits are improved efficiency and less rework, both of which result in cost reductions.

Furthermore, graphic representations of downtime and efficiency were added, and, according to the machine standards we had previously defined, we were able to surpass the previous standards by reducing scrap and increasing output so that all machines were running at + efficiency. This allowed for improved planning, and it was also possible to shut down ten machines which reduced energy costs and provided Mattel with spare capacity in case of breakdowns.
Hardware Design

The system is being designed based upon a PC system with master and slaves. Utilised hardware includes the industrial PC (IPC) range from Festo along with actuator-sensor interface technology (AS), which has simplified wiring of the system, and the Festo Field PC Net for data communication between master and slaves.

With the AS interface system, the number of I/Os can be easily increased by adding up to 31 AS interface nodes to the same line. Each node has 4 input signals. The Festo Field PC Net is an Ethernet based, data communications system which allows for the transfer of data from the master to the slaves, or vice versa. Slaves are located in the field, and the master is located in close proximity the Festo server in the MIS department.

Each slave has 2 controllers. One controller is used to process data, and the other for the display of variables. Each machine in each production line has 4 input signals. One signal is taken from the machine using a relay. Each time the machine completes a cycle, a signal is sent to the IPC. Three signals are used for the scrap control panel with 7 segment display. The control panel has 2 keys for entering scrap quantities, and 1 key for the acknowledgement of data entry. The operator can enter the number of scrapped parts by pressing the appropriate scroll key, and acknowledging transmission of the data to the IPC.

Software Design

The system software is made up of three components:

- The software at the IPC
- The software at the Festo server
- The software at the remote Mattel server

The Software at the IPC

The Festo IPC runs with Festo Software Tools (FST). FST is a special software program for the Festo PLC system. The FST software handles all signal processing within the AS interface system and the calculation of data, and enables communication between the master and the slaves. The display software for the slaves has been especially created to access the variables in the IPC. The IPC master is linked to the Festo server by means of a special driver created in C language. All communications between the master and the slaves are stored to the communications file in binary format. The IPC master is a gateway between the Field PC Net and the Festo server.

Software at the Festo Server

An NT based operating system has been installed to the Festo server. The files from the slaves are accessed and managed by a special program which runs in a visual basic application under Windows 95. The data base system is managed by Microsoft Access.

Software at the Mattel Server

The Mattel server runs the software required for displaying real-time data from the slaves in the field. The software reads all data in the tables which are provided by the Festo server. Visual basic provides all of the data and calculations in the form of inputs for Mattel’s server software. The software was created in visual C language under Windows 95.

How the System Works

Two types of data are collected by the monitoring process during each shift. These include "shift to date data" and "current 15 minute data".

Shift to date data indicate machine activity from the shift start time until present time. Current 15 minute data indicate machine activity for the last 15 minutes. Based upon these data, the supervisor can take direct action if any of the machines have had problems within the last 15 minutes.

The system monitors scrap levels, efficiency, and downtime for each of the machines in the field. In addition to these three parameters, machine utilisation is monitored in the office as well.

In order to calculate these parameters, the system requires certain input data.

The input data required by the system are taken from the bill of materials which is already being used by Mattel in building 1. The original data are created in Microsoft Excel format. These data are converted to Microsoft Access format and are used by visual basic in the Festo server.

Some data are sent to the IPC slaves via the IPC Master such as standard cycle time and number of cavities in the mould. Some data are used by the office monitoring
program such as MMPN (mixed material part no.), mould number and series, part name and colour etc. These data have to be related to the actual production lines. Whenever there is a mould change for any of the machines, the operator has to input the mould type and series number. Based upon the mould data, the system automatically reads in the appropriate data from the bill of materials, and then transfers them to IPC slaves via the IPC master.

Data communication is carried out by the IPC slaves and the IPC master via the Field PC Net. The slaves provide the raw "shift to date" data, and calculated "current 15 minute" data. Raw data include the number of scrapped parts, total cycle, target, downtime duration and downtime frequency.

Visual basic at the Festo server calculates the scrap rate as a percentage, efficiency, downtime and machine utilisation based on these data.

All of the shift to date data are calculated and logged at the Festo server for each 30 minute time period.

**Scrap Calculation**
The operator keys in the number of scrapped parts at the control panel for the evaluation of product quality.

There are 3 keys at the control panel, as well as a two-place 7 segment display. Two of the keys are used for entering the number of scrapped parts, one for tens and one for single units, and the third button acknowledges and transmits the entered quantity. The operator must enter the number manually with the keys, and the entered data appears at the 7 segment display. The data entry keys scroll through the numbers 1 through 0, so that a new quantity can be easily selected before activating the acknowledge / send key if an error has been made.

The following formula is used for calculating scrap:

\[
\% \text{ Scrap} = \left( \frac{\text{Total Scrap}}{\text{Total Output}} \right) \times 100\%
\]

Total Output = Number of shots x number of cavities

**Efficiency Calculation.**

Efficiency is a parameter which indicates how fast the machine runs when it runs. With the efficiency parameter we compare actual cycle time to standard cycle time when the machine runs.

The formula for efficiency is as follows:

\[
\% \text{Efficiency} = \left( \frac{\text{TotalCycle}}{\text{TotalTarget}} \right) \times 100\% - 100\%
\]

Total target is an internal variable which is increased depending upon cycle time. Total target is calculated from the beginning of the shift through current time. When the machine is stopped, this variable is no longer increased.

**Downtime Calculation**

Downtime indicates how long the machine has been stopped.

The machine is considered stopped if the system has not received a cycle signal for more than 2 times the standard cycle time.

The machine is assumed to be running until the above condition has been fulfilled.

The following formula is used for calculating downtime:

\[
\% \text{Downtime} = \left( \frac{\text{Downtime Duration [in seconds]}}{\text{(Cur. SS) [in seconds]}} \right) \times 100\%
\]

Cur = Current Time (in seconds)

SS = Shift Start Time (in seconds)

When the machine is stopped, downtime is increased relative to actual time.

**Utilisation Calculation**

Utilisation is used to monitor how well the machine is exploited, and is based upon scrap rate, efficiency and downtime.

The formula for calculating utilisation is:

\[
\% \text{Utilisation} = \left( \frac{100\% - \% \text{Scrap}}{100\%} \right) \times \left( \frac{100\% + \% \text{Efficiency}}{100\%} \right) \times \left( \frac{100\% - \% \text{Downtime}}{100\%} \right) \times 100\%
\]

The Screen Display.

The monitoring system displays performance of the individual machines as follows:
Each machine displays the scrap rate as a percentage, efficiency, downtime and utilisation. The actual status of the machine is displayed as a run/stop status. If the machine is stopped, a red indicator is displayed. The designation of the part which is currently being produced by the machine is displayed as well. The part designation is read by the system from the bill of materials. If the no utilisation of the machine has been planned for, the MMPN in tools change can be set to 000. The machine designation "Not Planned" is then displayed, and the parameters display is disabled.

If MMPN has not been set to 000, and if the machine is stopped, the last scrap rate and efficiency values are displayed. The percentage of downtime is increased according to actual time.

All machine activities are measured and calculated in line performance variables. Line performance picture is the average performance of all machines in any given line.

As is apparent from the application description, the following requirements were of utmost importance:

- A system was required which could be easily implemented as a retrofit.
- A flexible system was required which would allow for expansion at any time.
- A secure (data transmission) and reliable (high rate of utilisation) system was required.

**Reliability, Diagnosis and Troubleshooting**

Reliability in this context means that a system will deliver reliable data and diagnostic values continuously and on-time under all specified conditions, especially where severe electromagnetic noise prevails.

In order to define reliability we must take a closer look at the AS interface, especially as regards data security measures, and we must differentiate between the following situations:

A. Reliability of the AS interface during data transmission as regards corrupted telegrams, as well as the recognition and defined management of system or sub-system failures

B. Error conditions which result from system operation

**Reliability**

Despite its simplicity, the AS interface is a bus system and is subject to the same data transmission requirements as is the case with bus systems at superordinate levels.

A structural weakness where reliability is concerned is the use of a common bus line for data transmission. If the serial transmission system should fail, several, or even all of the bus users are affected, and not just an isolated sensor or actuator with a defective cable. Special attention must be allotted to data transmission reliability in order to counter this problem when systems of this type are used. A certain amount of tolerance as regards interference which may lead to corrupted telegrams, as well as the recognition and defined management of system and sub-system failures fall into this category.

The following elements were implemented in order to achieve the desired level of reliability:

- An integrated circuit was developed whose reliability is many times greater than that of a discrete circuit.
- The AS interface has a highly symmetrical layout, which minimises the effects of electromagnetic interference at the physical level.
- Signal alterations which may lead to corrupted telegrams are effectively recognised by means of code monitoring.
- The master continuously monitors for correct functioning of the network.
- Automatically stored configurations for peripheral devices are monitored by the master as well.

Peripheral devices can be expanded with self-monitoring units which enable them to report their status to the master.

Because we are unable to address all of the implemented measures in this paper, I would like to consider only two functions, namely the modulation process and data security.

**The Modulation Process**

In our search for a suitable modulation process it was not possible to make use of already known processes, because these did not fulfil all of the requirements, such as message signals which are free of direct current. A new modulation process was developed: alternating pulse modulation (APM). This process allows for serial transmission within the baseband.

The transmission bit sequence used for the APM modulation process is first re-coded into a sequence for which the phase is reversed each time the transmission signal is changed. A transmitting current is then generated from this signal, which in turn generates the desired signal level at the AS interface cable by means of differentiation in combination with an inductivity which occurs within the system only once.
Every increase in transmission current thus results in a negative voltage pulse, and every drop in a positive voltage pulse.

The voltage signals are detected within the cable at the receiver side, and are reconstructed into the originally transmitted bit sequence. The receiver synchronises itself to the first detected negative pulse, which is interpreted as the start bit of a message.

**Data Security**
Reliable error recognition in the "receive" condition is of great significance for error-free communications via the unshielded AS interface cable.
Due to the fact that data security with the AS interface is based upon a principle which is not common to most known bus systems (it demonstrates a number of regularities and redundancies which have been implemented in a targeted fashion), the following errors are already recognised during bit transmission:

Every query from the master and every response from the slaves is subjected to these tests.

**What does this mean in the event of a faulty transmission?**
If one of the above listed errors is recognised, the query is rejected as faulty and no response is generated at the slave.
If the query from the master is received in an error-free fashion, the slave responds with a positive acknowledgement to the master.
If the master receives a faulty response, or no response at all, within a given period of time, the query is simply repeated. If the second response is also faulty, communication with the slave is discontinued and a corresponding flag is set.
An error at the slave, e.g. the continuous transmission of a slave response, would paralyse data transfer. In order to avoid this possibility, the slave is equipped with a monitoring circuit which functions independent of the master, and which recognises continuous transmission and switches the slave off if this error should occur.
Other causes such as undervoltages (< 14V) trigger a reset, or interrupt operation.

**Error Conditions Resulting from Operation of the System**
If the network has been set up properly, transmission is reliable, even in industrial environments with high levels of interference.

"Nothing is perfect."
This also applies to automation systems with AS interface. However, the AS interface offers users advantages which are not available with conventional technology:
- The master provides a quick and simple diagnosis for the entire system.
- Errors at individual slaves do not necessarily paralyse the entire system, depending upon the requirement.
- Only errors which involve the entire network result in a breakdown of communications from the master.
Slaves can be replaced without shutting down the entire system (live insertion).
A restart is only required if the entire network is defective.

Start-Up Phases
The master runs through the following phases after supply power has been switched on:
- Off-line phase
- Warm-up period
- Normal operation

Initialisation takes place during the off-line phase, and the master's default conditions are established. Preset parameters are copied to the parameter fields, so that subsequent activation with the pre-selected parameters can be executed.

The warm-up period consists of a recognition phase and an activation phase.
A search for existing slaves is conducted during the recognition phase by reading I/O configurations and ID codes. Slaves which respond to both queries are added to the list of recognised slaves. Previously recognised slaves are activated during the activation phase by writing parameter outputs. The input image is filled with valid values by means of initial data exchange. The target configuration and the actual configuration are checked for agreement at the end of the activation phase, and the "config OK" flag is set if appropriate.

Actual data exchange with the interconnected sensors and actuators takes place during normal operation. One cycle consists of data exchange, a management and a recording phase.

If the system does not enter into normal operation, the cause of the error can be determined by means of various diagnostic possibilities.

Error Conditions and Diagnosis
From the standpoint of the user, differentiation can be made between two types of error:
- Operating errors
- Design and maintenance errors

Operating errors are errors which occur during normal operation of the system. These may be caused by device failure, mechanical destruction, extreme interference emission or other ambient influences.

Setup and maintenance errors are caused by improper or incorrect handling of the system.

The most common errors are incorrectly poled slaves, address errors, and layout and assembly errors. These errors may occur during setup of the system, as well as during maintenance.

The following checklist can be used to avoid such errors:

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are the addresses of all slaves set to 1 ... 31?</td>
<td></td>
</tr>
<tr>
<td>Have slave addresses been clearly assigned (no double addressing)?</td>
<td></td>
</tr>
<tr>
<td>Has an EMERGENCY STOP been provided where required for all relevant slaves in accordance with prevailing specifications?</td>
<td></td>
</tr>
<tr>
<td>Has a separate (auxiliary) power supply been connected to all slaves with higher current consumption?</td>
<td></td>
</tr>
<tr>
<td>Is it possible to switch on the power supply in parallel fashion, or in the correct sequence?</td>
<td></td>
</tr>
<tr>
<td>Have AS interface specifications been complied with?</td>
<td></td>
</tr>
<tr>
<td>- Max. current consumption</td>
<td></td>
</tr>
<tr>
<td>- Max. cable length of 100 metres without repeater</td>
<td></td>
</tr>
<tr>
<td>Has the AS interface bus been configured?</td>
<td></td>
</tr>
<tr>
<td>- Has a nominal list been compiled with AS interface slave addresses, ID codes and I/O codes?</td>
<td></td>
</tr>
</tbody>
</table>

Diagnosis
The status flags are the most important diagnostic tool (e.g. "config OK"), and they usually indicate system status via the LEDs at the master. Diagnostic software available from the manufacturer of the controls or the master offers additional possibilities, by means of which system diagnosis is performed through the use of functional building blocks. Service and diagnosis devices are also available as external tools for the monitoring of telegrams. However, device characteristics are manufacturer specific.

The following online diagnostic possibilities are provided in combination with Festo's FST software:

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>?DLES</td>
<td>Display list of current slaves</td>
</tr>
<tr>
<td>?DSxx</td>
<td>Display slave (I/O, ID+Inputs) xx = 1..1F</td>
</tr>
<tr>
<td>?DOUF</td>
<td>Display control flags</td>
</tr>
<tr>
<td>?DHARD</td>
<td>Display hardware/software reaction (0 = soft / 1 = hard)</td>
</tr>
<tr>
<td>?MMA</td>
<td>Modify master address (= 1..4)</td>
</tr>
<tr>
<td></td>
<td>(valid for all other commands)</td>
</tr>
<tr>
<td>?MSA</td>
<td>Modify slave address</td>
</tr>
</tbody>
</table>

472
Additional possibilities are offered as functional building blocks which, when used within the control program, are capable of inducing appropriate performance or triggering reactions when errors are recognised.

<table>
<thead>
<tr>
<th>Functional Building Block Designation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASI Mode</td>
<td>Sets AS interface response to configuration errors.</td>
</tr>
<tr>
<td>ASI STAT</td>
<td>Queries flags at the sequential control level.</td>
</tr>
<tr>
<td>ASI PARA</td>
<td>Transmission of a parameter to an AS interface slave.</td>
</tr>
<tr>
<td>ASI RES</td>
<td>Restarts cyclical update.</td>
</tr>
</tbody>
</table>

The status querying building block was integrated into the control program in our application. Changes to the network, as well as problems with slaves, can thus be recognised at any time.

Conclusions

As is apparent from the application, the AS interface offers wide ranging possibilities for use due to its flexibility. This has been proven by a great number of varying applications.

The AS interface is being influenced by a series of technical advances which will open up new applications possibilities. For example, version 2.1 of the AS interface has been expanded to include the following features:

- Number of slaves increased to 62
- Transmission of analogue values
- Distinction between peripheral and communications errors

Additional options are currently being considered such as a "safety bus". Slave ICs with new features like "EEPROM on board", display ports etc. will have a positive effect on product development.

Let's wait and see what the future holds in store for us.

Bibliography


