High-precision Gap Measurement with an Advanced Light Barrier via ET 200eco PN IO Link Master

SIMATIC

Application Description • October 2013
Caution:
The functions and solutions described in this article confine themselves to the realization of the automation task predominantly. Please take into account furthermore that corresponding protective measures have to be taken up in the context of Industrial Security when connecting your equipment to other parts of the plant, the enterprise network or the Internet. Further information can be found under the Entry ID 50203404.

SIEMENS

SIMATIC
High-precision Gap Measurement

Task
Solution
Basics About the SICK Sensor and IO Link
Function Mechanisms of this Application
Installation
Starting the Application
Operating the Application
Related Literature
History
Warranty and Liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use sound practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document. Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of fundamental contractual obligations are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.
Preface

Objective of this application

An innovative development of the sensors and the use of the open standard IO link ensure consistency from the sensor level to the control level, which is much easier to achieve as compared to former times and offers new possibilities. The application demonstrates these possibilities with a typical example from production technology and the resulting advantages for the user.

Main topics of this application

For a high-precision gap measurement between two neighboring products on a moving conveyor belt, the gap is determined and evaluated. This is done by means of innovative features in the sensor (Advanced functions), which are explained in this Application Example.

The determination of the gap is based on a time measurement the sensor carries out independently. The evaluation and decision whether the gap is sufficient is done by a SIMATIC controller which compares the actual value with the target value.

What is decisive for the use of the data consistency from the sensor level to the control level is the open standard IO link. Therefore, this Application Examples basically deals

- with a sensor, which enables a high-precision gap measurement in the first place with the Advanced functions (such as the internal time measurement).
- a powerful controller, which can read but also write configuration data into the sensor.
- with the function principle of IO Link.

The focus is on the interaction of sensor and controller via IO Link. In order to also address readers who are not completely familiar with the above aspects, the latter will be briefly discussed in a separate chapter ("Basics").
Table of Contents

Warranty and Liability................................................................. 4
Preface......................................................................................... 5
1 Task.......................................................................................... 7
  1.1 Overview................................................................................ 7
  1.2 Requirements......................................................................... 8
2 Solution...................................................................................... 9
  2.1 Solution overview ............................................................... 9
  2.2 Description of the HMI........................................................ 10
  2.3 Description of the core functionality................................... 12
  2.3.1 Time duration of the gap measurement............................ 13
  2.3.2 Reading and writing E part lists and sensor configuration data 15
  2.4 Hardware and software components used.......................... 17
3 Basics about the SICK Sensor and IO Link............................ 18
  3.1 Basics about the SICK sensor used...................................... 18
    3.1.1 Standard functions....................................................... 18
    3.1.2 Advanced functions..................................................... 19
  3.2 Basics about IO Link............................................................ 21
4 Function Mechanisms of this Application.............................. 24
  4.1 Program coordination by the FB MAIN (FB3)......................... 25
  4.2 Initialization by the FB INI (FB4).......................................... 25
  4.3 Detection of the gap by the FB CALCULATE (FB2)................. 26
  4.4 Environment of the SICK-FB.............................................. 28
5 Installation.................................................................................. 29
6 Starting the Application............................................................ 30
  6.1 Edit Ethernet nodes.............................................................. 30
  6.2 Loading the hardware configuration of STEP 7...................... 31
  6.3 IO Link configuration........................................................... 31
  6.4 Loading the S7 project into the SIMATIC CPU....................... 33
  6.5 Loading the WinCC flexible project into the HMI.................... 33
7 Operating the Application.......................................................... 34
  7.1 Requirements........................................................................ 34
  7.2 Initialization.......................................................................... 34
  7.3 Carry out measurement......................................................... 35
  7.4 Minimum gap not met as a fault scenario............................. 36
  7.4 Reading configuration data from the SICK sensor.................. 36
  7.5 Writing configuration data into the server.............................. 36
8 Related Literature...................................................................... 37
  8.1 Bibliography......................................................................... 37
  8.2 Internet link specifications.................................................. 37
9 History...................................................................................... 37
1 Task

1.1 Overview

Introduction

In the following, a concrete automation task is described. The principle of the solution demonstrated later is the same for a great number of similar tasks. Even if you have different requirements, you will notice that this Application Example can be adapted to your needs by simple modifications.

Overview of the automation task

The figure below provides an overview of the automation task.

![Figure 1-1](image)

Description of the automation problem

The conveyor belt transports products (single items). The gap between neighboring products must not fall below a defined value. One of the reasons for such a requirement might be that the claw of the robot enters into such a gap between two products to fix a handle to the product.
### 1.2 Requirements

#### Requirements of the automation task

In this chapter the specified requirements are defined concretely.

#### Table 1-1

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>During the measurement the velocity of the conveyor belt is considered constant.</td>
<td>The conventional solution</td>
</tr>
<tr>
<td>High-precision gap measurement</td>
<td>• detection of a product through a sensor</td>
</tr>
<tr>
<td></td>
<td>• transmission of the detection signal to the controller</td>
</tr>
<tr>
<td></td>
<td>• measuring the time with a timer</td>
</tr>
<tr>
<td></td>
<td>• measuring the gap between two products</td>
</tr>
<tr>
<td></td>
<td>is not practical for a high-precision gap measurement.</td>
</tr>
<tr>
<td>E parts list and configuration data of the sensor to be used should</td>
<td>Data to be read out of the sensor:</td>
</tr>
<tr>
<td>• be read out of the sensor by the controller</td>
<td>• E parts list</td>
</tr>
<tr>
<td>• be written out of the controller into the sensor.</td>
<td>- Manufacturer</td>
</tr>
<tr>
<td></td>
<td>- Product</td>
</tr>
<tr>
<td></td>
<td>- Serial number</td>
</tr>
<tr>
<td></td>
<td>• Configuration data</td>
</tr>
<tr>
<td></td>
<td>- Switch-on delay</td>
</tr>
<tr>
<td></td>
<td>- Switch-off delay</td>
</tr>
<tr>
<td></td>
<td>- Trigger mode</td>
</tr>
<tr>
<td></td>
<td>- Timer mode</td>
</tr>
<tr>
<td>Operation / visualization via an HMI</td>
<td>Data to be written into the sensor</td>
</tr>
<tr>
<td></td>
<td>• Configuration data</td>
</tr>
<tr>
<td></td>
<td>- Switch-on delay</td>
</tr>
<tr>
<td></td>
<td>- Switch-off delay</td>
</tr>
<tr>
<td></td>
<td>The following data are to be visualized on an HMI:</td>
</tr>
<tr>
<td></td>
<td>Operation:</td>
</tr>
<tr>
<td></td>
<td>• Starting the measurement</td>
</tr>
<tr>
<td></td>
<td>• Stopping the measurement</td>
</tr>
<tr>
<td></td>
<td>• Error acknowledgment</td>
</tr>
<tr>
<td></td>
<td>• Read command to the sensor for the</td>
</tr>
<tr>
<td></td>
<td>- E parts list</td>
</tr>
<tr>
<td></td>
<td>- Configuration data</td>
</tr>
<tr>
<td></td>
<td>• Write command to the sensor for configuration data</td>
</tr>
<tr>
<td></td>
<td>Visualization:</td>
</tr>
<tr>
<td></td>
<td>• Time duration of the gap measurement</td>
</tr>
<tr>
<td></td>
<td>• Length of the gap between two products</td>
</tr>
<tr>
<td></td>
<td>• Status messages</td>
</tr>
<tr>
<td></td>
<td>- Measurement active / inactive</td>
</tr>
<tr>
<td></td>
<td>- Error (detection of a gap between two products that is too small)</td>
</tr>
</tbody>
</table>
2 Solution

2.1 Solution overview

Display

The following figure gives a schematic overview of the most important components of the solution:

Advantages

This application offers you the following advantages:

- high-precision gap measurement through the use of
  - an innovative SICK sensor WSE4SC-3F2230A70 with Advanced functions,
  - the open IO Link interface ensuring data consistency from the sensor level to the control level,
  - a SIMATIC CPU S7-300 for processing the sensor data,
- comfortable operation and visualization by SIMATIC HMI,
- configuration of the sensor can be done on the HMI,
- increased productivity.

  - Due to the Advanced functions in the sensor, the data can be pre-processed in the sensor (in this case: independent determination of the time gap between two products).
2 Solution

2.2 Description of the HMI

- This increases the throughput (as it is possible to run on a higher conveyor velocity) and therefore an increased productivity.

- Simple commissioning,
- quick adaptation to changed tasks

- for example:
  Instead of measuring the gap between two products, you wish to determine the length of one product (the position of the product is incorrect - lengthwise or sidewise).

### 2.2 Description of the HMI

**Overview and description of the user interface**

The application is operated via the HMI. Four screens have been prepared for this:

<table>
<thead>
<tr>
<th>Note</th>
<th>Only fields with a white background can be edited.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Table 2-1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Screenshot</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC_INI</td>
<td><strong>START INITIALIZATION</strong>&lt;br&gt;When you press this button, the pre-assigned configuration values are written into the sensor. The values are displayed in the column “Value”. You will be informed in plain text if the initialization was successful or if it failed. It is only possible to start the gap measurement after a successful initialization.</td>
</tr>
<tr>
<td><img src="image_url" alt="Screenshot" /></td>
<td><strong>START MEASUREMENT</strong>&lt;br&gt;Changes to the screen PIC_MEASURE, from where you can start the gap measurement.</td>
</tr>
</tbody>
</table>

Table 2-1: Description and usage of the user interface screens.
## 2.2 Description of the HMI

### Screenshot

#### PIC_MEASURE

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACK</strong></td>
<td>Acknowledgment after an error (the gap between two products on the conveyor belt is less than the defined minimum).</td>
</tr>
<tr>
<td><strong>START</strong></td>
<td>When you press this button, the gap measurement starts.</td>
</tr>
<tr>
<td><strong>STOP</strong></td>
<td>When you press this button, the gap measurement stops.</td>
</tr>
<tr>
<td><strong>RD DATA</strong></td>
<td>Changes to the picture PIC_RD_DATA</td>
</tr>
<tr>
<td><strong>WR DATA</strong></td>
<td>Changes to the picture PIC_WR_DATA</td>
</tr>
</tbody>
</table>

### Command

- **Limit Gap**
  Here you can edit the minimum gap value in millimeters which must be met during gap measurement.

- **Velocity conveyor**
  Here you can edit the velocity of the conveyor belt. With these values, the SIMATIC CPU calculates the gap between two products. It is assumed that the velocity of the conveyor belt always remains constant.

### Results

- **Gap**
  Display of the determined value of the gap between two neighboring products on the conveyor belt.

- **Measured time gap**
  Display of the time value of the last measured gap between two products on the conveyor belt.

### Status

- **MEASUREMENT STOP**
  The measurement is inactive because
  - the STOP button has been pressed or
  - the minimum value for the gap between two neighboring products on the conveyor belt was not reached.

- **MEASUREMENT ACTIVE**
  The measurement is active

- **ERROR GAP**
  The minimum gap between two products on the conveyor belt was not reached. An acknowledgment is required to start a new measurement.
## 2.3 Description of the core functionality

### Principle of the core functionality

**Note** All the functionalities described in the following, run **automatically** once the application has been started. You will need the information if you want to modify the application example. For operating the application, however, the information is not imperative.

The core functions of this application example are:

- High-precision gap measurement between two neighboring products on a moving conveyor belt.
- Reading and writing E parts lists and sensor configuration data from and to the SICK sensor.

<table>
<thead>
<tr>
<th>Screenshot</th>
<th>Description</th>
</tr>
</thead>
</table>
| **PIC_RD_DATA** | **Read Data from sensor** By pressing a button (for example “Vendor name”), the  
- E parts list or  
- the respective configuration date is read from the sensor and displayed on the right side next to this button.  
**RESET ALL DATA** This button deletes the display, but not the data in the sensor. This function can be used to check the correctness of a read command.  
**BACK** Changes to PIC_MEASURE |
| **PIC_WR_DATA** | **Write Data to the sensor** You can write a switch-on or switch-off delay as a configuration date into the sensor (edit the value in milliseconds and press the corresponding button).  
**BACK** Changes to PIC_MEASURE  
**Note:** A Button RESET ALL DATA does not make any sense here. For checking the data, use the read function (see PIC_RD_DATA). |
2.3.1 Time duration of the gap measurement

Please find the description of the core functionality

- as an overview in form of a flowchart,
- followed by an explanation

Flow chart

Figure 2-2

Initialization

The initialization is done via the HMI and comprises writing the respective configuration values from the SIMATIC CPU into the SICK sensor. The configuration values are mandatory for the intended time measurement in the SICK sensor. There the following two parameters:

- On delay (default: 5 ms)
- Off delay (default: 3 ms)
- Trigger mode (time basis; in this application fixed at 1 ms)
- Timer mode (setting to measure the time for the gaps between two products on the conveyor belt, and not the products themselves, for example)

Note

The decentralized debouncing is identical to the On and Off delay.

To write the four desired parameter values into the sensor, four commands have to be triggered one after the other in the S7 program (in this application example, this happens automatically when the button INITIALIZATION in the HMI is pressed). This is done by setting a certain input bit (e.g. for the On delay) at a function block.
2 Solution

2.3 Description of the core functionality

developed by SICK (SICK-FB). This bit triggers this command automatically in the FB IO_CALL.

To submit commands, the respective indexes (e.g. index for an On delay) are normally necessary, which must be transmitted to the FB IO_CALL and can be found in the data sheet about the sensor. With the function block developed by SICK, this is no longer necessary. As a user, you simply have to submit the input bit as explained above.

Should the initialization fail, you will be informed about that on the HMI. Only after a successful initialization can the gap measurement begin.

Measuring the gap between two products

Due to the set timer mode, the sensor automatically begins measuring the time after a falling edge (end of product detection). The time measurement ends at after a rising edge (a product is detected by the sensor).

The information about the measured time (process data) reaches the IO Link Master (of the ET 200eco PN) from the SICK sensor (IO Link Device) and is present in two bytes there within one counter value.

The SIMATIC CPU can access both of these bytes via the PIB (Periphery Input Byte). The required addresses are in the hardware configuration of STEP 7, when the IO Link has been configured. The following figure shows this connection:

Figure 2-3
To see the time in milliseconds, the bits 0 and 1 from byte 1 need to be masked and the complete word must be shifted by two digits to the right.

Since the conveyor velocity $v_{\text{belt}}$ is known, the gap $s_{\text{gap}}$ between two products on the conveyor belt can be easily calculated in the S7 program:

$$s_{\text{gap}} = v_{\text{belt}} \cdot t_{\text{gap}}$$

Since $t_{\text{gap}}$ it is automatically indicated in milliseconds, and the conveyor velocity $V_{\text{belt}}$ is indicated in meters per second, the result for this length $s_{\text{gap}}$ is automatically the dimension millimeters.

**Advantage of this solution**

The great advantage of this solution is the use of the Advanced function of the SICK sensor. With the internal time detection directly in the sensor, and the possibility to read it out via IO Link directly into the process image of the SIMATIC CPU, quicker measurements become possible, which leads to an increased throughput.

### 2.3.2 Reading and writing E part lists and sensor configuration data

E part lists and configuration data can be read out by the SIMATIC CPU from the SICK sensor or written into it. The following paragraph explains this functionality in more detail.

**Basic principle**

The data exchange is done via IO Link Master. You need the FB IO_CALL for this, which you can download free of charge from the SIEMENS Industry Online Support.

You submit information via the input parameter of the FB IO_CALL, for example

- whether you wish to read data from the sensor (IO Link Device) or write data into it.
- at which port your IO Link Device is connected.
- in which data section (e.g. DBx from Byte 0 to Byte 500) you wish to save the data read by the sensor (IO Link Device).

You now have to inform the IO with an index which data you wish to read out, for example the sensor serial number as an E part list, or the On delay as a configuration date. For the examples used here, this would be the index 0x15 for the serial number, or index 0x54 for the On delay of the SICK sensor. Please refer to the manufacturer's data sheet to see which index you need.

Submit the command to the FB IO_Call via an edge call. Among other things with the output bits

- **BUSY**
- **DONE_VALID**
- **ERROR**

the FB IO_CALL informs about the submitted command. In case of an error, additional error information is put out.
2 Solution

2.3 Description of the core functionality

**Simplification by the SICK function block**

The application example uses a function block in the S7 program developed by SICK (SICK-FB). Instead of communicating the respective index for a certain command, it is sufficient to call the respective input in the SICK-FB with the corresponding bit. The SICK-FB makes the assignment to the required index within the “Black Box”.

Figure 2-4

For you as a user this means that you need the FB IO_CALL to be present in the SIMATIC Manager. The know-how-protected SICK-FB automatically calls the FB IO_CALL. The corresponding STEP 7 project is prepared accordingly.
2.4 **Hardware and software components used**

The application was set up with the following components:

### Hardware components

Table 2-2

<table>
<thead>
<tr>
<th>Component</th>
<th>No.</th>
<th>Order number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMATIC S7-300 CPU 315-2 PN/DP</td>
<td>1</td>
<td>6ES7315-2EH14-0AB0</td>
<td></td>
</tr>
<tr>
<td>ET 200ECO PN, IO-LINK MASTER</td>
<td>1</td>
<td>6ES7148-6JA00-0AB0</td>
<td></td>
</tr>
<tr>
<td>MULTI PANEL MP 277 10'' Touch 1</td>
<td>1</td>
<td>6AV6643-0CD01-1AX0</td>
<td></td>
</tr>
<tr>
<td>WSE4SC-3F2230A70</td>
<td>1</td>
<td>1060206</td>
<td>Unit consisting of a sensor and a receiver</td>
</tr>
</tbody>
</table>

### Standard software components

Table 2-3

<table>
<thead>
<tr>
<th>Component</th>
<th>No.</th>
<th>Order number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMATIC STEP 7: V5.5 SP2</td>
<td>1</td>
<td>6ES7810-4CC10-0YA7</td>
<td></td>
</tr>
<tr>
<td>WinCC flexible 2008 SP3</td>
<td>1</td>
<td>6AV6613-0AA51-3CU8</td>
<td></td>
</tr>
<tr>
<td>WinCC flexible 2008 SP3 Runtime</td>
<td>1</td>
<td>6AV6613-1XA51-3CU8</td>
<td></td>
</tr>
<tr>
<td>PCT</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Sample files and projects

The following list includes all files and projects used in this example.

Table 2-4

<table>
<thead>
<tr>
<th>Component</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>80198907_Gap_measurement_v10.zip</td>
<td>This zip file contains the STEP 7 project.</td>
</tr>
<tr>
<td>80198907_Gap_measurement_v10_d.pdf</td>
<td>This document.</td>
</tr>
</tbody>
</table>
3 Basics about the SICK Sensor and IO Link

3.1 Basics about the SICK sensor used

Independent from the interaction of the individual components, this chapter contains basic information about the sensor systems used and the topic IO Link. The aim is to allow even readers who have only little experience - or none at all - with sensor systems or IO Link, to understand what this application example is all about.

3.1 Basics about the SICK sensor used

Standard and Advanced functions

The following explains which concrete functionalities are behind the expressions.

3.1.1 Standard functions

Device exchange

An exchange of the sensor is quick and easy. Since the configuration data of the SIMATIC controller are known, they can be written into the SICK sensor (IO Link Device) via the open standard IO Link by means of the SICK-FB.

Writing into the SICK sensor (IO Link Device) is a realized core functionality of this application example.

Diagnosis

With the integration of the SICK sensor into the automation level via IO Link and the use of the SICK-FB, in addition to internal diagnoses (e.g. short circuit), also those diagnoses can be evaluated that do not concern the sensor directly, but the application. Please get respective error information at the SICK-FB via the respective output parameters (see figure below).

Figure 3-1

Please find the meaning of an “ERROR_CODE” in the manual to the respective manual about the SICK-FB (see link list at the end of this document).
E parts list

An electronic documentation of all sensors in the condition in which the machine or the plant is delivered can be created quickly and automatically. The IO Link technology allows for the automatic detection of IO Link sensors in a machine and presents them by name.

You as user have the following advantages:

- Redundancies and additional costs and expenditures are avoided.
- Increased transparency for the electronic documentation of installed sensors.
- Complicated trouble-shooting because of different documentation states.
- Error-free and simple documentation of the condition in which the machine or the plant is delivered.

Figure 3-2

3.1.2 Advanced functions

Decentralized debouncing

In some plants and machines it is essential for productivity that a sensor knows which detection signals are interferences and that it suppresses them using other detection information. With a reliable suppression of interference signals, the controller is not overloaded with information that interferes with the process, but receives an interference-free signal.

The debouncing times that can be implemented are independent from the cycle times, bus running times and switch-on delays. Only a debouncing value depending on production must be submitted to the sensor.

The detection of the interference signals to be debounced can be carried out with the velocity of the sensor controller.
3 Basics about the SICK Sensor and IO Link

3.1 Basics about the SICK sensor used

Possible uses for the decentralized debouncing:

The option of a decentralized debouncing offers the following advantages:
- The sensor does not debounce the switching signal, but only what it actually detects.
- Precise detection.
- Increased machine speed.

Time measurement

The time measurement detected in this Application Example is needed for calculating the gap between two products on the moving conveyor belt. With a simple adaptation of this application example, it is possible to solve other tasks, too, such as a length measurement, by using the Advanced function “Time measurement”.

Length measurement of the product

The sensor detects the passing product with a high level of precision. The time \( t_{Product} \) between the rising and falling edge (with the maximum internal cycle frequency) is evaluated. In this Application Example, the length of the product \( v_{Band} \) is determined in the same way by means of the known conveyor velocity \( l \):

\[
l = v_{Belt} \cdot t_{Product}
\]
### 3.2 Basics about IO Link

#### Introduction

The IO Link point-to-point interface through which process data and parameters are transferred was especially designed for the connection of any type of sensors or actuators to a control system. IO Link does not use the classic bus wiring but maintains the usual parallel wiring for sensors and actuators. The new communication standard below the field bus level enables central error diagnostics and error localization up to the sensor/actuator level and facilitates commissioning and maintenance by being able to dynamically change parameterization data directly from the application. The IO link can be integrated as an open interface in all common field bus and automation systems. Consequent interoperability provides high investment protection. This also applies within the framework of existing field bus and automation systems. It is a storage of existing machine concepts for the further use of sensors that have no IO Link interface.

#### Configuration of IO Link Master

The configuration of the IO Link Master and the devices connected to it (up to four) is done in an independent tool, the Port Configuration Tool, in short: PCT. In the PCT it is defined:

- which IO Link Device (e.g. a sensor) is connected to which port of the IO Link Master module,
- which addresses are assigned by the IO Link Devices in the IO area of the SIMATIC S7-CPU
- what the configuration for the devices looks like.

The following screenshot shows the SICK sensor of this application example in PCT:

![SICK Sensor Example in PCT](image)
To configure an IO Link Device in the PCT, it must be available as a descriptive file (IODD) in the catalog. If the required link device is not available in the catalog, the IODD can be imported with the menu command “Options > Import IODD”.

The following screenshot shows the IODD of the IO Link Device used in the application example imported into the catalog.

Figure 3-6

IO Link components by Siemens

Siemens offers a comprehensive product portfolio and support for IO Link:

- Master
- Device, IO module
- Device, industrial switchgear
- Device, RFID systems (RFID = Radio-Frequency IDentification)
- Port Configuration Tool - S7-PCT
- Function block "IOL_CALL" for SIMATIC S7-300 and S7-400
- IO Link Application Example

Please find more detailed information about the products in the Internet.

Advantages of IO Link advantages

The following advantages are in favor of IO Link:

- Open standard according to IEC 61131-9.
  - Devices can be integrated into all customary field bus devices and automation systems in the same way.
- Tool-supported parameter settings and central data storage.
  - Quick configuration and startup.
  - Simple creation of the up-to-date plant documentation also for sensors/actuators.
3 Basics about the SICK Sensor and IO Link

3.2 Basics about IO Link

- Simple, standardized wiring and significantly reduced interface variety at the sensors/actuators.
  - Reduction of the number of types and stock-keeping
  - Rapid commissioning
  - Reduction of space needed
  - Any combination of IO Link Devices and sensors/actuators without IO Link possible at the IO Link Master

- Consistent communication between sensors/actuators and the CPU.
  - Access to all process data, diagnosis data and device information
  - Access to device-specific data, e.g. energy data
  - Remote diagnosis possible.

- Consistent diagnosis information down to the sensor/actuator level
  - Quicker and easier trouble-shooting
  - Minimizing downtime risk
  - Preventive maintenance and optimization of maintenance and servicing planning

- Dynamic changes of the sensor/actuator parameters by the controller or the operator at the HMI.
  - Reduction of downtimes during product change
  - Increase of product variations at the machine

- Automatic reconfiguration when devices were changed during operation.
  - Minimizing downtimes
  - Device exchange by untrained personnel without additional tools
  - Avoiding wrong settings
4 Function Mechanisms of this Application

Program structure

The function mechanisms of this application can be explained with the implemented program structure.

Various kinds of code modules were used for the solution of the automation task:

- a Siemens code we developed ourselves without a know-how protection
- a Siemens code with a know-how protection
- a SICK code with a know-how protection

The following figure describes the program structure:

The start OP (OB100) with some parameters preset and the OB1, which only calls the FB MAIN (FB3) in very program cycle, are not explained any more in the following.

Note

The FB WEx_TMS_DBC_PB (FB16) is identical to the previously used name “SICK-FB”.

4.1 Program coordination by the FB MAIN (FB3)

The FB MAIN (FB3) calls the FB INI (FB4) first. As long as the latter has not carried out the initialization, the FB MAIN (FB3) does not call the FB CALCULATE (FB4).

4.2 Initialization by the FB INI (FB4)

Initialization means: Configuration data are written from the SIMATIC CPU via the IO Link Master to the SICK sensor (IO Link Device). The configuration data are intended for the following parameters:

- Counter mode
- Trigger mode
- Off delay
- On delay

The configuration data are in the data sheet of the sensor. They are stored in the FB INI (FB4):

- For the counter mode: 0x26
  - According to the sensor data sheet, this means a “MODE B”. With the setting of the “MODE B”, the SICK sensor measures the time after a falling edge to a rising edge, i.e. for how long there is a gap between two products on the conveyor belt.
  - The counter mode can be read, but not changed on the HMI.
- For the trigger mode: 0x5
  - According to the sensor data sheet, this means a time basis of 1 ms.
  - The trigger mode can be read, but not changed on the HMI.
- For the switch-off delay: 3
  - The Off delay can be entered directly in milliseconds in a range from 0 to 30,000.
  - The Off delay can be read and changed on the HMI.
- For the switch-on delay: 5
  - The On delay can be entered directly in milliseconds in a range from 0 to 30,000.
  - The On delay can be read and changed on the HMI.

The configuration data are submitted to the SICK sensor one after the other by means of the SICK-FB over several program cycles. The following flow chart demonstrates the function:
4.3 Detection of the gap by the FB CALCULATE (FB2)

Preconditions for starting the time measurement

Network 1 checks whether the time measurement can be started. Preconditions for starting the time measurements:

- Successful initialization (see chapter 4.2)
- Pressing the START button on the HMI

Reasons why a time measurement is not processed:

- The initialization has not taken place.
- The START button on the HMI has not been pressed.
- During a time measurement, the STOP button on the HMI was pressed.
- Pending error: During a time measurement a value for a gap between two products was determined, which is below the minimum (Bit ERR_GAP = TRUE)

Edge evaluation

Bit 0 of the counter value provides information whether the SICK sensor has detected a product on the conveyor belt.

- Bit 0 = FALSE: Sensor does not detect any product
- Bit 0 = TRUE: Sensor has detected a product
Network 2 separates the Bit 0 from the counter value and saves the information into the output variable DETECTION.

Figure 4-3

For the required time measurement, this means: As soon as DETECTION = TRUE, a product has been detected by the sensor. Exactly at this point, the time measurement of the gap ends. In other words: At the beginning of the positive edge of DETECTION, the gap measurement.

**Determination \( t_{Gap} \) from the counter value**

Network 3 masks Bit 0 and Bit 1 of the counter value and shifts all 16 Bits to the right by 2 digits. Therefore, \( t_{Gap} \) is in milliseconds (see figure 2-3).

**Measuring the gap between two products**

Since the velocity of the conveyor belt of the type REAL is known, the time \( t_{Gap} \) must be converted too.

- With the command ITD from 16 Bit (whole number) to 32 Bit (whole number).
- With the command DTR from 32 Bit (whole number) to 32 Bit (floating point number).

The gap between two neighboring products on the conveyor belt can be determined in this way.

\[
S_{Gap} = v_{Belt} \cdot t_{Gap}
\]

Since \( t_{Gap} \) it is automatically indicated in milliseconds, and the conveyor velocity \( v_{Belt} \) is indicated in meters per second, the result for this length \( S_{Gap} \) is automatically the dimension millimeters. The HMI displays the millimeter value as a whole number.
4 Function Mechanisms of this Application

4.4 Environment of the SICK-FB

Error handling

$S_{\text{Gap}}$ is compared to a limit value. If $S_{\text{Gap}}$ is smaller than the limit value, the gap between two products on the conveyor belt is too small. The error bit ERR_GAP is set accordingly.

With ERR_GAP = TRUE another time/gap measurement can be carried out. The next measurement can only be started after having acknowledged the message on the HMI and having pressed the button START.

4.4 Environment of the SICK-FB

The following image shows the SICK-FB (FB WEx_TMS_DBC_PB (FB16)) together with the DB SAVE_DATA (DB21) and the DB ACT_PAR_IO_LINK (DB20).

Figure 4-4

The submission of the commands via the SICK-FB is done via the DB ACT_PAR_IO_LINK (DB20). It contains the actual parameters for the SICK-FB.

The data structure of the SICK-FB is represented in an UDT16. The DB SAVE_DATA uses this structure of the UDT16.
5 Installation

4.4 Environment of the SICK-FB

5 Installation

Hardware installation

The figure below shows the hardware setup of the application.

Figure 5-1

Note

Setup guidelines for the individual components must generally be followed.

Software installation

Table 5-1

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Install STEP 7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>To be able to use the S7 project that is also offered:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Download the zip file into a local directory of the Windows Explorer.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• In the SIMATIC Manager, go to “File -&gt; Retrieve” and select the zip file. Follow the instructions.</td>
<td></td>
</tr>
</tbody>
</table>
6 Starting the Application

Preconditions for starting the application example:
The hardware and software have been installed according to chapter 5.

6.1 Edit Ethernet nodes

The Application Example uses the following IP addresses:

Table 6-1

<table>
<thead>
<tr>
<th>Hardware component</th>
<th>IP address</th>
<th>Subnet mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMI</td>
<td>192.168.0.1</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>SIMATIC S7-300 CPU</td>
<td>192.168.0.2</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>ET 200eco PN IO-Link</td>
<td>192.168.0.3</td>
<td>255.255.255.0</td>
</tr>
</tbody>
</table>

For setting the IP addresses, please proceed as follows:

Table 6-2

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In the menu of the SIMATIC Manager: “Target system &gt; Edit Ethernet nodes”</td>
<td>The screen depicted under number 2 opens up.</td>
</tr>
<tr>
<td>2</td>
<td>Click the “Search” button.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>With the button “Assign IP configuration”, you can transmit the IP addresses to the corresponding hardware component.</td>
<td></td>
</tr>
</tbody>
</table>
6.2 Loading the hardware configuration of STEP 7

Table 6-3

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In the SIMATIC Manager, doubleclick on “Hardware”.</td>
<td>The hardware configuration of STEP 7 opens up.</td>
</tr>
<tr>
<td>2</td>
<td>Click on the highlighted icon.</td>
<td>Follow the instructions.</td>
</tr>
</tbody>
</table>

6.3 IO Link configuration

Note

In PCT you need IODD. You can get the IODD via download from the sensor producer (see also chapter 3.2).

Table 6-4

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>In the SIMATIC Manager, doubleclick on “Hardware”.</td>
<td>The hardware configuration of STEP 7 opens up.</td>
</tr>
<tr>
<td>2</td>
<td>Select the ET 200eco PN.</td>
<td></td>
</tr>
</tbody>
</table>
6 Starting the Application

6.3 IO Link configuration

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Select the row of slot 1.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Right mouse button &gt; Configure IO Link</td>
<td>The PCT opens up.</td>
</tr>
<tr>
<td>5</td>
<td>Select the ET 200eco PN and press the icon “Load with devices”</td>
<td>The configuration is transmitted.</td>
</tr>
<tr>
<td>6</td>
<td>Close the PCT.</td>
<td></td>
</tr>
</tbody>
</table>

Check whether the connection of the IO Link fits the parameters of the SICK FB. In this Application Example, the SICK sensor (IO Link Device) is connected to the Port 1 of the IO Link Master. Should that not be the case in your system, change the entry at the input PORT of the SICK-FB.

Figure 6-1

IO Link Master
6.4 Loading the S7 project into the SIMATIC CPU

Table 6-5

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select the SIMATIC station in the SIMATIC Manager.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Press the “Load” icon</td>
<td>Download the system data and the S7 program</td>
</tr>
</tbody>
</table>

6.5 Loading the WinCC flexible project into the HMI

Tabelle 6-6

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Aktion</th>
<th>Anmerkung</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open the WinCC flexible und load the project into the HMI.</td>
<td></td>
</tr>
</tbody>
</table>
7 Operating the Application

7.1 Initialization

NOTICE
All the instructions given here, especially the values given as examples (e.g. for the limit value between two products), assume that you make sure that they do not lead to undesired results.

Requirements
- Start-up according to chapter 6.
- Sender and receiver (SICK sensor) are aligned accordingly.

7.1 Initialization

The successful initialization is a precondition for starting the measurement. To carry out the initialization, please follow the instructions.

Table 7-1

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On the HMI, press the flashing button “START INITIALIZATION”.</td>
<td>After a successful initialization:</td>
</tr>
<tr>
<td></td>
<td>Should you have pressed the button “START MEASUREMENT” by accident,</td>
<td>Plain text information “Initialization successful”. In addition, you</td>
</tr>
<tr>
<td></td>
<td>please proceed as follows to return to the PIC_INI screen.</td>
<td>will be informed which configuration data were written to the sensor.</td>
</tr>
<tr>
<td></td>
<td>&gt; Press the button “RUNTIME OFF”</td>
<td>After a faulty initialization:</td>
</tr>
<tr>
<td></td>
<td>&gt; Press START</td>
<td>Plain text information “Initialization not successful”. In this case,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>press the button “START INITIALIZATION” again.</td>
</tr>
<tr>
<td>2</td>
<td>On the HMI, press the flashing button “START MEASUREMENT”</td>
<td>to change to the HMI screen PIC_MEASUREMENT</td>
</tr>
</tbody>
</table>
7.2 Carry out measurement

Requirements

The initialization was successfully carried out, and the PIC_MEASURE is on the HMI.

Figure 7-1

Table 7-2

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Now set the following parameters on the HMI:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Limit Gap</td>
<td>Limit Gap: Minimum distance between two products, which must be attained.</td>
</tr>
<tr>
<td></td>
<td>- Velocity conveyor</td>
<td>Velocity conveyor</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>Velocity of the conveyor (constant)</td>
</tr>
<tr>
<td></td>
<td>Limit Gap = 50 mm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Velocity conveyor = 1 m/s</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Press the START button.</td>
<td>The measurement begins</td>
</tr>
<tr>
<td>3</td>
<td>Put an obstacle (hand, product etc.) between the sender and the receiver of the SICK sensor.</td>
<td>The sensor detects the beginning of the object to be measured.</td>
</tr>
<tr>
<td>4</td>
<td>Removing the obstacle.</td>
<td>The sensor detects the beginning of the gap and starts the internal time measurement.</td>
</tr>
<tr>
<td>5</td>
<td>Repeat no. 3.</td>
<td>The sensor stops the internal time measurement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The measured time value and the gap calculated by the SIMATIC CPU are displayed on the HMI under &quot;Results&quot;.</td>
</tr>
</tbody>
</table>

7.3 Minimum gap not met as a fault scenario

Table 7-3

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Carry out all the activities listed in the table 7-2. You can produce a fault (limit value between two products not met) as follows:</td>
<td>After a fault (limit value not met), the following applies: “Limit Gap” &gt; “Gap”. The status on the HMI shows: - MEASUREMENT STOP - ERROR GAP The ACK button flashes (asking for an acknowledgment of the error)</td>
</tr>
</tbody>
</table>
|     | • Either carry out the actions of number 3 to 5 from table 7-2 quicker or  
|     | • enter a bigger limit for “Limit Gap” first.                                                                                                                                                           |                                                                                                                                                                                                      |
| 2   | To carry on with the measurement, you have to press the ACK button for acknowledgment.                                                                                                                 | On the HMI, only the status - MEASUREMENT STOP is pending. The ACK button does not flash any more. The START button pops up.                                                                         |
| 3   | Carry out the actions number 2 to 5 from table 7-2.                                                                                                                                                     | Measurements are carried out.                                                                                                                                                                       |

7.4 Reading configuration data from the SICK sensor

Table 7-4

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Press the button “RD DATA” to change to the screen PIC_WR_DATA.</td>
<td>to change to the screen PIC_WR_DATA.</td>
</tr>
<tr>
<td>2</td>
<td>Press a button (for example “Vendor name”).</td>
<td>The respective information will be read out from the sensor (in this case: “SICK AG”).</td>
</tr>
</tbody>
</table>

7.5 Writing configuration data into the server

Table 7-5

<table>
<thead>
<tr>
<th>No.</th>
<th>Action</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Press the button “WR DATA”.</td>
<td>to change to the screen PIC_WR_DATA.</td>
</tr>
<tr>
<td>2</td>
<td>Enter a value (for example 4 ms for “On Delay”).</td>
<td>The value (in this case: 4 ms) is written into the sensor. You can check the value by reading the value out of the sensor (see table 7-4).</td>
</tr>
<tr>
<td>3</td>
<td>Press the respective button (in this case: “On Delay”).</td>
<td>The value (in this case: 4 ms) is written into the sensor. You can check the value by reading the value out of the sensor (see table 7-4).</td>
</tr>
</tbody>
</table>
8 Related Literature

8.1 Bibliography

This list is not complete and only represents a selection of relevant literature.

Table 8-1

<table>
<thead>
<tr>
<th>Topic</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>/1/ STEP 7 SIMATIC S7-300/400</td>
<td>Automating with STEP 7 in STL and SCL</td>
</tr>
<tr>
<td></td>
<td>Author: Hans Berger</td>
</tr>
<tr>
<td></td>
<td>Publicis MCD Verlag</td>
</tr>
<tr>
<td></td>
<td>ISBN: 978-3-89578-397-5</td>
</tr>
</tbody>
</table>

8.2 Internet link specifications

The following list is not complete and only represents a selection of relevant information.

Table 8-2

<table>
<thead>
<tr>
<th>Topic</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>/1/ Reference to the entry</td>
<td><a href="http://support.automation.siemens.com/WW/view/en/80198907">http://support.automation.siemens.com/WW/view/en/80198907</a></td>
</tr>
<tr>
<td>/2/ Siemens Industry</td>
<td><a href="http://support.automation.siemens.com">http://support.automation.siemens.com</a></td>
</tr>
<tr>
<td>Online Support</td>
<td></td>
</tr>
<tr>
<td>(Function Manual)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>communication/io-link/pages</td>
</tr>
<tr>
<td>/5/ “IOL_Call” for the</td>
<td><a href="http://support.automation.siemens.com/WW/view/en/38487085">http://support.automation.siemens.com/WW/view/en/38487085</a></td>
</tr>
<tr>
<td>acyclic data exchange</td>
<td></td>
</tr>
<tr>
<td>with an IO link device</td>
<td></td>
</tr>
<tr>
<td>ET 200 IO-Link Master</td>
<td></td>
</tr>
<tr>
<td>/7/ Website SICK AG</td>
<td><a href="http://www.sick.com">http://www.sick.com</a></td>
</tr>
</tbody>
</table>

9 History

Table 9-1

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>10/2013</td>
<td>First version</td>
</tr>
</tbody>
</table>