Industrial Data Communication- PLC and External Application

Tanoy Kumar Paul
DF&PD, Siemens
Bangalore, India
tanoy.paul@siemens.com

Manoj B
DF&PD, Siemens
Bangalore, India
b.manoj@siemens.com

Abstract— In the field of industrial automation and advanced robotics, data communication between hardware and user application is always important and complex in nature. This research paper presents an industrial solution to connect Programmable Logic Controller (PLC) with user application and describes the detailed description for configuration of PLCs, communication between PLCs, user application, and different databases. Next sections will explain the results of the research in terms of programming the PLCs and preparing the unique solution to configure a PLC connection and executing it to transfer data from PLC to PLC, PLC to database, or database to PLC.

Keywords— Industry automation; PLC; Robotics; Data communication;

I. INTRODUCTION

Current days, the trends in manufacturing industries are building the fully automated plants using different robotic applications to achieve better performance with robust behavior. To get better productivity and quality manufacturing, industries are investing a big amount of their budget in automation for the plants. As a result, human machine communication in the field of production management is obvious. To control the complex hardware in industries like automobile or steel plants, use of microcontroller or microprocessor is not enough. To manage those requirements in a tough environment, use of Programmable Logic Controllers (PLCs) with a user-friendly human machine interface (HMI) is globally accepted now.

PLCs are able to handle -20° Centigrade to +70° centigrade temperature. It requires to configure a PLC properly and programming it properly by following the user requirements. PLCs are able to send and receive data from other databases by using client application. Client application needs proper interface to talk with PLCs. Client application build a bridge between PLC and other database or with another PLC. For this documentation, a PLC SIMATIC S7 300, 314C-2 DP compact CPU with MPI, 24DI/16DO and with 24DC power supply is used.

II. PLC

A. PLC Configuration

Configuration [1] is required for both hardware and software. In hardware configuration, it is obvious to add PLC in a rack or rail (base of PLC) and then adding of power module and I/O devices, based on requirement. For this research, Siemens PLCs are used. For software configuration it is also required to configure the PLCs in Step7 project. Step7 project allows user to add the PLCs with CPU number and rack or rail number with communication profile manually. After proper configuration of hardware and software, user can upload the hardware details in the physical PLC [2, 4].

B. PLC Programming

To send and receive data to/from PLC, it is required to program PLC properly. For this current solution, Simatic S7-300 from Siemens is used and configured using Step7 programming. To create a basic application for sending and receiving data to PLC or from PLC, user needs to create data block(s), function block(s) and organization block(s) [5]. There are few steps to program a PLC for sending and receiving data properly.

Organization Block (OB): The operating system of the S7 CPU executes OB periodically. When OB has been executed, the operating system starts it again. Cyclic execution of OB starts after the startup has been completed. It is possible to call other function blocks (FBs, SFBs) or functions (FCs, SFCs) in OB.
Data Block (DB): DBs are data areas for storing user data. In addition to the data that are assigned to a function block, shared data can also be defined and used by any blocks.

Function Block (FB): FBs are blocks with a "memory" which you can program itself.

UDT: User-defined data types are special data structures that is require to create by user. By creating this, user can use the data types in the whole S7 program, once they have been defined.

VAT: The variable tables (VAT) that user can create to monitor and modify variables for debugging own program. Variable tables are not downloaded to the CPU.

III. USER APPLICATION

User application is responsible to create an interface between different mediums and is able to send and receive data. A unique solution from this research is used here. Data communication is possible between any data bases, SCADA solutions and PLCs. Application allows the flow of data in any direction among these components.

User application can be divided in two sections; first one is configuration and second is runtime part [3]. Configuration part is responsible to take required input from user regarding source of data and provider of data. Based on configuration, runtime acts accordingly. As a result user application sends and receives data by communicating PLCs or other databases.

Fig. 1. Configuration and Runtime Data Flow

A. Configuration

Configuration part is completely dependent on the requirement from users, it takes the inputs for source (from where data comes) and target (destination of data) and about their connection details. A database file or markup language file is the output of configuration module. This file holds all the information about the source and target of data. Runtime module reads this file and gives the final output as data transfer.

Total configuration or the output of configuration section can be explained in a tree structure.

1. Body: Hold a new structure (relation between source and target) or connection details.

2. Relation: Relation indicates the details configuration between a single data source and data target. A single body can hold multiple relations with different name. Relations have all information of data source and target with group details.
Relation node in above structure holds three nodes for configuration. Data source node keeps all the configuration data related to the source. Similarly Data Target node holds the data for the destination. Group node is dedicated for the individual connections between the particular fields of data source and data targets. Every fields need to be connected with each other to send the data from one position to another position.

3. Data Source: A source can be a PLC or any database. For example if PLC is the source of data then this tag holds information like IP address, protocol type, monitor cycle etc required to make a secure communication.

<table>
<thead>
<tr>
<th>Data Source</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Protocol</td>
</tr>
<tr>
<td>Local IP</td>
<td>Remote IP</td>
</tr>
<tr>
<td>Connection Type</td>
<td>Monitoring Cycle</td>
</tr>
</tbody>
</table>

4. Data Target: A target structure can be same as source structure but it is configured to receive data only. For example if Oracle data base is the target then this tag will hold the information like IP address of server, database name, table name, communication protocol etc required for oracle communication. Structure example of source is same as target if PLC is the receiving point as data target.

<table>
<thead>
<tr>
<th>Data Target</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>Protocol</td>
</tr>
<tr>
<td>Local IP</td>
<td>Remote IP</td>
</tr>
<tr>
<td>Connection Type</td>
<td>Monitoring Cycle</td>
</tr>
</tbody>
</table>

5. Group: Group structure holds the property of entire relation. Like - update time, data field information with their data types, server information etc. Group also holds the details of the communication. A communication indicates the details of mapping between the fields. Fields indicates the single tag option in PLCs or a single column name in the database table. This is used for to give the particular of the communication like which particular field of source will be connected with which particular field of target. A group can hold multiple communications.

<table>
<thead>
<tr>
<th>Group</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Name</td>
<td>Group Description</td>
</tr>
<tr>
<td>Update Cycle</td>
<td></td>
</tr>
</tbody>
</table>

Communication1

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td></td>
</tr>
<tr>
<td>Memory Location</td>
<td>Data Type</td>
</tr>
<tr>
<td>Data Target</td>
<td></td>
</tr>
<tr>
<td>Memory Location</td>
<td>Data Type</td>
</tr>
</tbody>
</table>

Communication2

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td></td>
</tr>
<tr>
<td>Memory Location</td>
<td>Data Type</td>
</tr>
<tr>
<td>Data Target</td>
<td></td>
</tr>
<tr>
<td>Memory Location</td>
<td>Data Type</td>
</tr>
</tbody>
</table>
B. Runtime

Runtime module is responsible to follow the configuration structure, which is the output of configuration module. By reading structure information, it connects with different PLCs or databases.

1. PLC configuration data for Runtime - In time of PLC configuration, selecting a protocol type, data length in bytes and the type is required for connection establishment.

The following protocol types are available for selection:
   - TCP/IP
   - UDP

2. Properties and settings

To communicate with PLC and the external application there are few fields which need to be configured properly. The fields are –

- Data Size - Data length indicates the size of the variable which needs to be in bytes. It needs to be mapped with PLC memory and application data. It is basically the tag lengths in byte.
- Connection Time – Connection time is responsible to check the connection in a cyclic manner. Following parameters are used to configure -
  - Cycle time: This refers to the time after which a ping will be set to check for whether the connection to the partner station still exists. Depending on the network load, it is required to select the monitoring cycle larger (with higher network load) or smaller.
  - Timeout: Maximum response time for a ping before an error is indicated. This unit needs to be specified in ms.

Addresses –

In the addresses section, connection data (IP address) for the source and target system need to store.

- Application IP address: IP address of the machine where external application is installed. The IP address” must be configured if the computer has multiple network cards or IP addresses.
- PLC IP address: Depending on the set protocol, further settings are required:
  - ISO on TCP: TSAP (Transfer Service Access Point) in ASC format for application. Configure the TSAP with a length from 1 to 16 bytes.
  - TSAP PLC: TSAP in ASC format for the PLC device. It can be configured by TSAP with a length of from 1 to 16 bytes.

UDP, TCP/IP -

Indicates the hosting application port number; value for port is in the range of 2000 to 65534. PLC Port address can be set to the value for port in the range 2000 to 65534.

3. Memory mapping

Application configuration allows user to map the variables based on the memory allocation. If a PLC is holding a 2bytes of memory space in application configuration user have to give the same space for that corresponding variable. In Fig. 2, image is presenting the one-to-one mapping mechanism within PLC memory and application configuration. The first image is representing the PLC memory allocation and second one is presenting the application configuration section.
Application mapping is done using the same variable used in Step 7 and application. Data length can be considered for the variables in bytes. As shown in Fig. 3, for a 2-byte integer it takes the first two bytes, then for next double integer four bytes and for next real variable another four bytes gets mapped. Total ten bytes of data are mapped for these three variables.

![Diagram of Data Size and Mapping]

For string contents, application allows the starting position of binary string and the size of total bytes as input. For a string array of size 200, application allocates 200 bytes of memory to get the data from data source or to send the data to a data source. As shown in Fig. 4, this data source can be any database.

![Diagram of String Data and Mapping]

1. **Socket Communication** – Application handle the communication between PLCs and other source or destination using secure socket communication.
2. **Protocols** – External application is used for sending and receiving data to or from Siemens PLC. A secure communication is created over TCP protocol and enables reading or writing data to or from the PLC memory. This part acts both as a data source and data target.
3. **Connect / Disconnect** – A connect and disconnect module works to make a communication between PLC and the application. This helps to control the communication from application.

### IV. DATA RESULT

The performance graph of the modules are shown in the below figure. Data transfer between PLCs and other modules are dependent on memory availability. A module for different source and destination in application consumes memory and runs
continuously for several days in real plant environment. Below figure shows the memory performance (stability) when data transfer happen for weeks or months.

Fig. 5. Memory Usage and Performance

V. CONCLUSION

Data communication between PLC and external application is necessary. External application plays the role of a bridge between hardware and database. SCADA system depends on flow of data between engineering platform and other systems. This communication helps to create a secure and high speed connection and results in smooth data transfer.

REFERENCES


BIOGRAPHY

Tanoy Kumar Paul is involved with product design, development and technical writing in Industrial Automation domain and doing active research in the areas of human behavior detection in critical situations for robotic intelligence. He is also involved in research of futuristic solution and proposing innovative business solution. He did his MCA from BIT Mesra, Ranchi, India and MBA(Project Management) from SMU, Sikkim, India. He started his career with research in Jadavpur University, India and now working in area of product design and development for Siemens, Bangalore, India.

Manoj B is currently working as a Project Manager in Siemens Technology and Services Private Limited at Bangalore, India. Mr. Manoj B is a Master of Computer Application graduate from Bharathiyar University. He holds a Bachelor of Science degree in Mathematics from Kerala University. He has 17 years of experience in Software Development and Project management.