Chapter 2  Processing I/O

Introduction

After an introduction of PLCs in chapter one, various characteristics of the PLC need to be discussed. The purpose now is to discuss what parts constitute a modern PLC and how these parts interface. The topology of a PLC system is also discussed as to how PLCs are distributed in a manufacturing environment to best control a process efficiently.

Since each is electronic and operates using one or more microprocessors, a 5 volt power supply and CPU (central processing unit) are the core of the PLC. Included in the CPU is a computer with memory and communications hardware to communicate to a programming panel, the I/O, and to a network which is either peer-to-peer or a multimode network.

Many PLC vendors divide the work of the PLC between multiple microprocessors with coordination handled by a master microprocessor. One processor may be assigned to handle the I/O. Another may handle the networking and communication to the programming panel. A supervisory microprocessor handles the logic, scan, arithmetic, and other instructions solution of the program.

Inputs and outputs complete the PLC with inputs reporting the status of the system and outputs controlling the sequencing of the process. Inputs and outputs are of many types and forms. A simple switch can be an input. Also, a high-speed pulse input can be an input providing speed information from a motor. Inputs and outputs alike can be simple or complex in nature. Both the simple I/O as well as the more complex will be discussed in the chapter and through the rest of the book.

Overview of the PLC

Inputs form the portion of the PLC connecting switches, sensors, transducers and other devices to the processor. Typically, the input is tied to a screw terminal. The PLC program reads the status of the inputs and solves logic based on this status.

The CPU stores the program and controls communication with all peripherals including programming devices as well as the I/O. The CPU executes the programs in an orderly manner and guarantee that I/O responds per the program. The guarantee is not trivial if one is experienced with most computer operating systems.

Outputs are connected to devices that control the process. Relays, motors, solenoids and other outputs are some examples. A pulse wave PWM is one that controls stepper motors and positional movement. The PLC program can control the status of this output and thus control the motor speed and movement.

In the figures below, the simplified PLC is shown first followed by an expanded view of the PLC. Each view shows the importance of the CPU (Central Processing Unit) as well as the interconnection of the CPU to I/O (Inputs and Outputs). Other devices interact in such a way that the program executes and solves logic in a timely manner.
The second view gives a greater detail of data flow into and out of the PLC. As the devices and programs become more involved, the flow of data must both increase while being as secure as with simple systems. All systems may not use communication modules or specialty modules. Some may use a great number. All will use some kind of programming terminal and “you” will be responsible for providing the program to run it.

Most PLCs also have a table reserved for health status of the CPU, the I/O, and the software. This table can be monitored to find if processor errors have occurred. Status tables may be ignored for the most part until something goes wrong. When an error occurs, their use is extremely important to the programmer and to the recovery of the processor. The programmer must monitor the status table in order to determine what went wrong and to restore the processor to running condition again.

Also along-side the PLC’s CPU is a watch dog timer (WDT). The WDT monitors health throughout the PLC and shuts down the I/O and program if there is a danger that the program or hardware has caused a major breakdown of the PLC’s integrity to process the program and control the process. The WDT is helpless to shut down the machine being controlled if the program in the PLC is not functioning correctly due to poor programming. Care must be taken to consider all possible conditions of a program. The proper control of the machine or process under all conditions and circumstances is critical.
How is the Program Processed in the PLC?

The program is processed in the PLC cyclically, in the following sequence:

1. First, the status is transferred from the process image of the outputs (PIQ) to the outputs, and switched on or off.
2. Then the processor -which is practically the PLC’s brain- inquires whether the individual inputs are carrying voltage. This status of the inputs is stored in the process image of the inputs (PII). For the inputs that carry voltage, the information 1 or "High“ is stored, for those that don’t the information 0 or "Low“.
3. This processor then processes the program stored in the program memory. The program consists of a list of logic operations and instructions that are processed one after the other. For the required input information, the processor accesses the PII that was entered previously, and the result of the logic operation (RLO) is written into a process image of the outputs (PIQ). If necessary, the processor also accesses other memory areas during program processing; for example, for local data of sub-programs, data blocks and flags.
4. Then, internal operating system tasks such as self tests and communication are performed. Then we continue with Item 1.

1. Transfer the status from the PIQ to the outputs.
2. Store the status of the inputs in the PII.

3. Processing the program instruction by instruction with access to PII and PIQ

4. Perform internal operating system tasks (communication, selftest, etc…)

Note: The time the processor needs for this sequence is called cycle time. In turn, the cycle time depends on the number and type of instructions and the processor capacity.
What Happens Electrically

Figure 2-2 demonstrates the flow of current in a simple circuit. The battery provides power to the lamp but is blocked in Fig. 2-2b because the switch is open. With an open switch, no current flows and the circuit is incomplete. When the switch closes, however, current flows and the lamp is illuminated (Fig. 2-2c). As simple as this circuit is, it contains the fundamental principle of input and output flow in a control circuit and the PLC.

![Simple Electrical Components](image1)

Fig. 2-2a Simple Electrical Components

![Simple Electrical Circuit (Open, No Flow)](image2)

Fig. 2-2b Simple Electrical Circuit (Open, No Flow)

![Simple Electrical Circuit (Closed, Flow)](image3)

Fig. 2-2c Simple Electrical Circuit (Closed, Flow)
The schematic for these circuits resembles the circuit below (Fig. 2-2d). Symbols have replaced their physical devices but the functionality remains the same.

Figure 2-3 shows the PLC solving logic in a similar manner to the simple circuit above. The complication of additional circuits solving logic adds to the sophistication of the circuit. This allows much more sophistication in the defining of how a circuit will perform under all conditions.
In addition to simple PLC networks such as that above, the PLC may contain network I/O allowing inputs and outputs to be communicated with at remote locations. Figure 2-4 demonstrates this type of system.

Fig. 2-4  PLC Circuit with Remote I/O

Inputs and outputs may even be communicated over wireless networks and this type of network is becoming increasingly more popular as wiring costs continue to rise and the equipment is designed for safe operation in all environments. Safe wireless networks are the latest advances in PLC equipment and offer expansion of logic into areas formerly off-limits to the PLC.

The PLC program is generated on a PC using the software STEP 7, and temporarily stored there. After the PC is connected with the TCP/IP interface of the PLC, the program can be transferred with a load function to the PLC’s memory.

The PC is no longer needed for further program processing in the PLC.
The Generic PLC

How does the PLC replace relay logic from a ladder logic diagram? Consider the following example. Pictured below is a simple generic PLC with four inputs and four outputs. One input is wired to a push button and one output is wired to an indicator light. While not exactly the same as our PLC processor, the steps of installing a program and wiring the PLC are the same.

![Generic PLC Layout](image)

Fig. 2-5  Generic PLC Layout
Notice when wiring an input and energizing the button that the green indicator light for the input comes on:

![Fig. 2-6  PLC Inputs](image)

In the program, contacts referring to the input conduct as shown below:

![Fig. 2-7  PLC Internal Logic](image)

If a program exists in the PLC similar to the following:

![Fig. 2-7  PLC Internal Logic](image)

and the Run light is on:  

![Run](image)

then the output will turn on and the light will turn on.

When the program shows the output on, the output LED turns on and the output terminal energizes the light as shown:

![Fig. 2-8a  Output Power On](image)
When the program is turned from Run to Program, the output LED turns off and the output turns off. The outputs also turn off and the Run light goes off if a fault occurs.

![Diagram of Q0 Indicator Light](image)

Fig. 2-8b  Output Power Off

Program is loaded in the PLC’s memory......

The PLC controls the process as follows: through the PLC connections called outputs, so-called actuators are wired with a control voltage of 24V for example. This allows for switching motors on and off, opening and closing valves, turning lamps on and off.

![Diagram of PLC, Outputs, and Machine](image)
How does the PLC Get the Information about the Process States?

The PLC receives information about the process from the so-called **signal transmitters** that are wired to the **inputs** of the PLC. These signal transmitters can be, for example, sensors that recognize whether a work piece is in a certain position, or they can be simple switches and pushbuttons that may be open or closed. Here, we differentiate between **break contact elements** that are closed if not operated, and **make contact elements** that are open if not activated.

```
+ 24V
Switch closed
+ 24V
Inputs
+ 0V
The inputs of the PLC record the information about the states in the process
+ 24V
Switch open
```

www.infoPLC.net
From Liptak’s *Process Control*：“Input Systems

Inputs are defined as real-world signals giving the controller real-time status of process variables. These signals can be analog or digital, low or high frequency, maintained or momentary. Typically they are presented to the programmable controller as a varying voltage, current, or resistance value.”

Analog signals include thermocouple and resistance temperature devices. Digital signals include on-off signals from relay contacts or push buttons. Signals such as flowmeters provide frequency input with the frequency varying with the flow.

Signals to the programmable controller are input from single wire devices or from parallel signals. A thumb-wheel switch or scale system can input a four-digit number from four BCD parallel digits. Many signals such as the scale system also require a synchronization signal before data can be read.

Inputs include the following types or attributes:

- DC voltage
- AC voltage, ranges of 50 Hz or 60 Hz available
- True High or True Low DC voltages
- Analog inputs, ranges 0-10V or 4-20 ma most popular
- BCD, Binary Coded Decimal
- Thermocouple
- Scale/load cells/LVDT, weight and force sensors
- RTD, Resistance Temperature Detector
- Latching
- Isolated or Common Neutral
- Intelligent (Smart with own CPU on board I/O card)
- Resolver
- Encoder
- Serial Communications Port
Some examples of PLC inputs:

![Limit Switch](Image)

**Fig. 2-9a** Picture of Limit Switch (Input)

![Selector Switch](Image)

**Fig. 2-9b** Picture of Selector Switch (Input)

![Selector Switch](Image)

**Fig. 2-9c** Picture of Selector Switch (Input)

The example of Fig. 2-9c shows a pushbutton with many contact blocks stacked on it. Is this practical? See how hard Eddie must push to energize the pushbutton.
From Liptak’s *Process Control*:

“Outputs

There are three common categories of outputs: discrete, register, and analog. Discrete outputs can be pilot lights, solenoid valves, or annunciator windows (lamp box). Register outputs can drive panel meters or displays; analog outputs can drive signals to variable speed drives or to I/P (current to air) converters and thus to control valves.”

Output signals are similar to input signals in that signals can be either analog or digital. Digital signals can be either single data or a parallel arrangement of bits. Most modules are ordered in arrangements of 4, 8, 16, or 32 devices per card.

Both input and output signals are optically isolated in designs for the US market. This protects signals from entering the interior of the PLC and allows the designer to wire circuits less carefully than in circuits without optical isolation. One main design difference between US and European PLC design is the lack of optical isolation in the European design.

Outputs include the following types or attributes:

- DC voltage
- AC voltage, ranges of 50 Hz or 60 Hz available
- Isolated or Common Source
- True High or True Low DC voltages
- Analog Output
- Serial Communications Port
- Intelligent (Smart with own CPU on board I/O card)
- Servo Controller

While I/O modules vary in type and number, recent developments have caused even these general rules to change. Distributed I/O is an example of a small number of inputs and outputs isolated at a machine that control a portion of a machine remotely from the PLC. Typical remote I/O requires a rack, power supply and a large number of cards while distributed I/O is pre-configured for only a small number of inputs and outputs. A number of advantages occur with the use of distributed I/O in that the machine can be wired and tested in one facility, broken down and shipped to a second facility, and re-connected with very little change in the wiring. This leads to quicker start-ups and cheaper overall wiring costs. Typical distributed I/O is controlled over a communications network that is daisy-chained from device to device.
Some examples of PLC outputs include:

Fig. 2-10a  Picture of Solenoid Valve (Output)

Fig. 2-10b  Picture of Relay (Output)

The relay pictured may provide input contacts but is primarily used to turn on or off various other signals from the PLC and is connected to a PLC output to accomplish this task.
The SIMATIC S7-1200

Pictured below is a S7-1200 PLC from Siemens. It is a powerful new controller with many capabilities only available in more expensive models until recently.

The S7-1200 is referred to as a micro PLC and is programmed in STEP 7 Basic, the newest software offering from Siemens. The processor has capabilities of adding additional I/O to the basic unit shown above. The processor communicates to a programming panel through an Ethernet port referred to as the PROFINET interface found on the bottom of the unit. This port offers access to other controllers, a programmer’s console and various HMI (Human Machine Interface) units.
Capabilities of this model – the CPU 1214C - include:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Memory</td>
<td>Work memory: 50 Kbytes</td>
</tr>
<tr>
<td></td>
<td>Load memory: 2 Mbytes</td>
</tr>
<tr>
<td></td>
<td>Retentive memory: 2 Kbytes</td>
</tr>
<tr>
<td>On-board digital I/O</td>
<td>14 inputs, 10 outputs</td>
</tr>
<tr>
<td>On-board analog I/O</td>
<td>2 inputs</td>
</tr>
<tr>
<td>Process image size</td>
<td>Inputs: 1024 bytes, Outputs: 1024 bytes</td>
</tr>
<tr>
<td>Bit memory (M)</td>
<td>8192 bytes</td>
</tr>
<tr>
<td>SM modules expansion</td>
<td>8 SMs max</td>
</tr>
<tr>
<td>SB expansion</td>
<td>1 SB max</td>
</tr>
<tr>
<td>CM expansion</td>
<td>3 CMs max</td>
</tr>
<tr>
<td>High-speed counters</td>
<td>6 total</td>
</tr>
<tr>
<td></td>
<td>Single phase: 3 at 100 kHz and 3 at 30 kHz</td>
</tr>
<tr>
<td></td>
<td>Quadrature phase: 3 at 80 kHz and 3 at 20 kHz</td>
</tr>
<tr>
<td>Pulse outputs</td>
<td>2</td>
</tr>
<tr>
<td>Pulse catch inputs</td>
<td>14</td>
</tr>
<tr>
<td>Timedelay/cyclic interrupts</td>
<td>4 total with 1 ms resolution</td>
</tr>
<tr>
<td>Edge interrupts</td>
<td>12 rising and 12 falling</td>
</tr>
<tr>
<td>Real time clock</td>
<td>accuracy: +/- 60 sec/mon</td>
</tr>
<tr>
<td>Execution speed</td>
<td>boolean: 0.1 microsec/instruction</td>
</tr>
<tr>
<td></td>
<td>Move Word: 12 µsec/instruction</td>
</tr>
<tr>
<td></td>
<td>Real Math: 18 µsec/instruction</td>
</tr>
<tr>
<td>Communication</td>
<td>1 Ethernet port</td>
</tr>
<tr>
<td></td>
<td>Data rate: 10/100Mb/s</td>
</tr>
<tr>
<td></td>
<td>Isolation: xfmr isolated</td>
</tr>
<tr>
<td></td>
<td>Cable type: CAT5e shielded</td>
</tr>
<tr>
<td>Connections</td>
<td>HMI: 3</td>
</tr>
<tr>
<td></td>
<td>PG: 1</td>
</tr>
<tr>
<td></td>
<td>User program: 8</td>
</tr>
<tr>
<td></td>
<td>CPU to CPU: 3</td>
</tr>
</tbody>
</table>

Sample wiring for the S7-1200 CPU is provided below (refer to the S7-1200 Systems Manual for additional details):

![Wiring Layout of S7-1200 (CPU 1214C)](www.infoPLC.net)
Specifications of the PLC include:

<table>
<thead>
<tr>
<th>Digital Inputs</th>
<th>Number of inputs</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Sink/Source</td>
<td></td>
</tr>
<tr>
<td>Rated voltage</td>
<td>24 VDC at 4 mA, nominal</td>
<td></td>
</tr>
<tr>
<td>Continuous permissible</td>
<td>30 VDC, max</td>
<td></td>
</tr>
<tr>
<td>Surge voltage</td>
<td>35 VDC for 0.5 sec</td>
<td></td>
</tr>
<tr>
<td>Logic 1 min.</td>
<td>15 VDC</td>
<td></td>
</tr>
<tr>
<td>Logic 0 max.</td>
<td>5 VDC</td>
<td></td>
</tr>
<tr>
<td>Isolation</td>
<td>500 VAC for 1 min.</td>
<td></td>
</tr>
<tr>
<td>High Speed Clock</td>
<td>Single phase rate</td>
<td>100 kHz and 30 kHz</td>
</tr>
<tr>
<td>Analog Inputs</td>
<td>Number</td>
<td>2 Voltage (single-ended)</td>
</tr>
<tr>
<td>Range</td>
<td>0 to 10 V</td>
<td></td>
</tr>
<tr>
<td>Full-scale range</td>
<td>0 to 27648</td>
<td></td>
</tr>
<tr>
<td>Overshoot range</td>
<td>27649 to 32511</td>
<td></td>
</tr>
<tr>
<td>Overflow</td>
<td>32512 to 32767</td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>10 bits</td>
<td></td>
</tr>
<tr>
<td>Max withstand voltage</td>
<td>35 VDC</td>
<td></td>
</tr>
<tr>
<td>Smoothing</td>
<td>None, weak, medium or strong</td>
<td></td>
</tr>
<tr>
<td>Noise rejection</td>
<td>10, 50, or 60 Hz</td>
<td></td>
</tr>
<tr>
<td>Impedance</td>
<td>&gt;= 100 KΩ</td>
<td></td>
</tr>
<tr>
<td>Isolation</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>3.0% - 3.5%</td>
<td></td>
</tr>
<tr>
<td>Common mode rejection</td>
<td>40 dB, DC to 60 Hz</td>
<td></td>
</tr>
<tr>
<td>Operational signal range</td>
<td>Signal plus common mode voltage less than 12V and greater than -12 V</td>
<td></td>
</tr>
<tr>
<td>Cable length</td>
<td>100 m, twisted and shielded</td>
<td></td>
</tr>
</tbody>
</table>

Descriptions of S7-1200 Modules:

- Central modules CPU with different capacity, integrated inputs/outputs and PROFINET interface (for example, CPU1214C)
Power supply PM with input AC 120/230V, 50Hz/60Hz, 1.2A/0.7A, and output DC 24V/2.5A

- Signal boards SB for adding analog or digital inputs/outputs; whereby the size of the CPU does not change.
  (signal boards can be used with the CPUs 1211C/1212C and 1214C)

- Signal modules SM for digital and analog inputs and outputs
  (for CPUs 1212C a maximum of 2 SMs can be used, for 1214C a maximum of 8)
- Communication modules CM for serial communication RS 232/RS 485
  (for CPUs 1211C/1212C and 1214C, up to 3 CMs can be used)

- Compact Switch Module CSM with 4x RJ45 socket connectors 10/100 MBit/s

- SIMATIC memory cards 2MB or 24MB for storing program data and simple CPU replacement for maintenance

**Note:** For this module M01, any CPU with integrated digital inputs and digital outputs is sufficient.
The SIMATIC Memory Card (MC) stores the program, data, system data, files and projects. It can be used for the following:

- Transferring a program to several CPUs
- Firmware update of CPUs, signal modules SM and communication modules CM

Operating Modes of the CPU

The CPU has the following operating modes:

● In the operating mode **STOP**, the CPU does not execute the program, and you can load a project
● In the operating mode **STARTUP**, the CPU performs a startup.
● In the operating mode **RUN**, the program is executed cyclically. Projects cannot be loaded in the CPU’s RUN mode.

The CPU does not have a physical switch for changing the operating mode. The operating mode (**STOP** or **RUN**) is changed by using the button on the operator panel of the software STEP7 Basic. In addition, the operator panel is provided with the button **MRES** to perform a general memory reset and displays the status LEDs of the CPU.

The color of the **status LED RUN/STOP** on the front of the CPU indicates its current operating mode.
In addition, there are the LEDs **ERROR** to indicate errors and **MAINT** to indicate that maintenance is required.

**Power Supply**

The power supply of the PLC is a very critical component of the design. If the power supply lets noise into the rest of the PLC, the PLC can be interrupted and fault. Popular among the power supply designs is the SOLA design. This design allows only the 120VAC signal into the processor from around the peak of the sine wave, eliminating any signal at the zero-voltage cross-over where noise can be a problem (a low signal/noise ratio - SNR). Peak wave forms are then transformed down to the 5-volt level and the waveform is rectified with the output signal made available to the PLC's buss.

Care must be taken to not put a SOLA rectifier in front of a PLC with the power supply circuit already present. Be careful to look at the power specifications before putting a power conditioner or uninterruptible power supply (UPS) ahead of any PLC.

Many power supplies are also used to power I/O. Most PLC manufacturers have their own power supplies and should be used as recommended with a 10% margin for error. Use the tables provided by the manufacturer to properly size the power supply before installing. Many times proper choice of power supplies will determine the lay-out of the entire PLC system.

In addition to the 115 VAC standard, the following voltages are common and must be considered in any design:

- 24 VDC
- 24 VAC
- 48 VDC

While these voltages are most common, some equipment is powered by 9-12 VDC as well as 5 VDC. Some PLCs are powered by 220 VAC. Some PLCs are powered by a floating 115 VAC system (no ground). This type of power is not common and should be considered only if the end user insists on using it.

Isolation of power for the PLC is critical. Separation of AC line voltages and low voltage circuits is required. In general, optical couplers, capacitors, transformers and relays are used to provide isolation.

Grounding of the PLC including the S7-1200 is accomplished by providing a common ground wire to
a single point near the PLC. The ground wire should be run directly to earth ground. Use NEC recommended sizes for ground wiring.

Network

Connecting to the CPU by means of TCP/IP, andResetting to Factory Setting

To program the SIMATIC S7-1200 from the PC, the PG or a laptop, you need a TCP/IP connection.

For the PC and the SIMATIC S7-1200 to communicate with each other, it is important also that the IP addresses of both devices match.

First, we show you how to set the computer’s IP address.

1. From the 'System control', call the 'Network connections'. Then, select the 'Properties' of the LAN connection (→ Start → Settings → System control → Network connections → Local Area Connection → Properties)

2. Select the 'Properties' from the 'Internet Protocol (TCP/IP)' (→ Internet Protocol (TCP/IP) → Properties)

3. You can now set the 'IP address' and the 'Subnet screen form', and accept with 'OK' (→ Use the following IP address → IP address: 192.168.0.99 → Subnet screen form 255.255.255.0 → OK → Close)

Notes on networking on the Ethernet (additional information is provided in Appendix V of the training document):

MAC address:
The MAC address consists of a permanent and a variable part. The permanent part ("Basic MAC Address") identifies the manufacturer (Siemens, 3COM, ...). The variable part of the MAC address differentiates the various Ethernet stations and should be assigned uniquely world-wide. On each module, a MAC address is imprinted specified by the factory.

Value range for the IP-address:
The IP address consists of 4 decimal numbers from the value range 0 to 255, separated by a period. For example, 141.80.0.16

Value range for the subnet screen form:
This screen form is used to recognize whether a station or its IP address belongs to the local subnetwork, or can be accessed only by means of a router.

The subnet screen form consists of four decimal numbers from the value range 0 to 255, separated by a period. For example, 255.255.0.0

In their binary representation, the 4 decimal numbers of the subnet screen form have to contain -from the left- a series of gapless values "1" and from the right a series of gapless values "0".
The values "1" specify the area of the IP address for the network number. The values "0" specify the area of the IP address for the station address.

Example:
Correct values: 255.255.0.0 Decimal = 1111 1111.1111 1111.0000 0000 0000 0000 binary
255.255.128.0 Decimal = 1111 1111.1111 1111.1000 0000 0000 0000 binary
255.254.0.0 Decimal = 1111 1111.1111 1110.0000 0000 0000.0000 binary
Wrong value: 255.255.1.0 Decimal = 1111 1111.1111 1111.0000 0000 0000 binary

Value range for the address of the gateway (Router):
The address consists of 4 decimal numbers from the value range 0 to 255, separated by a period. For example, 141.80.0.1.

Relationship of IP addresses, router address, and subnet screen form:
The IP address and the gateway address are to differ only at positions where an "0" is located in the subnet screen form.
Example: You entered the following: for the subnet screen form 255.255.255.0, for the IP address 141.30.0.5 and for the router address 141.30.128.1.
The IP address and the gateway address must have a different value only in the 4th decimal number. However, in the example, the 3rd position already differs.
That means, in the example you have to change alternatively:
- the subnet screen form to: 255.255.0.0 or
- the IP address to: 141.30.128.5 or
- the gateway address to: 141.30.0.1

The SIMATIC S7-1200 IP address is set as follows:

4. Select the "Totally Integrated Automation Portal"; it is called here with a double click (→ Totally Integrated Automation Portal V10)
5. Then, select the ‘Project View’. (→ Project view)

6. Next, in project navigation, select under ‘Online accesses’ the network card that was already set beforehand. If you click here on ‘Update accessible stations’ <<Erreichbare Teilnehmer aktualisieren>>, you will see the the MAC address of the connected SIMATIC S7-1200. Select ‘Online & Diagnosis’. (→ Online accesses → … Network Connection → Update accessible stations → MAC= ….. → Online & Diagnosis)
Note: If an IP address was set previously at the CPU, you will see this address instead of the MAC address.
7. Under ‘Functions’ you will see the item ‘Assign IP address’. Here, enter ‘IP address’ and ‘Subnet screen form’. Then, click on ‘Assign IP address’, and your SIMATIC S7-1200 will be assigned this new address (→ Functions → Assign IP address → IP address: 192.168.0.1 → Subnet screen form: 255.255.255.0 → Assign IP address)
8. Under ‘Functions’, select 'Reset to factory settings'. Keep this setting on 'Retain IP address' and click on 'Reset'. (→ Functions → Reset to factory settings → Retain IP address → Reset)

9. Confirm the query whether you want to go through with a reset to the factory setting with 'OK' (→ OK)
Why Use PLCs

Believing that it is advantageous to use the PLC by now should be complete but let’s review again why PLCs are better than wired relay logic:

<table>
<thead>
<tr>
<th>PLC:</th>
<th>Hard-wired Logic:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiring required only for inputs and outputs</td>
<td>Wiring required for all devices including all logic</td>
</tr>
<tr>
<td>Panels tend to be smaller and cost less</td>
<td>Panels are larger and tend to cost more</td>
</tr>
<tr>
<td>Wiring changes can be handled via programming on a terminal</td>
<td>Wiring changes must be done wire by wire</td>
</tr>
<tr>
<td>Fewer devices are required resulting in fewer failure points</td>
<td>Typically more points capable of failure present in hard-wired panel logic</td>
</tr>
<tr>
<td>Linking to computers and HMI devices relatively easy</td>
<td>Linking to computers impossible. No HMI devices usually allowed</td>
</tr>
</tbody>
</table>