An educational software project in the field of process control

Michael Ritzschke
Institute of Informatics
Humboldt University
Berlin
Contents

- Motivation and goals of the project
- Configuration of the programmable logic controller (PLC) SIMATIC
- Steps to develop a PLC program
- Outlook: Automatically generate a PLC program
Process control with programmable logic controller (PLC)

- PLC: digital computer used for automation of industrial processes (SIMATIC S7-200 ... -400, SIMATIC S5)

- For instance control of machinery on factory assembly lines

- PLC are special PC’s for using in severe conditions (dust, moisture, heat, cold, etc)

- PLC have the facility for extensive input/output arrangements, connecting the PLC to sensors and actuators with different modules
Process control with PLC – an industrial example

http://www.insystems.de

2008

InSystems on Exhibitions
July 2008

InSystems will be present on two photovoltaic exhibitions this year.

Fully automatic pallet storage system

M. Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
Process control of a production model in our CAM_Lab at Humboldt University
The production model

High-bay storage

Supply parts storage (gibs)

Machines

Conveyors and rotary tables

Assambly robot

M. Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
Goals of the project – what our students can learn

- Architecture of PLC and their configuration
- Why PLC is used in process control
- Develop different models from a production process (state machines, Petri Net etc)
- Write a control program
- Test the program with a simulation tool
- Test the program with our production model
- Find mistakes and eliminate these
- Work with different development tools
Contents

- Motivation and goals of the project
- Configuration of the programmable logic controller (PLC) SIMATIC
- Steps to develop a PLC program
- Outlook: Automatically generate a PLC program
Hardware configuration of the PLC SIMATIC

Basis: Numbers and types of the sensorsignals and the actuatorsignals

- **84 Input-Signals**
  - 50 Switches, 28 Metal detectors, 1 photoelectric barrier → binary signal
  - 1 Potentiometer → analogue signal
  - 4 incremental position encoder → digital signal

- **88 Output-Signals**
  - 84 Motors, 4 Magnets → binary signal
1995: Use a catalogue

M. Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
Today: STEP 7-Lite
(via internet freely available Software /1/)

Easy configure the hardware with drag&drop

M.Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
S135U: Our configuration

- 3 CPU’s
- 1 Coordination
- 4 Digit input
- 4 Digit output
- 1 Analogue input
- 2 Special modules with own CPU (Incrementer)
- 2 Electricity supply
Contents

- Motivation, goals of the project
- Configuration of the PLC SIMATIC
- Steps to develop a PLC program
  - Develop models of the Process
  - Write the program
  - Test the program with simulation PLC
  - Test the program in practice
- Outlook: Automatically generate a PLC program
Factory door – a teaching example

Static model:
Layout of a factory door

- 2 sensors (opened / closed)
- actuator: electrical motor
- 2 key switches, 1 telekey
- Flashing light (door moves)
- Broken photoelectric key → door opens
- Door closes after 20 sec
Signal-Table

<table>
<thead>
<tr>
<th>Signal</th>
<th>I/O</th>
<th>Kennz</th>
<th>Logische Zuordnung</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schlüsselschalter außen</td>
<td>E 1.0</td>
<td>S1</td>
<td>Betätigt S1=1</td>
</tr>
<tr>
<td>Schlüsselschalter innen</td>
<td>E 1.1</td>
<td>S2</td>
<td>Betätigt S2=1</td>
</tr>
<tr>
<td>Funkempfänger</td>
<td>E 1.2</td>
<td>S3</td>
<td>Code empfangen S3=1</td>
</tr>
<tr>
<td>Endschalter Tor zu</td>
<td>E 1.3</td>
<td>S4</td>
<td>Tor zu S4=1</td>
</tr>
<tr>
<td>Endschalter Tor auf</td>
<td>E 1.4</td>
<td>S5</td>
<td>Tor auf S5=1</td>
</tr>
<tr>
<td>Lichtschranke</td>
<td>E 1.5</td>
<td>L1</td>
<td>Unterbrochen L1=0 (!) → Öffner</td>
</tr>
<tr>
<td>Blinklicht</td>
<td>A 1.0</td>
<td>H1</td>
<td>Licht an H1=1</td>
</tr>
<tr>
<td>Motorschütz Tor auf</td>
<td>A 1.1</td>
<td>K1</td>
<td>Schütz angezogen K1=1</td>
</tr>
<tr>
<td>Motorschütz Tor zu</td>
<td>A 1.2</td>
<td>K2</td>
<td>Schütz angezogen K2=1</td>
</tr>
</tbody>
</table>

M. Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
Static model: Finite state machine

Mealy-Automat \( y = f(e,s) \)

- **Tor ist geschlossen**
  - \( S1 \lor S2 \lor S3 \)
  - \( K1 := 1 \)
  - \( H1 := 1 \)

- **Tor wird geöffnet**
  - \( \overline{L1} \)
  - \( K2 := 0 \)
  - \( K1 := 1 \)

- **Tor ist offen**
  - \( S5 \)
  - \( K1 := 0 \)
  - \( T1 := 20\text{sek} \)

- **Tor wird geschlossen**
  - \( S4 \)
  - \( K2 := 0 \)
  - \( H1 := 0 \)
  - Zustandsübergang
  - Bedingung
  - Aktion
  - Zustand

M. Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
Dynamic model: Petri Net

Software /2/: Visual Object Net
Some student solutions

M.Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
Editing the program
(IEC 61131-3: Function block diagram and statement list)

Software /3/: S5/S7 for Windows

M.Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
Simulation and test the program with simulation PLC /3/
Student project: Development of a PLC program for a part of the production model

M. Ritzschke, DAAD Workshop Durres, September, 8th - 13th, 2008
Example:
Cycle transport pallets with conveyors

Static model: Layout
The program as a paper (statement list) and ...
The program
... in action
Contents

- Motivation, goals of the project
- Configuration of the programmable logic controller (PLC) SIMATIC
- Steps to develop a PLC program
- Outlook: Automatically generate a PLC program
Model:
Signal interpreted Petri Net (SIPN)

\[
\begin{align*}
E & \rightarrow \text{Control system} \\
A & \rightarrow \text{Process} \\
E & : \text{sensor signals} \\
A & : \text{control signals}
\end{align*}
\]

\[
\begin{align*}
A4.2 &= 0 \\
A4.3 &= 0 \\
E5.3 &= 1 \\
E5.4 &= 0 \\
A4.4 &= 1
\end{align*}
\]

SIPN = Petri Net + input signals + output signals
Example: Control of 3 conveyors
SIPN for control of 3 conveyors /4/

Firing condition: pallet on conveyor 9, conveyor 10 free

Output signals: 2 motors work parallel

Firing condition: pallet on conveyor 10
PROGRAM Main
VAR
PV_1 : BOOL := TRUE ; (* M0.0 *)
PV_2 : BOOL := FALSE ; (* M0.1 *)
PV_3 : BOOL := FALSE ; (* M0.2 *)
PV_4 : BOOL := FALSE ; (* M0.3 *)
PCK_0 : BYTE := 1 ; (* Packed place variable *)
END_VAR
VAR_GLOBAL
E5.3 at %IXO.0 : BOOL;
E5.4 at %IXO.1 : BOOL;
E5.5 at %IXO.2 : BOOL;
A4.4 at %XQO.0 : BOOL;
A4.6 at %XQO.1 : BOOL;
A5.0 at %XQO.0 : BOOL;
END_VAR

(***** Transition T1 *****)
l_0:  LD  PV_1  (* pre place M0.0 *)
        ANDN PV_3  (* post place M0.2 *)
        ANDN PV_2  (* post place M0.1 *)
                JMPCN l_1
                AND  (E5.3
                        ANDN E5.4
                        )
                JMPCN l_1
                    R  PV_1  (* pre place M0.0 *)
                    S  PV_3  (* post place M0.2 *)
                    S  PV_2  (* post place M0.1 *)
                (* Update packed place variables *)
                    LD  PCK_0
                    AND  254
                    OR  6
                    ST  PCK_0

(***** Transition T2 *****)
l_1:  LD  PV_1  (* pre place M0.0 *)
        ANDN PV_3  (* post place M0.2 *)
        ANDN PV_4  (* post place M0.3 *)

(***** Transition T3 *****)
l_3:  LD  FV_3  (* pre place M0.2 *)
        AND  FV_4  (* pre place M0.3 *)
        ANDN FV_1  (* post place M0.0 *)
                JMPCN l_4
                AND  E5.5
                JMPCN l_4
                    R  FV_3  (* pre place M0.2 *)
                    R  FV_4  (* pre place M0.3 *)
                    S  FV_1  (* post place M0.0 *)
                (* Update packed place variables *)
                    LD  PCK_0
                    AND  243
                    OR  1
                    ST  PCK_0

(***** Place M0.0 *****)
l_4:  LD  FV_1
        JMPCN l_3
        R  A4.4
        R  A4.6
        R  A5.0

(***** Place M0.1 *****)
l_5:  LD  FV_2
        JMPCN l_6
        S  A4.4

(***** Place M0.2 *****)
l_6:  LD  FV_3
        JMPCN l_7
        S  A4.6

(***** Place M0.3 *****)
l_7:  LD  FV_4
        JMPCN l_8
        S  A5.0

l_8:  RET
END_PROGRAM
Summary and references

References

/1/ STEP 7-Lite. Siemens AG, http://support.automation.siemens.com
/2/ Drath, R., Visual Object Net++, Technical University Ilmenau