Control of Hydraulic Pulse System Based on the PLC and State Machine Programming †

Juraj Pančík 1,* and Pavel Maxera 2

1 Institute of Forensic Engineering, Brno University of Technology, Purkyněova, 464/118, Brno 61200, Czech Republic
2 Institute of Forensic Engineering, Brno University of Technology, Purkyněova, 464/118, Brno 61200, Czech Republic; pavel.maxera@usi.vutbr.cz
* Correspondence: juraj.pancik@gmail.com; Tel.: +421-908-963-289

Received: 17 September 2018; Accepted: 14 November 2018; Published: 20 November 2018

Abstract: In this paper, we deal with a simple embedded electronic system for an industrial pneumatic–hydraulic system, based on a low-cost programmable logic controller (PLC) and industrial electronic parts with 24 V logic. The developed system is a hydraulic pulse system and generates a series of high-pressure hydraulic pulses with up to a max. 200 bar output pressure level and with up to a max. 2 Hz output hydraulic pulses frequency. In this paper we are describing requirements, the concept of the embedded control system in a diagram, security features and its industrial network connectivity (CAN bus, MODBUS). In description of the software solution we describe the implementation of the program threads approach in this low-cost PLC. The PLC programming with threads generate two layers of services—physical and application layer, and as a result, the threads create the main control state machine. In conclusion, we describe the calibration method of the system and the calibration curves. For further study we offer readers the full programming code written in sequential function charts to be used as PLC language. The cost of the described industrial networked control system with industry standard optoelectronic insulated interfaces and certified industrial safety relay does not exceed €1000 Euros.

Keywords: PLC programming; hydraulic pulse system; state machine programming

1. Introduction

Our built-in embedded system for the pneumatic–hydraulic pulse system is based on the use of a modern low-cost programmable logic controller (PLC). A modern PLC allows you to program parallel running programs (threads), each of them being programmed for PLC programmers well-known ladder diagrams or as functional block diagrams [1,2]. In our work, we prepared for the PLC robust, external error-free control program, so as to control the series high pressure hydraulic pulses based at one PLC. Each thread represents a separate task and independent state of system. Parallel running threads cooperate with each other and this set of independent threads creates a unique reactive control state machine [3,4].

2. Requirements for Pneumatic–Hydraulic Pulse System

Since we have limited space in this contribution, we are only trying to describe the requirements for the overall machine and requirements for creating a specific series of impulses.

The system controls up to four independent series, each of them defined by four adjustable parameters; the number of pulses (N), the maximum hydraulic pressure (p), the hydraulic pulse
duration (T1) and the duration of the pause between pulses (T0). Each WAGON of this TRAIN has four passengers: N, p, T1, T2, see Figure 1, where WAGONs build a so-called “PNEUMATIC–HYDRAULIC PULSE TRAIN”. The pneumatic–hydraulic pulse train can cyclically repeat its defined circle path by the number of times which is set in the TRAIN SESSION number. The number of WAGONs may be set by the user from 1 to 4. The embedded control system allows each user to set the output of the hydraulic pressure in bars within the range 0–200 bar, the number of WAGONs, and number of cyclic reruns of the train’s path. The control system must allow system calibration (dependence of the output hydraulic pressure in units of [bar] at the user’s desired hydraulic pressure in [ADU] units (ADU unit = Analog Digital Unit)). It must ensure the generation of system and error messages.

Figure 1. Pneumatic–hydraulic pulse train-schematic represents of cyclic repeating sequence pulses for hydraulic pulse system as cyclic paths of train with four wagons, each with four passengers.

3. Hardware Solution

3.1. Pneumatic–Hydraulic Parts

The schematic diagram of the developed pneumatic–hydraulic system with PLC based control system is shown in Figure 2 and a photo of the prototype is shown in Figure 3. The proportional pneumatic valve regulates input air pressure. Regulated air is then fed into pneumatic four-way valve circuit and it controls the two-position pneumatic pump. This pneumatic circuit controls the two-direction movement of the hydraulic cylinder with, resulting amplified hydraulic pressure as its output [5].
Figure 2. Schematic diagram of the developed pneumatic–hydraulic system.
3.2. PLC Based Control System

The embedded control system was implemented on a cheap programmable logic controller type (PLC) Nanoline from Phoenix Contact [6] and on additional PLC’s modules (analog input–output module, LCD display and keypad user interface module, USB, RS232 and Ethernet communication module). The PLC works with industrial logic levels and they both have two levels 0 and 24 V; PLC’s analog outputs are in the range 0–10 V, and PLC’s analog inputs are in range 0–5 V. The used PLC Nanoline is built around an ARM 32 processor and its operating system allows for the running of several independent program threads simultaneously. The run of each thread is conditional on the value of the hardware input signal or by the program setting of value of the internal program flag. This interesting ability of the PLC’s operating system creates a challenge for us to generate a master program for control of hydraulic pulses based on states machines.

3.3. User Interface Elements

HMI (human–machine interface) of the pneumatic–hydraulic pulse system consists of keypad, LCD display (4 rows × 20 characters) and few switches. The manual switch strains between two basic modes of equipment (MANUAL mode and AUTO mode), the potentiometer serves for manual adjustment of output hydraulic pressure, the button allows for starting the series hydraulic pulsation (PULSE mode), another button serves to stop the pulses (the RESET button), the security (or emergency) button serves to turn off the power of the appliance (EMERGENCY). The required values for WAGON and TRAIN parameters and the choosing of other equipment modes (e.g., calibration of the system) are entered from the PLC build keypad.

3.4. Electrical Parts of System

Electrical parts of system consist a 24 V power supply, a safety relay, electrical and optical relays, switches, terminals and wires as is seen in Figures 2 and 3.
3.5. Safety Measures

The basis of ensuring the safety operation of high pressure pneumatic–hydraulic pulse system is the certified safety relay [7]. The safety relay can detect system errors that switch off the power for the whole electrical power part of the equipment (but not for control PLC). Key safety sensors are connected in the series. In the case of detection of problems by dedicated sensors with any media (hardware signals AIR or LIQUID) or detection of mechanical problems (hardware signals as DISTANCE ERROR or COVER EMERGENCY SWITCH), the series of connected sensor’s contacts is interrupted, the safety relay responds and sets the signal ERROR and switches off the power for the hydraulic–pneumatic pulse system in order to automatically reduce the output hydraulic pressure to minimum. The mechanical error with name DISTANCE ERROR is generated by a mechanical contact switch mounted near the hydraulic cylinder and serves to indicate an attempt to generate high hydraulic pressure. To ensure of the safe operation of the high pressure pneumatic–hydraulic pulse system, we used a certified safety relay. The safety relay can, in the case of detecting the system errors, switch off the power for the whole electrical power part of the equipment (except of power for control PLC).

3.6. Network Connectivity of the Pneumatic–Hydraulic System

Network connectivity of our control system is ensured with the PLC expansion Ethernet communication module [8]. This expansion module allows for external control of all program elements in the PLC program (e.g., PLC’s signals, flags, registers ...) via industrial standard MODBUS protocol (via TCP-IP protocols). For our special purpose, where the pneumatic–hydraulic pulse system is a part of a test bench, we have developed the communication embedded server—a converter between the CAN bus (CAN is used by test bench’s protocol) and RS232 (as physical base for MODBUS protocol). The CAN converter consists of the hardware module OLIMEX occupied with 8bit ATMEL AVR-CAN microcontroller [9]. The developed protocol and BUS converter allows us to control and monitor pneumatic–hydraulic pulses system, even though the CAN bus.

4. Software Solution

4.1. Basic PLC Functions

The programming approaches for modern PLC are described in standards and books, for example by the authors of [10,11]. The ladder diagram or graphic function block diagram programming approach is still the focus of PLC’s programmers. The IEC 61131-3 (IEC 1131-3: The International Programmable Controller Language Standard) is the third part (of 10) of the open international standard IEC 61131 for programmable logic controllers and was first published in December 1993 by the IEC [6]. The current (third) edition was published in February 2013. Part 3 of IEC 61131 deals with the basic software architecture and programming languages of the control program within the PLC. It defines two graphical and two textual programming language standards:

- Ladder diagram (LD), graphical
- Function block diagram (FBD), graphical
- Structured text (ST), textual
- Instruction list (IL), textual (deprecated in 3rd edition of the standard)
- Sequential function chart (SFC), has elements to organize programs for sequential and parallel control processing (IEC 1131-3: The International Programmable Controller Language Standard and)

The modern low-cost PLC based on powerful ARM architecture such as Nanoline from Phoenix Contact allows for the running of several program threads. It gives to the PLC programmer the possibility to program the main control program without creating one long and complicated PLC program. Each independent PLC program thread can be initiated by hardware signal(s) (the PLC use
24 V logic at input) or by internal programming flag with boolean data type (it can be set by another PLC program thread). Each PLC program thread can be programmed with ladder diagram or graphic functional block diagram programming approach and the PLC programmer may not to know about the concurrent program techniques or about the use of special concurrent program library for PLC.

4.2. List of Hardware Inputs and Outputs for PLC

The list of all digital and analog inputs and outputs used in the project is in the Table 1. Each hardware input signal (ERROR, WAIT, MANUAL, AUTO, PULSE and RESET) is processed in the PLC by individual program thread (see Table 2). These signals are grouped in the so-called physical layer. Some hardware signal inputs are used for indicating the error’s cause in the safety power shutdown of the pneumatic–hydraulic system via safety relay (Figure 2).

<table>
<thead>
<tr>
<th>n.</th>
<th>Name of Hardware Signal</th>
<th>Signal Direction/Digital/Analog</th>
<th>Signal is Processed in the Independent Thread</th>
<th>Description of Physical Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ERROR</td>
<td>I0-input</td>
<td>Physical layer 4_ERROR</td>
<td>From safety relay</td>
</tr>
<tr>
<td>2</td>
<td>MANUAL</td>
<td>I1-input</td>
<td>Physical layer 1_MANUAL</td>
<td>User input, manual control</td>
</tr>
<tr>
<td>3</td>
<td>AUTO</td>
<td>I2-input</td>
<td>Physical layer 2.AUTO</td>
<td>User input, pulse gen.</td>
</tr>
<tr>
<td>4</td>
<td>PULSE</td>
<td>I3-input</td>
<td>Physical layer 3_PULSE</td>
<td>User input, start of pulsing</td>
</tr>
<tr>
<td>5</td>
<td>RESET</td>
<td>I4-input</td>
<td>Physical layer 5_RESET</td>
<td>User input, stop of pulsing</td>
</tr>
<tr>
<td>6</td>
<td>AIR_ERROR</td>
<td>I5-input</td>
<td>Application layer 4_State_Func</td>
<td>From pressure air tank switch</td>
</tr>
<tr>
<td>7</td>
<td>LIQUID_ERROR</td>
<td>I6-input</td>
<td>Application layer 4_State_Func</td>
<td>From hydraulic tank switch</td>
</tr>
<tr>
<td>8</td>
<td>DISTANCE_ERROR</td>
<td>I7-input</td>
<td>Application layer 4_State_Func</td>
<td>From distance contact switch</td>
</tr>
<tr>
<td>9</td>
<td>PNEU_PUMP</td>
<td>Q0-output</td>
<td>1_State_Func and 3_State_Func</td>
<td>For pneumatic pump control</td>
</tr>
<tr>
<td>10</td>
<td>PULSE_RELAY</td>
<td>Q1-output</td>
<td>Application layer 3_State_Func</td>
<td>For pulse control generation</td>
</tr>
<tr>
<td>11</td>
<td>HYDR_PRES_SENSOR</td>
<td>A10-input</td>
<td>All application layers</td>
<td>From hydraulic pressure sensor</td>
</tr>
<tr>
<td>12</td>
<td>MANUAL_PRES_SET</td>
<td>A11-input</td>
<td>Application layer 1_State_Func</td>
<td>From user man. potentiometer</td>
</tr>
<tr>
<td>13</td>
<td>PNEUMATIC_VALVE</td>
<td>AU0-output</td>
<td>1_State_Func and 3_State_Func</td>
<td>For proportional valve control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n.</th>
<th>Name of Thread</th>
<th>Signal Condition for Perform of the Thread</th>
<th>Program Switches the Flag to ON</th>
<th>Conditionally can Set also</th>
<th>Description of Program Thread</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4_ERROR</td>
<td>Signal ERROR = ON</td>
<td>State_FLAGS_4_ERROR</td>
<td>PNEU_PUMP(OFF) PULSE_RELAY(OFF)</td>
<td>Safety relay output processing</td>
</tr>
<tr>
<td>2</td>
<td>0_WAIT</td>
<td>Signal ERROR = ON</td>
<td>State_FLAGS_0_WAIT</td>
<td></td>
<td>Wait state</td>
</tr>
<tr>
<td>3</td>
<td>1_MANUAL</td>
<td>Signal MANUAL = ON AND ERROR = OFF</td>
<td>State_FLAGS_1_MAN</td>
<td>State_FLAGS_0_WAIT</td>
<td>User input, manual control mode</td>
</tr>
<tr>
<td>4</td>
<td>2_AUTO</td>
<td>Signal AUTO = ON AND ERROR = OFF</td>
<td>State_FLAGS_2_AUTO</td>
<td>State_FLAGS_0_WAIT</td>
<td>User input, pulse generation mode</td>
</tr>
<tr>
<td>5</td>
<td>3_PULSE</td>
<td>Signal PULSE = ON AND ERROR = OFF</td>
<td>State_FLAGS_3_PULSE</td>
<td>State_FLAGS_0_WAIT</td>
<td>User input, start of pulsing</td>
</tr>
<tr>
<td>6</td>
<td>5_RESET</td>
<td>Signal RESET = ON AND ERROR = OFF</td>
<td>State_FLAGS_2_AUTO</td>
<td>START_BUTT_ON</td>
<td>User input, stop of pulsing</td>
</tr>
</tbody>
</table>

4.3. Physical Layer Threads

Threads of the physical layer (Table 2) run in the PLC’s operating system after their initializing by hardware input signals. The physical layer ensures processing of hardware signals and as a result, this layer sets outputs flags, which are processed in the application layer. These output flags manage transitions between states in the master control state machine (literally by start or by stop of the respective threads in the application layer). These program threads also process aforementioned inputs hardware signals and set of hardware output signals by aligning the required settling time or also, the next set of parameters of elements of pneumatic or hydraulic hardware (e.g., delay of relays, setting time of input valve, etc.).
4.4. Application Layer Threads

The running of the application layer thread (listed in Table 3) begin by changes of program flags (one or two flags, then in AND or in OR mutual combination) which are received from the physical layer. Some threads in application layer create of individual states in the master control state diagram (Figure 3). In addition, these threads enable the realization of sub states. For example, thread 2_State_Func can call of sub state 6_State_Func. This thread serves for the manual input of parameters in the hydraulic pulse train. Next threads also underpin subroutine for user input interaction such the 22_State_Func. This sub state provides the possibility of the manual input of calibration constants.

<table>
<thead>
<tr>
<th>n.</th>
<th>Name of Thread</th>
<th>Flag Condition for Perform of the Thread</th>
<th>Description of Program Thread, see Figure 4 Master Control State Diagram for Hydraulic Pulse System</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0_state_Func</td>
<td>State_FLAGS_0_WAIT – ON AND ERROR – OFF</td>
<td>WAIT STATE, Initialization of calibration constants, zeroing of the ATE pressure sensor</td>
</tr>
<tr>
<td>2</td>
<td>1_state_Func</td>
<td>State_FLAGS_1_MANUAL – ON AND ERROR – OFF</td>
<td>MANUAL STATE, manual control of output hydraulic pressure via manual potentiometer</td>
</tr>
<tr>
<td>3</td>
<td>2_state_Func</td>
<td>State_FLAGS_2_AUTO – ON AND ERROR – OFF</td>
<td>AUTO STATE, setting of parameters for hydraulic pulse train: WAGON parameters numbers of TRAIN WAGON and number of TRAIN SEASON, manual calibration constants inputs (sub state: 22_state_Func)</td>
</tr>
<tr>
<td>4</td>
<td>3_state_Func</td>
<td>State_FLAGS_3_PULSE – ON AND ERROR – OFF</td>
<td>PULSE STATE, performing of hydraulic pulsation according of parameter settings, after finish of pulsing goes automatic to AUTO MODE (FLAGS_2_AUTO – ON, FLAGS_3_PULSE – OFF)</td>
</tr>
<tr>
<td>5</td>
<td>4_state_Func</td>
<td>State_FLAGS_4_ERROR</td>
<td>ERROR STATE, Safety relay determined, determination of the error source</td>
</tr>
<tr>
<td>6</td>
<td>5_state_Func</td>
<td>State_FLAGS_5</td>
<td>Sub state of PULSE STATE, measurement of the maximum of the hydraulic pulse peak during PULSE MODE (during the pause time and during the pulse time)</td>
</tr>
<tr>
<td>7</td>
<td>6_state_Func</td>
<td>State_FLAGS_21_QSET</td>
<td>Sub state of AUTO STATE, user setting of parameters for four TRAIN (T0, T1, COUNT, PRESSURE)</td>
</tr>
<tr>
<td>8</td>
<td>7_state_Func</td>
<td>State_FLAGS_7</td>
<td>Sub state of PULSE STATE, actual modification of parameters after one TRAIN SESSION</td>
</tr>
</tbody>
</table>

4.5. Description of the Master Control State Machine

The main control state machine (Figure 4) for hydraulic pulse system consists of 7 separate threads running in the application layer, each thread represents the individual state of pneumatic–hydraulic pulsing system and is described in Table 3. Transitions between states in the application layer is controlled by the program flags and they are isolated from the hardware inputs signals (Table 1) processed by the physical layer (Table 2). This solution allows for the reliable function of high-pressure hydraulic pulse system, with reliable answers to user inputs or to the error status. We observed the reliable automatically generated ends of each hydraulic train pulsation cycle.

4.6. Auxiliary Functions

The control of pulses sequences according to saved parameters in the hydraulic pulse train ensures thread 7_state_Func in the application layer, which is called from the state PULSE (3_state_Func). The thread 6_state_Func ensures the users inputs for controlling of the actual content of the hydraulic pulse train and is called from AUTO STATE (2_state_Func).

4.7. Calibration of the System

The thread 6_state_Func in the application layer ensures the manual input of calibration constants from the user. The reason for the calibration can be described as follows: user works at input or at output with pressure units in [bar], but the PLC works with an internal representation of pressure in ADU units (12 bit). The user can perform the calibration process in MANUAL STATE which means static mode (no pulse condition, manually controlled output hydraulic pressure). The user obtains, during the calibration procedure, two calibration curves. The first calibration curve is dependence between the output hydraulic pressure (measured with portable calibrated hand-held hydraulic pressure measurement equipment from WIKA [10]) and the output voltage in 12bit ADU (measured via PLC itself) from the sensor output hydraulic pressure ATE PS60. ATE PS60 is a commercial produced hydraulic pressure sensor used in the cars, the manufacturer is ATE. In our case, in calibration process
we obtained a linear relationship, which leads to the linear Equation (1) derived by regression analysis from the graph of the measured values:

\[ PWIKA \ [\text{bar}] = A_{ATE\ PS60} \times PATE\ PS60 \ [\text{ADU}] + B_{ATE\ PS60} \] (1)

By the calibration-determined value of the constant \( A_{ATE\ PS60} \) for pressure sensor ATE PS60 is 0.1021 [bar/ADU] and the value of the constant \( B_{ATE\ PS60} \) is equal to 4.8074 [bar] (Figure 5).

\[ PWIKA \ [\text{bar}] = A_{SYS} \times PSetInputValve \ [\text{ADU}] + B_{SYS} \] (2)

where the value of the constant \( A_{SYS} \) is 0.053 [bar/ADU] and the value of the constant \( B_{SYS} \) is 0.0626 [bar] (Figure 6). Both linear Equations (1) and (2) provide of four calibration constants, which user can enter in to the control system (in the thread 6_State_Func). Then, after the static calibration of pneumatic hydraulic pulse system, the user input entered value are in bar units obtained at the output of pneumatic–hydraulic system, as is required in both states—MANUAL and PULSE STATE.
We developed an embedded control for pneumatic–hydraulic system which produces a series of high-pressure hydraulic pulses (up to 200 bar). The pneumatic–hydraulic system can be calibrated, it is networked in industrial networks (MODBUS over TCP/IP and CAN bus) and generates accurately hydraulic pulses (time and pressures characteristics). The system proved to be reliable. This hydraulic impulse unit is a part of a test machine and it can be used in the industry for hydraulic parts dynamic testing by series of pulses with maximal frequency pulses 2 Hz. From the point of view of software
architecture, the embedded control system is based on a low-cost PLC architecture and control program is a set of parallel program threads. Each programming thread can be identified as one state in a master control state diagram. The PLC multi-thread program is not designed as one large and opaque linear program, but rather as a group of mutually communicating PLCs programs, which ultimately forms one master control state diagram. The cost of the described industrial networked control system with industry standard optoelectronic insulated interfaces and certified industrial safety relay does not exceed €1000 Euros.

**Supplementary Materials:** The following are available online at [http://www.mdpi.com/2411-9660/2/4/48/s1](http://www.mdpi.com/2411-9660/2/4/48/s1), Source code S1: PLC software project (NanoNavigator IDE).

**Author Contributions:** Conceptualization, J.P.; Data curation, P.M.; Methodology, J.P. and P.M.; Software, J.P.; Validation, J.P. and P.M.; Writing – original draft, J.P.; Writing – review & editing, J.P.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**


© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).