

User Manual

DCU
CONTROL SYSTEM
PLATFORM
-
INSTALLATION
AND
CONFIGURATION

DCU  **PROSYSTEMY**
CONTROL SYSTEM

User Manual

DCU Control System Platform – Installation and configuration

04/10/2024

Contents

1 DCU Control System Platform installation.....	3
1.1 Platform installation without installer.....	3
1.2 Configuring DCU platform.....	4
1.3 Setting Linux visualization server.....	4
2 Special settings.....	6
2.1 Changing MySQL data directory.....	6
2.2 Change MySQL root password.....	7
2.3 Remove Unlock keyring popup in web-browser.....	7
3 Description of visualization server configuration tables and setting Alarm values.....	9

1 DCU Control System Platform installation

Complete DCU Control System Platform can be installed on any computing device with any of the following operating system:

- Windows XP or higher,
- Mac OS X,
- Linux,

DCU platform software package contains the following software tools:

- JAVA Runtime Environment
- MySQL database
- Apache webserver
- PHP
- PHPMyAdmin tool for MySQL database management
- Scilab computational platform
- User Command Center – DCU platform service
- DCU Design Tool – design tool for management of control system projects

For Windows and Linux Ubuntu operating systems installers are available that will install automatically required packages. Linux installer requires connection to internet.

On Windows, run installer by double-click and follow instructions.

On Linux Ubuntu:

1. Unpack .tar.gz installer to your home directory
2. open terminal
3. go into installer directory (cd directory)
4. run installer as a root (type: sudo ./dcu_install.sh)

Android 4.1 (Jelly Bean) or higher operating system is supported for server or operator panel deployment. It means that any Android device can serve as an operator panel or server if properly installed. The android application DCU User Command Center must be installed.

1.1 Platform installation without installer

The software tools may be installed separately without installer. The installation procedure for Winows, Linux, Mac OS X is as follows:

1. Install Java Runtime Environment (JRE)
2. Install Apache-MySQL-PHP (WAMP/LAMP/MAMP) package.

3. Download and unzip User Command Center (communication service) and DCU Design Tool (design package) from www.prosystemy.sk.

To start DCU Control System Platform

1. Start AMP package if not running automatically (according to installation).
2. Start ucc.jar, enter password to MySQL database and click “Start” button to start UCC service. Running UCC for the first time will create configuration database for defining standard visualization.
3. Start Scilab or Matlab computational platform.
4. Change directory in Scilab/Matlab console to DCU design tool and type 'exec dcudesign.sci' (in case of Scilab) or 'dcu' (in case of Matlab) to start DCU Design tool.

1.2 Configuring DCU platform

Some configuration parameters are advised to set to obtain optimal performance of DCU platform:

1. Set memory limit of Scilab to approximately 70% of available RAM: Start Scilab, go to menu Edit-> Preferences and set Java Heap Memory to about 70% of RAM size (for example with 8GB RAM available set value to 6GB).
2. Modify parameters of PHP: Inside php.ini file (initialization file of PHP) set the parameters
 1. max_input_vars = 10000,
 2. memory_limit = 512M,
 3. max_execution_time = 180,
 4. display_errors = On.
 5. opcache.enable=0

This will allow large number of parameters to be transferred from/to webpage and to use more memory (needed for display of graphs with large number of samples).

3. Optionally inside my.ini (MySQL initialization file) check/increase key_buffer_size, table_cache parameters to improve database performance.

1.3 Setting Linux visualization server

To set the visualization server for a specific procejt follow this simple procedure:

1. Set computer to start automatically after power failure. Also operating system needs to start automatically without waiting on user boot selection. Example of setting these parameters for NUC mini PC with Linux Ubuntu is shown below.
 1. Press F2 during minPC startup to call Bios. In Bios go to Advanced

Setup - Power and for the parameter "After Power Failure" select the option "Power ON".

2. Open grub file as a root by typing in terminal : `sudo gedit /etc/default/grub`
3. Set parameter `GRUB_TIMEOUT=3` (wait only 3 seconds with boot menu)
4. Add parameter `GRUB_RECORDFAIL_TIMEOUT=3` (wait only 3 seconds with boot menu after a failure)
2. Copy control design project to the server.
3. Start DCU platform and in Design Tool open the project.
4. In the design project, set parameters of the visualization, if not done previously. Visualization and data recordign parameters may be set comfortably using interactive windows directly from DCU Design Tool, see Design and Programming Guide.
5. Click "Set Visualization Server" → Locally
6. Set group privilages of the Apache web directory to var-www. For LAMP package the director is typically in "var/www" or "var/www/html".
7. Copy visualization files from your project "Visualization" folder into the web-page Apache directory. The file "index.php" of the visualization will be now called when user types "127.0.0.1" in web-browser on visualization server or when remote computer will access IP address of the server via Apache TCP port (typically http port 80).
8. Set operating system so that, AMP package and UCC starts automatically after restart. In Linux, AMP is started automatically after restart when installed. To set UCC to start automatically in Linux Ubuntu follow these steps:
 1. Copy `ucc.jar` file and `autostart_ucc.bat` file into Home directory.
 2. Set correct database password inside `autostart_ucc.bat` file.
 3. Run Startup Applications from Dashboard
 4. Click Add button in Startup Applications window and enter Name: UCC, Command: (browse for `autostart_ucc.bat`)
 5. Save and close Startup Applications window.

Note: UCC will start and log in to database if it is called with database login, database password, and database URL as input parameters. Example of the command to call in terminal is (see `autostart_ucc.bat` file):

```
java -Xmx512M -jar ucc.jar "jdbc:mysql: : 127.0.0.1:3306/mysql" "root" "rootpassword"
```

2 Special settings

2.1 Changing MySQL data directory

To change data directory of MySQL database for example in linux ubuntu from default directory `/var/lib/mysql` to some usb disk storage at `/media/usbdisk/data` follow these steps:

1. Format the disk as EXT4 (linux type) partitions to keep linux permissions for all mysql files and folders.

2. Stop mysql server

```
>> sudo service mysql stop
```

3. Copy database files to a new location with all privileges (-a option)

```
>> cp -a /var/lib/mysql /media/usbdisk/data
```

3. Set usbdisk privileges to mysql group.

4. Open configuration file of MySQL as superuser

```
>>sudo gedit /etc/mysql/mysql.conf.d/mysql.cnf
```

and change the line

```
datadir=/var/lib/mysql
```

to

```
datadir=/media/usbdisk/data/mysql
```

5. Open AppArmor Alias configuration file as superuser

```
>>sudo gedit /etc/apparmor.d/tunables/alias
```

and add the following line:

```
alias /var/lib/mysql/ -> /media/usbdisk/data/mysql/,
```

6. Restart AppArmor

```
>> sudo systemctl restart apparmor
```

7. Restart MySQL database server

```
>> sudo service mysql start
```

8. Check if MySQL running

```
>> sudo service mysql status
```

2.2 Change MySQL root password

To change MySQL root password follow the following procedure:

1. KILL PROCESSES

```
ps -A | grep sql
sudo service mysql stop
sudo service mysqld stop
sudo /etc/init.d/mysql stop
killall -KILL mysql mysqld_safe mysqld
```

2. START SPECIAL PROCESS AND CONNECT

```
sudo mysqld --user=mysql --skip-networking --skip-grant-tables &
mysql -uroot mysql
```

3. CHANGE PASSWORD

```
select * from mysql.user;
UPDATE mysql.user SET authentication_string = PASSWORD('password') WHERE
User = 'root';
FLUSH PRIVILEGES;
quit
```

4. KILL SPECIAL PROCESSES

```
ps -A | grep sql
sudo service mysql stop
sudo service mysqld stop
sudo /etc/init.d/mysql stop
killall -KILL mysql mysqld_safe mysqld
```

5. RESTART NORMAL PROCESS

```
sudo service mysql start
```

2.3 Remove Unlock keyring popup in web-browser

On Linux Ubuntu:

1. Open Passwords and Keys (in Dashboard type Passwords and Keys)

2. Click with mouse right button on Login a click “Change password”
3. Enter root password to unlock setting and New Password leave empty.
4. Restart web-browser.

3 Description of visualization server configuration tables and setting Alarm values

Visualization server has all configuration parameters stored in MySQL database called “dcuUserCommConfig”. This database is automatically set using DCU Design Tool, see Design and Programming Guide. Structure of the configuration database is detailed in this section for experts.

All Dynamic Control Units that will stream data into server are defined in “permanentHistRec” table. Example of one DCU definition:

- cUnitIP=10.0.0.10 (IP address of DCU)
- cUnitKey= NULL (encryption key of DCU)
- cUnitTcpPort=1033 (user command TCP port of DCU, if local connection, then always 1033)
- userId=8 (ID number of user defined inside DCU, visualization will communicate with DCU via this user account for configuration purposes so it should be user with high security level)
- userSecurity=15 (security level of DCU user, max. is 15 this gives access to all parameters)
- userPassword=”root” (password of DCU user)
- histServTcpPort=42250 (TCP port on which will listen visualization server, each DCU must have its own port. This port number is defined inside DCU in general application parameters as well.)
- histServKey=NULL (unused)
- visualName=Default unit (Label of DCU that will appear in visualization in Control System section)
- ruccIP=NULL (IP address of remote server with running user command center (UCC), if communication with this DCU should pass remotely via another UCC)
- ruccTcpPort=NULL (TCP port of MySQL database that is installed on remote server, on local network, it is typically 3306)
- ruccLogin=NULL (login into remote MySQL database)
- ruccPassword=NULL (password into remote MySQL database)

All visualization users are defined in “visualUsers” table. Example of one user definition:

- userId=1 (ID of visualization user)
- visualLogin=operator (login of visualization user)
- visualPassword=operator (password of visualization user)
- dcuUserId=8 (ID number of user defined inside DCU, visualization will communicate with DCU via this user account)

- dcuUserSecurity=15 (security level of DCU user, max. is 15 this gives access to all parameters)
- dcuUserPassword=root (password of DCU user)
- showControlSystem=1 (enable(1) / disable(0) view of Control System menu)
- allowEventAck=1 (enable(1) / disable(0) acknowledge of alarms)
- showView1 ... showView10 = 1 (enable(1) / disable(0) view inside Process section)

Groups (chapters) of process views are defined in “processViews” table. Example:

- viewId=1 (ID number of process view group)
- visualName='Default view group' (label of process view group that will appear in Process section)
- visualFile=NULL (unused)

Process view pages are define in “processViewsPages” table. Example of definition is as follows:

- groupId=1 (ID number of process view group to which belongs the page)
- pageId=1 (ID number of the process view page)
- visualName='Main view page' (Title of process view page)
- visualFile='cpanel_1.php' (PHP file with content definition of the process view page)

Groups (chapters) of event views are defined in “eventViews” table. Example:

- viewId=1 (ID number of event view group)
- visualName=All events/alarms (label of event view that will appear in Events/Alarms section)
- visualFile=NULL (unused)

Event view pages are defined in “eventViewsPages” table. Example of definition is as follows:

- groupId=1 (ID number of event view group to which belongs the page)
- pageId=1 (ID number of the event view page)
- eventsAlarms=1 (=1 if alarms as well as events are displayed in the page, =2 if alarms only are displayed)
- dcuIds='1,2,3' (ID numbers of DCU controllers whose alarms/events will be displayed in the page)
- visualName='Event view page' (Title of event view page)
- visualFile='events_displalarms.php' (PHP file with content definition of the event view page)

DCU clock synchronization service synchronizes periodically clock of specified DCU

with visualization server PC time. The service is defined in the “dcuUsercommConfig” database in the table “rtRtcSync”. Example of definition is as follows:

- dcuId=1 (ID number of dynamic control unit to synchronize. The unit must be defined in “permanentHistRec” table)
- syncCycleHr=1 (synchronization period in hours)

Alarm watch service checks permanently for pre-specified type of event/alarm. When event/alarm occurs, a message is sent to printer or/and via email. The service is defined in the “dcuUsercommConfig” database in the table “rtAlarmWatch”. Example of definition is as follows:

- dcuId=1 (ID number of dynamic control unit where alarm will occur)
- cVarId=0 (ID number of control variable. To be defined, if an alarm of control variable value is to be checked. Value -1 means, that control variable is defined by its name.)
- cVarName=”cfm:: AI1 pressure_out0” (control variable name. To be defined, if an alarm of control variable value is to be checked. It must be defined if cVarId is set to -1.)
- alarmId=0 (ID number of control variable alarm. To be defined, if an alarm of control variable value is to be checked. Each control variable may have defined up to 16 alarms with ID number from 0 to 15.)
- eventId=0 (ID number of event. To be defined if a DCU event other than control variable alarm is to be checked. For example, control unit re-start, see Annexe C. Otherwise set to 0.)
- message=”Alarm on pressure//pressure above 10bar//check process” (Message to be sent in case of event/alarm occurrence. The text string “//” indicates new line.)
- commType=”email” (defines by what type of communication channel will the message be sent. The type can be: “email”, “printer”, “email+printer”.)
- commSettings=”awemailto:email1@test.com,email2@test.com//awemailfrom:emailfrom@test.com//awemailsubject:Alarm on cooling//awemailhost:localhost//awemaillogin:test//awemailpassword:test//” (Defines email settings if an email must be sent. The settings must have structure as shown in the example.)

Virtual input / output service receives or sends RT UDP messages containing real-time process data. The service is defined in the table “rtVirtualIO”. Values received or to be sent are in table “rtVirtualIOvalue” created at the time of starting virtual I/O services inside the database “dcuUsercommConfig”. Example of definition is as follows:

- in0out1=0 (It defines if the service is input (0) or output (1))
- remotecFunId=1100 (defines ID number of remote control function receiving/sending values)
- remoteIp (IP address of remote communicating partner)

- remoteUdpPort (UDP port of remote communicating partner for this service)
- localUdpPort (local UDP port for this service)
- cFunId=1100 (ID number of local control function receiving/sending values)
- samplingms=1000 (It defines sampling time of the virtual I/O in miliseconds)
- ioSize=4 (It defines number of inputs/outputs of the control function)
- real1bin2int3='1,1,2,2' (It defines a format of transported value for each input/output 1=real double-precision value, 2=binary 1/0 value, 3=64-bit long integer value.)
- initialValue='0.0, 0.0 ,0.0, 0.0' (It defines initial values for each input/output)
- encryptKey=NULL (Encryption key in case an encryption is used for RT UDP communication.)

Global alarm definition is in table “globalAlarmVariable”. The table defines control variable that indicates global alarm. If unused, alarm will be indicated in the visualization every time an unacknowledged alarm event is received from any DCU controller. Example of definition is as follows:

- dcuId=0 (It defines ID number of DCU controller in which control variable that indicates global alarm is present)
- cvId=0 (It defines ID number of control variable that indicates global alarm)
- cvName="cfn::var0_out0" (name of control variable that indicates global alarm. Global alarm is checked according to control variable name.)

Archive definition is in “histRecArchive” table. The table is used to define archiving service for all DCU controllers. Each DCU has its own archiving service. Example of definition is as follows:

- dcuId=1 (It defines ID number of DCU controller that will have archiving service)
- timeSpan=86400 (Time span of the archive in seconds)
- samplingSec=600 (Sampling time of the archive service in seconds)
- prePostTimeSec=600 (Time period for serious events recording. Serious events are defined in “histRecArchiveEvents”, see below. If a serious event occurs, archiving service records every sample received from DCU, samplingSec sampling time is ignored. prePostTimeSec is time period how long before serious event occurrence and how long after the occurrence the archiving records all samples.)

Archive serious events are defined in “histRecArchiveEvents” table. The table is used to define serious events for all archiving services. Example of definition is as follows:

- dcuId=1 (It defines ID number of corresponding DCU controller)
- cvId=0 (Control variable ID if alarm on control variable is serious event, otherwise -1 is set)
- cvName (Name of control variable if alarm on control variable is serious event)
- eventType (If alarm on control variable is serious event, index of alarm 1...16 is defined here. If DCU event is serious event, ID of the event is defined here.)

The table “visualizationPath” contains path to actual control design project visualization. Example :

- path='C:/home/dcu/project1/Visualization'

The table “uccParams” contains actual parameters of the User Command Center service. Example :

- id=3 (ID number of the parameter)
- value="600000" (value of the parameter)
- name="histRecCvarIdNameCheck" (name of the parameter)
- note="[msec] interval for updating control variable names and IDs in history recording service" (detailed description of the parameter)

The table “operatorPanels” contains definitions of operator panels. Example of definition is as follows:

- opId=0 (ID number of the operator panel)
- opActual=0 (=0 if configuration inside server, =1 if configuration inside this operator panel)
- opName="cfn.: OP1 23xRTUDP User VI5001" (title of the operator panel, by default the name is generated from the name of virtual input control function that defines the panel)
- opIP=10.0.0.101 (IP address of the operator panel)
- opMode=1 (Mode of streaming process data from server to panel. =1 history recording, =2 monitoring data recording)
- opSamplingMsec=1000 (sampling time of the process data stream from server to panel)
- opTimeSpanMsec=0 (history recording time span of the operator panel)
- opEncryption=0 (Encryption mode of the process data stream =0 off, =1 on)
- opEncryptionKey='xyzxyz' (Encryption key)
- opUdpPort=1060 (UDP port of the process data stream)
- opStreamedUnits (ID numbers of DCU controllers from which process data are streamed to panel)

Alarm values for any variable may be defined in the file called “alarms_definition_server.xml”. This file must be located in “Visualization/process” folder of the DCU control design project. Example of the file content is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<alarms_def>
  <email_set id="setting1">
    <to>hp@prosystemy.sk</to>
    <to>hynek.prochazka@prosystemy.sk</to>
    <from>alarm@prosystemy.sk</from>
    <host>smtp.exohosting.com:465,SSL</host>
    <login>alarm@prosystemy.sk</login>
    <password>Alarm2018prosystemy</password>
    <subject>Alarm vseobecny subject</subject>
    <text_line>Alarm text line 1 vseobecny.</text_line>
    <text_line>Alarm text line 2 vseobecny.</text_line>
  </email_set>

  <email_set id="setting2">
    <to>hp@prosystemy.sk</to>
    <from>alarm@prosystemy.sk</from>
    <host>smtp.exohosting.com:465,SSL</host>
    <login>alarm@prosystemy.sk</login>
    <password>Alarm2018prosystemy</password>
    <subject>Alarm ExDom</subject>
    <text_line>Alarm Experimentalny Dom.</text_line>
  </email_set>

  <dcu id="I1">
    <variable name="cfn:: Real.Zero_out0">
      <alarm>
        <on_text> alarm on 1</on_text>
        <off_text> alarm off 1</off_text>
        <high_low> low</high_low>
        <value> -10</value>
        <delay_ms> 1000</delay_ms>
        <email>
          <email_set_id>setting1</email_set_id>
          <subject>Alarm RealZero</subject>
          <text_line>Alarm realzero 1.</text_line>
          <text_line>Alarm realzero 2.</text_line>
        </email>
      </alarm>
      <alarm>
        <on_text> alarm on 2</on_text>
        <off_text> alarm off 2</off_text>
        <high_low> high</high_low>
        <value> 95</value>
        <delay_ms> 2000</delay_ms>
      </alarm>
    </variable>

    <variable name="cfn:: Bin.Zero_out0">
      <alarm>
        <on_text> Teplota nizka</on_text>
        <off_text> Teplota OK</off_text>
        <high_low> high</high_low>
        <value> 1</value>
        <delay_ms> 3000</delay_ms>
        <email>
          <email_set_id>setting1</email_set_id>
          <subject>Alarm bin.gen</subject>
          <text_line>Alarm bin gen 1 specificky.</text_line>
          <text_line>Alarm bin gen 2 specificky.</text_line>
        </email>
      </alarm>
    </variable>

    <variable name="cfn:: Real1_out0">
      <alarm>
        <on_text> Real 1 alarm on</on_text>
        <off_text> Real 1 alarm off 1</off_text>
        <high_low> low</high_low>
        <value> -10</value>
        <delay_ms> 1000</delay_ms>
        <email>
          <email_set_id>setting2</email_set_id>
          <subject>Alarm Real1</subject>
          <text_line>Alarm real1.</text_line>
        </email>
      </alarm>
    </variable>

    <variable name="cfn:: Real2_out0">
      <alarm>
        <on_text> Real 2 alarm on</on_text>
        <off_text> Real 2 alarm off 1</off_text>
        <high_low> low</high_low>
      </alarm>
    </variable>
  </dcu id="I1">
</alarms_def>
```

```
<value> -10</value>
<delay_ms> 1000</delay_ms>
<email>
    <email_set_id>setting2</email_set_id>
    <subject>Alarm Real2</subject>
    <text_line>Alarm real2.</text_line>
</email>
</alarm>
</variable>
<variable name="cfn:: Real3_out0">
    <alarm>
        <on_text> Real 3 alarm on</on_text>
        <off_text> Real 3 alarm off 1</off_text>
        <high_low> low</high_low>
        <value> -10</value>
        <delay_ms> 1000</delay_ms>
        <email>
            <email_set_id>setting2</email_set_id>
            <subject>Alarm Real3</subject>
            <text_line>Alarm real3.</text_line>
        </email>
    </alarm>
</variable>
<variable name="cfn:: Real4_out0">
    <alarm>
        <on_text> Real 4 alarm on</on_text>
        <off_text> Real 4 alarm off 1</off_text>
        <high_low> low</high_low>
        <value> -10</value>
        <delay_ms> 1000</delay_ms>
        <email>
            <email_set_id>setting2</email_set_id>
            <subject>Alarm Real4</subject>
            <text_line>Alarm real4.</text_line>
        </email>
    </alarm>
</variable>
<variable name="cfn:: Real5_out0">
    <alarm>
        <on_text> Real 5 alarm on</on_text>
        <off_text> Real 5 alarm off 1</off_text>
        <high_low> low</high_low>
        <value> -10</value>
        <delay_ms> 1000</delay_ms>
        <email>
            <email_set_id>setting2</email_set_id>
            <subject>Alarm Real5</subject>
            <text_line>Alarm real5.</text_line>
        </email>
    </alarm>
</variable>
<variable name="cfn:: Real6_out0">
    <alarm>
        <on_text> Real 6 alarm on</on_text>
        <off_text> Real 6 alarm off 1</off_text>
        <high_low> low</high_low>
        <value> -10</value>
        <delay_ms> 1000</delay_ms>
        <email>
            <email_set_id>setting2</email_set_id>
            <subject>Alarm Real6</subject>
            <text_line>Alarm real6.</text_line>
        </email>
    </alarm>
</variable>
<variable name="cfn:: Real7_out0">
    <alarm>
        <on_text> Real 7 alarm on</on_text>
        <off_text> Real 7 alarm off 1</off_text>
        <high_low> low</high_low>
        <value> -10</value>
        <delay_ms> 1000</delay_ms>
        <email>
            <email_set_id>setting2</email_set_id>
            <subject>Alarm Real7</subject>
            <text_line>Alarm real7.</text_line>
        </email>
    </alarm>
</variable>
<variable name="cfn:: Real8_out0">
    <alarm>
        <on_text> Real 8 alarm on</on_text>
        <off_text> Real 8 alarm off 1</off_text>
```

```
<high_low> low</high_low>
<value> -10</value>
<delay_ms> 1000</delay_ms>
<email>
    <email_set_id>setting2</email_set_id>
    <subject>Alarm Real8</subject>
    <text_line>Alarm real8.</text_line>
</email>
</alarm>
</variable>
<variable name="cfn: Real9_out0">
    <alarm>
        <on_text> Real 9 alarm on</on_text>
        <off_text> Real 9 alarm off 1</off_text>
        <high_low> low</high_low>
        <value> -10</value>
        <delay_ms> 1000</delay_ms>
        <email>
            <email_set_id>setting2</email_set_id>
            <subject>Alarm Real9</subject>
            <text_line>Alarm real9.</text_line>
        </email>
    </alarm>
</variable>
</dcu>

<dcu id="12">
    <variable name="cfn: Real21_out0">
        <alarm>
            <on_text> Real 21 alarm on</on_text>
            <off_text> Real 21 alarm off 1</off_text>
            <high_low> low</high_low>
            <value> -10</value>
            <delay_ms> 1000</delay_ms>
            <email>
                <email_set_id>setting2</email_set_id>
                <subject>Alarm Real21</subject>
                <text_line>Alarm real21.</text_line>
            </email>
        </alarm>
    </variable>
</dcu>
```