

## LiQt interface

0

Generated by Doxygen 1.8.11

## Contents

<b>1</b>	<b>README</b>	<b>2</b>
<b>2</b>	<b>Namespace Index</b>	<b>2</b>
2.1	Namespace List . . . . .	2
<b>3</b>	<b>Hierarchical Index</b>	<b>2</b>
3.1	Class Hierarchy . . . . .	2
<b>4</b>	<b>Class Index</b>	<b>4</b>
4.1	Class List . . . . .	4
<b>5</b>	<b>File Index</b>	<b>6</b>
5.1	File List . . . . .	6
<b>6</b>	<b>Namespace Documentation</b>	<b>7</b>
6.1	Ui Namespace Reference . . . . .	7
<b>7</b>	<b>Class Documentation</b>	<b>7</b>
7.1	CheckBoxIndexed Class Reference . . . . .	7
7.1.1	Constructor & Destructor Documentation . . . . .	8
7.1.2	Member Function Documentation . . . . .	8
7.1.3	Member Data Documentation . . . . .	9
7.2	ComSettings Struct Reference . . . . .	9
7.2.1	Member Function Documentation . . . . .	9
7.3	CountersDialog Class Reference . . . . .	10
7.3.1	Constructor & Destructor Documentation . . . . .	11
7.3.2	Member Function Documentation . . . . .	11
7.3.3	Member Data Documentation . . . . .	11
7.4	CrcCalc Class Reference . . . . .	12
7.4.1	Member Function Documentation . . . . .	12
7.5	CyclesDialog Class Reference . . . . .	13
7.5.1	Constructor & Destructor Documentation . . . . .	14

7.5.2	Member Function Documentation . . . . .	14
7.5.3	Member Data Documentation . . . . .	16
7.6	EmailSettings Struct Reference . . . . .	18
7.6.1	Member Data Documentation . . . . .	18
7.7	ExitDialog Class Reference . . . . .	19
7.7.1	Member Typedef Documentation . . . . .	20
7.7.2	Member Enumeration Documentation . . . . .	20
7.7.3	Constructor & Destructor Documentation . . . . .	20
7.7.4	Member Function Documentation . . . . .	21
7.7.5	Member Data Documentation . . . . .	21
7.8	GeneralSettingDialog Class Reference . . . . .	22
7.8.1	Constructor & Destructor Documentation . . . . .	23
7.8.2	Member Function Documentation . . . . .	24
7.8.3	Member Data Documentation . . . . .	27
7.9	HeadActivationDialog Class Reference . . . . .	28
7.9.1	Constructor & Destructor Documentation . . . . .	28
7.9.2	Member Function Documentation . . . . .	29
7.9.3	Member Data Documentation . . . . .	30
7.10	HeadSetting::HeadComands_ Struct Reference . . . . .	31
7.10.1	Member Data Documentation . . . . .	31
7.11	HeadForm Class Reference . . . . .	32
7.11.1	Member Typedef Documentation . . . . .	33
7.11.2	Member Enumeration Documentation . . . . .	33
7.11.3	Constructor & Destructor Documentation . . . . .	34
7.11.4	Member Function Documentation . . . . .	34
7.11.5	Member Data Documentation . . . . .	35
7.12	HeadSetting::HeadParameters_ Struct Reference . . . . .	37
7.12.1	Member Function Documentation . . . . .	37
7.12.2	Member Data Documentation . . . . .	38
7.13	Register::HeadReg_ Union Reference . . . . .	39

7.13.1	Member Data Documentation . . . . .	39
7.14	HeadSetting Class Reference . . . . .	40
7.14.1	Member Typedef Documentation . . . . .	42
7.14.2	Member Enumeration Documentation . . . . .	42
7.14.3	Constructor & Destructor Documentation . . . . .	44
7.14.4	Member Function Documentation . . . . .	44
7.14.5	Member Data Documentation . . . . .	44
7.15	HeadSettingButton Class Reference . . . . .	45
7.15.1	Constructor & Destructor Documentation . . . . .	45
7.15.2	Member Function Documentation . . . . .	45
7.15.3	Member Data Documentation . . . . .	45
7.16	IndexerLiftSettings Class Reference . . . . .	46
7.16.1	Member Typedef Documentation . . . . .	47
7.16.2	Member Enumeration Documentation . . . . .	47
7.16.3	Constructor & Destructor Documentation . . . . .	48
7.16.4	Member Function Documentation . . . . .	48
7.16.5	Member Data Documentation . . . . .	49
7.17	Register::IndexerReg_ Union Reference . . . . .	50
7.17.1	Member Data Documentation . . . . .	50
7.18	IndexerSettingDialog Class Reference . . . . .	51
7.18.1	Constructor & Destructor Documentation . . . . .	52
7.18.2	Member Function Documentation . . . . .	52
7.18.3	Member Data Documentation . . . . .	55
7.19	IndexerWidget Class Reference . . . . .	56
7.19.1	Constructor & Destructor Documentation . . . . .	57
7.19.2	Member Function Documentation . . . . .	57
7.19.3	Member Data Documentation . . . . .	58
7.20	IndexerLiftSettings::IndexParameters_ Struct Reference . . . . .	59
7.20.1	Member Function Documentation . . . . .	59
7.20.2	Member Data Documentation . . . . .	59

7.21	InfoWidget Class Reference . . . . .	60
7.21.1	Constructor & Destructor Documentation . . . . .	60
7.21.2	Member Function Documentation . . . . .	61
7.21.3	Member Data Documentation . . . . .	61
7.22	KeyboardButton Class Reference . . . . .	62
7.22.1	Constructor & Destructor Documentation . . . . .	62
7.22.2	Member Function Documentation . . . . .	62
7.22.3	Member Data Documentation . . . . .	62
7.23	KeyboardDialog Class Reference . . . . .	63
7.23.1	Member Typedef Documentation . . . . .	64
7.23.2	Member Enumeration Documentation . . . . .	64
7.23.3	Constructor & Destructor Documentation . . . . .	64
7.23.4	Member Function Documentation . . . . .	64
7.23.5	Member Data Documentation . . . . .	65
7.24	MachineSettings::LastRevolverWarm_ Union Reference . . . . .	66
7.24.1	Member Data Documentation . . . . .	66
7.25	IndexerLiftSettings::LiftParameters_ Struct Reference . . . . .	67
7.25.1	Member Function Documentation . . . . .	67
7.25.2	Member Data Documentation . . . . .	67
7.26	Register::LiftReg_ Union Reference . . . . .	68
7.26.1	Member Data Documentation . . . . .	68
7.27	LoginDialog Class Reference . . . . .	69
7.27.1	Constructor & Destructor Documentation . . . . .	69
7.27.2	Member Function Documentation . . . . .	70
7.27.3	Member Data Documentation . . . . .	70
7.28	LogoDialog Class Reference . . . . .	71
7.28.1	Constructor & Destructor Documentation . . . . .	71
7.28.2	Member Function Documentation . . . . .	71
7.28.3	Member Data Documentation . . . . .	72
7.29	MachineSettings::MachineHeadType_ Union Reference . . . . .	73

7.29.1	Member Data Documentation . . . . .	73
7.30	MachineSettings::MachineIndexLiftType_ Union Reference . . . . .	74
7.30.1	Member Data Documentation . . . . .	74
7.31	MachineSettings::MachineParameters_ Struct Reference . . . . .	75
7.31.1	Member Function Documentation . . . . .	75
7.31.2	Member Data Documentation . . . . .	75
7.32	MachineSettings Class Reference . . . . .	76
7.32.1	Member Typedef Documentation . . . . .	77
7.32.2	Member Enumeration Documentation . . . . .	78
7.32.3	Constructor & Destructor Documentation . . . . .	78
7.32.4	Member Function Documentation . . . . .	78
7.32.5	Member Data Documentation . . . . .	80
7.33	MachineSettings::MachineState_ Union Reference . . . . .	81
7.33.1	Member Data Documentation . . . . .	81
7.34	MailSender Class Reference . . . . .	82
7.34.1	Constructor & Destructor Documentation . . . . .	82
7.34.2	Member Function Documentation . . . . .	83
7.34.3	Member Data Documentation . . . . .	83
7.35	MaintanceDialog Class Reference . . . . .	84
7.35.1	Member Enumeration Documentation . . . . .	85
7.35.2	Constructor & Destructor Documentation . . . . .	85
7.35.3	Member Function Documentation . . . . .	86
7.35.4	Member Data Documentation . . . . .	87
7.36	MaintanceElement Class Reference . . . . .	88
7.36.1	Constructor & Destructor Documentation . . . . .	88
7.36.2	Member Data Documentation . . . . .	88
7.37	MaintanceWidget Class Reference . . . . .	89
7.37.1	Constructor & Destructor Documentation . . . . .	89
7.37.2	Member Function Documentation . . . . .	90
7.37.3	Member Data Documentation . . . . .	90

7.38	MainWindow Class Reference . . . . .	91
7.38.1	Constructor & Destructor Documentation . . . . .	93
7.38.2	Member Function Documentation . . . . .	93
7.38.3	Member Data Documentation . . . . .	100
7.39	Register::MasterReg_ Union Reference . . . . .	106
7.39.1	Member Data Documentation . . . . .	106
7.40	ModData_ Union Reference . . . . .	107
7.40.1	Member Data Documentation . . . . .	107
7.41	NumpadButton Class Reference . . . . .	108
7.41.1	Constructor & Destructor Documentation . . . . .	108
7.41.2	Member Function Documentation . . . . .	108
7.41.3	Member Data Documentation . . . . .	108
7.42	NumpadDialog Class Reference . . . . .	109
7.42.1	Constructor & Destructor Documentation . . . . .	110
7.42.2	Member Function Documentation . . . . .	110
7.42.3	Member Data Documentation . . . . .	110
7.43	Register::MasterReg_::reg Struct Reference . . . . .	111
7.43.1	Member Data Documentation . . . . .	112
7.44	Register::IndexerReg_::reg Struct Reference . . . . .	115
7.44.1	Member Data Documentation . . . . .	116
7.45	Register::HeadReg_::reg Struct Reference . . . . .	118
7.45.1	Member Data Documentation . . . . .	119
7.46	Register::LiftReg_::reg Struct Reference . . . . .	121
7.46.1	Member Data Documentation . . . . .	121
7.47	Register Class Reference . . . . .	123
7.47.1	Member Typedef Documentation . . . . .	126
7.47.2	Member Enumeration Documentation . . . . .	126
7.47.3	Constructor & Destructor Documentation . . . . .	129
7.47.4	Member Function Documentation . . . . .	129
7.47.5	Member Data Documentation . . . . .	130

7.48 ReprogramDialog Class Reference . . . . .	131
7.48.1 Member Typedef Documentation . . . . .	131
7.48.2 Member Enumeration Documentation . . . . .	131
7.48.3 Constructor & Destructor Documentation . . . . .	132
7.48.4 Member Function Documentation . . . . .	132
7.48.5 Member Data Documentation . . . . .	132
7.49 SerialPort Class Reference . . . . .	133
7.49.1 Constructor & Destructor Documentation . . . . .	134
7.49.2 Member Function Documentation . . . . .	134
7.49.3 Member Data Documentation . . . . .	135
7.50 SerialSettingsDialog Class Reference . . . . .	136
7.50.1 Constructor & Destructor Documentation . . . . .	136
7.50.2 Member Function Documentation . . . . .	137
7.50.3 Member Data Documentation . . . . .	137
7.51 SettingDialog Class Reference . . . . .	138
7.51.1 Constructor & Destructor Documentation . . . . .	139
7.51.2 Member Function Documentation . . . . .	139
7.51.3 Member Data Documentation . . . . .	141
7.52 UdpSocket Class Reference . . . . .	142
7.52.1 Constructor & Destructor Documentation . . . . .	142
7.52.2 Member Function Documentation . . . . .	142
7.52.3 Member Data Documentation . . . . .	143
7.53 UserSettingDialog Class Reference . . . . .	144
7.53.1 Constructor & Destructor Documentation . . . . .	145
7.53.2 Member Function Documentation . . . . .	145
7.53.3 Member Data Documentation . . . . .	147
7.54 WarmingWidget Class Reference . . . . .	147
7.54.1 Constructor & Destructor Documentation . . . . .	148
7.54.2 Member Function Documentation . . . . .	148
7.54.3 Member Data Documentation . . . . .	148



<b>8 File Documentation</b>	<b>149</b>
8.1 countersdialog.cpp File Reference	149
8.2 countersdialog.h File Reference	149
8.3 crc16.h File Reference	149
8.3.1 Variable Documentation	149
8.4 cyclesdialog.cpp File Reference	149
8.5 cyclesdialog.h File Reference	150
8.6 exitdialog.cpp File Reference	150
8.7 exitdialog.h File Reference	150
8.8 generalsettingdialog.cpp File Reference	150
8.9 generalsettingdialog.h File Reference	151
8.9.1 Function Documentation	151
8.10 headactivationdialog.cpp File Reference	151
8.11 headactivationdialog.h File Reference	152
8.12 headform.cpp File Reference	152
8.13 headform.h File Reference	152
8.13.1 Typedef Documentation	153
8.13.2 Enumeration Type Documentation	153
8.14 headsettingdialog.cpp File Reference	153
8.15 headsettingdialog.h File Reference	153
8.16 indexersettingdialog.cpp File Reference	154
8.17 indexersettingdialog.h File Reference	154
8.18 indexerwidget.cpp File Reference	154
8.19 indexerwidget.h File Reference	154
8.20 infowidget.cpp File Reference	155
8.21 infowidget.h File Reference	155
8.22 keyboarddialog.cpp File Reference	155
8.23 keyboarddialog.h File Reference	155
8.24 logindialog.cpp File Reference	155
8.25 logindialog.h File Reference	156

8.26	logodialog.cpp File Reference . . . . .	156
8.27	logodialog.h File Reference . . . . .	156
8.28	mailsender.cpp File Reference . . . . .	156
8.29	mailsender.h File Reference . . . . .	156
8.30	main.cpp File Reference . . . . .	157
8.30.1	Function Documentation . . . . .	157
8.31	maintancedialog.cpp File Reference . . . . .	157
8.32	maintancedialog.h File Reference . . . . .	157
8.33	maintancewidget.cpp File Reference . . . . .	157
8.34	maintancewidget.h File Reference . . . . .	158
8.34.1	Function Documentation . . . . .	158
8.35	mainwindow.cpp File Reference . . . . .	158
8.36	mainwindow.h File Reference . . . . .	159
8.37	numpaddialog.cpp File Reference . . . . .	159
8.38	numpaddialog.h File Reference . . . . .	159
8.39	README.md File Reference . . . . .	160
8.40	reprogramdialog.cpp File Reference . . . . .	160
8.41	reprogramdialog.h File Reference . . . . .	160
8.42	serialport.cpp File Reference . . . . .	160
8.43	serialport.h File Reference . . . . .	160
8.43.1	Typedef Documentation . . . . .	161
8.44	serialsettingsdialog.cpp File Reference . . . . .	161
8.44.1	Variable Documentation . . . . .	161
8.45	serialsettingsdialog.h File Reference . . . . .	161
8.45.1	Function Documentation . . . . .	162
8.46	settings.cpp File Reference . . . . .	162
8.47	settings.h File Reference . . . . .	162
8.47.1	Typedef Documentation . . . . .	163
8.48	udpsocket.cpp File Reference . . . . .	163
8.49	udpsocket.h File Reference . . . . .	163
8.50	usersetting.cpp File Reference . . . . .	164
8.51	usersetting.h File Reference . . . . .	164
8.52	warmingwidget.cpp File Reference . . . . .	164
8.53	warmingwidget.h File Reference . . . . .	164

## 1 README

test0

L\_Aries-progect

## 2 Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

[Ui](#) 7

## 3 Hierarchical Index

### 3.1 Class Hierarchy

Structures, enums, lists and specific functions:

Registers:

<b>Register::MasterReg_</b>	<a href="#">106</a>
<b>Register::IndexerReg_</b>	<a href="#">50</a>
<b>Register::LiftReg_</b>	<a href="#">68</a>
<b>Register::HeadReg_</b>	<a href="#">39</a>

Indexer and lift settings:

<b>IndexerLiftSettings</b>	<a href="#">46</a>
<b>IndexerLiftSettings::IndexParameters_</b>	<a href="#">59</a>
<b>IndexerLiftSettings::LiftParameters_</b>	<a href="#">67</a>

Machine settings:

<b>MachineSettings::MachineHeadType_</b>	<a href="#">73</a>
<b>MachineSettings::MachineIndexLiftType_</b>	<a href="#">74</a>
<b>MachineSettings::MachineParameters_</b>	<a href="#">75</a>
<b>MachineSettings</b>	<a href="#">76</a>
<b>MachineSettings::MachineState_</b>	<a href="#">81</a>
<b>MachineSettings::LastRevolverWarm_</b>	<a href="#">66</a>
<b>MaintanceElement</b>	<a href="#">88</a>

Head settings:

<b>HeadSetting</b>	<b>40</b>
<b>HeadSetting::HeadComands_</b>	<b>31</b>
<b>HeadSetting::HeadParameters_</b>	<b>37</b>
Other:	
<b>ModData_</b>	<b>107</b>
<b>ComSettings</b>	<b>9</b>
<b>CrcCalc</b>	<b>12</b>
QDialog items	
<b>CyclesDialog</b>	<b>13</b>
<b>ExitDialog</b>	<b>19</b>
<b>GeneralSettingDialog</b>	<b>22</b>
<b>HeadActivationDialog</b>	<b>28</b>
<b>KeyboardDialog</b>	<b>63</b>
<b>LoginDialog</b>	<b>69</b>
<b>LogoDialog</b>	<b>71</b>
<b>MaintanceDialog</b>	<b>84</b>
<b>MaintanceWidget</b>	<b>89</b>
<b>NumpadDialog</b>	<b>109</b>
<b>ReprogramDialog</b>	<b>131</b>
<b>SerialSettingsDialog</b>	<b>136</b>
<b>UserSettingDialog</b>	<b>144</b>
QFrame items	
<b>InfoWidget</b>	<b>60</b>
QMainWindow items	
<b>MainWindow</b>	<b>91</b>
QObject items	
<b>MailSender</b>	<b>82</b>
<b>SerialPort</b>	<b>133</b>
QPushButton items	
<b>HeadSettingButton</b>	<b>45</b>
<b>KeyboardButton</b>	<b>62</b>
<b>NumpadButton</b>	<b>108</b>
QWidget items	
<b>HeadForm</b>	<b>32</b>
<b>IndexerSettingDialog</b>	<b>51</b>

IndexerWidget	56
SettingDialog	138
UdpSocket	142
Register::LiftReg_::reg	121
Register::IndexerReg_::reg	115
Register::MasterReg_::reg	111
Register::HeadReg_::reg	118
Register	123

## 4 Class Index

### 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CheckBoxIndexed	7
ComSettings	9
CrcCalc	12
CyclesDialog	13
EmailSettings	18
ExitDialog	19
GeneralSettingDialog	22
HeadActivationDialog	28
HeadSetting::HeadComands_	31
HeadForm	32
HeadSetting::HeadParameters_	37
Register::HeadReg_	39
HeadSetting	40
HeadSettingButton	45
IndexerLiftSettings	46
Register::IndexerReg_	50
IndexerSettingDialog	51
IndexerWidget	56
IndexerLiftSettings::IndexParameters_	59

InfoWidget	60
KeyboardButton	62
KeyboardDialog	63
MachineSettings::LastRevolverWarm_	66
IndexerLiftSettings::LiftParameters_	67
Register::LiftReg_	68
LoginDialog	69
LogoDialog	71
MachineSettings::MachineHeadType_	73
MachineSettings::MachineIndexLiftType_	74
MachineSettings::MachineParameters_	75
MachineSettings	76
MachineSettings::MachineState_	81
MailSender	82
MaintanceDialog	84
MaintanceElement	88
MaintanceWidget	89
MainWindow	91
Register::MasterReg_	106
ModData_	107
NumpadButton	108
NumpadDialog	109
PasswordDialog	??
Register::LiftReg_::reg	121
Register::IndexerReg_::reg	115
Register::MasterReg_::reg	111
Register::HeadReg_::reg	118
Register	123
ReprogramDialog	131
SerialPort	133
SerialSettingsDialog	136
SettingDialog	138

<a href="#">UdpSocket</a>	142
<a href="#">UserSettingDialog</a>	144

## 5 File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<a href="#">crc16.h</a>	149
<a href="#">cyclesdialog.cpp</a>	149
<a href="#">cyclesdialog.h</a>	150
<a href="#">exitdialog.cpp</a>	150
<a href="#">exitdialog.h</a>	150
<a href="#">generalsettingdialog.cpp</a>	150
<a href="#">generalsettingdialog.h</a>	151
<a href="#">headactivationdialog.cpp</a>	151
<a href="#">headactivationdialog.h</a>	152
<a href="#">headform.cpp</a>	152
<a href="#">headform.h</a>	152
<a href="#">headsettingdialog.cpp</a>	153
<a href="#">headsettingdialog.h</a>	153
<a href="#">indexersettingdialog.cpp</a>	154
<a href="#">indexersettingdialog.h</a>	154
<a href="#">indexerwidget.cpp</a>	154
<a href="#">indexerwidget.h</a>	154
<a href="#">infowidget.cpp</a>	155
<a href="#">infowidget.h</a>	155
<a href="#">keyboarddialog.cpp</a>	155
<a href="#">keyboarddialog.h</a>	155
<a href="#">logindialog.cpp</a>	155
<a href="#">logindialog.h</a>	156
<a href="#">logodialog.cpp</a>	156
<a href="#">logodialog.h</a>	156

<a href="#">mailsender.cpp</a>	156
<a href="#">mailsender.h</a>	156
<a href="#">main.cpp</a>	157
<a href="#">maintancedialog.cpp</a>	157
<a href="#">maintancedialog.h</a>	157
<a href="#">maintancewidget.cpp</a>	157
<a href="#">maintancewidget.h</a>	158
<a href="#">mainwindow.cpp</a>	158
<a href="#">mainwindow.h</a>	159
<a href="#">numpaddialog.cpp</a>	159
<a href="#">numpaddialog.h</a>	159
<a href="#">reprogramdialog.cpp</a>	160
<a href="#">reprogramdialog.h</a>	160
<a href="#">serialport.cpp</a>	160
<a href="#">serialport.h</a>	160
<a href="#">serialsettingsdialog.cpp</a>	161
<a href="#">serialsettingsdialog.h</a>	161
<a href="#">settings.cpp</a>	162
<a href="#">settings.h</a>	162
<a href="#">udpsocket.cpp</a>	163
<a href="#">udpsocket.h</a>	163
<a href="#">usersetting.cpp</a>	164
<a href="#">usersetting.h</a>	164

## 6 Namespace Documentation

### 6.1 Ui Namespace Reference

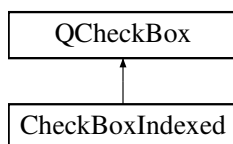
## 7 Class Documentation

### 7.1 CheckBoxIndexed Class Reference

```
#include <headactivationdialog.h>
```

Inheritance diagram for CheckBoxIndexed:





## Signals

- void `clicked` (int `index`)

## Public Member Functions

- `CheckBoxIndexed` (int `index`, QString `text`, QWidget `*parent=0`)
- virtual `~CheckBoxIndexed` ()

## Private Slots

- void `chClicked` (bool)

## Private Attributes

- int `index`

## 7.1.1 Constructor & Destructor Documentation

### 7.1.1.1 `CheckBoxIndexed()`

```
CheckBoxIndexed::CheckBoxIndexed (
    int index,
    QString text,
    QWidget * parent = 0 ) [inline]
```

### 7.1.1.2 `~CheckBoxIndexed()`

```
virtual CheckBoxIndexed::~~CheckBoxIndexed ( ) [inline], [virtual]
```

## 7.1.2 Member Function Documentation

## 7.1.2.1 chClicked

```
void CheckBoxIndexed::chClicked (
    bool ) [inline], [private], [slot]
```

## 7.1.2.2 clicked

```
void CheckBoxIndexed::clicked (
    int index ) [signal]
```

## 7.1.3 Member Data Documentation

## 7.1.3.1 index

```
int CheckBoxIndexed::index [private]
```

The documentation for this class was generated from the following file:

- [headactivationdialog.h](#)

## 7.2 ComSettings Struct Reference

```
#include <serialsettingsdialog.h>
```

## Public Member Functions

- [ComSettings operator= \(ComSettings nSett\)](#)

## Public Attributes

Class of serial port setting. Class constructor have standard values of parameters to allow start program without setting file.

- QString [name](#) = "/dev/ttyS0"
- qint32 [baudRate](#) = 38400
- QString [stringBaudRate](#) = "38400"
- QSerialPort::DataBits [dataBits](#) = QSerialPort::Data8
- QString [stringDataBits](#) = "8"
- QSerialPort::Parity [parity](#) = QSerialPort::OddParity
- QString [stringParity](#) = "Odd"
- QSerialPort::StopBits [stopBits](#) = QSerialPort::OneStop
- QString [stringStopBits](#) = "1"
- QSerialPort::FlowControl [flowControl](#) = QSerialPort::NoFlowControl
- QString [stringFlowControl](#) = "None"
- bool [localEchoEnabled](#)

## 7.2.1 Member Function Documentation

```
ComSettings ComSettings::operator=() (
    ComSettings nSett ) - Reimplemented standard operator. Allow copy parameters between two
```

class object in simply way.

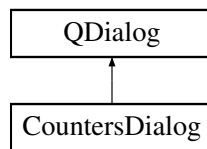
The documentation for this struct was generated from the following files:

- [serialsettingsdialog.h](#)
- [serialsettingsdialog.cpp](#)

### 7.3 CountersDialog Class Reference

```
#include <countersdialog.h>
```

Inheritance diagram for CountersDialog:



#### Signals

- void [resetSkipped](#) ()
- void [resetRemaining](#) ()
- void [remainingValChanged](#) (int val)

#### Public Member Functions

- [CountersDialog](#) (QWidget \*parent=0)
- [~CountersDialog](#) ()
- void [setRemaining](#) (int val)

#### Protected Member Functions

- bool [event](#) (QEvent \*e)
- void [showEvent](#) (QShowEvent \*ev)
- bool [eventFilter](#) (QObject \*watched, QEvent \*event)
- void [changeEvent](#) (QEvent \*event)

#### Private Slots

- void [on\\_pushButtonRemainReset\\_clicked](#) ()
- void [on\\_pushButtonSkippedReset\\_clicked](#) ()
- void [on\\_pushButtonHide\\_clicked](#) ()
- void [doubleSpinBoxRemain\\_valueChanged](#) (double arg1)

#### Private Attributes

- Ui::CountersDialog \* [ui](#)
- bool [acceptOnDeactivationEn](#)

### 7.3.1 Constructor & Destructor Documentation

7.3.1.1 CountersDialog::CountersDialog (  
    QWidget \* *parent* = 0 ) [explicit]

7.3.1.2 CountersDialog::~CountersDialog ( )

### 7.3.2 Member Function Documentation

7.3.2.1 void CountersDialog::changeEvent (  
    QEvent \* *event* ) [protected]

7.3.2.2 void CountersDialog::doubleSpinBoxRemain\_valueChanged (  
    double *arg1* ) [private],[slot]

7.3.2.3 bool CountersDialog::event (  
    QEvent \* *e* ) [protected]

7.3.2.4 bool CountersDialog::eventFilter (  
    QObject \* *watched*,  
    QEvent \* *event* ) [protected]

7.3.2.5 void CountersDialog::on\_pushButtonHide\_clicked ( ) [private],[slot]

7.3.2.6 void CountersDialog::on\_pushButtonRemainReset\_clicked ( ) [private],[slot]

7.3.2.7 void CountersDialog::on\_pushButtonSkippedReset\_clicked ( ) [private],[slot]

7.3.2.8 void CountersDialog::remainingValChanged (  
    int *val* ) [signal]

7.3.2.9 void CountersDialog::resetRemaining ( ) [signal]

7.3.2.10 void CountersDialog::resetSkipped ( ) [signal]

7.3.2.11 void CountersDialog::setRemaining (  
    int *val* )

7.3.2.12 void CountersDialog::showEvent (  
    QShowEvent \* *ev* ) [protected]

### 7.3.3 Member Data Documentation

7.3.3.1 bool CountersDialog::acceptOnDeactilationEn [private]

7.3.3.2 Ui::CountersDialog\* CountersDialog::ui [private]

The documentation for this class was generated from the following files:

- [countersdialog.h](#)
- [countersdialog.cpp](#)

## 7.4 CrcCalc Class Reference

```
#include <crc16.h>
```

Class contain two static function to calculate CRC16 value in two way: with or without CRC table. Functions are used without creating class object and are inline to optimize program.

### Public Member Functions

- [CrcCalc\(\)](#)

### Static Public Member Functions

- static [uint16\\_t CalculateCRC16](#) ([uint16\\_t](#) crc, [QByteArray](#) c\_ptr)
- static [uint16\\_t CalculateCRC16](#) ([QByteArray](#) inpArr)

### 7.4.1 Member Function Documentation

#### 7.4.1.1 [CalculateCRC16\(\)](#) [1/2]

Calculate CRC without table. Used for communication between PC and machine master board.

```
static uint16\_t CrcCalc::CalculateCRC16 (  
    uint16\_t crc,  
    QByteArray c_ptr ) [inline], [static]
```

#### 7.4.1.2 [CalculateCRC16\(\)](#) [2/2]

Calculate CRC with written table. Can be used to communicate with HMI or other similar devices.

```
static uint16\_t CrcCalc::CalculateCRC16 (  
    QByteArray inpArr ) [inline], [static]
```

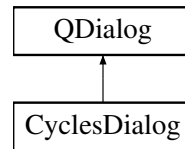
The documentation for this class was generated from the following file:

- [crc16.h](#)

## 7.5 CyclesDialog Class Reference

```
#include <cyclesdialog.h>
```

Inheritance diagram for CyclesDialog:



### Signals

- void [sendCommand](#) (QByteArray cmdArr)

### Public Member Functions

- [CyclesDialog](#) (QWidget \*parent=0)
- [CyclesDialog](#) (int [headCount](#), QWidget \*parent=0)
- [~CyclesDialog](#) ()

### Protected Member Functions

- void [showEvent](#) (QShowEvent \*ev)
- bool [eventFilter](#) (QObject \*watched, QEvent \*event)
- void [changeEvent](#) (QEvent \*event)

### Private Slots

- void [on\\_pButtonOK\\_clicked](#) ()
- void [on\\_pButtonON\\_clicked](#) ()
- void [on\\_pButtonPrev\\_clicked](#) ()
- void [on\\_pButtonNext\\_clicked](#) ()
- void [on\\_pButtonC1\\_clicked](#) ()
- void [on\\_pButtonC2\\_clicked](#) ()
- void [on\\_pButtonC3\\_clicked](#) ()
- void [on\\_pButtonC4\\_clicked](#) ()
- void [on\\_pButtonC5\\_clicked](#) ()
- void [on\\_pButtonC6\\_clicked](#) ()
- void [on\\_pButtonC7\\_clicked](#) ()
- void [on\\_pButtonC8\\_clicked](#) ()
- void [loadValues](#) (int \*vals)
- void [saveValues](#) (void)

## Private Attributes

- Ui::CyclesDialog \* [ui](#)
- int [headCount](#)
- QList< QDoubleSpinBox \* > [spinBoxList](#)
- QList< QLabel \* > [labelList](#)
- int [lastCycleSel](#)
- QList< int \* > [cycleValues](#)
- int [cycleState](#)
- QSettings \* [cycleSettings](#)
- QList< uint32\_t > [headStateList](#)
- QList< uint32\_t > [headStrokList](#)

## 7.5.1 Constructor & Destructor Documentation

### 7.5.1.1 CyclesDialog() [1/2]

Empty constructor (default). Not used in project.

```
CyclesDialog::CyclesDialog (
    QWidget * parent = 0 ) [explicit]
```

### 7.5.1.2 CyclesDialog() [2/2]

Class constructor. Sets count of heads of machine to create and layout widget of dialog.

```
CyclesDialog::CyclesDialog (
    int headCount,
    QWidget * parent = 0 ) [explicit]
```

### 7.5.1.3 ~CyclesDialog()

```
CyclesDialog::~CyclesDialog ( )
```

## 7.5.2 Member Function Documentation

```
void CyclesDialog::changeEvent ( ) (
    QEvent * event ) [protected] - reimplementation of default event for QDialog to enable user interface translation. Function called automatically.
```

### 7.5.2.1 eventFilter()

Mouse event filter to call on-screen numeric key to enter values to appropriate QSpinBoxes. Function called automatically.

```
bool CyclesDialog::eventFilter (
    QObject * watched,
    QEvent * event ) [protected]
```

### 7.5.2.2 loadValues

Function load values to appropriate QSpinBoxes form `QList<int* > cycleValues` list with consideration of selected cycle by variable `lastCycleSel`.

```
void CyclesDialog::loadValues (
    int * vals ) [private], [slot]
```

### 7.5.2.3 A set of set of on\_pButtonCX\_clicked() functions where number under 'C' X = 1..8

Functions which are invoked by push buttons (pButtonC(1..8)) to select number of cycle to set it parameters. Invoke any of that functions call `loadValues` (int \*vals) after set `lastCycleSel` to fill QSpinBox with previous (saved) values and set pButtonOn to appropriate state.

```
void CyclesDialog::on_pButtonC1_clicked() [private], [slot]
void CyclesDialog::on_pButtonC2_clicked() [private], [slot]
void CyclesDialog::on_pButtonC3_clicked() [private], [slot]
void CyclesDialog::on_pButtonC4_clicked() [private], [slot]
void CyclesDialog::on_pButtonC5_clicked() [private], [slot]
void CyclesDialog::on_pButtonC6_clicked() [private], [slot]
void CyclesDialog::on_pButtonC7_clicked() [private], [slot]
void CyclesDialog::on_pButtonC8_clicked() [private], [slot]
```

### 7.5.2.4 on\_pButtonNext\_clicked() and on\_pButtonPrev\_clicked()

Functions to change number of selected cycle. Invoke any of that functions call `loadValues` (int \*vals) after set `lastCycleSel` to fill QSpinBox with previous (saved) values.

Changing numbers work in cycle: call 'Next' at last select first and call 'Prev' at first select last data list.

```
void CyclesDialog::on_pButtonNext_clicked() [private], [slot]
void CyclesDialog::on_pButtonPrev_clicked() [private], [slot]
```

### 7.5.2.5 on\_pButtonON\_clicked

Function which invoke by push button pButtonON. pButtonON is two-state button which allow user turn on (or turn off) selected cycle. State of cycle (enabled or disabled) is displayed on this button and on appropriate button from top buttons menu.

```
void CyclesDialog::on_pButtonON_clicked ( ) [private], [slot]
```

### 7.5.2.6 saveValues

Function invoke by every `on_pButtonCX_clicked()` function, functions `on_pButtonPrev_clicked()` and `on_pButtonNext_clicked()`. It used to save spinBox values to `QList<int* > cycleValues` and write to "cycles.ini" setting file to appropriate section which are determinate by cycle number.

```
void CyclesDialog::saveValues (
    void ) [private], [slot]
```

### 7.5.2.7 sendCommand

Signal contain QByteArray type variable to transmit it to serial port. Signal receive by class parent.

```
void CyclesDialog::sendCommand (
    QByteArray cmdArr ) [signal]
```





- `QList<QDoubleSpinBox*> CyclesDialog::spinBoxList[private]` - List of `QDoubleSpinBox` class objects. Used for set strokes count for heads in selected sequence. Fill in [loadValues](#) (int \*vals) function for every sequence when variable [lastCycleSel](#) changed.

The documentation for this class was generated from the following files:

- [cyclesdialog.h](#)
- [cyclesdialog.cpp](#)

## 7.6 EmailSettings Struct Reference

```
#include <generalsettingdialog.h>
```

### Public Attributes

- QString [senderAddress](#) = "sender@server.com"
- QString [senderPassword](#) = "sender\_PSW"
- QString [receiverAddress](#) = "receiver@server.com"
- QString [emailSubject](#) = "Subject"
- bool [mailEnable](#) = false

### 7.6.1 Member Data Documentation

All variables contain data appropriate that names.

- `QString EmailSettings::emailSubject = "Subject"`
- `bool EmailSettings::mailEnable = false`
- `QString EmailSettings::receiverAddress = "receiver@server.com"`
- `QString EmailSettings::senderAddress = "sender@server.com"`
- `QString EmailSettings::senderPassword = "sender_PSW"`

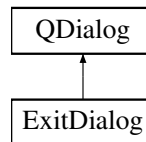
The documentation for this struct was generated from the following file:

- [generalsettingdialog.h](#)

## 7.7 ExitDialog Class Reference

```
#include <exitdialog.h>
```

Inheritance diagram for ExitDialog:



### Public Types

- enum `ExitCode_` {  
    *Continue*,  
    *LogOut*,  
    *Shutdown*,  
    *ExitFromProgram*,  
    *ReprogramMachine*,  
    *ServiceMode*,  
    *RestartProgram*,  
    *RestartMachine* }  
• typedef enum `ExitDialog::ExitCode_` `ExitCode`

### Public Member Functions

- `ExitDialog` (`QWidget *parent=0`)
- `~ExitDialog` ()
- void `setLogOutEn` (bool en)

### Static Public Member Functions

- static `ExitCode` `tryExit` (`QWidget *parent`, bool logOutEn=false)

### Protected Member Functions

- void `showEvent` (`QShowEvent *ev`)
- void `changeEvent` (`QEvent *event`)

### Private Slots

- void `on_pButtonTurnOff_clicked` ()
- void `on_pButtonRestartApp_clicked` ()
- void `on_pButtonRastartAll_clicked` ()
- void `on_pButtonServiceMode_clicked` ()
- void `on_pButtonExitApp_clicked` ()
- void `on_pButtonReprogram_clicked` ()
- void `on_pButtonLogOut_clicked` ()
- void `on_pushButtonCancel_clicked` ()

## Private Attributes

- [ExitCode](#) `exitCode`
- `Ui::ExitDialog` \* `ui`

## 7.7.1 Member Typedef Documentation

### 7.7.1.1 ExitCode

`typedef enum ExitDialog::ExitCode\_ ExitDialog::ExitCode` - type of variable which will be returned from function [tryExit](#). Created just for good code look.

## 7.7.2 Member Enumeration Documentation

### 7.7.2.1 ExitCode\_

`enum ExitDialog::ExitCode\_` - created for [ExitDialog::ExitCode\\_](#)

#### Enumerator

Continue	Flag to continue work without any changes. <a href="#">ExitCode</a>
LogOut	Flag to change user.
Shutdown	Flag to exit from program and send shutdown signal to operating system.
ExitFromProgram	Flag to exit from program and go to graphical user interface of operating system.
ReprogramMachine	Send signal disconnect serial port and connect with other parameters. After that open <a href="#">ReprogramDialog</a> class object (or create and open it).
ServiceMode	Flag to disconnect serial port connection (break data line between PC and master board) to give possibility press buttons without machine moving.
RestartProgram	Flag to re-initialization of program.
RestartMachine	Flag to exit from program and send restart signal to operating system. After system restart program will start in usual mode.

## 7.7.3 Constructor & Destructor Documentation

### 7.7.3.1 ExitDialog()

```
ExitDialog::ExitDialog (
    QWidget * parent = 0 ) [explicit]
```

### 7.7.3.2 ~ExitDialog()

```
ExitDialog::~ExitDialog ( )
```

## 7.7.4 Member Function Documentation

`void ExitDialog::changeEvent() (`  
`QEvent * event ) [protected]` - Reimplementation of default event for QDialog to enable user interface translation. Function called automatically.

---

All function in itemizer are called by click signals of appropriate buttons and assigns appropriate to button (function) name value (from `ExitCode_` enumerator) to `exitCode` variable. That variable will return by `tryExit(...)` function.

- `void ExitDialog::on_pButtonExitApp_clicked ( ) [private], [slot]`
  - `void ExitDialog::on_pButtonLogOut_clicked ( ) [private], [slot]`
  - `void ExitDialog::on_pButtonRastartAll_clicked ( ) [private], [slot]`
  - `void ExitDialog::on_pButtonReprogram_clicked ( ) [private], [slot]`
  - `void ExitDialog::on_pButtonRestartApp_clicked ( ) [private], [slot]`
  - `void ExitDialog::on_pButtonServiceMode_clicked ( ) [private], [slot]`
  - `void ExitDialog::on_pButtonTurnOff_clicked ( ) [private], [slot]`
  - `void ExitDialog::on_pushButtonCancel_clicked ( ) [private], [slot]`
- 

`void ExitDialog::setLogOutEn() (`  
`bool en )` - Function to enable or disable `pButtonLogOut` at exit dialog. Call with `true` value to show button and with `false` to hide it.

`void ExitDialog::showEvent() (`  
`QShowEvent * ev ) [protected]` - Reimplementation of default event for QDialog. In this reimplementation added hiding (or showing) buttons, which are not available for all users. Function called automatically on every window call.

`ExitDialog::ExitCode ExitDialog::tryExit (`  
`QWidget * parent,`  
`bool logOutEn = false ) [static]` - static function to call exit dialog end get `ExitDialog::Exit↵`  
**Code.**

## 7.7.5 Member Data Documentation

`exitCode ExitDialog::exitCode [private]`

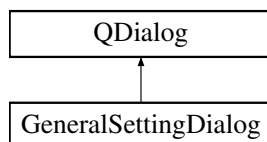
The documentation for this class was generated from the following files:

- `exitdialog.h`
- `exitdialog.cpp`

## 7.8 GeneralSettingDialog Class Reference

```
#include <generalsettingdialog.h>
```

Inheritance diagram for GeneralSettingDialog:



### Signals

- void [emailSettingsChanged](#) ([EmailSettings](#))
- void [machineParamChanged](#) (QByteArray machineParamArr)
- void [serialPortSettingsDialogRequested](#) ()
- void [styleChangedIndex](#) (int index)
- void [iconsChangedIndex](#) (int index)
- void [langChangedIndex](#) (int index)
- void [serviceSettingRequest](#) ()
- void [headActivationRequest](#) ()
- void [usersSettingRequest](#) ()
- void [directionChanged](#) (int dir)
- void [unloadStateChanged](#) (bool state)
- void [sendCommand](#) (QByteArray command)
- void [imageRequest](#) (bool enable, bool req=false)

### Public Member Functions

- [GeneralSettingDialog](#) (QWidget \*parent=0)
- [~GeneralSettingDialog](#) ()
- void [setIconFolder](#) (QString path)
- void [setEmailSettings](#) ([EmailSettings](#) emailSett)
- void [setMachineSetting](#) ([MachineSettings::MachineParameters](#) machineParam)
- void [setFocusLossAccept](#) (bool flag)
- void [setPasswords](#) (uint16\_t serialPass, uint16\_t mailPass, uint16\_t userPass)
- void [setStyle](#) (QStringList stList, int curSelect, QStringList iconList, int iconSel, bool backGrEn)
- void [setLangList](#) (QStringList langList, int curSelect)
- void [showPortInfo](#) ([ComSettings](#) comSett)

### Protected Member Functions

- bool [event](#) (QEvent \*e)
- bool [eventFilter](#) (QObject \*watched, QEvent \*event)
- void [showEvent](#) (QShowEvent \*ev)
- void [changeEvent](#) (QEvent \*event)

## Private Slots

- void [accept](#) ()
- void [reject](#) ()
- void [lockUnlockEmail](#) ()
- void [hideShowPassword](#) ()
- void [eventFilterSetup](#) ()
- void [changeSerialPortSettingsClicked](#) ()
- void [changeServiceStateClicked](#) ()
- void [userSettingClicked](#) ()
- void [styleChanged](#) (int index)
- void [iconChanged](#) (int index)
- void [langChanged](#) (int index)
- void [changeDirection](#) ()
- void [changeCyclesState](#) ()
- void [useUnloadStateChanged](#) ()
- void [headCountChanged](#) (double arg1)
- void [machineTypeChange](#) (int index)
- void [warningTimeChanged](#) (double arg1)
- void [on\\_pButtonHeadsActivation\\_clicked](#) ()
- void [on\\_checkBoxUseBackgr\\_clicked](#) ()
- void [on\\_pButtonSelectImg\\_clicked](#) ()

## Private Attributes

- Ui::GeneralSettingDialog \* [ui](#)
- [MachineSettings::MachineParameters](#) [machineParamGl](#)
- bool [acceptOnDeactilationEn](#)
- bool [acceptEnable](#)
- bool [loggedInSerial](#)
- bool [loggedInMail](#)
- bool [loggedInUserSett](#)
- [uint16\\_t](#) [serialPassword](#)
- [uint16\\_t](#) [mailPassword](#)
- [uint16\\_t](#) [usersPassword](#)
- QIcon [directionIcon](#)
- QString [pathIcon](#)

## 7.8.1 Constructor &amp; Destructor Documentation

`GeneralSettingDialog::GeneralSettingDialog() ( QWidget * parent = 0 ) [explicit]` - constructor of class. Connect signals to slots, fill comboBoxes with data and initialize variables to zero values.

## 7.8.1.1 ~GeneralSettingDialog()

`GeneralSettingDialog::~GeneralSettingDialog ( )` - standard destructor function.



### 7.8.2 Member Function Documentation

`void GeneralSettingDialog::accept ( ) [private], [slot]` - gather [EmailSettings](#) and [MachineSettings](#) and [MachineParameters](#) to appropriate structures. After that function generate signals [emailSettingsChanged\(...\)](#) and [machineParamChanged\(...\)](#) which are call appropriate function in parent object ([getEmailSettings\(...\)](#) and [getMachineParam\(...\)](#)).

`void GeneralSettingDialog::changeCyclesState ( ) [private], [slot]` - send command with [sendCommand\(...\)](#) function to turn on/off revolver function of machine and show/hide button "Cycles" on main interface (That button call [CyclesDialog](#) class object).

`void GeneralSettingDialog::changeDirection ( ) [private], [slot]` - send command with [sendCommand\(...\)](#) function to change rotate direction. Also function generate signal [directionChanged\(...\)](#).

`void GeneralSettingDialog::changeEvent ( QEvent * event ) [protected]` - Reimplementation of default event for QDialog to enable user interface translation. Function called automatically.

`void GeneralSettingDialog::changeSerialPortSettingsClicked ( ) [private], [slot]` - function to call [SerialSettingsDialog](#) class object to setup serial port. Function ask password to call setting dialog.

`void GeneralSettingDialog::changeServiceStateClicked ( ) [private], [slot]` - function to generate signal which are call appropriate function in parent object ([serviceStateChange\(...\)](#)).

`void GeneralSettingDialog::directionChanged ( int dir ) [signal]` - signal what are generated by [changeDirection\(...\)](#) and call appropriate function ([getDirection\(...\)](#)) in parent object.

`void GeneralSettingDialog::emailSettingsChanged ( EmailSettings ) [signal]` - signal what are generated by [accept\(...\)](#) and call appropriate function ([getEmailSettings\(...\)](#)) in parent object.

`bool GeneralSettingDialog::event() ( QEvent * e ) [protected]` - reimplementation of standard function to handle *QEvent::WindowDeactivate* or *QEvent::Leave* for automatic invoke of [accept\(...\)](#) function on window deactivation.

`bool GeneralSettingDialog::eventFilter() ( QObject * watched, QEvent * event ) [protected]` - reimplementation of standard function to handle *QEvent::MouseButtonClick* or *QEvent::MouseButtonRelease* events to call [NumpadDialog](#) or [KeyboardDialog](#) to enter data to appropriate widgets. Widgets which will call this function are defined in [eventFilterSetup\(...\)](#) function. In this function checked is *QObject \* watched* object of *QLineEdit* class or *QDoubleSpinBox* class and call appropriate dialog to enter data.

`void GeneralSettingDialog::eventFilterSetup ( ) [private], [slot]` - function to configure widgets on dialog to use [eventFilter \(...\)](#).

`void GeneralSettingDialog::headActivationRequest ( ) [signal]` - signal what are generated by [on\\_pushButtonHeadsActivation\\_clicked\(...\)](#) and call [HeadActivationDialog](#) class object. Connection is made in parent class.

`void GeneralSettingDialog::headCountChanged ( double arg1 ) [private], [slot]` - function to set up heads count on machine. Send appropriate data to master board with [sendCommand\(...\)](#) function.

`void GeneralSettingDialog::hideShowPassword ( ) [private], [slot]` - function to show or hide symbols line edit for sender password on mail setup tab.

```

void GeneralSettingDialog::iconChanged (
    int index )[private], [slot] - function invoked by QListView widget on changing selected item
and emit signal iconsChangedIndex\(...\) to call appropriate function in parent object (setIconFolder\(...\)) to load other
icons to interface.

void GeneralSettingDialog::iconsChangedIndex (
    int index )[signal] - signal which are invoked by iconChanged\(...\) function to call appropriate
function in parent object (setIconFolder\(...\)) to load other icons to interface.

void GeneralSettingDialog::imageRequest (
    bool enable,
    bool req = false )[signal] - signal what can be generated by checkBoxUseBackgr and pButtonSelectImg. Parameter bool enable used to enable or disable background on a main interface, and parameter
bool req used to invoke (or not if false) QFileDialog to get background image file name. Handled in parent object
with setBackground\(...\) function.

void GeneralSettingDialog::langChanged (
    int index )[private], [slot] - function invoked by QListView widget on changing selected item
and emit signal langChangedIndex\(...\) to call appropriate function in parent object (getLangFile\(...\)) to change
interface language.

void GeneralSettingDialog::langChangedIndex (
    int index )[signal] - signal which invoked by langChanged\(...\) function to call appropriate func-
tion in parent object (getLangFile\(...\)) to change interface language.

void GeneralSettingDialog::lockUnlockEmail ( ) [private], [slot] - function called by QPushButton on
mail setting tab and used to unlock (or lock) mail settings.

void GeneralSettingDialog::machineParamChanged (
    QByteArray machineParamArr )[signal] - signal which is emitted by function accept\(...\) and
handle by getMachineParam\(...\) in parent object.

void GeneralSettingDialog::machineTypeChanget (
    int index )[private], [slot] - empty function.

void GeneralSettingDialog::on_checkBoxUseBackgr_clicked ( ) [private], [slot] - function connected
to clicked() signal of check box to send signal imageRequest\(...\) with appropriate parameters.

void GeneralSettingDialog::on_pButtonHeadsActivation_clicked ( ) [private], [slot] - function to
emit signal headActivationRequest\(...\) which handle in parent object and call HeadActivationDialog class object.

void GeneralSettingDialog::on_pButtonSelectImg_clicked ( ) [private], [slot] - function connected
to clicked() signal of button to send signal imageRequest\(...\) with appropriate parameters.

void GeneralSettingDialog::reject ( ) [private], [slot] - function to hide dialog without saving settings.
Invoked by Cancel button.

void GeneralSettingDialog::sendCommand (
    QByteArray command )[signal] - signal to send data in QByteArray format to handle in parent
object and send it to serial port.

void GeneralSettingDialog::serialPortSettingsDialogRequested ( ) [signal] - signal which handle in
parent object and call SerialSettingsDialog class object.

void GeneralSettingDialog::serviceSettingRequest ( ) [signal] - signal emit by changeServiceState←
Clicked\(...\) function and handle in parent object (serviceStateChange\(...\)).

void GeneralSettingDialog::setEmailSettings() (
    EmailSettings emailSett ) - function to set mail parameters and fill appropriate field at dialog.

```

```

void GeneralSettingDialog::setFocusLossAccept() (
    bool flag ) - function to enable (or disable) hiding widget when user click beside widget.

void GeneralSettingDialog::setIconFolder() (
    QString path ) - function set folder with icons to change icons theme of current dialog.

void GeneralSettingDialog::setLangList() (
    QStringList langList,
    int curSelect ) - function to set language list to appropriate field (QListView) and select item
which are used in present moment.

void GeneralSettingDialog::setMachineSetting() (
    MachineSettings::MachineParameters machineParam ) - function to set machine parameters to
appropriate fields and set two-state buttons to appropriate state which are used in present moment.

void GeneralSettingDialog::setPasswords() (
    uint16_t serialPass,
    uint16_t mailPass,
    uint16_t userPass ) - function to set system passwords to private variables, which are used to
confirm password what user enter to access to some specific functions.

void GeneralSettingDialog::setStyle() (
    QStringList stList,
    int curSelect,
    QStringList iconList,
    int iconSel,
    bool backGrEn ) - function to fill widgets for selecting icons and style (QListView) and select item
which are used in present moment.

void GeneralSettingDialog::showEvent() (
    QShowEvent * ev ) [protected] - reimplementation of standard function. Used to hide or show
some widgets which are not available for user.

void GeneralSettingDialog::showPortInfo() (
    ComSettings comSett ) - function to set serial port parameters to appropriate fields at Serial Port
tab.

void GeneralSettingDialog::styleChanged (
    int index ) [private], [slot] - function which are called by QListView currentRowChanged(int)
signal and emit styleChangedIndex\(...\) signal.

void GeneralSettingDialog::styleChangedIndex (
    int index ) [signal] - signal which invoked by styleChanged\(...\) function to call appropriate func-
tion in parent object (getVeiwSettings\(...\)) to change interface colors and style.

void GeneralSettingDialog::unloadStateChanged (
    bool state ) [signal] - signal emitted by useUnloadStateChanged\(...\) function and handle in
parent object by getLoadState\(...\) function.

void GeneralSettingDialog::userSettingClicked ( ) [private], [slot] - function to emit usersSetting↔
Request\(...\) signal which are handle in parent object to show UserSettingDialog class object.

void GeneralSettingDialog::usersSettingRequest ( ) [signal] - signal invoked by useUnloadState↔
Changed\(...\) function which are handle in parent object to show UserSettingDialog class object.

void GeneralSettingDialog::useUnloadStateChanged ( ) [private], [slot] - function to emit unload↔
StateChanged\(...\) signal which are handle in parent object.

void GeneralSettingDialog::warningTimeChanged (
    double arg1 ) [private], [slot] - function to send warning time to machine using machine↔
ParamChanged\(...\) function.

```

### 7.8.3 Member Data Documentation

```
bool GeneralSettingDialog::acceptEnable[private]

bool GeneralSettingDialog::acceptOnDeactilationEn[private]

QIcon GeneralSettingDialog::directionIcon[private]

bool GeneralSettingDialog::loggedInMail[private]

bool GeneralSettingDialog::loggedInSerial[private]

bool GeneralSettingDialog::loggedInUserSett[private]

MachineSettings::MachineParameters GeneralSettingDialog::machineParamGl[private]

uint16_t GeneralSettingDialog::mailPassword[private]

QString GeneralSettingDialog::pathIcon[private]

uint16_t GeneralSettingDialog::serialPassword[private]

uint16_t GeneralSettingDialog::usersPassword[private]
```

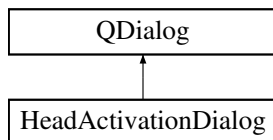
The documentation for this class was generated from the following files:

- [generalsettingdialog.h](#)
- [generalsettingdialog.cpp](#)

## 7.9 HeadActivationDialog Class Reference

```
#include <headactivationdialog.h>
```

Inheritance diagram for HeadActivationDialog:



### Signals

- void [sendCommand](#) (QByteArray cmd)

### Public Member Functions

- [HeadActivationDialog](#) (int headCount, QWidget \*parent=0)
- [~HeadActivationDialog](#) ()
- void [setHeadActivState](#) (uint32\_t state)
- [uint32\\_t](#) [getHeadActivState](#) ()
- bool [getHeadActivAtIndex](#) (int index)

### Protected Member Functions

- bool [event](#) (QEvent \*e)
- void [showEvent](#) (QShowEvent \*ev)
- void [changeEvent](#) (QEvent \*event)

### Private Slots

- void [pushButtonOK\\_clicked](#) ()
- void [pushButtonCancel\\_clicked](#) ()
- void [headStateChanged](#) (int index)

### Private Attributes

- Ui::HeadActivationDialog \* [ui](#)
- QList< [CheckBoxIndexed](#) \* > [checkBoxList](#)
- int [headCount](#)
- [uint32\\_t](#) [headActivState](#)
- bool [needSendReset](#)

#### 7.9.1 Constructor & Destructor Documentation

## 7.9.1.1 HeadActivationDialog()

`HeadActivationDialog::HeadActivationDialog (`  
     `int headCount,`  
     `QWidget * parent = 0 ) [explicit]` - class constructor. Used to create widget with [headCount](#) `QCheckBox` widgets to allow user activate (or deactivate) heads.

## 7.9.1.2 ~HeadActivationDialog()

`HeadActivationDialog::~HeadActivationDialog ( )` - standard class destructor.

## 7.9.2 Member Function Documentation

`void HeadActivationDialog::changeEvent() (`  
     `QEvent * event ) [protected]` - reimplementation of default event for `QDialog` to enable user interface translation. Function called automatically.

`bool HeadActivationDialog::event() (`  
     `QEvent * e ) [protected]` - reimplementation of standard function to handle `QEvent::WindowDeactivate` or `QEvent::Leave` for automatic invoke [pushButtonOK\\_clicked\(...\)](#) function on window deactivation.

`bool HeadActivationDialog::getHeadActivAtIndex() (`  
     `int index )` - function to get state of head according to given index. Return *bool* value: *true* if head active and *false* if it is not.

`uint32_t HeadActivationDialog::getHeadActivState() ( )` - function return state of heads in 32-bit variable. Position of bit describe head number.

`void HeadActivationDialog::headStateChanged (`  
     `int index ) [private], [slot]` - function which called by checkboxes on window (from [checkBoxList](#)). Function update [headActivState](#) value and send command to activate (or deactivate) heads to master board with [sendCommand](#) (`QByteArray cmd`) signal.

`void HeadActivationDialog::pushButtonCancel_clicked ( ) [private], [slot]` - function call by 'Cancel' button and used to hide window. Button, which call function, can be disabled if user must apply changes in settings.

`void HeadActivationDialog::pushButtonOK_clicked ( ) [private], [slot]` - function call by 'OK' button and used to gather data and hide window.

`void HeadActivationDialog::sendCommand (`  
     `QByteArray cmd ) [signal]` - signal to send data in `QByteArray` format to handle in parent object and send it to serial port.

`void HeadActivationDialog::setHeadActivState() (`  
     `uint32_t state )` - function to set [headActivState](#) value and set checkboxes on window (from [checkBoxList](#)) to appropriate state.

`void HeadActivationDialog::showEvent() (`  
     `QShowEvent * ev ) [protected]` - reimplementation of standard event handler; used for update state of [checkBoxList](#) items.

### 7.9.3 Member Data Documentation

`QList<CheckBoxIndexed*> HeadActivationDialog::checkBoxList [private]` - list of redefined QCheckBox class objects. All list items are showed on dialog and used to activate (or deactivate) heads.

`uint32_t HeadActivationDialog::headActivState [private]`

`int HeadActivationDialog::headCount [private]`

`bool HeadActivationDialog::needSendReset [private]`

The documentation for this class was generated from the following files:

- [headactivationdialog.h](#)
- [headactivationdialog.cpp](#)

## 7.10 HeadSetting::HeadComands\_ Struct Reference

```
#include <settings.h>
```

### Public Attributes

- [HeadCommandsEn headCmd](#)

### 7.10.1 Member Data Documentation

[HeadCommandsEn](#) `HeadSetting::HeadComands_::headCmd` - enumerator of commands which are send to head PCB's

The documentation for this struct was generated from the following file:

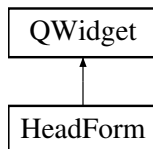
- [settings.h](#)



## 7.11 HeadForm Class Reference

```
#include <headform.h>
```

Inheritance diagram for HeadForm:



### Public Types

- enum [HeadformType\\_](#) { [HeadPuttingOn](#), [HeadRemoving](#), [HeadProcessing](#) }
- enum [HeadformState\\_](#) { [shirtOn](#), [shirtOff](#), [shirtProcessing](#) }
- enum [SettBtnPos\\_](#) { [AtLeftUp](#), [AtLeftDown](#), [AtRightUp](#), [AtRightDown](#) }
- typedef enum [HeadForm::HeadformType\\_](#) [HeadformType](#)
- typedef enum [HeadForm::HeadformState\\_](#) [HeadformState](#)
- typedef enum [HeadForm::SettBtnPos\\_](#) [SettBtnPos](#)

### Signals

- void [settingButtonCliced](#) (int [index](#))
- void [loadStateChanged](#) (int [index](#), [LoadState](#) state)

### Public Member Functions

- [HeadForm](#) (QWidget \*parent=0)
- [~HeadForm](#) ()
- void [setIndex](#) (int i)
- void [setIndexLabelPosition](#) ([SettBtnPos](#) position)
- void [setPixmap](#) ([HeadformState](#) state, QString stStr)
- void [setPixmap](#) ([HeadformState](#) state)
- void [setRagOn](#) (bool state)
- void [setIconPath](#) (QString path)
- void [setRagColor](#) (QColor color)
- void [setStrokCount](#) (int val)
- void [setStepBkStrCnt](#) (int val)
- void [setDryPower](#) (int val)
- void [setOff](#) ()
- [HeadformState](#) [getRagState](#) ()
- void [setHeadformType](#) ([HeadformType](#) type)
- [HeadformType](#) [getHeadformType](#) ()
- QSize [getLabelSize](#) ()

### Protected Member Functions

- void [mousePressEvent](#) (QMouseEvent \*event)

## Private Attributes

- `Ui::HeadForm * ui`
- `QLabel * labelIndex`
- `QLabel * labelStrokCnt`
- `QLabel * labelStBkStr`
- `QImage pixShirtShow`
- `QImage pixShirtHide`
- `QImage pixShirtAnimate`
- `QGraphicsColorizeEffect * graphEffect`
- `QString pathImage`
- `int index`
- `HeadformState headformState`
- `HeadformType headformType`
- `QGraphicsScene * grScene`
- `QGraphicsPixmapItem * grPixLogoItem`
- `QGraphicsRectItem * grRectBkgItem`
- `QRect grSize`
- `HeadForm::SettBtnPos labelPos`

## 7.11.1 Member Typedef Documentation

```
typedef enum HeadForm::HeadformState_ HeadForm::HeadformState
```

```
typedef enum HeadForm::HeadformType_ HeadForm::HeadformType
```

```
typedef enum HeadForm::SettBtnPos_ HeadForm::SettBtnPos
```

## 7.11.2 Member Enumeration Documentation

```
enum HeadForm::HeadformState_
```

## Enumerator

shirtOn	show rag image on form.
shirtOff	hide rag image on form.
shirtProcessing	show rag image animation on form.

```
enum HeadForm::HeadformType_
```

## Enumerator

HeadPuttingOn	paint load pallet to appropriate color.
HeadRemoving	paint unload pallet to appropriate color.
HeadProcessing	paint processing pallet to appropriate color.

```
enum HeadForm::SettBtnPos_ - describe position of labelIndex, labelStrokCnt and labelStBkStr
```

## Enumerator

AtLeftUp	
AtLeftDown	
AtRightUp	
AtRightDown	

### 7.11.3 Constructor & Destructor Documentation

#### 7.11.3.1 HeadForm()

`HeadForm::HeadForm ( QWidget * parent = 0 )` - constructor of class. Create QWidget class object, initialize QLabel 's, QGraphicsScene etc.

#### 7.11.3.2 ~HeadForm()

`HeadForm::~~HeadForm ( )` - standard class destructor.

### 7.11.4 Member Function Documentation

`HeadForm::HeadformType HeadForm::getHeadformType ( )` - function to get head type. Return one of `HeadformType_` values.

`QSize HeadForm::getLabelSize ( )` - return size of QGraphicsView object in pixels.

`HeadForm::HeadformState HeadForm::getRagState ( )` - return state of pallet (`HeadformState_` value).

`void HeadForm::loadStateChanged ( int index, LoadState state )` [signal] - signal what emit by `mousePressEvent` function if `headformType` is `HeadProcessing` on mouse click. Signal handle in parent object.

`void HeadForm::mousePressEvent ( QMouseEvent * event )` [protected] - reimplementation of standard mouse event handler. Used to change state of pallet with `setPixmap(...)` function and emit `loadStateChanged(...)` signal to handle in parent object by `getLoadState(...)` function.

`void HeadForm::setDryPower ( int val )` - set text to `labelStrokCnt` to display on screen.

`void HeadForm::setHeadformType ( HeadForm::HeadformType type )` - set variable `headformType` value to paint form to appropriate style and use it in future work.

`void HeadForm::setIconPath ( QString path )` - function to change folder with icons which used on view and update icons.

```
void HeadForm::setIndex (
    int i ) - function to set index value and put number on labelIndex. Used in future.
```

```
void HeadForm::setIndexLabelPosition (
    HeadForm::SetBtnPos position ) - function to set labelPos and set labelIndex, labelStrokCnt
and labelStBkStr labels positions.
```

```
void HeadForm::setOff ( ) - print "OFF" text on labelStrokCnt to indicate that head is turned off.
```

```
void HeadForm::setPixmap() [1/2] (
    HeadformState state,
    QString stStr ) - set value state to headformState variable and style of widget with stStr.
```

```
void HeadForm::setPixmap() [2/2] (
    HeadForm::HeadformState state ) - polymorphic function. Call setPixmap(HeadformState state,
QString stStr) with state and current style.
```

```
void HeadForm::setRagColor() (
    QColor color ) - set QGraphicsColorizeEffect to rag image.
```

```
void HeadForm::setRagOn() (
    bool state ) - function to set or remove rag image from pallet using setPixmap (...) function.
```

```
void HeadForm::setStepBkStrCnt() (
    int val ) - function to set text on labelStBkStr or hide it if val = 0.
```

```
void HeadForm::setStrokCount() (
    int val ) - function to set text on labelStrokCnt or hide it if val = 0.
```

```
void HeadForm::settingButtonCliced (
    int index ) [signal] - unused signal.
```

### 7.11.5 Member Data Documentation

[QGraphicsColorizeEffect\\*](#) [HeadForm::graphEffect](#) [private] - object of [QGraphicsColorizeEffect](#) class effect. Allow user to colorize items. Used under [grPixLogoItem](#) object.

[QGraphicsPixmapItem\\*](#) [HeadForm::grPixLogoItem](#) [private] - object of [QGraphicsPixmapItem](#) which are used to display image on [grScene](#).

[QGraphicsRectItem\\*](#) [HeadForm::grRectBkgrItem](#) [private] - object of [QGraphicsRectItem](#) contained by [grScene](#) to allow put background color (or color gradient) to graphic scene.

[QGraphicsScene\\*](#) [HeadForm::grScene](#) [private] - main graphic object on widget contained by [graphicsView](#) ([graphicsView](#) created in QtDesigner). Allow add all graphic object to widget, colorize them and change styles.

[QRect](#) [HeadForm::grSize](#) [private] - [QRect](#) class object which to create [grRectBkgrItem](#) object in constructor [HeadForm\(...\)](#)

[HeadformState](#) [HeadForm::headformState](#) [private] - variable to contain state of headForm. Sets by [setPixmap\(...\)](#) polymorphic function, get by [getRagState](#) function.

[HeadformType](#) [HeadForm::headformType](#) [private] - variable to contain type of headForm. Sets by [setHeadformType\(...\)](#) function, get by [getHeadformType](#) function.

[int](#) [HeadForm::index](#) [private] - variable to contain number of headForm. Used in [loadStateChanged](#) (...) to send it to parent object. Sets by [setIndex](#) function.

`QLabel* HeadForm::labelIndex[private]` - QLabel class object. Used to display number of headForm ([index](#) variable).

`HeadForm::SettBtnPos` `HeadForm::labelPos[private]` - variable to contain position of [labelIndex](#), [labelStrokCnt](#) and [labelStBkStr](#) labels positions. Sets by [setIndexLabelPosition\(...\)](#) function.

`QLabel * HeadForm::labelStBkStr[private]` - QLabel class object. Used to display strokes count in step back mode for current head.

`QLabel * HeadForm::labelStrokCnt[private]` - QLabel class object. Used to display strokes count for current head.

`QString HeadForm::pathImage[private]` - variable to contain path to image which are load to [pixShirtShow](#), [pixShirtHide](#) and [pixShirtAnimate](#). Sets by [setIconPath\(...\)](#) function.

- `QImage HeadForm::pixShirtAnimate[private]`
  - `QImage HeadForm::pixShirtHide[private]`
  - `QImage HeadForm::pixShirtShow[private]`
- QImage objects. Contain images to set it to [grPixLogolItem](#)

The documentation for this class was generated from the following files:

- [headform.h](#)
- [headform.cpp](#)

## 7.12 HeadSetting::HeadParameters\_ Struct Reference

```
#include <settings.h>
```

### Public Member Functions

- QByteArray [toByteArray](#) ()

### Public Attributes

- [uint16\\_t](#) headOnType
- [uint16\\_t](#) powerOn
- [uint16\\_t](#) speedRear
- [uint16\\_t](#) speedFront
- [uint16\\_t](#) stroksCount
- [uint16\\_t](#) stroksSBCount
- [uint16\\_t](#) limitRear
- [uint16\\_t](#) limitFront
- [uint16\\_t](#) dwellFLTime
- [uint16\\_t](#) dwellSQTime
- [uint16\\_t](#) heatTime1
- [uint16\\_t](#) heatTime2
- [uint16\\_t](#) heatPower
- [uint16\\_t](#) heatLimit
- [uint16\\_t](#) heatTime1Q
- [uint16\\_t](#) heatTime2Q
- [uint16\\_t](#) dryPowerQ
- [uint16\\_t](#) stepbackDryTimeQ
- [uint16\\_t](#) temperatureSetQ
- [uint16\\_t](#) dryTimeQ
- [uint16\\_t](#) standbyTimeQ
- [uint16\\_t](#) standbyPowerQ
- [uint16\\_t](#) warmFlashTimeQ
- [uint32\\_t](#) inkColor

### 7.12.1 Member Function Documentation

`QByteArray HeadSetting::HeadParameters_::toByteArray ( )` - function to gather all structure items to QByteArray to have possibility save data on hard drive in compact format.

### 7.12.2 Member Data Documentation

Fields of structure which contain parameters of head of machine.

- [uint16\\_t](#) `HeadSetting::HeadParameters_::dryPowerQ`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::dryTimeQ`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::dwellFLTime`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::dwellSQTime`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::headOnType`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::heatLimit`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::heatPower`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::heatTime1`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::heatTime1Q`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::heatTime2`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::heatTime2Q`
- [uint32\\_t](#) `HeadSetting::HeadParameters_::inkColor`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::limitFront`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::limitRear`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::powerOn`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::speedFront`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::speedRear`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::standbyPowerQ`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::standbyTimeQ`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::stepbackDryTimeQ`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::strokesCount`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::strokesSBCount`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::temperatureSetQ`
- [uint16\\_t](#) `HeadSetting::HeadParameters_::warmFlashTimeQ`

The documentation for this struct was generated from the following files:

- [settings.h](#)
- [settings.cpp](#)

## 7.13 Register::HeadReg\_ Union Reference

```
#include <settings.h>
```

### Classes

- struct [reg](#)

### Public Attributes

- struct [Register::HeadReg\\_::reg](#) field
- [uint16\\_t](#) memBeg

#### 7.13.1 Member Data Documentation

`struct Register::HeadReg\_::reg` `Register::HeadReg_::field` - fields of [Register::HeadReg\\_::reg](#) structure to give possibility to access to memory by name.

`uint16_t` `Register::HeadReg_::memBeg` - variable to notice beginning of memory region of structure.

The documentation for this union was generated from the following file:

- [settings.h](#)



## 7.14 HeadSetting Class Reference

```
#include <settings.h>
```

### Classes

- struct [HeadComands\\_](#)
- struct [HeadParameters\\_](#)

### Public Types

- enum {
  - [HeadDeviceAdrOffcet](#) = 0x0002
- enum {
  - [HeadOn](#) = 0x01,
  - [HeadSpeedRear](#) = 0x02,
  - [HeadSpeedFront](#) = 0x03,
  - [HeadFIDwellTime](#) = 0x04,
  - [HeadSqDwellTime](#) = 0x05,
  - [HeadStroksCount](#) = 0x06,
  - [HeadFlashTime1Q](#) = 0x07,
  - [HeadFlashTime2Q](#) = 0x08,
  - [HeadSBStroksCount](#) = 0x0E,
  - [HeadHoldOnOff](#) = 0x14,
  - [HeadRangeLimit1](#) = 0x16,
  - [HeadRangeLimit2](#) = 0x17,
  - [HeadHeatTemper](#) = 0x18,
  - [HeadHeatTimeIR](#) = 0x19,
  - [HeadFlashPowerStBy](#) = 0x1A,
  - [HeadFlashTimeStBy](#) = 0x1B,
  - [HeadFlashPowerWtoutIR](#) = 0x1D,
  - [HeadHeatWarmTime](#) = 0x1F,
  - [HeadFlashWarmTime](#) = 0x20,
  - [HeadHeadType](#) = 0x21,
  - [HeadPowerOn](#) = 0x01,
  - [HeadHeatDryRange](#),
  - [HeadHeatTime1IR](#),
  - [HeadHeatTime2IR](#)
- enum [HeadOnType\\_](#) {
  - [PrintHeadOff](#) = 0x0001,
  - [PrintHeadOn](#),
  - [QuartzHeadOff](#),
  - [QuartzHeadOn](#),
  - [InfraRedHeadOff](#),
  - [InfraRedHeadOn](#)
- enum [HeadCommandsEn\\_](#) {
  - [Idle](#) = 0,
  - [Head\\_OnOff](#) = 0x01F4,
  - [ChangeRagState](#) = 0x0190,
  - [MoveRear](#) = 0x0672,
  - [MoveFront](#),

```

    - MoveTest,
    - SQ,
    - FL,
    - SQ_FL,
    - Hold_Off,
    - Hold_On,
    - TeachPosition1,
    - TeachPosition2,
    - SQ_FL_UpDown,
    - MPT_Move,
    - PressureSQ = 0x005A,
    - AirRelease = 0x005A,
    - TimeSelect_1 = 0x025B,
    - TimeSelect_2,
    - TimeSelect_3,
    - Stepback,
    - Plast_water,
    - Preheat,
    - IndexHere,
    - SensorOn_Off,
    - TemperatureUnit,
    - WarmFlash,
    - HeatTest
}
• typedef enum HeadSetting::HeadOnType_ HeadOnType
• typedef struct HeadSetting::HeadParameters_ HeadParameters
• typedef enum HeadSetting::HeadCommandsEn_ HeadCommandsEn
• typedef struct HeadSetting::HeadComands_ HeadComands

```

#### Public Member Functions

- void [fromByteArray](#) (QByteArray hParamArr)
- [HeadParameters](#) operator= ([HeadParameters](#) hParam)
- [HeadSetting](#) ([HeadParameters](#) hParam)
- [HeadSetting](#) ()
- [~HeadSetting](#) ()

#### Static Public Member Functions

- static [uint16\\_t](#) [getHeadStateLo](#) ()
- static [uint16\\_t](#) [getHeadStateHi](#) ()
- static bool [getHeadStateAtIndex](#) ([uint8\\_t](#) index)
- static void [setHeadOn\\_OffStateInd](#) ([uint8\\_t](#) index, bool state)

#### Public Attributes

- [HeadParameters](#) [headParam](#)

#### Static Public Attributes

- static [uint32\\_t](#) [headStateAll](#)

### 7.14.1 Member Typedef Documentation

#### 7.14.1.1 HeadComands

```
typedef struct HeadSetting::HeadComands_ HeadSetting::HeadComands
```

#### 7.14.1.2 HeadCommandsEn

```
typedef enum HeadSetting::HeadCommandsEn_ HeadSetting::HeadCommandsEn
```

#### 7.14.1.3 HeadOnType

```
typedef enum HeadSetting::HeadOnType_ HeadSetting::HeadOnType
```

#### 7.14.1.4 HeadParameters

```
typedef struct HeadSetting::HeadParameters_ HeadSetting::HeadParameters
```

### 7.14.2 Member Enumeration Documentation

anonymous enum

Describe address of zero head. First head will have address 0x0003 and so on.

#### Enumerator

HeadDeviceAdrOffcet	0x0002
---------------------	--------

anonymous enum

Describe address of data places for every head. Name of enumerator item describe parameter which are sends to head PCB.

#### Enumerator

HeadOn	0x01
HeadSpeedRear	0x02
HeadSpeedFront	0x03
HeadFIDwellTime	0x04
HeadSqDwellTime	0x05
HeadStroksCount	0x06
HeadFlashTime1Q	0x07
HeadFlashTime2Q	0x08
HeadSBStroksCount	0x0E

## Enumerator

HeadHoldOnOff	0x14
HeadRangeLimit1	0x16
HeadRangeLimit2	0x17
HeadHeatTemper	0x18
HeadHeatTimeIR	0x19
HeadFlashPowerStBy	0x1A
HeadFlashTimeStBy	0x1B
HeadFlashPowerWtoutIR	0x1D
HeadHeatWarmTime	0x1F
HeadFlashWarmTime	0x20
HeadHeadType	0x21
HeadPowerOn	0x01
HeadHeatDryRange	ND
HeadHeatTime1IR	ND
HeadHeatTime2IR	ND

```
enum HeadSetting::HeadCommandsEn_
```

Enumerator, which written to create names of data what must be send to execute commands for head.

## Enumerator

Idle	0
Head_OnOff	0x01F4
ChangeRagState	0x0190
MoveRear	0x0762
MoveFront	0x0763
MoveTest	0x0764
SQ	0x0765
FL	0x0766
SQ_FL	0x0767
Hold_Off	0x0768
Hold_On	0x0769
TeachPosition1	0x076A
TeachPosition2	0x076B
SQ_FL_UpDown	0x076C
MPT_Move	0x076D
PressureSQ	0x005A
AirRelease	0x005B
TimeSelect_1	0x005C
TimeSelect_2	0x005D
TimeSelect_3	0x005E
Stepback	0x005F
Plast_water	0x0060
Preheat	0x0061
IndexHere	0x0062
SensorOn_Off	0x0063
TemperatureUnit	0x0064
WarmFlash	0x0065
HeatTest	0x0066

```
enum HeadSetting::HeadOnType_
```

Enumerator describe data which put to [headOnType](#) variable to set head type and head state.

#### Enumerator

PrintHeadOff	0x0001
PrintHeadOn	0x0002
QuartzHeadOff	0x0003
QuartzHeadOn	0x0004
InfraRedHeadOff	0x0005
InfraRedHeadOn	0x0006

### 7.14.3 Constructor & Destructor Documentation

```
HeadSetting::HeadSetting[1/2] (
```

[HeadParameters](#) *hParam* ) - constructor of class. In this variant of constructor new object fields of [headParam](#) variable will fill with *hParam* structure fields.

```
HeadSetting::HeadSetting[2/2] ( )
```

- constructor of class. In this variant of constructor new object fields of [headParam](#) variable will fill with default values.

```
HeadSetting::~~HeadSetting ( )
```

- standard class destructor.

### 7.14.4 Member Function Documentation

```
void HeadSetting::fromByteArray (
```

[QByteArray](#) *hParamArr* ) - function to fill fields of [headParam](#) variable with data obtained from *hParamArr* array.

```
bool HeadSetting::getHeadStateAtIndex (
```

[uint8\\_t](#) *index* ) [static] - function to get state (on or off) of head. State is reading from [headStateAll](#) variable. Parameter *index* describe number of bit in [headStateAll](#) which will be returned.

```
uint16_t HeadSetting::getHeadStateHi ( )
```

[static] - function to get two highest bytes of [headStateAll](#) variable.

```
uint16_t HeadSetting::getHeadStateLo ( )
```

[static] - function to get two lowest bytes of [headStateAll](#) variable.

```
HeadSetting::HeadParameters HeadSetting::operator= (
```

[HeadSetting::HeadParameters](#) *hParam* ) - reimplement of equality operator to copy *hParam* fields to fields of [headParam](#) variable.

```
void HeadSetting::setHeadOn_OffStateInd (
```

```
uint8_t index,
```

```
bool state ) [static] - function to set one bit of headStateAll to state at index position.
```

### 7.14.5 Member Data Documentation

```
HeadParameters HeadSetting::headParam
```

```
uint32\_t HeadSetting::headStateAll [static]
```

The documentation for this class was generated from the following files:

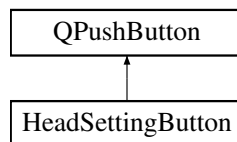
- [settings.h](#)
- [settings.cpp](#)

## 7.15 HeadSettingButton Class Reference

Reimplementation of class of QPushButton. Added [index](#) variable and signal [thisClicked\(...\)](#)

```
#include <headform.h>
```

Inheritance diagram for HeadSettingButton:



### Signals

- void [settingButtonCliced](#) (int [index](#))

### Public Member Functions

- [HeadSettingButton](#) (unsigned int inputNumber, QWidget \*parent=0)
- void [setIconPath](#) (QString path)

### Private Slots

- void [thisClicked](#) ()

### Private Attributes

- unsigned int [index](#)

#### 7.15.1 Constructor & Destructor Documentation

```
HeadSettingButton::HeadSettingButton (
    unsigned int inputNumber,
    QWidget * parent = 0 ) [inline] - constructor of class. Create widget and set index variable
```

#### 7.15.2 Member Function Documentation

```
void HeadSettingButton::setIconPath (
    QString path ) [inline] - function to update icon on button.
```

```
void HeadSettingButton::settingButtonCliced (
    int index ) [signal] signal what emitted by thisClicked function and handle in parent object by headSettingRequest\(...\) function.
```

```
void HeadSettingButton::thisClicked ( ) [inline], [private], [slot] - function which called by standard QPushButton clicked() signal.
```

#### 7.15.3 Member Data Documentation

```
unsigned int HeadSettingButton::index [private]
```

The documentation for this class was generated from the following file:

- [headform.h](#)

## 7.16 IndexerLiftSettings Class Reference

```
#include <settings.h>
```

### Classes

- struct [IndexParameters\\_](#)
- struct [LiftParameters\\_](#)

### Public Types

- enum [Devices](#) {
  - [LiftDevice](#) = 0x0002,
  - [IndexerDevice](#) = 0x0001
- enum [IndexerLiftDataPlaces](#) {
  - [IndexHomeOffset](#) = 0x01,
  - [IndexDistOffcet](#) = 0x02,
  - [IndexSpeed](#) = 0x03,
  - [IndexDirection](#) = 0x04,
  - [IndexStepTimeDelay](#) = 0x06,
  - [LiftDelayDown](#) = 0x1C,
  - [LiftDelayUp](#) = 0x07,
  - [IndexDistance](#) = 0x08,
  - [IndexLastRevolvWarm](#) = 0x09,
  - [IndexAcceleration](#) = 0x0A,
  - [IndexAccelerationRet](#) = 0x0B,
  - [IndexSpeedRet](#) = 0x0D,
  - [WarningTime](#) = 0x0E,
  - [LiftHomeOffcet](#) = 0x15,
  - [LiftDistance](#) = 0x16,
  - [LiftSpeed](#) = 0x17,
  - [LiftAcceleration](#) = 0x18,
  - [LoadHeadState](#) = 0x1F
- enum [IndexerCommandsEn\\_](#) {
  - [Machine\\_Reset](#) = 0x0001,
  - [Machine\\_Home](#) = 0x0009,
  - [IndexLock](#) = 0x000A,
  - [IndexUnLock](#) = 0x000A,
  - [MoveUp\\_Down](#) = 0x0006,
  - [MoveLeft](#) = 0x0005,
  - [MoveRight](#) = 0x0007,
  - [MoveLeftHalf](#) = 0x0005,
  - [MoveRightHalf](#) = 0x0007,
  - [MoveFull\\_Half](#) = 0x0004,
  - [Auto\\_Manual](#) = 0x0008,
  - [PrintStart](#) = 0x000B,
  - [PrintStop](#) = 0x000B,
  - [AirRelease](#) = 0x005A,
  - [EasySetup](#) = 0x08,
  - [IndexMoveHome](#) = 0x005B,
  - [IndexMoveEnd](#) = 0x005C,
  - [LiftMoveHome](#) = 0x00B8,
  - [LiftMoveEnd](#) = 0x00B9,
  - [IndexDirChange](#) = 0x005D
- typedef struct [IndexerLiftSettings::LiftParameters\\_](#) [LiftParameters](#)
- typedef struct [IndexerLiftSettings::IndexParameters\\_](#) [IndexParameters](#)
- typedef enum [IndexerLiftSettings::IndexerCommandsEn\\_](#) [IndexerCommandsEn](#)

## Public Member Functions

- [IndexerLiftSettings](#) ([IndexParameters](#) indParam, [LiftParameters](#) lifParam)
- [IndexerLiftSettings](#) ()
- void [fromByteArray](#) (QByteArray indParamArr, QByteArray lifParamArr)

## Public Attributes

- [IndexParameters](#) [indexerParam](#)
- [LiftParameters](#) [liftParam](#)

## 7.16.1 Member Typedef Documentation

```
typedef enum IndexerLiftSettings::IndexerCommandsEn_ IndexerLiftSettings::IndexerCommandsEn
```

```
typedef struct IndexerLiftSettings::IndexParameters_ IndexerLiftSettings::IndexParameters
```

```
typedef struct IndexerLiftSettings::LiftParameters_ IndexerLiftSettings::LiftParameters
```

## 7.16.2 Member Enumeration Documentation

enum [IndexerLiftSettings::Devices](#) - enumerator which contain device addresses of lift and indexer.

## Enumerator

<a href="#">LiftDevice</a>	0x0002
<a href="#">IndexerDevice</a>	0x0001

enum [IndexerLiftSettings::IndexerCommandsEn\\_](#) - enumerator, which written to create names of data what must be send to execute commands for indexer and lift device.

## Enumerator

<a href="#">Machine_Reset</a>	0x0001
<a href="#">Machine_Home</a>	0x0009
<a href="#">IndexLock</a>	0x000A
<a href="#">IndexUnLock</a>	0x000A
<a href="#">MoveUp_Down</a>	0x0006
<a href="#">MoveLeft</a>	0x0005
<a href="#">MoveRight</a>	0x0007
<a href="#">MoveLeftHalf</a>	0x0005
<a href="#">MoveRightHalf</a>	0x0007
<a href="#">MoveFull_Half</a>	0x0004
<a href="#">Auto_Manual</a>	0x0008
<a href="#">PrintStart</a>	0x000B
<a href="#">PrintStop</a>	0x000B
<a href="#">AirRelease</a>	0x005A
<a href="#">EasySetup</a>	0x0008
<a href="#">IndexMoveHome</a>	0x005B



### Enumerator

IndexMoveEnd	0x005C
LiftMoveHome	0x00B8
LiftMoveEnd	0x00B9
IndexDirChange	0x005D

enum `IndexerLiftSettings::IndexerLiftDataPlaces` - Enumerator describe address of data places for indexer and lift device. Name of enumerator item describe parameter which are sends to PCB.

### Enumerator

IndexHomeOffset	0x01
IndexDistOffcet	0x02
IndexSpeed	0x03
IndexDirection	0x04
IndexStepTimeDelay	0x06
LiftDelayDown	0x1C
LiftDelayUp	0x07
IndexDistance	0x08
IndexLastRevolvWarm	0x09
IndexAcceleration	0x0A
IndexAccelerationRet	0x0B
IndexSpeedRet	0x0D
WarningTime	0x0E
LiftHomeOffcet	0x15
LiftDistance	0x16
LiftSpeed	0x17
LiftAcceleration	0x18
LoadHeadState	0x1F

## 7.16.3 Constructor & Destructor Documentation

```
IndexerLiftSettings::IndexerLiftSettings[1/2] (
    IndexerLiftSettings::IndexParameters indParam,
    IndexerLiftSettings::LiftParameters lifParam ) - class constructor with data initialization.
Initialize indexerParam and liftParam with given data.
```

```
IndexerLiftSettings::IndexerLiftSettings[2/2] ( ) - empty class constructor. Initialize indexerParam and
liftParam with default parameters.
```

## 7.16.4 Member Function Documentation

```
void IndexerLiftSettings::fromByteArray (
    QByteArray indParamArr,
    QByteArray lifParamArr ) - fill fields of indexerParam and liftParam with data parsed from given
QByteArray's.
```

### 7.16.5 Member Data Documentation

[IndexParameters](#) IndexerLiftSettings::indexerParam

[LiftParameters](#) IndexerLiftSettings::liftParam

The documentation for this class was generated from the following files:

- [settings.h](#)
- [settings.cpp](#)

## 7.17 Register::IndexerReg\_ Union Reference

```
#include <settings.h>
```

### Classes

- struct [reg](#)

### Public Attributes

- struct [Register::IndexerReg\\_::reg](#) field
- [uint16\\_t](#) [memBeg](#)

#### 7.17.1 Member Data Documentation

[struct \[Register::IndexerReg\\\_::reg\]\(#\)](#) [Register::IndexerReg\\_::field](#) - fields of [Register::IndexerReg\\_::reg](#) structure to give possibility to access to memory by name.

[uint16\\_t](#) [Register::IndexerReg\\_::memBeg](#) - variable to notice beginning of memory region of structure.

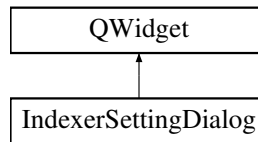
The documentation for this union was generated from the following file:

- [settings.h](#)

## 7.18 IndexerSettingDialog Class Reference

```
#include <indexersettingdialog.h>
```

Inheritance diagram for IndexerSettingDialog:



### Signals

- void [indexerParamChanged](#) (QByteArray indexParamArr)
- void [liftParamChanged](#) (QByteArray liftParamArr)
- void [sendCommand](#) (QByteArray command)
- void [liftDistanceChanged](#) (float distance)

### Public Member Functions

- [IndexerSettingDialog](#) (QWidget \*parent=0)
- [~IndexerSettingDialog](#) ()
- void [setRegisters](#) (Register \*reg)
- void [setIndexerSetting](#) (bool disconnect=true)
- void [setLiftSetting](#) (bool disconnect=true)
- void [setIndexerSetting](#) (IndexerLiftSettings::IndexParameters indexParam, bool disconnect=true)
- void [setLiftSetting](#) (IndexerLiftSettings::LiftParameters liftParam, bool disconnect=true)
- void [setLiftDistance](#) (float distance, int liftGearRatio)
- float [getLiftDistance](#) ()
- void [setLiftGearRatio](#) (uint32\_t liftGearRatio)

### Protected Member Functions

- bool [event](#) (QEvent \*e)
- bool [eventFilter](#) (QObject \*watched, QEvent \*event)
- void [showEvent](#) (QShowEvent \*ev)
- void [changeEvent](#) (QEvent \*event)

### Private Slots

- void [connectAll](#) ()
- void [disconnectAll](#) ()
- void [accept](#) ()
- void [reject](#) ()
- void [eventFilterSetup](#) ()
- void [dSpinBoxIndexDistance\\_valueChanged](#) (double arg1)
- void [spinBoxIndexHomeOffset\\_valueChanged](#) (double arg1)
- void [spinBoxIndexDistanceOffcet\\_valueChanged](#) (double arg1)
- void [spinBoxIndexSpeed\\_valueChanged](#) (double arg1)
- void [dSpinBoxIndexAccel\\_valueChanged](#) (double arg1)

- void [spinBoxindexSpeedRet\\_valueChanged](#) (double arg1)
- void [dSpinBoxIndexAccelRet\\_valueChanged](#) (double arg1)
- void [dSpinBoxLiftDownDelay\\_valueChanged](#) (double arg1)
- void [dSpinBoxLiftUpDelay\\_valueChanged](#) (double arg1)
- void [dSpinBoxLiftDistance\\_valueChanged](#) (double arg1)
- void [spinBoxLiftHomeOffset\\_valueChanged](#) (double arg1)
- void [spinBoxLiftSpeed\\_valueChanged](#) (double arg1)
- void [dSpinBoxLiftAccel\\_valueChanged](#) (double arg1)
- void [on\\_pButtonLiftMove\\_clicked](#) ()
- void [on\\_pButtonLiftHome\\_clicked](#) ()
- void [on\\_pButtonIndexMove\\_clicked](#) ()
- void [on\\_pButtonIndexHome\\_clicked](#) ()

#### Private Attributes

- Ui::IndexerSettingDialog \* [ui](#)
- bool [acceptOnDeactilationEn](#)
- bool [acceptEnable](#)
- [uint32\\_t](#) [liftGearRatio](#)
- [Register](#) \* [registers](#)

#### 7.18.1 Constructor & Destructor Documentation

`IndexerSettingDialog::IndexerSettingDialog ( QWidget * parent = 0 ) [explicit]` - standard QWidget constructor.

`IndexerSettingDialog::~~IndexerSettingDialog ( )` - standard QWidget destructor.

#### 7.18.2 Member Function Documentation

`void IndexerSettingDialog::accept ( ) [private], [slot]` - function to collect all data from QSpinBoxes to save it at hard drive. Function emit two signals with data [indexerParamChanged\(...\)](#) and [liftParamChanged\(...\)](#) which are handle in parent object. Function invoke by *OK* button and from [event \(...\)](#) when window lost focus or deactivated.

`void IndexerSettingDialog::changeEvent ( QEvent * event ) [protected]` - Reimplementation of default event for QWidget to enable user interface translation. Function called automatically.

`void IndexerSettingDialog::connectAll ( ) [private], [slot]` - function what is used to connect all wid-gets signals to appropriate functions.

`void IndexerSettingDialog::disconnectAll ( ) [private], [slot]` - function what is used to disconnect all widgets signals to appropriate functions.

---

All functions in next block used to send parameters to master PCB to configure machine. Method of data packing described in [Register](#) class reference.

---

- void IndexerSettingDialog::dSpinBoxIndexAccel\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::dSpinBoxIndexAccelRet\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::dSpinBoxIndexDistance\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::dSpinBoxLiftAccel\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::dSpinBoxLiftDistance\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::dSpinBoxLiftDownDelay\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::dSpinBoxLiftUpDelay\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::spinBoxIndexDistanceOffset\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::spinBoxIndexHomeOffset\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::spinBoxIndexSpeed\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::spinBoxIndexSpeedRet\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::spinBoxLiftHomeOffset\_valueChanged ( double *arg1* ) [private], [slot]
- void IndexerSettingDialog::spinBoxLiftSpeed\_valueChanged ( double *arg1* ) [private], [slot]

bool IndexerSettingDialog::event ( QEvent \* *e* ) [protected] - reimplementation of standard function to handle *QEvent::WindowDeactivate* or *QEvent::Leave* for automatic invoke of [accept\(...\)](#) function on window deactivation.

bool IndexerSettingDialog::eventFilter ( QObject \* *watched*, QEvent \* *event* ) [protected] - reimplementation of standard function to handle *QEvent::MouseButtonClick* or *QEvent::MouseButtonRelease* events to call [NumpadDialog](#) or [KeyboardDialog](#) to enter data to appropriate widgets. Widgets which will call this function are defined in [eventFilterSetup\(...\)](#) function.

void IndexerSettingDialog::eventFilterSetup ( ) [private], [slot] - function to configure widgets on dialog to use [eventFilter](#) (...).

float IndexerSettingDialog::getLiftDistance ( ) - function to get lift distance parameter. Used in parent object.

void IndexerSettingDialog::indexerParamChanged ( QByteArray *indexParamArr* ) [signal] - signal which is emitted by function [accept\(...\)](#) and handle by [getIndexerParam\(...\)](#) in parent object.

void IndexerSettingDialog::liftDistanceChanged ( float *distance* ) [signal] - signal which is emitted by function [dSpinBoxLiftDistance\\_valueChanged\(...\)](#) and handle by [getLiftDistance\(...\)](#) in parent object to update value at pallet distance spinBox.

void IndexerSettingDialog::liftParamChanged ( QByteArray *liftParamArr* ) [signal] - signal which is emitted by function [accept\(...\)](#) and handle by [getLiftParam\(...\)](#) in parent object.

Functions in next block used to send commands to master PCB to move parst of machine. Method of commands data packing described in [Register](#) class reference.

- `void IndexerSettingDialog::on_pButtonIndexHome_clicked ( ) [private], [slot]`
- `void IndexerSettingDialog::on_pButtonIndexMove_clicked ( ) [private], [slot]`
- `void IndexerSettingDialog::on_pButtonLiftHome_clicked ( ) [private], [slot]`
- `void IndexerSettingDialog::on_pButtonLiftMove_clicked ( ) [private], [slot]`

`void IndexerSettingDialog::reject ( ) [private], [slot]` - function to hide dialog without data saving.

`void IndexerSettingDialog::sendCommand ( QByteArray command ) [signal]` - signal to send data in QByteArray format to handle in parent object and send it to serial port.

`void IndexerSettingDialog::setIndexerSetting[1/2] ( bool disconnect = true )` - polymorphic function to set(or update) spinBox values on indexer tab of dialog . In this variant of function data will take from [Register::IndexerReg\\_::reg](#) and set to appropriate widgets at dialog. Boolean variable *disconnect* describe does program must call [disconnectAll\(...\)](#) function or not.

`void IndexerSettingDialog::setIndexerSetting() [2/2] ( IndexerLiftSettings::IndexParameters indexParam, bool disconnect = true )` - polymorphic function to set(or update) spinBox values on indexer tab of dialog. In this variant of function data will take from *indexParam* variable, given in parameters. Boolean variable *disconnect* describe does program must call [disconnectAll\(...\)](#) function or not.

`void IndexerSettingDialog::setLiftDistance() ( float distance, int liftGearRatio )` - function to update *lift distance* parameter with *distance* parameter.

`void IndexerSettingDialog::setLiftGearRatio ( uint32_t liftGearRatio )` - function to set value of [liftGearRatio](#) variable.

`void IndexerSettingDialog::setLiftSetting [1/2] ( bool disconnect = true )` - polymorphic function to set(or update) spinBox values on lift tab of dialog . In this variant of function data will take from [Register::LiftReg\\_::reg](#) and set to appropriate widgets at dialog. Boolean variable *disconnect* describe does program must call [disconnectAll\(...\)](#) function or not.

`void IndexerSettingDialog::setLiftSetting [2/2] ( IndexerLiftSettings::LiftParameters liftParam, bool disconnect = true )` - polymorphic function to set(or update) spinBox values on indexer tab of dialog. In this variant of function data will take from *liftParam* variable, given in parameters. Boolean variable *disconnect* describe does program must call [disconnectAll\(...\)](#) function or not.

`void IndexerSettingDialog::setRegisters ( Register * reg )` - function to set pointer to [Register](#) class object. In a whole program used only one sample of class and pointer ([registers](#)) set from parent object.

`void IndexerSettingDialog::showEvent ( QShowEvent * ev ) [protected]` - reimplementation of standard function. Used to hide or show some widgets which are not available for user.

### 7.18.3 Member Data Documentation

`bool IndexerSettingDialog::acceptEnable[private]`

`bool IndexerSettingDialog::acceptOnDeactilationEn[private]`

`uint32_t IndexerSettingDialog::liftGearRatio[private]`

`Register* IndexerSettingDialog::registers[private]`

The documentation for this class was generated from the following files:

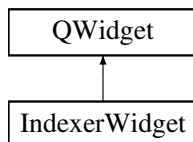
- [indexersettingdialog.h](#)
- [indexersettingdialog.cpp](#)



## 7.19 IndexerWidget Class Reference

```
#include <indexerwidget.h>
```

Inheritance diagram for IndexerWidget:



### Public Slots

- void [printFinish](#) ()
- void [printStart](#) ()
- void [setState](#) (u\_int16\_t state)
- void [resetWidget](#) ()

### Signals

- void [settingButtonClicked](#) ()
- void [sendCommand](#) (QByteArray command)
- void [resetRequest](#) ()
- void [startPrint](#) (bool isAuto)
- void [stopPrint](#) ()

### Public Member Functions

- [IndexerWidget](#) (QWidget \*parent=0)
- [~IndexerWidget](#) ()
- bool [getIsAutoPrint](#) ()
- void [setIconFolder](#) (QString path)
- void [clickButton](#) (QByteArray data)

### Protected Member Functions

- virtual void [resizeEvent](#) (QResizeEvent \*e)
- void [changeEvent](#) (QEvent \*event)

### Private Slots

- void [on\\_pButtonLock\\_clicked](#) ()
- void [on\\_pButtonMove\\_clicked](#) ()
- void [on\\_pButtonAuto\\_clicked](#) ()
- void [on\\_pButtonPrint\\_clicked](#) ()
- void [on\\_pButtonMoveLeft\\_clicked](#) ()
- void [on\\_pButtonMoveUp\\_clicked](#) ()
- void [on\\_pButtonMoveRight\\_clicked](#) ()
- void [on\\_pButtonReset\\_clicked](#) ()
- void [on\\_pButtonHome\\_clicked](#) ()
- void [settingPButtonClicSlot](#) ()
- void [on\\_pButtonAir\\_clicked](#) ()

## Private Attributes

- `Ui::IndexerWidget * ui`
- `QPushButton * pButtonSets`
- `int halfCounter`
- `bool isAutoPrintEnable`
- `QString pathIcon`
- `MachineSettings::MachineState machineState`

## 7.19.1 Constructor &amp; Destructor Documentation

`IndexerWidget::IndexerWidget() ( QWidget * parent = 0 ) [explicit]` - standard QWidget constructor. In constructor initialized `pButtonSets`, resize it and move it to it's place.

`IndexerWidget::~IndexerWidget() ( )` - standard QWidget destructor.

## 7.19.2 Member Function Documentation

`void IndexerWidget::changeEvent ( QEvent * event ) [protected]` - Reimplementation of default event for QWidget to enable user interface translation. Function called automatically.

`void IndexerWidget::clickButton ( QByteArray data )` - function to imitate clicking of button. Which button will be clicked is selecting by contents of `data`.

`bool IndexerWidget::getIsAutoPrint ( )` - return state of `pButtonAuto`. Used in parent object.

---

Functions in next block used to send commands to master PCB to control machine. Data in package tacked from `IndexerCommandsEn` enumerator. Method of commands data packing described in `Register` class reference.

- `void IndexerWidget::on_pButtonAir_clicked ( ) [private], [slot]` -
  - `void IndexerWidget::on_pButtonAuto_clicked ( ) [private], [slot]`
  - `void IndexerWidget::on_pButtonHome_clicked ( ) [private], [slot]`
  - `void IndexerWidget::on_pButtonLock_clicked ( ) [private], [slot]`
  - `void IndexerWidget::on_pButtonMove_clicked ( ) [private], [slot]`
  - `void IndexerWidget::on_pButtonMoveLeft_clicked ( ) [private], [slot]`
  - `void IndexerWidget::on_pButtonMoveRight_clicked ( ) [private], [slot]`
  - `void IndexerWidget::on_pButtonMoveUp_clicked ( ) [private], [slot]`
  - `void IndexerWidget::on_pButtonPrint_clicked ( ) [private], [slot]`
  - `void IndexerWidget::on_pButtonReset_clicked ( ) [private], [slot]`
-

```
void IndexerWidget::printFinish ( ) [slot] - unused function.

void IndexerWidget::printStart ( ) [slot] - unused function.

void IndexerWidget::resetRequest ( ) [signal] - signal which emit by on\_pButtonReset\_clicked\(...\) function
and handle in parent object by resetMachine\(...\) function.

void IndexerWidget::resetWidget ( ) [slot] - function to return buttons on widget to initial state.

void IndexerWidget::resizeEvent() (
    QResizeEvent * e ) [protected], [virtual] - reimplementation of standard function to move
setting button to new place after widget resize. Function call automatically.

void IndexerWidget::sendCommand (
    QByteArray command ) [signal] - signal to send data in QByteArray format to handle in parent
object and send it to serial port.

void IndexerWidget::setIconFolder (
    QString path ) - function to change folder with icons which used on view and update icons. Icon
path written to pathIcon variable.

void IndexerWidget::setState (
    u_int16_t state ) [slot] - function to set state of widget. State describe which buttons will be
enabled or disabled, visible or hidden. State of widget sets by master board of machine, when Register::Master↔
Reg\_::reg field masterReg\_EKR updates its value.

void IndexerWidget::settingButtonClicked ( ) [signal] - signal which emitted by settingPButtonClickSlot\(...\)
function and handle in parent object by indexerLiftSettingRequest\(...\) function.

void IndexerWidget::settingPButtonClickSlot ( ) [private], [slot] - function which called by pButton↔
Sets click signal and emit settingButtonClicked\(...\) signal.

void IndexerWidget::startPrint (
    bool isAuto ) [signal] - signal which handle in parent object by startPrintProcess\(...\) function.

void IndexerWidget::stopPrint ( ) [signal] - signal which handle in parent object by stopPrintProcess\(...\)
function.
```

### 7.19.3 Member Data Documentation

```
int IndexerWidget::halfCounter[private] - variable which contain count of steps in half mode to hide/show
pButtonMove.

bool IndexerWidget::isAutoPrintEnable[private]

MachineSettings::MachineState IndexerWidget::machineState[private] - variable to contain state of ma-
chine.

QString IndexerWidget::pathIcon[private] - variable to contain path to icon folder.

QPushButton* IndexerWidget::pButtonSets[private] - QPushButton class object.
```

The documentation for this class was generated from the following files:

- [indexerwidget.h](#)
- [indexerwidget.cpp](#)

## 7.20 IndexerLiftSettings::IndexParameters\_ Struct Reference

```
#include <settings.h>
```

### Public Member Functions

- [QByteArray toByteArray \(\)](#)

### Public Attributes

- [uint16\\_t distance](#)
- [int16\\_t homeOffset](#)
- [int16\\_t distOffcet](#)
- [uint16\\_t speed](#)
- [uint16\\_t acceleration](#)
- [uint16\\_t speedRet](#)
- [uint16\\_t accelerationRet](#)

#### 7.20.1 Member Function Documentation

`QByteArray IndexerLiftSettings::IndexParameters_::toByteArray ()` - function to gather all structure items to QByteArray to have possibility save data on hard drive in compact format.

#### 7.20.2 Member Data Documentation

`uint16_t IndexerLiftSettings::IndexParameters_::acceleration`

`uint16_t IndexerLiftSettings::IndexParameters_::accelerationRet`

`uint16_t IndexerLiftSettings::IndexParameters_::distance`

`int16_t IndexerLiftSettings::IndexParameters_::distOffcet`

`int16_t IndexerLiftSettings::IndexParameters_::homeOffset`

`uint16_t IndexerLiftSettings::IndexParameters_::speed`

`uint16_t IndexerLiftSettings::IndexParameters_::speedRet`

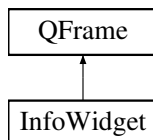
The documentation for this struct was generated from the following files:

- [settings.h](#)
- [settings.cpp](#)

## 7.21 InfoWidget Class Reference

```
#include <infowidget.h>
```

Inheritance diagram for InfoWidget:



### Public Member Functions

- [InfoWidget](#) (QWidget \*parent=0)
- [~InfoWidget](#) ()
- void [setRegisterPointer](#) ([Register](#) \*regPtr)
- void [setPrinted](#) (int val)
- void [setTotal](#) (int val)
- void [setIconFolder](#) (QString path)
- void [setIndicatorState](#) (u\_int16\_t state)
- void [setErrorText](#) ([MachineSettings::MachineParameters](#) machineParameters, [uint16\\_t](#) val)
- void [setText](#) (QString text)

### Protected Member Functions

- void [changeEvent](#) (QEvent \*event)

### Private Attributes

- Ui::InfoWidget \* [ui](#)
- QImage [imageHome](#)
- QImage [imageLock](#)
- QImage [imageUp](#)
- QImage [imageArrows](#)
- QImage [imageEmerg](#)
- QImage [imageWarning](#)
- QImage [imageStopHand](#)
- QString [pathIcon](#)
- QTime [lastTime](#)
- QGraphicsOpacityEffect \* [effect](#) [7]
- [Register](#) \* [localRegisters](#)
- QSettings \* [errMasages](#)

#### 7.21.1 Constructor & Destructor Documentation

`InfoWidget::InfoWidget (`  
     `QWidget * parent = 0 ) [explicit]` - standard class constructor. Function initialize QImages and load pictures into them, create QGraphicsOpacityEffects and set it to appropriate images, load error messages file to QSettings class object.

`InfoWidget::~~InfoWidget ( )` - standard class destructor.

## 7.21.2 Member Function Documentation

`void InfoWidget::changeEvent (   
     QEvent * event ) [protected]` - reimplementation of default event for QWidget to enable user interface translation. Function called automatically.

`void InfoWidget::setErrorText (   
     MachineSettings::MachineParameters machineParameters,   
     uint16_t val )` - function to display error message at *labelInfo*. Parameter *machineParameters* used to select text from *errMasages* object. Parameter *val* is unused. Error code, device and other info takes from *localRegisters :: masterReg\_DEVERR*, *masterReg\_ERR*, *masterReg\_ERROR\_MESSAGE* data at function invoke.

`void InfoWidget::setIconFolder (   
     QString path )` - function to change folder with icons which used on view and update icons. Icon path written to *pathIcon* variable.

`void InfoWidget::setIndicatorState (   
     u_int16_t state )` - function to set state of indicators on widget. State describe which indicators will be visible or hidden. State sets by master board of machine, when *Register::MasterReg\_::reg* field *masterReg\_EKR* updates its value.

`void InfoWidget::setPrinted() (   
     int val )` - function to update values on *labelPrinted* and on *labelDZH*. Value *labelDZH* updates automatically by time delay calculation between invokes of function.

`void InfoWidget::setRegisterPointer (   
     Register * regPtr )` - function to set pointer to *Register* class object. In a whole program used only one sample of class and pointer (*registers*) set from parent object.

`void InfoWidget::setText (   
     QString text )` - function to set custom text on *labelInfo*.

`void InfoWidget::setTotal (   
     int val )` - function to update values on *labelTotal*.

## 7.21.3 Member Data Documentation

`QGraphicsOpacityEffect* InfoWidget::effect[7] [private]`

`QSettings* InfoWidget::errMasages [private]`

`QImage InfoWidget::imageArrows [private]`

`QImage InfoWidget::imageEmerg [private]`

`QImage InfoWidget::imageHome [private]`

`QImage InfoWidget::imageLock [private]`

`QImage InfoWidget::imageStopHand [private]`

`QImage InfoWidget::imageUp [private]`

`QImage InfoWidget::imageWarning [private]`

`QTime InfoWidget::lastTime [private]`

`Register* InfoWidget::localRegisters [private]`

`QString InfoWidget::pathIcon [private]`

The documentation for this class was generated from the following files:

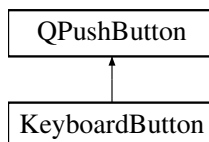
- [infowidget.h](#)
- [infowidget.cpp](#)

## 7.22 KeyboardButton Class Reference

Reimplementation of class of QPushButton. Added [character](#) variable and signal [clicked\(...\)](#)

```
#include <keyboarddialog.h>
```

Inheritance diagram for KeyboardButton:



### Signals

- void [clicked](#) (QString [character](#))

### Public Member Functions

- [KeyboardButton](#) (QString [character](#))
- QString [getCharacter](#) ()
- void [setCharacter](#) (QString str)

### Private Slots

- void [thisClicked](#) ()

### Private Attributes

- QString [character](#)

#### 7.22.1 Constructor & Destructor Documentation

```
KeyboardButton::KeyboardButton (
    QString character ) - constructor of class with character variable initialization. Value of parameter
sets like a text on button.
```

#### 7.22.2 Member Function Documentation

```
void KeyboardButton::clicked (
    QString character ) [signal] - signal which emit when button is clicked by thisClicked\(..\) func-
tion.
```

```
QString KeyboardButton::getCharacter ( ) - function to get character variable.
```

```
void KeyboardButton::setCharacter (
    QString str ) - function to set character variable.
```

```
void KeyboardButton::thisClicked ( ) [inline], [private], [slot] - function which called by standard
QPushButton::clicked() signal and emit clicked\(...\) signal.
```

#### 7.22.3 Member Data Documentation

```
QString KeyboardButton::character [private]
```

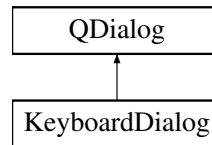
The documentation for this class was generated from the following files:

- [keyboarddialog.h](#)
- [keyboarddialog.cpp](#)

## 7.23 KeyboardDialog Class Reference

```
#include <keyboarddialog.h>
```

Inheritance diagram for KeyboardDialog:



### Public Types

- enum `KeyboardPosition_` { `Bottom`, `Custom` }
- typedef enum `KeyboardDialog::KeyboardPosition_` `KeyboardPosition`

### Signals

- void `keyClicked` ()

### Public Member Functions

- `KeyboardDialog` (`QWidget *parent=0`, `QString windowTitle="Keyboard"`)
- `~KeyboardDialog` ()
- void `buttonsToUpper` ()
- void `buttonsToLower` ()

### Static Public Member Functions

- static `QString getText` (`QWidget *parent=0`, `QString windowTitle="Keyboard"`, `KeyboardPosition position=Custom`)

### Protected Member Functions

- void `changeEvent` (`QEvent *event`)

### Private Slots

- void `appendToLineEdit` (`QString str`)
- void `backspace` ()
- void `submitText` ()
- void `shift` ()
- void `deshift` ()
- void `capsLock` ()
- void `someKeyClicked` ()
- void `dashUnderscoreSwitch` (`bool shifted=false`)



## Private Attributes

- QStringList [characters](#)
- Ui::KeyboardDialog \* [ui](#)
- QList< [KeyboardButton](#) \* > [buttons](#)
- [KeyboardButton](#) \* [dashUnderscoreButton](#)
- QString [text](#)
- bool [capsLockFlag](#)
- bool [shiftFlag](#)

### 7.23.1 Member Typedef Documentation

```
typedef enum KeyboardDialog::KeyboardPosition\_ KeyboardDialog::KeyboardPosition
```

### 7.23.2 Member Enumeration Documentation

```
enum KeyboardDialog::KeyboardPosition\_
```

#### Enumerator

Bottom	
Custom	

### 7.23.3 Constructor & Destructor Documentation

```
KeyboardDialog::KeyboardDialog (
    QWidget * parent = 0,
    QString windowTitle = "Keyboard" ) [explicit] - constructor of class. In function fill charac-
ters and buttons lists, create connection for functional buttons, and make layout of that widget on window. Parameter
windowTitle used to put it on window top panel.
```

```
KeyboardDialog::~~KeyboardDialog ( ) - standard QDialog destructor.
```

### 7.23.4 Member Function Documentation

```
void KeyboardDialog::appendToLineEdit (
    QString str ) [private], [slot] - function which is connected to every object in buttons list and
put appropriate symbol at last position at lineEdit. Symbol contains in parameter str.
```

```
void KeyboardDialog::backspace ( ) [private], [slot] - function for removing last symbol at lineEdit. in-
voked by pbBackspace.
```

```
void KeyboardDialog::buttonsToLower ( ) - function to change text on keyboard and symbol to lower register.
```

```
void KeyboardDialog::buttonsToUpper ( ) - function to change text on keyboard and symbol to higher register.
```

```
void KeyboardDialog::capsLock ( ) [private], [slot] - function to call buttonsToUpper\(...\) or buttonsTo↵
Lower(...) function appropriate to state of capsLockFlag and invert capsLockFlag value.
```

`void KeyboardDialog::changeEvent (   
     QEvent * event ) [protected]` - reimplementation of default event for QWidget to enable user interface translation. Function called automatically.

`void KeyboardDialog::dashUnderscoreSwitch (   
     bool shifted = false ) [private], [slot]` - function to change [dashUnderscoreButton](#) text and character.

`void KeyboardDialog::deshift ( ) [private], [slot]` - function for set [shiftFlag](#) value to false (disable "Shift" on keyboard), and set text on buttons to appropriate state

`QString KeyboardDialog::getText (   
     QWidget * parent = 0,   
     QString windowTitle = "Keyboard",   
     KeyboardPosition position = Custom ) [static]` - static function of [KeyboardDialog](#) class. Used to get *QString* text value without visible object creation. Return text which was putted with [buttons](#) to *QLineEdit* after call [submitText\(...\)](#) function.

`void KeyboardDialog::keyClicked ( ) [signal]` - signal which emitted at clicking of any object from [buttons](#) list.

`void KeyboardDialog::shift ( ) [private], [slot]` - function for set [shiftFlag](#) value to true (enable "Shift" on keyboard), and set text on buttons to appropriate state.

`void KeyboardDialog::someKeyClicked ( ) [private], [slot]` - function which called by any object from [buttons](#) list. Emit [keyClicked \(...\)](#) signal.

`void KeyboardDialog::submitText ( ) [private], [slot]` - function which invoked by *click()* signal from *pbEnter*. Used to copy text to [text](#) variable and call standard *QDialog::accept()* function to exit from keyboard dialog.

### 7.23.5 Member Data Documentation

`QList<KeyboardButton>` `KeyboardDialog::buttons` [private] - array of [KeyboardButton](#) class objects. Used to add keyboard buttons to dialog interface.

`bool` `KeyboardDialog::capsLockFlag` [private]

`QStringList` `KeyboardDialog::characters` [private]

`KeyboardButton*` `KeyboardDialog::dashUnderscoreButton` [private]

`bool` `KeyboardDialog::shiftFlag` [private]

`QString` `KeyboardDialog::text` [private] - variable for containing text from *lineEdit*. Used in [getText\(...\)](#) functions.

The documentation for this class was generated from the following files:

- [keyboarddialog.h](#)
- [keyboarddialog.cpp](#)

## 7.24 MachineSettings::LastRevolverWarm\_ Union Reference

Data union for simplify writing data to [Register::IndexerReg::reg](#) at [indexerReg\\_TM](#) field.

```
#include <settings.h>
```

### Public Attributes

- struct {
  - [uint8\\_t](#) multiple:1
  - [uint8\\_t](#) last:1
  - [uint8\\_t](#) revolver:1
  - [uint8\\_t](#) warm:1
  - [uint8\\_t](#) res:4
- [uint8\\_t](#) all

### 7.24.1 Member Data Documentation

[uint8\\_t](#) MachineSettings::LastRevolverWarm\_::all - variable which contain all bits in one byte in right positions.

```
struct { ... } MachineSettings::LastRevolverWarm_::field
```

Fields of structure.

- [uint8\\_t](#) MachineSettings::LastRevolverWarm\_::last
- [uint8\\_t](#) MachineSettings::LastRevolverWarm\_::multiple
- [uint8\\_t](#) MachineSettings::LastRevolverWarm\_::revolver
- [uint8\\_t](#) MachineSettings::LastRevolverWarm\_::warm

[uint8\\_t](#) MachineSettings::LastRevolverWarm\_::res - reserved bits.

The documentation for this union was generated from the following file:

- [settings.h](#)

## 7.25 IndexerLiftSettings::LiftParameters\_ Struct Reference

```
#include <settings.h>
```

### Public Member Functions

- [QByteArray](#) [toByteArray](#) ()

### Public Attributes

- [uint16\\_t](#) [distance](#)
- [int16\\_t](#) [homeOffcet](#)
- [uint16\\_t](#) [speed](#)
- [uint16\\_t](#) [acceleration](#)
- [uint16\\_t](#) [delayDown](#)
- [uint16\\_t](#) [delayUp](#)

### 7.25.1 Member Function Documentation

[QByteArray](#) [IndexerLiftSettings::LiftParameters\\_::toByteArray](#) ( ) - function to gather all structure items to [QByteArray](#) to have possibility save data on hard drive in compact format.

### 7.25.2 Member Data Documentation

[uint16\\_t](#) [IndexerLiftSettings::LiftParameters\\_::acceleration](#)

[uint16\\_t](#) [IndexerLiftSettings::LiftParameters\\_::delayDown](#)

[uint16\\_t](#) [IndexerLiftSettings::LiftParameters\\_::delayUp](#)

[uint16\\_t](#) [IndexerLiftSettings::LiftParameters\\_::distance](#)

[int16\\_t](#) [IndexerLiftSettings::LiftParameters\\_::homeOffcet](#)

[uint16\\_t](#) [IndexerLiftSettings::LiftParameters\\_::speed](#)

The documentation for this struct was generated from the following files:

- [settings.h](#)
- [settings.cpp](#)

## 7.26 Register::LiftReg\_ Union Reference

```
#include <settings.h>
```

### Classes

- struct [reg](#)

### Public Attributes

- struct [Register::LiftReg\\_::reg](#) field
- [uint16\\_t](#) [memBeg](#)

#### 7.26.1 Member Data Documentation

`struct Register::LiftReg\_::reg` `Register::LiftReg_::field` - fields of [Register::LiftReg\\_::reg](#) structure to give possibility to access to memory by name.

`uint16_t` `Register::LiftReg_::memBeg`- variable to notice beginning of memory region of structure.

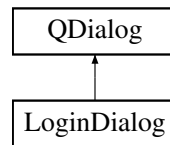
The documentation for this union was generated from the following file:

- [settings.h](#)

## 7.27 LoginDialog Class Reference

```
#include <logindialog.h>
```

Inheritance diagram for LoginDialog:



### Public Member Functions

- [LoginDialog](#) (QWidget \*parent=0)
- [~LoginDialog](#) ()
- void [setUserNames](#) (QStringList names)

### Public Attributes

- QString [userName](#)
- QString [userPassword](#)

### Protected Member Functions

- bool [eventFilter](#) (QObject \*watched, QEvent \*event)
- void [changeEvent](#) (QEvent \*event)

### Private Slots

- void [loginAccepted](#) ()

#### 7.27.1 Constructor & Destructor Documentation

```
LoginDialog::LoginDialog() (
    QWidget * parent = 0 ) [explicit] - standard QDialog constructor. In constructor connected
    pButtonLogin to loginAccepted\(...\) function and installed eventFilter to lineEditPassword.
```

```
LoginDialog::~~LoginDialog() ( ) - standard destructor.
```

### 7.27.2 Member Function Documentation

`void LoginDialog::changeEvent (`  
    `QEvent * event ) [protected]` - Reimplementation of default event for QWidget to enable user interface translation. Function called automatically.

`bool LoginDialog::eventFilter (`  
    `QObject * watched,`  
    `QEvent * event ) [protected]` - reimplementation of standard function to handle `QEvent::MouseButtonDblClick` or `QEvent::MouseButtonRelease` events to call [KeyboardDialog](#) to enter data to *lineEditPassword*.

`void LoginDialog::loginAccepted ( ) [private], [slot]` - function which called by *pButtonLogin* button and save use name and password to [userName](#) and [userPassword](#) variables and call standard `QDialog::accept()` function to exit from dialog. Parameters [userName](#) and [userPassword](#) are checked in [userLogin\(...\)](#) function.

`void LoginDialog::setUserNames (`  
    `QStringList names )` - function used to fill *listWidget* with *names*. Used to give possibility select user name from list.

### 7.27.3 Member Data Documentation

`QString LoginDialog::userName`

`QString LoginDialog::userPassword`

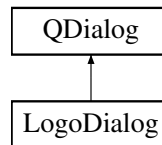
The documentation for this class was generated from the following files:

- [logindialog.h](#)
- [logindialog.cpp](#)

## 7.28 LogoDialog Class Reference

```
#include <logodialog.h>
```

Inheritance diagram for LogoDialog:



### Public Member Functions

- [LogoDialog](#) (QWidget \*parent=0)
- [~LogoDialog](#) ()

### Protected Member Functions

- void [showEvent](#) (QShowEvent \*ev)
- void [changeEvent](#) (QEvent \*event)

### Private Slots

- void [timerTimeOut](#) ()

### Private Attributes

- Ui::LogoDialog \* [ui](#)
- QTimer \* [timer](#)
- QMovie \* [movie](#)

#### 7.28.1 Constructor & Destructor Documentation

`LogoDialog::LogoDialog() (`  
     `QWidget * parent = 0 ) [explicit]` - standard dialog constructor. At constructor is loading *gif* animation to [movie](#) and creates [timer](#) which is connected to [timerTimeOut\(...\)](#) function.

`LogoDialog::~~LogoDialog() ( )` - standard QDialog destructor.

#### 7.28.2 Member Function Documentation

`void LogoDialog::changeEvent() (`  
     `QEvent * event ) [protected]` - reimplementation of default event for QWidget to enable user interface translation. Function called automatically.

`void LogoDialog::showEvent() (`  
     `QShowEvent * ev ) [protected]` - reimplementation of standard QDialog function. Used to start playing [movie](#) and start [timer](#).

`void LogoDialog::timerTimeOut ( ) [private], [slot]` - function to close dialog after some count of function calling.



### 7.28.3 Member Data Documentation

QMovie\* LogoDialog::movie [private]

QTimer\* LogoDialog::timer [private]

The documentation for this class was generated from the following files:

- [logodialog.h](#)
- [logodialog.cpp](#)

## 7.29 MachineSettings::MachineHeadType\_ Union Reference

Data union for simplify writing data about type of machine.

```
#include <settings.h>
```

### Public Attributes

- struct {
  - uint8\_t servoDriveType:3
  - uint8\_t carriageType:3
  - uint8\_t sqFlType:2 } field
- uint8\_t all

### 7.29.1 Member Data Documentation

uint8\_t MachineSettings::MachineHeadType\_::all - variable which contain all bits in one byte in right positions.

```
struct { ... } MachineSettings::MachineHeadType_::field
```

### Fields of structure

- uint8\_t MachineSettings::MachineHeadType\_::carriageType
- uint8\_t MachineSettings::MachineHeadType\_::servoDriveType
- uint8\_t MachineSettings::MachineHeadType\_::sqFlType

The documentation for this union was generated from the following file:

- [settings.h](#)

### 7.30 MachineSettings::MachineIndexLiftType\_ Union Reference

Data union for simplify writing data about type of machine.

```
#include <settings.h>
```

#### Public Attributes

- struct {
  - [uint16\\_t](#) mainServoDriveType:3
  - [uint16\\_t](#) indexerType:3
  - [uint16\\_t](#) liftType:3
  - [uint16\\_t](#) lockType:3
  - [uint16\\_t](#) hmilsConnected:1
  - [uint16\\_t](#) keypadIsConnected:1
  - [uint16\\_t](#) res\_:2
- [uint16\\_t](#) all

#### 7.30.1 Member Data Documentation

[uint16\\_t](#) MachineSettings::MachineIndexLiftType\_::all- variable which contain all bits in one byte in right positions.

```
struct { ... } MachineSettings::MachineIndexLiftType_::field
```

Fields of structure.

- [uint16\\_t](#) MachineSettings::MachineIndexLiftType\_::hmiIsConnected
- [uint16\\_t](#) MachineSettings::MachineIndexLiftType\_::indexerType
- [uint16\\_t](#) MachineSettings::MachineIndexLiftType\_::keypadIsConnected
- [uint16\\_t](#) MachineSettings::MachineIndexLiftType\_::liftType
- [uint16\\_t](#) MachineSettings::MachineIndexLiftType\_::lockType
- [uint16\\_t](#) MachineSettings::MachineIndexLiftType\_::mainServoDriveType

[uint16\\_t](#) MachineSettings::MachineIndexLiftType\_::res\_ - reserved bits.

The documentation for this union was generated from the following file:

- [settings.h](#)

## 7.31 MachineSettings::MachineParameters\_ Struct Reference

```
#include <settings.h>
```

### Public Member Functions

- [QByteArray toByteArray \(\)](#)

### Public Attributes

- [uint16\\_t headCount](#)
- [uint16\\_t warningTime](#)
- [int16\\_t direction](#)
- [MachineType machineType](#)
- [MachineHeadType headType](#)
- [MachineIndexLiftType indexeLiftType](#)
- [uint16\\_t headMaxRange](#)
- [uint16\\_t liftGearRatio](#)
- [uint16\\_t indexerScrewPinch](#)
- [uint8\\_t useUnloadHead](#)
- [uint8\\_t stepTimeDelay](#)
- [LastRevolverWarm lastRevWarm](#)

#### 7.31.1 Member Function Documentation

[QByteArray MachineSettings::MachineParameters\\_::toByteArray \(\)](#) - function to gather all structure items to QByteArray to have possibility save data on hard drive in compact format.

#### 7.31.2 Member Data Documentation

```
int16_t MachineSettings::MachineParameters_::direction
uint16_t MachineSettings::MachineParameters_::headCount
uint16_t MachineSettings::MachineParameters_::headMaxRange
MachineHeadType MachineSettings::MachineParameters_::headType
MachineIndexLiftType MachineSettings::MachineParameters_::indexeLiftType
uint16_t MachineSettings::MachineParameters_::indexerScrewPinch
LastRevolverWarm MachineSettings::MachineParameters_::lastRevWarm
uint16_t MachineSettings::MachineParameters_::liftGearRatio
MachineType MachineSettings::MachineParameters_::machineType
uint8_t MachineSettings::MachineParameters_::stepTimeDelay
uint8_t MachineSettings::MachineParameters_::useUnloadHead
uint16_t MachineSettings::MachineParameters_::warningTime
```

The documentation for this struct was generated from the following files:

- [settings.h](#)
- [settings.cpp](#)

## 7.32 MachineSettings Class Reference

```
#include <settings.h>
```

### Classes

- union [LastRevolverWarm\\_](#)
- union [MachineHeadType\\_](#)
- union [MachineIndexLiftType\\_](#)
- struct [MachineParameters\\_](#)
- union [MachineState\\_](#)

### Public Types

- enum [Devices](#) { [MasterDevice](#) = 0x0000 }
- enum [MasterDataPlaces](#) {
  - [MasterHeadCount](#) = 0x0001,
  - [MasterLastButton](#) = 0x0003,
  - [MasterIndexLiftCommand](#) = 0x0004,
  - [MasterHeadStateLo](#) = 0x0005,
  - [MasterHeadStateHi](#) = 0x0009,
  - [MasterPaletStateLo](#) = 0x000A,
  - [MasterPaletStateHi](#) = 0x0012,
  - [MasterMachineType](#) = 0x0011
- }
- enum [MachineType\\_](#) {
  - [VoltServo](#) = 0x0000,
  - [VoltAC](#),
  - [Vector](#),
  - [TitanASE](#),
  - [TitanASA](#),
  - [TitanAAA](#)
- }
- typedef union [MachineSettings::MachineState\\_ MachineState](#)
- typedef enum [MachineSettings::MachineType\\_ MachineType](#)
- typedef union [MachineSettings::MachineHeadType\\_ MachineHeadType](#)
- typedef union [MachineSettings::MachineIndexLiftType\\_ MachineIndexLiftType](#)
- typedef union [MachineSettings::LastRevolverWarm\\_ LastRevolverWarm](#)
- typedef struct [MachineSettings::MachineParameters\\_ MachineParameters](#)

### Public Member Functions

- [MachineSettings](#) ([MachineParameters](#) mParam)
- [MachineSettings](#) ()
- void [fromByteArray](#) (QByteArray machineParamArray)

## Static Public Member Functions

- static bool [getServiceWidgEn](#) ()
- static void [setServiceWidgEn](#) (bool servEn)
- static [MachineType](#) [getMachineType](#) ()
- static void [setMachineType](#) ([MachineType](#) mType)
- static bool [getMachineIdle](#) ()
- static void [setMachineIdle](#) (bool idle)
- static [uint16\\_t](#) [getHeadMaxRange](#) ()
- static [uint16\\_t](#) [getHeadType](#) ()
- static [uint16\\_t](#) [getIndexLiftType](#) ()
- static [uint16\\_t](#) [getHeadPalStateLo](#) ()
- static [uint16\\_t](#) [getHeadPalStateHi](#) ()
- static void [setHeadMaxRange](#) ([uint16\\_t](#) val)
- static void [setHeadType](#) ([uint16\\_t](#) val)
- static void [setIndexLiftType](#) ([uint16\\_t](#) val)
- static void [setHeadPalStateLo](#) ([uint16\\_t](#) val)
- static void [setHeadPalStateHi](#) ([uint16\\_t](#) val)
- static void [setHeadPalStateIndex](#) (int index, bool state)

## Public Attributes

- QStringList [machineTypeList](#)
- QList< int > [machineTypeData](#)
- [MachineParameters](#) [machineParam](#)

## Static Public Attributes

- static bool [serviceWidgetsEn](#)
- static [MachineType](#) [machineTypeStat](#)
- static bool [machineIdle](#)

## Static Private Attributes

- static [uint16\\_t](#) [headMaxRangeStat](#)
- static [uint16\\_t](#) [headTypeStat](#)
- static [uint16\\_t](#) [indexerLiftTypeStat](#)
- static [uint32\\_t](#) [headPalStateStat](#)

## 7.32.1 Member Typedef Documentation

typedef union [MachineSettings::LastRevolverWarm\\_](#) [MachineSettings::LastRevolverWarm](#) - union of data fields to write to [indexerReg\\_TM](#) by parameters names.

typedef union [MachineSettings::MachineHeadType\\_](#) [MachineSettings::MachineHeadType](#) - union of data to declare parameters of head.

typedef union [MachineSettings::MachineIndexLiftType\\_](#) [MachineSettings::MachineIndexLiftType](#) - union of data to declare parameters of lift.

typedef struct [MachineSettings::MachineParameters\\_](#) [MachineSettings::MachineParameters](#) - structure of data to save machine parameters to save it on hard drive.

typedef union [MachineSettings::MachineState\\_](#) [MachineSettings::MachineState](#) - union to save state of machine to use in program.

typedef enum [MachineSettings::MachineType\\_](#) [MachineSettings::MachineType](#) - enumerator to save type of machine to hide or show widgets on dialogs.

### 7.32.2 Member Enumeration Documentation

enum `MachineSettings::Devices` - enumerator to declare address of devices to send data to appropriated fields.

#### Enumerator

MasterDevice	0x0000
--------------	--------

enum `MachineSettings::MachineType_`

#### Enumerator

VoltServo	0x0000
VoltAC	0x0001
Vector	0x0002
TitanASE	0x0003
TitanASA	0x0004
TitanAAA	0x0005

enum `MachineSettings::MasterDataPlaces`

#### Enumerator

MasterHeadCount	0x0001
MasterLastButton	0x0003
MasterIndexLiftCommand	0x0004
MasterHeadStateLo	0x0005
MasterHeadStateHi	0x0009
MasterPaletStateLo	0x000A
MasterPaletStateHi	0x0012
MasterMachineType	0x0011

### 7.32.3 Constructor & Destructor Documentation

`MachineSettings::MachineSettings[1/2] (   
 MachineSettings::MachineParameters mParam )` - constructor of class to initialize fields of structure with given in *mParam* variable parameters.

`MachineSettings::MachineSettings[2/2] ( )` - constructor of class to initialize fields of structure with default parameters.

### 7.32.4 Member Function Documentation

`void MachineSettings::fromByteArray (   
 QByteArray machineParamArray )` - function to fill appropriated fields of class with data given in *machineParamArray*

`uint16_t MachineSettings::getHeadMaxRange ( ) [static]` - function to get value of `headMaxRangeStat` variable.

`uint16_t MachineSettings::getHeadPalStateHi ( ) [static]` - function to get highest 16 bit of `headPalStateStat` variable.

`uint16_t MachineSettings::getHeadPalStateLo ( ) [static]` - function to get lowest 16 bit of `headPalStateStat` variable.

`uint16_t MachineSettings::getHeadType ( ) [static]` - function to get value of `headTypeStat` variable.

`uint16_t MachineSettings::getIndexLiftType ( ) [static]` - function to get value of `indexerLiftTypeStat` variable.

`bool MachineSettings::getMachineIdle ( ) [static]` - function to get value of `machineIdle` variable.

`MachineSettings::MachineType MachineSettings::getMachineType ( ) [static]` - function to get value of `machineTypeStat` variable.

`bool MachineSettings::getServiceWidgEn ( ) [static]` - function to get value of `serviceWidgetsEn` variable.

`void MachineSettings::setHeadMaxRange ( uint16_t val ) [static]` - function to set value of `headMaxRangeStat` variable.

`void MachineSettings::setHeadPalStateHi ( uint16_t val ) [static]` - function to set highest 16 bit of `headPalStateStat` variable.

`void MachineSettings::setHeadPalStateIndex ( int index, bool state ) [static]` - function to set one bit (value *state*) of `headPalStateStat` variable at position given by *index*.

`void MachineSettings::setHeadPalStateLo ( uint16_t val ) [static]` - function to set lowest 16 bit of `headPalStateStat` variable.

`void MachineSettings::setHeadType ( uint16_t val ) [static]` - function to set value of `headTypeStat` variable.

`void MachineSettings::setIndexLiftType ( uint16_t val ) [static]` - function to set value of `indexerLiftTypeStat` variable.

`void MachineSettings::setMachineIdle ( bool idle ) [static]` - function to set value of `machineIdle` variable.

`void MachineSettings::setMachineType ( MachineSettings::MachineType mType ) [static]` - function to set value of `machineTypeStat` variable.

`void MachineSettings::setServiceWidgEn ( bool servEn ) [static]` - function to set value of `serviceWidgetsEn` variable.



### 7.32.5 Member Data Documentation

- `uint16_t` `MachineSettings::headMaxRangeStat[static]`, [private]
  - `uint16_t` `MachineSettings::headTypeStat[static]`, [private]
  - `uint16_t` `MachineSettings::indexerLiftTypeStat[static]`, [private]
- variables to describe types and parameters of machine parts.

`uint32_t` `MachineSettings::headPalStateStat[static]`, [private] - variable to describe state of pallets. Position of bit describe pallet number, state of bit at position mean information about containing rag on pallet.

`bool` `MachineSettings::machineIdle[static]` - contain information about machine state. Get *true* value when machine is ready to work, and *false* in any other case.

`MachineParameters` `MachineSettings::machineParam` - structure to contain parameters of the whole machine.

- `QList<int>` `MachineSettings::machineTypeData`
- `QStringList` `MachineSettings::machineTypeList`
- `MachineSettings::MachineType` `MachineSettings::machineTypeStat[static]`

block of variables which are used for filling comboBox with machine types.

`bool` `MachineSettings::serviceWidgetsEn[static]` - variable which used to describe state of service widgets and dialogs. Changing state need higher access privileges.

The documentation for this class was generated from the following files:

- [settings.h](#)
- [settings.cpp](#)

### 7.33 MachineSettings::MachineState\_ Union Reference

```
#include <settings.h>
```

Structure to save state of buttons of [IndexerWidget](#) object class. Created for future use.

#### Public Attributes

- struct {
  - uint8\_t printStop:1
  - uint8\_t reserved:1
  - uint8\_t lockUnLock:1
  - uint8\_t upDown:1
  - uint8\_t halfFull:1
  - uint8\_t reserved2:1
  - uint8\_t manualAuto:1
  - uint8\_t airRelease:1
  - uint8\_t easySetup:1
  - uint8\_t reserved3:7
- uint16\_t all

#### 7.33.1 Member Data Documentation

```
uint8_t MachineSettings::MachineState_::airRelease

uint16_t MachineSettings::MachineState_::all

struct { ... } MachineSettings::MachineState_::bit

uint8_t MachineSettings::MachineState_::easySetup

uint8_t MachineSettings::MachineState_::halfFull

uint8_t MachineSettings::MachineState_::lockUnLock

uint8_t MachineSettings::MachineState_::manualAuto

uint8_t MachineSettings::MachineState_::printStop

uint8_t MachineSettings::MachineState_::reserved

uint8_t MachineSettings::MachineState_::reserved2

uint8_t MachineSettings::MachineState_::reserved3

uint8_t MachineSettings::MachineState_::upDown
```

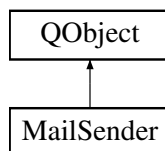
The documentation for this union was generated from the following file:

- [settings.h](#)

### 7.34 MailSender Class Reference

```
#include <mailsender.h>
```

Inheritance diagram for MailSender:



#### Public Slots

- void [setRecipientMailAddress](#) (QString mail="mishko.litvin@gmail.com")
- void [setEmailSubject](#) (QString subject="LiQt Machine interface autosend message")
- void [setRecipientName](#) (QString name="Customer")
- void [setSenderMailAddress](#) (QString mail="Liqt.autosend@gmail.com")
- void [setSenderPassword](#) (QString pswd="D7C5cqAz")
- void [setSenderName](#) (QString name="LiQt Interface")

#### Public Member Functions

- [MailSender](#) (QObject \*parent=0)
- int [sendMessage](#) (QString messageStr, QString subject, bool clear=true)
- int [sendMessage](#) (QString [message](#), bool clear=true)

#### Private Attributes

- SmtplibClient \* [smtp](#)
- MimeMessage \* [message](#)
- EmailAddress \* [senderAddr](#)
- EmailAddress \* [recipientAddr](#)
- MimeText \* [messageText](#)
- QString [messageSubject](#)

#### 7.34.1 Constructor & Destructor Documentation

```
MailSender::MailSender() (
    QObject * parent = 0 ) [explicit] - standard class constructor. Initialize mail sender class
with standard parameters.
```

## 7.34.2 Member Function Documentation

```
int MailSender::sendMessage [1/2] (
    QString messageStr,
    QString subject,
    bool clear = true ) - function to send message with messageStr and subject by e-mail.
```

```
int MailSender::sendMessage [2/2] (
    QString message,
    bool clear = true ) - function to send message with messageStr and subject recorded by
setEmailSubject\(...\) function with e-mail.
```

---

Block of function which used for setup parameters of e-mail messages.

- void MailSender::setEmailSubject (
 QString subject = "LiQt Machine interface autosend message" ) [slot]
  - void MailSender::setRecipientMailAddress (
 QString mail = "mishko.litvin@gmail.com" ) [slot]
  - void MailSender::setRecipientName (
 QString name = "Customer" ) [slot]
  - void MailSender::setSenderMailAddress (
 QString mail = "LiQt.autosend@gmail.com" ) [slot]
  - void MailSender::setSenderName (
 QString name = "LiQt Interface" ) [slot]
  - void MailSender::setSenderPassword (
 QString pswd = "D7C5cqAz" ) [slot]
- 

## 7.34.3 Member Data Documentation

```
MimeMessage* MailSender::message [private]

QString MailSender::messageSubject [private]

MimeText* MailSender::messageText [private]

EmailAddress* MailSender::recipientAddr [private]

EmailAddress* MailSender::senderAddr [private]

SmtpClient* MailSender::smtp [private]
```

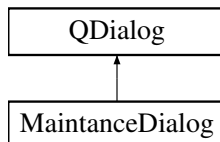
The documentation for this class was generated from the following files:

- [mailsender.h](#)
- [mailsender.cpp](#)

### 7.35 MaintanceDialog Class Reference

```
#include <maintancedialog.h>
```

Inheritance diagram for MaintanceDialog:



#### Public Types

- enum [MaintanceType](#) { [Warning](#), [Critical](#) }

#### Public Slots

- void [openMaintanceList](#) ()
- void [openDialog](#) ()
- void [solveItem](#) (int index)

#### Signals

- void [stopRequest](#) ()
- void [continueRequest](#) ()
- void [maintanceWorkEnable](#) (bool enabled)

#### Public Member Functions

- [MaintanceDialog](#) (QWidget \*parent=0)
- [MaintanceDialog](#) ([MaintanceType](#) mType, QWidget \*parent=0, QString name="trouble", QString troubleInfo="trouble info", QString machineInfo="machine info")
- [~MaintanceDialog](#) ()
- void [setTrableName](#) (QString name="trouble")
- void [setTrableInfo](#) (QString info="trouble info")
- void [setMachineInfo](#) (QString info="machine info")
- void [setMaintanceType](#) ([MaintanceType](#) mType=[Warning](#))
- QList< [MaintanceElement](#) > [getUnsolvedList](#) ()
- void [check](#) (int cyclesCount)

#### Static Public Member Functions

- static bool [execute](#) (QWidget \*parent=0, [MaintanceType](#) mType=[Warning](#), QString name="trouble", QString troubleInfo="trouble info", QString machineInfo="machine info")
- static bool [execute](#) ([MaintanceElement](#) maintance, QWidget \*parent=0)

#### Protected Member Functions

- void [changeEvent](#) (QEvent \*event)

## Private Slots

- void [acceptSlot](#) ()
- void [rejectSlot](#) ()

## Private Attributes

- Ui::MaintenanceDialog \* [ui](#)
- bool [doltNow](#)
- bool [maintenanceHaveWork](#)
- bool [maintenanceHaveWarning](#)
- QSettings \* [settings](#)
- QList< [MaintenanceElement](#) > [maintenanceList](#)
- QList< [MaintenanceElement](#) > [unsolvedList](#)
- QList< int > [unsolvedListIndex](#)
- [MaintenanceWidget](#) \* [maintenanceWidget](#)
- bool [firstCheck](#)
- int [lastCyclesCount](#)

## 7.35.1 Member Enumeration Documentation

## 7.35.1.1 MaintenanceType

enum [MaintenanceDialog::MaintenanceType](#)

## Enumerator

Warning	
Critical	

## 7.35.2 Constructor &amp; Destructor Documentation

[MaintenanceDialog](#)() [1/2] (  
     QWidget \* *parent* = 0 ) [explicit] - standard QDialog class constructor. Lists [maintenanceList](#), [unsolvedList](#) and [unsolvedListIndex](#) are fill in constructor.

[MaintenanceDialog](#)() [2/2] (  
     [MaintenanceDialog::MaintenanceType](#) *mType*,  
     QWidget \* *parent* = 0,  
     QString *name* = "trouble",  
     QString *trInfo* = "trouble info",  
     QString *machInfo* = "machine info" ) [explicit] - reimplemented constructor of class.  
 Used to fill appropriated fields on user interface.

[~MaintenanceDialog](#)() ( ) - standard QDialog class destructor.

### 7.35.3 Member Function Documentation

`void MaintanceDialog::acceptSlot ( ) [private], [slot]` - function used to set *true* to `doltNow` variable and exit from dialog;

`void MaintanceDialog::changeEvent ( QEvent * event ) [protected]` - reimplementation of default event for QDialog to enable user interface translation. Function called automatically.

`void MaintanceDialog::check ( int cyclesCount )` - function used to check does machine must stop for maintenance. Function called at every indexer step and check all maintenance items (from `maintenanceList`). Function emits `maintenanceWorkEnable(...)` signal with appropriate value.

`void MaintanceDialog::continueRequest ( ) [signal]` - unused.

`bool MaintanceDialog::execute [1/2] ( QWidget * parent = 0, MaintanceDialog::MaintanceType mType = Warning, QString name = "trouble", QString trInfo = "trouble info", QString machInfo = "machine info" ) [static]` - function used to call `MaintanceDialog(...)` constructor with appropriate parameters given in function.

`bool MaintanceDialog::execute [2/2] ( MaintanceElement maintenance, QWidget * parent = 0 ) [static]` - function used to call `MaintanceDialog(...)` constructor with parameters taken from `maintenance` variable.

`QList< MaintanceElement > MaintanceDialog::getUnsolvedList ( )` - function used to get `unsolvedList` variable value.

`void MaintanceDialog::maintenanceWorkEnable ( bool enabled ) [signal]` - signal which emitted by `check(...)` function and handle in parent object. Value which send with signal sets in function.

`void MaintanceDialog::openDialog ( ) [slot]` - function used to open dialogs fo every item in `unsolvedList`.

`void MaintanceDialog::openMaintenanceList ( ) [slot]`

`void MaintanceDialog::rejectSlot ( ) [private], [slot]` - function used to set *false* to `doltNow` variable and exit from dialog;

Block of function which used for setup parameters and fill appropriate fields of user interface.

- `void MaintanceDialog::setMachineInfo() ( QString info = "machine info" )`
- `void MaintanceDialog::setMaintenanceType() ( MaintanceDialog::MaintanceType mType = Warning )`
- `void MaintanceDialog::setTrableInfo ( QString info = "trouble info" )`
- `void MaintanceDialog::setTrableName ( QString name = "trouble" )`

`void MaintanceDialog::solveItem ( int index ) [slot]` - function used to remove one item (number *index*) from `unsolvedList` and save `unsolvedList` and `unsolvedListIndex` to settings. If `unsolvedListIndex` length is zero emits `maintenanceWorkEnable (...)` signal with *false* value.

`void MaintanceDialog::stopRequest ( ) [signal]` - unused.

#### 7.35.4 Member Data Documentation

```
bool MaintenanceDialog::doItNow [private]

bool MaintenanceDialog::firstCheck [private]

int MaintenanceDialog:lastCyclesCount [private]

bool MaintenanceDialog::maintenanceHaveWarning [private]

bool MaintenanceDialog::maintenanceHaveWork [private]

QList<MaintenanceElement> MaintenanceDialog::maintenanceList [private]

MaintenanceWidget* MaintenanceDialog::maintenanceWidget [private]

QSettings* MaintenanceDialog::settings [private]

QList<MaintenanceElement> MaintenanceDialog::unsolvedList [private]

QList<int> MaintenanceDialog::unsolvedListIndex [private]
```

The documentation for this class was generated from the following files:

- [maintancedialog.h](#)
- [maintancedialog.cpp](#)



## 7.36 MaintanceElement Class Reference

```
#include <maintancewidget.h>
```

Class which use to contain data about machine service. Variable names describe contents of the data stored in that variable.

### Public Member Functions

- [MaintanceElement](#) (int index=0, QString name="trouble", QString trInfo="trouble info", QString machInfo="machine info", int lastCount=0, int repeatCount=0)

### Public Attributes

- int [troubleIndex](#)
- int [troubleType](#)
- QString [troubleName](#)
- QString [troubleInfo](#)
- QString [machineInfo](#)
- int [lastCount](#)
- int [repeatCyclesCount](#)

### 7.36.1 Constructor & Destructor Documentation

#### 7.36.1.1 MaintanceElement()

```
MaintanceElement::MaintanceElement (
    int index = 0,
    QString name = "trouble",
    QString trInfo = "trouble info",
    QString machInfo = "machine info",
    int lastCount = 0,
    int repeatCount = 0 ) [inline]
```

### 7.36.2 Member Data Documentation

```
int MaintanceElement::lastCount

QString MaintanceElement::machineInfo

int MaintanceElement::repeatCyclesCount

int MaintanceElement::troubleIndex

QString MaintanceElement::troubleInfo

QString MaintanceElement::troubleName

int MaintanceElement::troubleType
```

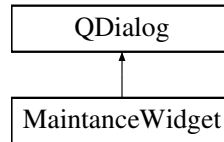
The documentation for this class was generated from the following file:

- [maintancewidget.h](#)

## 7.37 MaintenanceWidget Class Reference

```
#include <maintancewidget.h>
```

Inheritance diagram for MaintenanceWidget:



### Signals

- void [troubleSolved](#) (int troubleIndex)
- void [maintenanceClosed](#) (QList< [MaintenanceElement](#) > elemList)
- void [tutorialRequest](#) (int troubleIndex)

### Public Member Functions

- [MaintenanceWidget](#) (QWidget \*parent=0)
- [~MaintenanceWidget](#) ()
- void [clearText](#) ()
- void [setElemets](#) (QList< [MaintenanceElement](#) > elemList)

### Protected Member Functions

- void [changeEvent](#) (QEvent \*event)

### Private Slots

- void [solved](#) ()
- void [closeWindow](#) ()
- void [callTutorial](#) ()

### Private Attributes

- Ui::MaintenanceWidget \* [ui](#)
- QList< [MaintenanceElement](#) > [maintanceList](#)

#### 7.37.1 Constructor & Destructor Documentation

`MaintenanceWidget::MaintenanceWidget (`  
     `QWidget * parent = 0 )` [explicit] - standard QWidget constructor. Used to create connections from buttons to functions.

`MaintenanceWidget::~~MaintenanceWidget ( )` - standard QWidget destructor.

### 7.37.2 Member Function Documentation

`void MaintanceWidget::callTutorial ( ) [private], [slot]` - function created show (in some way) tutorial how to made maintenance for machine. It is not finished.

`void MaintanceWidget::changeEvent ( QEvent * event ) [protected]` - reimplementation of default event for QDialog to enable user interface translation. Function called automatically.

`void MaintanceWidget::clearText ( )` - function used to QListWidget (remove all items).

`void MaintanceWidget::closeWindow ( ) [private], [slot]` - used to hide QDialog.

`void MaintanceWidget::maintanceClosed ( QList< MaintanceElement > elemList ) [signal]` - signal what emits on window hide. Parameter send [maintanceList](#) to signal handler.

`void MaintanceWidget::setElemets ( QList< MaintanceElement > elemList )` - function which used to fill listWidget with maintenance elements and append *elemList* to [maintanceList](#).

`void MaintanceWidget::solved ( ) [private], [slot]` - function to remove selected at *listWidget* item from [maintanceList](#) and from *listWidget*. Also this function emit [troubleSolved\(...\)](#) signal with index at list to handle at parent object.

`void MaintanceWidget::troubleSolved ( int troubleIndex ) [signal]` - signal which handle at parent object (request redirected to [solvetem\(...\)](#) to change list of maintenance items at file system.

`void MaintanceWidget::tutorialRequest ( int troubleIndex ) [signal]` - signal which emitted by [callTutorial\(...\)](#) function. Handled in parent object (handler is not created at present moment).

### 7.37.3 Member Data Documentation

`QList<MaintanceElement> MaintanceWidget::maintanceList [private]`

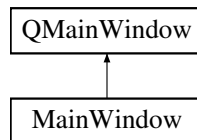
The documentation for this class was generated from the following files:

- [maintancewidget.h](#)
- [maintancewidget.cpp](#)

## 7.38 MainWindow Class Reference

```
#include <mainwindow.h>
```

Inheritance diagram for MainWindow:



## Public Member Functions

- [MainWindow](#) (QWidget \*parent=0)
- [~MainWindow](#) ()
- void [masterCodeCheck](#) ()
- void [userLogin](#) ()

## Public Attributes

- [ExitDialog::ExitCode](#) [exitCode](#)

## Protected Member Functions

- virtual void [resizeEvent](#) (QResizeEvent \*e)
- void [showEvent](#) (QShowEvent \*ev)
- bool [eventFilter](#) (QObject \*obj, QEvent \*ev)
- void [changeEvent](#) (QEvent \*event)

## Private Slots

- void [zeroStart](#) ()
- void [headsInit](#) ()
- void [watchDogTimeout](#) ()
- void [headSettingRequest](#) (int index)
- void [indexerLiftSettingRequest](#) ()
- void [generalSettingDialogRequest](#) ()
- void [changeHeadNo](#) (int index)
- void [resetMachine](#) ()
- void [getSerialData](#) ([ModData](#) modData)
- void [getUdpData](#) (QByteArray data)
- void [getHeadColor](#) (int index, QColor color)
- void [getHeadParam](#) (int index, QByteArray hParamArr)
- void [getAllHeadParam](#) (int index, QByteArray hParamArr)
- void [getHeadCommand](#) (int index, QByteArray commandArr)
- void [getHeadActivCommand](#) (QByteArray commandArr)
- void [getCyclesCommand](#) (QByteArray commandArr)
- void [getLoadState](#) (int index, [LoadState](#) state)
- void [getIndexerParam](#) (QByteArray indexerParamArr)
- void [getMachineParam](#) (QByteArray machineParamArr)

- void [getStepDelayTime](#) (double arg1)
- void [getDirection](#) (int direction)
- void [setUnloadState](#) (bool state)
- void [getLiftParam](#) (QByteArray liftParamArr)
- void [getLiftOffcet](#) (double arg1)
- void [getLiftDistance](#) (float distance)
- void [getIndexLiftSettComm](#) (QByteArray commandArr)
- void [getIndexLiftCommand](#) (QByteArray commandArr)
- void [getMachineCommand](#) (QByteArray commandArr)
- void [getSerialSetting](#) ([ComSettings](#) comSett)
- void [getEmailSettings](#) ([EmailSettings](#) emailSett)
- void [getVeiwSettings](#) (int stSheetIndex)
- void [getLangFile](#) (int langIndex)
- void [serviceStateChange](#) ()
- void [exitProgram](#) (bool restart=false)
- void [saveJob](#) ()
- void [loadJob](#) ()
- void [setHeadsPosition](#) ()
- void [indexerStepFinish](#) ()
- void [startPrintProcess](#) (bool autoPrint)
- void [stopPrintProcess](#) ()
- void [maintanceWorkSlot](#) (bool enable)
- void [showMaintananceWidget](#) ()
- void [setIconFolder](#) (int index)
- void [setBackground](#) (bool enable, bool request=false)
- void [updateTimeSlot](#) ()

#### Private Attributes

- Ui::MainWindow \* [ui](#)
- QList< [HeadForm](#) \* > [headButton](#)
- QList< [HeadSettingButton](#) \* > [headSettButton](#)
- [InfoWidget](#) \* [infoWidget](#)
- QSettings \* [settings](#)
- QSettings \* [masterCodes](#)
- [HeadSetting](#) [headSettings](#)
- [IndexerLiftSettings](#) [indexerLiftSettings](#)
- [MachineSettings](#) [machineSettings](#)
- [GeneralSettingDialog](#) \* [generalSettingDialog](#)
- [SettingDialog](#) \* [headSettingDialog](#)
- [IndexerSettingDialog](#) \* [indexerLiftSetDialog](#)
- [SerialSettingsDialog](#) \* [serialSettingsDialog](#)
- [IndexerWidget](#) \* [indexer](#)
- [MailSender](#) \* [mailSender](#)
- [UserSettingDialog](#) \* [usersSettingDialog](#)
- [MaintanceDialog](#) \* [maintanceDialog](#)
- [MaintanceWidget](#) \* [maintanceWidget](#)
- [CyclesDialog](#) \* [cycleDialog](#)
- [HeadActivationDialog](#) \* [headActDialog](#)
- [ReprogramDialog](#) \* [reprogramDialog](#)
- [SerialPort](#) \* [comPort](#)
- [UdpSocket](#) \* [udpHandler](#)
- QTimer \* [watchDog](#)
- QTimer \* [updateTime](#)

- QTime [timeProgramStart](#)
- QTime [timeProgramEnd](#)
- QTime [timeWorking](#)
- QString [truePassword](#)
- bool [loggedInHeadSettings](#)
- bool [loggedInIndexer](#)
- bool [loggedInGeneral](#)
- bool [loggedInService](#)
- uint16\_t [headsCount](#)
- bool [needCompleteReset](#)
- int [indexerCyclesSession](#)
- int [indexerCyclesAll](#)
- int [ragSessionCount](#)
- int [ragAllCount](#)
- uint32\_t [ragAllCountReg](#)
- int [ragAtHeadCount](#)
- bool [autoPrintEnabled](#)
- QString [userName](#)
- Register \* [registers](#)
- QTranslator [translator](#)
- QStringList [headStylesStr](#)

### 7.38.1 Constructor & Destructor Documentation

#### 7.38.1.1 MainWindow()

```
MainWindow::MainWindow (
    QWidget * parent = 0 ) [explicit]
```

#### 7.38.1.2 ~MainWindow()

```
MainWindow::~MainWindow ( )
```

### 7.38.2 Member Function Documentation

#### 7.38.2.1 changeEvent()

```
void MainWindow::changeEvent (
    QEvent * event ) [protected]
```

#### 7.38.2.2 changeHeadNo

```
void MainWindow::changeHeadNo (
    int index ) [private], [slot]
```

#### 7.38.2.3 eventFilter()

```
bool MainWindow::eventFilter (
    QObject * obj,
    QEvent * ev ) [protected]
```

#### 7.38.2.4 exitProgram

```
void MainWindow::exitProgram (
    bool restart = false ) [private], [slot]
```

#### 7.38.2.5 generalSettingDialogRequest

```
void MainWindow::generalSettingDialogRequest ( ) [private], [slot]
```

#### 7.38.2.6 getAllHeadParam

```
void MainWindow::getAllHeadParam (
    int index,
    QByteArray hParamArr ) [private], [slot]
```

#### 7.38.2.7 getCyclesCommand

```
void MainWindow::getCyclesCommand (
    QByteArray commandArr ) [private], [slot]
```

#### 7.38.2.8 getDirection

```
void MainWindow::getDirection (
    int direction ) [private], [slot]
```

#### 7.38.2.9 getEmailSettings

```
void MainWindow::getEmailSettings (
    EmailSettings emailSett ) [private], [slot]
```

#### 7.38.2.10 getHeadActivCommand

```
void MainWindow::getHeadActivCommand (
    QByteArray commandArr ) [private], [slot]
```

#### 7.38.2.11 getHeadColor

```
void MainWindow::getHeadColor (
    int index,
    QColor color ) [private], [slot]
```

#### 7.38.2.12 getHeadCommand

```
void MainWindow::getHeadCommand (
    int index,
    QByteArray commandArr ) [private], [slot]
```

#### 7.38.2.13 getHeadParam

```
void MainWindow::getHeadParam (
    int index,
    QByteArray hParamArr ) [private], [slot]
```

#### 7.38.2.14 getIndexerParam

```
void MainWindow::getIndexerParam (
    QByteArray indexerParamArr ) [private], [slot]
```

#### 7.38.2.15 getIndexLiftCommand

```
void MainWindow::getIndexLiftCommand (
    QByteArray commandArr ) [private], [slot]
```



#### 7.38.2.16 getIndexLiftSettComm

```
void MainWindow::getIndexLiftSettComm (
    QByteArray commandArr ) [private], [slot]
```

#### 7.38.2.17 getLangFile

```
void MainWindow::getLangFile (
    int langIndex ) [private], [slot]
```

#### 7.38.2.18 getLiftDistance

```
void MainWindow::getLiftDistance (
    float distance ) [private], [slot]
```

#### 7.38.2.19 getLiftOffcet

```
void MainWindow::getLiftOffcet (
    double arg1 ) [private], [slot]
```

#### 7.38.2.20 getLiftParam

```
void MainWindow::getLiftParam (
    QByteArray liftParamArr ) [private], [slot]
```

#### 7.38.2.21 getLoadState

```
void MainWindow::getLoadState (
    int index,
    LoadState state ) [private], [slot]
```

#### 7.38.2.22 getMachineCommand

```
void MainWindow::getMachineCommand (
    QByteArray commandArr ) [private], [slot]
```

**7.38.2.23 getMachineParam**

```
void MainWindow::getMachineParam (
    QByteArray machineParamArr ) [private], [slot]
```

**7.38.2.24 getSerialData**

```
void MainWindow::getSerialData (
    ModData modData ) [private], [slot]
```

**7.38.2.25 getSerialSetting**

```
void MainWindow::getSerialSetting (
    ComSettings comSett ) [private], [slot]
```

**7.38.2.26 getStepDelayTime**

```
void MainWindow::getStepDelayTime (
    double arg1 ) [private], [slot]
```

**7.38.2.27 getUdpData**

```
void MainWindow::getUdpData (
    QByteArray data ) [private], [slot]
```

**7.38.2.28 getVeiwSettings**

```
void MainWindow::getVeiwSettings (
    int stSheetIndex ) [private], [slot]
```

**7.38.2.29 headSettingRequest**

```
void MainWindow::headSettingRequest (
    int index ) [private], [slot]
```

**7.38.2.30 headsInit**

```
void MainWindow::headsInit ( ) [private], [slot]
```

**7.38.2.31 indexerLiftSettingRequest**

```
void MainWindow::indexerLiftSettingRequest ( ) [private], [slot]
```

**7.38.2.32 indexerStepFinish**

```
void MainWindow::indexerStepFinish ( ) [private], [slot]
```

**7.38.2.33 loadJob**

```
void MainWindow::loadJob ( ) [private], [slot]
```

**7.38.2.34 maintanceWorkSlot**

```
void MainWindow::maintanceWorkSlot (
    bool enable ) [private], [slot]
```

**7.38.2.35 masterCodeCheck()**

```
void MainWindow::masterCodeCheck ( )
```

**7.38.2.36 resetMachine**

```
void MainWindow::resetMachine ( ) [private], [slot]
```

**7.38.2.37 resizeEvent()**

```
void MainWindow::resizeEvent (
    QResizeEvent * e ) [protected], [virtual]
```

**7.38.2.38 saveJob**

```
void MainWindow::saveJob ( ) [private], [slot]
```

#### 7.38.2.39 serviceStateChange

```
void MainWindow::serviceStateChange ( ) [private], [slot]
```

#### 7.38.2.40 setBackGround

```
void MainWindow::setBackGround (
    bool enable,
    bool request = false ) [private], [slot]
```

#### 7.38.2.41 setHeadsPosition

```
void MainWindow::setHeadsPosition ( ) [private], [slot]
```

#### 7.38.2.42 setIconFolder

```
void MainWindow::setIconFolder (
    int index ) [private], [slot]
```

#### 7.38.2.43 setUnloadState

```
void MainWindow::setUnloadState (
    bool state ) [private], [slot]
```

#### 7.38.2.44 showEvent()

```
void MainWindow::showEvent (
    QShowEvent * ev ) [protected]
```

#### 7.38.2.45 showMaintananceWidget

```
void MainWindow::showMaintananceWidget ( ) [private], [slot]
```

#### 7.38.2.46 startPrintProcess

```
void MainWindow::startPrintProcess (
    bool autoPrint ) [private], [slot]
```

#### 7.38.2.47 stopPrintProcess

```
void MainWindow::stopPrintProcess ( ) [private], [slot]
```

#### 7.38.2.48 updateTimeSlot

```
void MainWindow::updateTimeSlot ( ) [private], [slot]
```

#### 7.38.2.49 userLogin()

```
void MainWindow::userLogin ( )
```

#### 7.38.2.50 watchDogTimeout

```
void MainWindow::watchDogTimeout ( ) [private], [slot]
```

#### 7.38.2.51 zeroStart

```
void MainWindow::zeroStart ( ) [private], [slot]
```

### 7.38.3 Member Data Documentation

#### 7.38.3.1 autoPrintEnabled

```
bool MainWindow::autoPrintEnabled [private]
```

#### 7.38.3.2 comPort

```
SerialPort* MainWindow::comPort [private]
```

#### 7.38.3.3 cycleDialog

```
CyclesDialog* MainWindow::cycleDialog [private]
```

#### 7.38.3.4 exitCode

```
ExitDialog::ExitCode MainWindow::exitCode
```

#### 7.38.3.5 generalSettingDialog

```
GeneralSettingDialog* MainWindow::generalSettingDialog [private]
```

#### 7.38.3.6 headActDialog

```
HeadActivationDialog* MainWindow::headActDialog [private]
```

#### 7.38.3.7 headButton

```
QList<HeadForm*> MainWindow::headButton [private]
```

#### 7.38.3.8 headsCount

```
uint16_t MainWindow::headsCount [private]
```

#### 7.38.3.9 headSettButton

```
QList<HeadSettingButton*> MainWindow::headSettButton [private]
```

#### 7.38.3.10 headSettingDialog

```
SettingDialog* MainWindow::headSettingDialog [private]
```

#### 7.38.3.11 headSettings

```
HeadSetting MainWindow::headSettings [private]
```

#### 7.38.3.12 headStylesStr

```
QStringList MainWindow::headStylesStr [private]
```

**7.38.3.13 indexer**

```
IndexerWidget* MainWindow::indexer [private]
```

**7.38.3.14 indexerCyclesAll**

```
int MainWindow::indexerCyclesAll [private]
```

**7.38.3.15 indexerCyclesSession**

```
int MainWindow::indexerCyclesSession [private]
```

**7.38.3.16 indexerLiftSetDialog**

```
IndexerSettingDialog* MainWindow::indexerLiftSetDialog [private]
```

**7.38.3.17 indexerLiftSettings**

```
IndexerLiftSettings MainWindow::indexerLiftSettings [private]
```

**7.38.3.18 infoWidget**

```
InfoWidget* MainWindow::infoWidget [private]
```

**7.38.3.19 loggedInGeneral**

```
bool MainWindow::loggedInGeneral [private]
```

**7.38.3.20 loggedInHeadSettings**

```
bool MainWindow::loggedInHeadSettings [private]
```

**7.38.3.21 loggedInIndexer**

```
bool MainWindow::loggedInIndexer [private]
```

#### 7.38.3.22 loggedInService

```
bool MainWindow::loggedInService [private]
```

#### 7.38.3.23 machineSettings

```
MachineSettings MainWindow::machineSettings [private]
```

#### 7.38.3.24 mailSender

```
MailSender* MainWindow::mailSender [private]
```

#### 7.38.3.25 maintanceDialog

```
MaintanceDialog* MainWindow::maintanceDialog [private]
```

#### 7.38.3.26 maintanceWidget

```
MaintanceWidget* MainWindow::maintanceWidget [private]
```

#### 7.38.3.27 masterCodes

```
QSettings* MainWindow::masterCodes [private]
```

#### 7.38.3.28 needCompleteReset

```
bool MainWindow::needCompleteReset [private]
```

#### 7.38.3.29 ragAllCount

```
int MainWindow::ragAllCount [private]
```

#### 7.38.3.30 ragAllCountReg

```
uint32_t MainWindow::ragAllCountReg [private]
```



**7.38.3.31 ragAtHeadCount**

```
int MainWindow::ragAtHeadCount [private]
```

**7.38.3.32 ragSessionCount**

```
int MainWindow::ragSessionCount [private]
```

**7.38.3.33 registers**

```
Register* MainWindow::registers [private]
```

**7.38.3.34 reprogramDialog**

```
ReprogramDialog* MainWindow::reprogramDialog [private]
```

**7.38.3.35 serialSettingsDialog**

```
SerialSettingsDialog* MainWindow::serialSettingsDialog [private]
```

**7.38.3.36 settings**

```
QSettings* MainWindow::settings [private]
```

**7.38.3.37 timeProgramEnd**

```
QTime MainWindow::timeProgramEnd [private]
```

**7.38.3.38 timeProgramStart**

```
QTime MainWindow::timeProgramStart [private]
```

**7.38.3.39 timeWorking**

```
QTime MainWindow::timeWorking [private]
```

**7.38.3.40 translator**

```
QTranslator MainWindow::translator [private]
```

**7.38.3.41 truePassword**

```
QString MainWindow::truePassword [private]
```

**7.38.3.42 udpHandler**

```
UdpSocket* MainWindow::udpHandler [private]
```

**7.38.3.43 ui**

```
Ui::MainWindow* MainWindow::ui [private]
```

**7.38.3.44 updateTime**

```
QTimer* MainWindow::updateTime [private]
```

**7.38.3.45 userName**

```
QString MainWindow::userName [private]
```

**7.38.3.46 usersSettingDialog**

```
UserSettingDialog* MainWindow::usersSettingDialog [private]
```

**7.38.3.47 watchDog**

```
QTimer* MainWindow::watchDog [private]
```

The documentation for this class was generated from the following files:

- [mainwindow.h](#)
- [mainwindow.cpp](#)

### 7.39 Register::MasterReg\_ Union Reference

```
#include <settings.h>
```

#### Classes

- struct [reg](#)

#### Public Attributes

- struct [Register::MasterReg\\_::reg](#) field
- [uint16\\_t](#) [memBeg](#)

#### 7.39.1 Member Data Documentation

`struct Register::MasterReg\_::reg` `Register::MasterReg_::field` - - fields of [Register::MasterReg\\_::reg](#) structure to give possibility to access to memory by name.

`uint16_t` `Register::MasterReg_::memBeg` - variable to notice beginning of memory region of structure.

The documentation for this union was generated from the following file:

- [settings.h](#)

## 7.40 ModData\_ Union Reference

```
#include <serialport.h>
```

## Public Attributes

- struct {
  - `uint16_t` `crc16Val`:16
  - `uint16_t` `data`:16
  - `uint8_t` `registerNo`:8
  - `uint8_t` `address`:8
 } `bits`
- struct {
  - `uint16_t` `crc16Val`:16
  - `uint16_t` `data`:16
  - `uint8_t` `registerNo`:7
  - `uint8_t` `rwBit`:1
  - `uint8_t` `address`:7
  - `uint8_t` `rwBit_`:1
 } `fileds`
- `u_int64_t` `all`:48

## 7.40.1 Member Data Documentation

`uint8_t` `ModData_::address` - field of union which give access to data at position where address of sender/receiver device stored. Part of `bits` structure.

`u_int64_t` `ModData_::all` - field of union which give access to all data stored in union. Used to written data to union and have access to parts of data.

struct { ... } `ModData_::bits` - structure (one of union fields) which used to have access to data in different places in data union.

`uint16_t` `ModData_::crc16Val` - field of union which give access to data at position where 2 bytes of data of package stored.

`uint16_t` `ModData_::data` - field of union which give access to data at position where control sum of package stored.

struct { ... } `ModData_::fileds` - structure (one of union fields) which used to access to data in different places in data union. Created to have possibility to access to `rwBit` (Used to check is got data must be written to appropriate place or it is data request).

`uint8_t` `ModData_::registerNo` - field of union which give access to data at position where register address stored.

`uint8_t` `ModData_::rwBit` - field of union (one bit) used to check is got data must be written to appropriate place or it is data request.

`uint8_t` `ModData_::rwBit_` - unused (reserved bit).

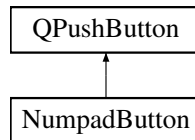
The documentation for this union was generated from the following file:

- [serialport.h](#)

## 7.41 NumpadButton Class Reference

```
#include <numpaddialog.h>
```

Inheritance diagram for NumpadButton:



### Signals

- void [clicked](#) (int [index](#))

### Public Member Functions

- [NumpadButton](#) (unsigned int inputNumber, QString name)

### Private Slots

- void [thisClicked](#) ()

### Private Attributes

- unsigned int [index](#)

#### 7.41.1 Constructor & Destructor Documentation

```
NumpadButton::NumpadButton (
    unsigned int inputNumber,
    QString name )
```

- reimplemented constructor of QPushButton class. Initialize button and set value of [index](#) with given parameter. Connect standard QPushButton::clicked() signal to [thisClicked](#)(...) function.

#### 7.41.2 Member Function Documentation

```
void NumpadButton::clicked (
    int index )
```

[signal] - signal which emitted when button is clicked with [thisClicked](#)(...) function. Signal transmit value of [index](#) variable in parameters. Signal handle in parent object.

```
void NumpadButton::thisClicked ( )
```

[inline], [private], [slot] - function called by QPushButton::clicked() signal and emit [clicked](#)(...) signal.

#### 7.41.3 Member Data Documentation

```
unsigned int NumpadButton::index
```

[private]

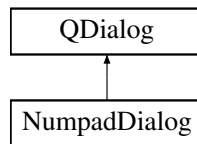
The documentation for this class was generated from the following files:

- [numpaddialog.h](#)
- [numpaddialog.cpp](#)

## 7.42 NumpadDialog Class Reference

```
#include <numpaddialog.h>
```

Inheritance diagram for NumpadDialog:



### Signals

- void [valueSubmitted](#) (QString [value](#))

### Public Member Functions

- [NumpadDialog](#) (QWidget \*parent=0, QString windowTitle="Numpad")
- [~NumpadDialog](#) ()

### Static Public Member Functions

- static double [getValue](#) (QWidget \*parent=0, QString windowTitle="Numpad")

### Protected Member Functions

- void [changeEvent](#) (QEvent \*event)

### Private Slots

- void [appendToLineEdit](#) (int number)
- void [submitValue](#) ()
- void [changeSign](#) ()
- void [appendZero](#) ()
- void [addDot](#) ()
- void [backspace](#) ()

### Private Attributes

- Ui::NumpadDialog \* [ui](#)
- [NumpadButton](#) \* [buttons](#) [10]
- const QString [stSheet](#)
- QAction \* [actionSubmit](#)
- double [value](#)

### 7.42.1 Constructor & Destructor Documentation

```
NumpadDialog::NumpadDialog (
    QWidget * parent = 0,
    QString windowTitle = "Numpad" ) [explicit] - standard QDialog constructor.
```

```
NumpadDialog::~NumpadDialog( ) - standard QDialog destructor.
```

### 7.42.2 Member Function Documentation

```
void NumpadDialog::addDot ( ) [private], [slot] - function to add dot to text field.
```

```
void NumpadDialog::appendToLineEdit (
    int number ) [private], [slot] - function which used to add number to text field. Function
called by 1..9 button from buttons array.
```

```
void NumpadDialog::appendZero ( ) [private], [slot] - function which used to add zero to text field. Func-
tion called by 0 button from buttons array.
```

```
void NumpadDialog::backspace ( ) [private], [slot] - function to remove last symbol from text field.
```

```
void NumpadDialog::changeEvent (
    QEvent * event ) [protected] - reimplementation of default event of QWidget to enable user
interface translation. Function called automatically.
```

```
void NumpadDialog::changeSign ( ) [private], [slot] - function to change sign of number which is st text
field.
```

```
double NumpadDialog::getValue (
    QWidget * parent = 0,
    QString windowTitle = "Numpad" ) [static] - static function of class used to show dialog and
get double value.
```

```
void NumpadDialog::submitValue ( ) [private], [slot] - function to transform text at text field to double
number and put it to value variable.
```

```
void NumpadDialog::valueSubmitted (
    QString value ) [signal] - unused.
```

### 7.42.3 Member Data Documentation

```
QAction* NumpadDialog::actionSubmit [private]
```

```
NumpadButton* NumpadDialog::buttons[10] [private]
```

```
const QString NumpadDialog::stSheet [private]
```

```
double NumpadDialog::value [private]
```

The documentation for this class was generated from the following files:

- [numpaddialog.h](#)
- [numpaddialog.cpp](#)

## 7.43 Register::MasterReg\_::reg Struct Reference

```
#include <settings.h>
```

Fields of structure describe data places at master board to setup machine states and parameters.

### Public Attributes

- [uint16\\_t masterReg\\_DEV\\_INF\\_H](#)
- [uint16\\_t masterReg\\_SIZE](#)
- [uint16\\_t \\_res](#)
- [uint16\\_t masterReg\\_PRZ](#)
- [uint16\\_t masterReg\\_STA](#)
- [uint16\\_t masterheadReg](#)
- [uint16\\_t masterReg\\_LAM](#)
- [uint16\\_t masterReg\\_TOTALH](#)
- [uint16\\_t masterReg\\_TOTALL](#)
- [uint16\\_t masterheadReg1](#)
- [uint16\\_t masterReg\\_PAL](#)
- [uint16\\_t masterReg\\_INPUT](#)
- [uint16\\_t masterReg\\_REMAINH](#)
- [uint16\\_t masterReg\\_REMAINL](#)
- [uint16\\_t masterReg\\_SPEED](#)
- [uint16\\_t masterReg\\_PRINTEDH](#)
- [uint16\\_t masterReg\\_PRINTEDL](#)
- [uint16\\_t masterReg\\_MACHINE\\_TYPE](#)
- [uint16\\_t masterReg\\_PAL1](#)
- [uint16\\_t masterReg\\_EKR](#)
- [uint16\\_t masterReg\\_ACTIVHEAD\\_L](#)
- [uint16\\_t masterReg\\_ACTIVHEAD\\_H](#)
- [uint16\\_t masterReg\\_DEVERR](#)
- [uint16\\_t masterReg\\_ERR](#)
- [uint16\\_t masterReg\\_KODH](#)
- [uint16\\_t masterReg\\_KODL](#)
- [uint16\\_t masterReg\\_DATH](#)
- [uint16\\_t masterReg\\_DATL](#)
- [uint16\\_t masterReg\\_KOD\\_ON](#)
- [uint16\\_t masterReg\\_DAT](#)
- [uint16\\_t REG\\_KOD\\_WPISZ](#)
- [uint16\\_t masterReg\\_HRW](#)
- [uint16\\_t masterReg\\_HRW1](#)
- [uint16\\_t masterReg\\_KOD\\_ON2](#)
- [uint16\\_t masterReg\\_KOD\\_ON3](#)
- [uint16\\_t masterReg\\_ERROR\\_MESSAGE](#)
- [uint16\\_t masterReg\\_DEV\\_INF\\_L](#)



### 7.43.1 Member Data Documentation

`uint16_t` `Register::MasterReg_::_res` - reserved field.

`uint16_t` `Register::MasterReg_::reg::masterheadReg` - States of the Heads On/Off buttons. Low Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterheadReg1` - States of the Head On/Off buttons. High Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterReg_ACTIVHEAD_H` - States of the Head Activation buttons. 0 means Head is Active, 1 means Head is Not Active. High Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterReg_ACTIVHEAD_L` - States of the Head Activation buttons. 0 means Head is Active, 1 means Head is Not Active. Low Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterReg_DAT` - Actual Date readed from system

`uint16_t` `Register::MasterReg_::reg::masterReg_DATH` - Date of the installment code activation. High Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterReg_DATL` - Date of the installment code activation. Low Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterReg_DEV_INF_H` - Number of PCB and version of software of MASTER. High Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterReg_DEV_INF_L` - Number of the PCB and version of software of MASTER. Low Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterReg_DEVERR` - Address of the device with error

`uint16_t` `Register::MasterReg_::reg::masterReg_EKR` - States of the information screen

- low-order byte:
  - bit 0 – e-stop icon
  - bit 1 – safety bar icon
  - bit 2 – error lamp
- high-order byte: (value)
  - 0 – you need to reset
  - 1 - resetting
  - 2 – you need to home
  - 3 - homing
  - 4 – you need to lock
  - 5 - locking
  - 6 - unlocking
  - 7 – ready to print
  - 8 - printing
  - 9 – you need to lower the pallets
  - 10 - half index – middle position
  - 11 - half index – extreme position

`uint16_t` Register::MasterReg\_::reg::masterReg\_ERR - Error signature

`uint16_t` Register::MasterReg\_::reg::masterReg\_ERROR\_MESSAGE - Number of the error message

`uint16_t` Register::MasterReg\_::reg::masterReg\_HRW - State of the HRW (flash station between two heads).  
Low Word of Double Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_HRW1 - State of the HRW (flash station between two heads).  
High Word of Double Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_INPUT - States of the PCB211 inputs.

`uint16_t` Register::MasterReg\_::reg::masterReg\_KOD\_ON - States of the installment code off/on buttons 1-st  
Word of Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_KOD\_ON2 - States of the installment code off/on buttons.  
2-nd Word of Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_KOD\_ON3 - States of the installment code off/on buttons.  
3-rd Word of Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_KODH - Value of the installment code. High Word of Double  
Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_KODL - Value of the installment code. Low Word of Double  
Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_LAM - States of the warning lamps (icons)

- bit 0 – HOME lamp
- bit 1 – flashing HOME lamp
- bit 2 – LOCK lamp
- bit 3 – flashing LOCK lamp
- bit 4 – UP lamp
- bit 5 – flashing UP lamp

`uint16_t` Register::MasterReg\_::reg::masterReg\_MACHINE\_TYPE - Machine type (because of one bad man  
have very interesting way to use.)

`uint16_t` Register::MasterReg\_::reg::masterReg\_PAL - States of the Palet buttons. Low Word of Double  
Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_PAL1 - States of the Pallets buttons. High Word of Double  
Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_PRINTEDH - Value of the PRINTED counter. High Word of  
Double Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_PRINTEDL - Value of the PRINTED counter. Low Word of  
Double Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_PRZ - Number of the last pressed button from application

`uint16_t` Register::MasterReg\_::reg::masterReg\_REMAINH - Value of the REMAIN counter. High Word of  
Double Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_REMAINL - Value of the REMAIN counter. Low Word of  
Double Word Register

`uint16_t` Register::MasterReg\_::reg::masterReg\_SIZE - Size of the machine – number of heads (not pallets)

`uint16_t` Register::MasterReg\_::reg::masterReg\_SPEED - Value of the production speed of the machine

`uint16_t` Register::MasterReg\_::reg::masterReg\_STA - States of 2-state buttons from bottom menu:

- bit 0 – print / stop
- bit 2 – lock / unlock
- bit 3 – up / down
- bit 4 – half / full
- bit 6 – manual / auto
- bit 7 – air release
- bit 8 – easy setup off / on

`uint16_t` `Register::MasterReg_::reg::masterReg_TOTALH` - Value of the TOTAL counter. High Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::masterReg_TOTALL` - Value of the TOTAL counter. Low Word of Double Word Register

`uint16_t` `Register::MasterReg_::reg::REG_KOD_WPISZ` - Number of installment code to enter

The documentation for this struct was generated from the following file:

- [settings.h](#)

## 7.44 Register::IndexerReg\_::reg Struct Reference

```
#include <settings.h>
```

Fields of structure describe data places at master board to setup indexer states and parameters.

### Public Attributes

- [uint16\\_t masterReg\\_DEV\\_INF\\_H](#)
- [uint16\\_t indexerReg\\_HOME\\_OFF](#)
- [uint16\\_t indexerReg\\_DIST\\_OFF](#)
- [uint16\\_t indexerReg\\_MAX\\_SPEED](#)
- [uint16\\_t indexerReg\\_DIR](#)
- [uint16\\_t \\_res0](#)
- [uint16\\_t indexerReg\\_CYCLE\\_DWELL](#)
- [uint16\\_t indexerliftReg\\_UP\\_DELAY](#)
- [uint16\\_t indexerReg\\_DIST](#)
- [uint16\\_t indexerReg\\_TM](#)
- [uint16\\_t indexerReg\\_ACC](#)
- [uint16\\_t indexerReg\\_RACC](#)
- [uint16\\_t \\_res1](#)
- [uint16\\_t indexerReg\\_RSPEED](#)
- [uint16\\_t indexerReg\\_WARN](#)
- [uint16\\_t \\_res2](#)
- [uint16\\_t \\_res3](#)
- [uint16\\_t \\_res4](#)
- [uint16\\_t \\_res5](#)
- [uint16\\_t \\_res6](#)
- [uint16\\_t \\_res7](#)
- [uint16\\_t liftReg\\_HOME\\_OFF](#)
- [uint16\\_t liftReg\\_DIST](#)
- [uint16\\_t liftReg\\_SPEED](#)
- [uint16\\_t liftReg\\_ACC](#)
- [uint16\\_t indexerReg\\_HRW\\_TIME\\_1](#)
- [uint16\\_t indexerReg\\_HRW\\_TIME\\_2](#)
- [uint16\\_t indexerReg\\_HRW\\_TIME\\_3](#)
- [uint16\\_t indexerReg\\_HRW\\_TIME\\_4](#)
- [uint16\\_t indexerReg\\_MODE](#)
- [uint16\\_t \\_res8](#)
- [uint16\\_t indexerReg\\_LOAD](#)
- [uint16\\_t indexerReg\\_WARM\\_CYCLES](#)
- [uint16\\_t indexerReg\\_WARM\\_TEMP](#)
- [uint16\\_t indexerReg\\_WARM\\_TIME](#)
- [uint16\\_t indexerReg\\_SKOK\\_SR](#)
- [uint16\\_t indexerReg\\_DEV\\_INF\\_L](#)

### 7.44.1 Member Data Documentation

- `uint16_t` `Register::IndexerReg::reg::_res0`
- `uint16_t` `Register::IndexerReg::reg::_res1`
- `uint16_t` `Register::IndexerReg::reg::_res2`
- `uint16_t` `Register::IndexerReg::reg::_res3`
- `uint16_t` `Register::IndexerReg::reg::_res4`
- `uint16_t` `Register::IndexerReg::reg::_res5`
- `uint16_t` `Register::IndexerReg::reg::_res6`
- `uint16_t` `Register::IndexerReg::reg::_res7`
- `uint16_t` `Register::IndexerReg::reg::_res8`

- reserved fields.

`uint16_t` `Register::IndexerReg::reg::indexerliftReg_UP_DELAY` - Value of the Lift Up delay

`uint16_t` `Register::IndexerReg::reg::indexerReg_ACC` - Value of the acceleration of indexer

`uint16_t` `Register::IndexerReg::reg::indexerReg_CYCLE_DWELL` - Value of the Cycle dwell time

`uint16_t` `Register::IndexerReg::reg::indexerReg_DEV_INF_L` - Number of the PCB and version of the IN-DEXER software. Low Word of Double Word Register

`uint16_t` `Register::IndexerReg::reg::indexerReg_DIR` - Work Direction

`uint16_t` `Register::IndexerReg::reg::indexerReg_DIST` - Value of the indexer distance

`uint16_t` `Register::IndexerReg::reg::indexerReg_DIST_OFF` - Value of the indexer distance offset

`uint16_t` `Register::IndexerReg::reg::indexerReg_HOME_OFF` - Value of the indexer home offset

`uint16_t` `Register::IndexerReg::reg::indexerReg_HRW_TIME_1` - Half Rapid Wave heating time #1

`uint16_t` `Register::IndexerReg::reg::indexerReg_HRW_TIME_2` - Half Rapid Wave heating time #2

`uint16_t` `Register::IndexerReg::reg::indexerReg_HRW_TIME_3` - Half Rapid Wave heating time #3

`uint16_t` `Register::IndexerReg::reg::indexerReg_HRW_TIME_4` - Half Rapid Wave heating time #4

`uint16_t` `Register::IndexerReg::reg::indexerReg_LOAD` - State of the load button

`uint16_t` `Register::IndexerReg::reg::indexerReg_MAX_SPEED` - Value of the maximal speed of indexer

`uint16_t` `Register::IndexerReg::reg::indexerReg_MODE` - States of the buttons from Mode Window

`uint16_t` `Register::IndexerReg::reg::indexerReg_RACC` - Value of the return acceleration of indexer

`uint16_t` `Register::IndexerReg::reg::indexerReg_RSPEED` - Value of the return speed of the indexer

`uint16_t` `Register::IndexerReg::reg::indexerReg_SKOK_SR` - Indexer screw pitch

`uint16_t` `Register::IndexerReg::reg::indexerReg_TM` - States of 2-states buttons

- bit 0 - multiprint table move (0 = off, 1 = on)
- bit 1 - last on/off
- bit 2 - revolver/step back
- bit 3 - warm flash

`uint16_t` Register::IndexerReg\_::reg::indexerReg\_WARM\_CYCLES - Number of the warming cycles

`uint16_t` Register::IndexerReg\_::reg::indexerReg\_WARM\_TEMP - Temperature of the warming pallets

`uint16_t` Register::IndexerReg\_::reg::indexerReg\_WARM\_TIME - Time of the warming pallets

`uint16_t` Register::IndexerReg\_::reg::indexerReg\_WARN - Value of the warning time

`uint16_t` Register::IndexerReg\_::reg::liftReg\_ACC - Value of the lift acceleration

`uint16_t` Register::IndexerReg\_::reg::liftReg\_DIST - Value of the lift distance

`uint16_t` Register::IndexerReg\_::reg::liftReg\_HOME\_OFF - Value of the lift home offset

`uint16_t` Register::IndexerReg\_::reg::liftReg\_SPEED - Value of the lift speed

`uint16_t` Register::IndexerReg\_::reg::masterReg\_DEV\_INF\_H - Number of the PCB and version of software of INDEXER. High Word of Double Word Register

The documentation for this struct was generated from the following file:

- [settings.h](#)

## 7.45 Register::HeadReg\_::reg Struct Reference

```
#include <settings.h>
```

### Public Attributes

- [uint16\\_t REG\\_DEV\\_INF\\_H](#)
- [uint16\\_t headReg\\_ON](#)
- [uint16\\_t headReg\\_RSPD](#)
- [uint16\\_t headReg\\_FSPD](#)
- [uint16\\_t headReg\\_FLDWE](#)
- [uint16\\_t headReg\\_SQDWE](#)
- [uint16\\_t headReg\\_NOSTR](#)
- [uint16\\_t headReg\\_RW1\\_TIME](#)
- [uint16\\_t headReg\\_RW2\\_TIME](#)
- [uint16\\_t headReg\\_RW3\\_TIME](#)
- [uint16\\_t headReg\\_REG\\_INP](#)
- [uint16\\_t headReg\\_ROZ](#)
- [uint16\\_t headReg\\_STAN](#)
- [uint16\\_t REG\\_KEY](#)
- [uint16\\_t headReg\\_SBSTR](#)
- [uint16\\_t REG\\_LED](#)
- [uint16\\_t headRegIsHeat](#)
- [uint16\\_t headReg\\_R](#)
- [uint16\\_t headReg\\_G](#)
- [uint16\\_t headReg\\_B](#)
- [uint16\\_t REG\\_SERVO\\_HOLD](#)
- [uint16\\_t REG\\_SHUTTLE\\_REAR\\_POS](#)
- [uint16\\_t headReg\\_RANGE\\_1](#)
- [uint16\\_t headReg\\_RANGE\\_2](#)
- [uint16\\_t REG\\_TEMP\\_SET](#)
- [uint16\\_t REG\\_SENSOR\\_TIME](#)
- [uint16\\_t REG\\_STANDBY\\_POWER](#)
- [uint16\\_t REG\\_STANDBY\\_TIME](#)
- [uint16\\_t headReg\\_CONFIG](#)
- [uint16\\_t REG\\_RW\\_POWER](#)
- [uint16\\_t REG\\_TEMP\\_MEAS](#)
- [uint16\\_t headReg\\_WARM\\_PAL\\_TIME](#)
- [uint16\\_t headReg\\_WARM\\_FLASH\\_TIME](#)
- [uint16\\_t headReg\\_MACHINE\\_TYPE](#)
- [uint16\\_t headReg\\_PRESSURE\\_1](#)
- [uint16\\_t headReg\\_PRESSURE\\_2](#)
- [uint16\\_t headReg\\_PRESSURE\\_3](#)
- [uint16\\_t headReg\\_PRESSURE\\_4](#)
- [uint16\\_t headReg\\_PRESSURE\\_5](#)
- [uint16\\_t headReg\\_ERROR\\_MESSAGE](#)
- [uint16\\_t REG\\_HMI\\_DATA](#)
- [uint16\\_t headReg\\_REVOLVER\\_STR\\_L](#)
- [uint16\\_t headReg\\_REVOLVER\\_STR\\_H](#)
- [uint16\\_t REG\\_DEV\\_INF\\_L](#)

## 7.45.1 Member Data Documentation

`uint16_t` Register::HeadReg\_::reg::headReg\_B - Ink color – Blue value

`uint16_t` Register::HeadReg\_::reg::headReg\_CONFIG - Additional head configuration

- bit 0 – IR sensor on / off
- bit 1 – warm flash on / off

`uint16_t` Register::HeadReg\_::reg::headReg\_ERROR\_MESSAGE - Number of the error message

`uint16_t` Register::HeadReg\_::reg::headReg\_FLDWE - Delay before FL

`uint16_t` Register::HeadReg\_::reg::headReg\_FSPD - Speed in the front direction

`uint16_t` Register::HeadReg\_::reg::headReg\_G - Ink color – Green value

`uint16_t` Register::HeadReg\_::reg::headReg\_MACHINE\_TYPE - Type of the head

`uint16_t` Register::HeadReg\_::reg::headReg\_NOSTR - Number of the strokes

`uint16_t` Register::HeadReg\_::reg::headReg\_ON - States of the information screen

- low-order byte: (value)
  - 0 – empty head
  - 1 – print head off
  - 2 – print head on
  - 3 – rapid head off
  - 4 – rapid head on
  - 5 – shuttle head off
  - 6 – shuttle head on
  - 7 – pilot off
  - 8 – pilot on
  - 9 – flock head off
  - 10 – flock head on
  - 11 – vacuum head off
  - 12 – vacuum head on
- high-order byte:
  - bit 0 – type of the ink: 0 = plastisol, 1 = water based
  - bit 1 & bit 2 – heating time selector: 00 = 1-st time, 01 = 2-nd time, 10 = 3-rd time
  - bit 3 – step back function off / on
  - bit 4 – preheat function off / on
  - bit 5 & bit 6 – number of rapid
  - bit 7 – cake flash unit

`uint16_t` Register::HeadReg\_::reg::headReg\_PRESSURE\_1 - Pressure of the 1-st stroke

`uint16_t` Register::HeadReg\_::reg::headReg\_PRESSURE\_2 - Pressure of the 2-nd stroke

`uint16_t` Register::HeadReg\_::reg::headReg\_PRESSURE\_3 - Pressure of the 3-rd stroke

`uint16_t` Register::HeadReg\_::reg::headReg\_PRESSURE\_4 - Pressure of the 4-th stroke

`uint16_t` Register::HeadReg\_::reg::headReg\_PRESSURE\_5 - Pressure of the 5-th stroke

`uint16_t` Register::HeadReg\_::reg::headReg\_R - Ink color – Red value



`uint16_t` Register::HeadReg\_:reg::headReg\_RANGE\_1 - 1-st position of the print range

`uint16_t` Register::HeadReg\_:reg::headReg\_RANGE\_2 - 2-nd position of the print range

`uint16_t` Register::HeadReg\_:reg::headReg\_REG\_INP - States of the Inpute

`uint16_t` Register::HeadReg\_:reg::headReg\_REVOLVER\_STR\_H - Number of the strokes in revolver mode.  
High Word of Double Word Register

`uint16_t` Register::HeadReg\_:reg::headReg\_REVOLVER\_STR\_L - Number of the strokes in revolver mode. Low  
Word of Double Word Register

`uint16_t` Register::HeadReg\_:reg::headReg\_ROZ - Command from MASTER PCB

`uint16_t` Register::HeadReg\_:reg::headReg\_RSPD - Speed in the rear direction

`uint16_t` Register::HeadReg\_:reg::headReg\_RW1\_TIME - 1-st time of the flash

`uint16_t` Register::HeadReg\_:reg::headReg\_RW2\_TIME - 2-st time of the flash

`uint16_t` Register::HeadReg\_:reg::headReg\_RW3\_TIME - 3-st time of the flash

`uint16_t` Register::HeadReg\_:reg::headReg\_SBSTR - Number of the strokes in step back

`uint16_t` Register::HeadReg\_:reg::headReg\_SQDWE - Delay before SQ

`uint16_t` Register::HeadReg\_:reg::headReg\_STAN - Execution status of the command

`uint16_t` Register::HeadReg\_:reg::headReg\_WARM\_FLASH\_TIME - Time of the flashes warming

`uint16_t` Register::HeadReg\_:reg::headReg\_WARM\_PAL\_TIME - Time of the pallets warming

`uint16_t` Register::HeadReg\_:reg::headRegIsHeat - Information about heating of the flash (0 – not heating,  
1 – heating)

`uint16_t` Register::HeadReg\_:reg::REG\_DEV\_INF\_H - Number of the PCB and version of the software. High  
Word of Double Word Register

`uint16_t` Register::HeadReg\_:reg::REG\_DEV\_INF\_L - Number of the PCB and version of the software. High  
Word of Double Word Register

`uint16_t` Register::HeadReg\_:reg::REG\_HMI\_DATA - Last pressed button of the HMI

`uint16_t` Register::HeadReg\_:reg::REG\_KEY - Last pressed keypad button

`uint16_t` Register::HeadReg\_:reg::REG\_LED - States of the keypad diodes

`uint16_t` Register::HeadReg\_:reg::REG\_RW\_POWER - Power of the flash heating when IR sensor is disabled

`uint16_t` Register::HeadReg\_:reg::REG\_SENSOR\_TIME - Heating time of the flash with IR sensor

`uint16_t` Register::HeadReg\_:reg::REG\_SERVO\_HOLD - Hold on / Hold off on servo head (State of the enable  
of the servo drive)

`uint16_t` Register::HeadReg\_:reg::REG\_SHUTTLE\_REAR\_POS - Shuttle rear position

`uint16_t` Register::HeadReg\_:reg::REG\_STANDBY\_POWER - Power of the flash standby

`uint16_t` Register::HeadReg\_:reg::REG\_STANDBY\_TIME - Time of the flash standby

`uint16_t` Register::HeadReg\_:reg::REG\_TEMP\_MEAS - Measured temperature

`uint16_t` Register::HeadReg\_:reg::REG\_TEMP\_SET - Heating set temperature

The documentation for this struct was generated from the following file:

- [settings.h](#)

## 7.46 Register::LiftReg\_::reg Struct Reference

```
#include <settings.h>
```

### Public Attributes

- [uint16\\_t \\_res0](#)
- [uint16\\_t \\_res1](#)
- [uint16\\_t \\_res2](#)
- [uint16\\_t \\_res3](#)
- [uint16\\_t \\_res4](#)
- [uint16\\_t \\_res5](#)
- [uint16\\_t liftReg\\_SEQU1\\_L](#)
- [uint16\\_t liftReg\\_SEQU1\\_H](#)
- [uint16\\_t liftReg\\_SEQU2\\_L](#)
- [uint16\\_t liftReg\\_SEQU2\\_H](#)
- [uint16\\_t liftReg\\_SEQU3\\_L](#)
- [uint16\\_t liftReg\\_SEQU3\\_H](#)
- [uint16\\_t liftReg\\_SEQU4\\_L](#)
- [uint16\\_t liftReg\\_SEQU4\\_H](#)
- [uint16\\_t liftReg\\_SEQU5\\_L](#)
- [uint16\\_t liftReg\\_SEQU5\\_H](#)
- [uint16\\_t liftReg\\_SEQU6\\_L](#)
- [uint16\\_t liftReg\\_SEQU6\\_H](#)
- [uint16\\_t liftReg\\_SEQU7\\_L](#)
- [uint16\\_t liftReg\\_SEQU7\\_H](#)
- [uint16\\_t liftReg\\_SEQU8\\_L](#)
- [uint16\\_t liftReg\\_SEQU8\\_H](#)
- [uint16\\_t liftReg\\_DIST\\_PULSE\\_L](#)
- [uint16\\_t liftReg\\_DIST\\_PULSE\\_H](#)
- [uint16\\_t REG\\_TEMP\\_UNIT](#)
- [uint16\\_t REG\\_GET\\_ZERO\\_OFF\\_CONTACT](#)
- [uint16\\_t REG\\_SKIPC\\_H](#)
- [uint16\\_t REG\\_SKIPC\\_L](#)
- [uint16\\_t liftReg\\_DOWN\\_DELAY](#)
- [uint16\\_t REG\\_PCB35\\_STAN](#)
- [uint16\\_t REG\\_PCB35\\_ROZ](#)
- [uint16\\_t REG\\_PCB35\\_SELECT](#)
- [uint16\\_t REG\\_PCB35\\_HEAT](#)
- [uint16\\_t REG\\_PCB35\\_ERR\\_DEV](#)
- [uint16\\_t REG\\_PCB35\\_MACHINE\\_TYPE](#)
- [uint16\\_t REG\\_PCB35\\_ERR\\_MESSAGE](#)

### 7.46.1 Member Data Documentation

- [uint16\\_t](#) [Register::LiftReg\\_::reg::\\_res0](#)
  - [uint16\\_t](#) [Register::LiftReg\\_::reg::\\_res1](#)
  - [uint16\\_t](#) [Register::LiftReg\\_::reg::\\_res2](#)
  - [uint16\\_t](#) [Register::LiftReg\\_::reg::\\_res3](#)
  - [uint16\\_t](#) [Register::LiftReg\\_::reg::\\_res4](#)
  - [uint16\\_t](#) [Register::LiftReg\\_::reg::\\_res5](#)
- reserved fields.

`uint16_t` Register::LiftReg\_:reg::liftReg\_DIST\_PULSE\_H - Eccentricity distance of lift. High Word of Double Word Register

`uint16_t` Register::LiftReg\_:reg::liftReg\_DIST\_PULSE\_L - Eccentricity distance of lift. Low Word of Double Word Register

`uint16_t` Register::LiftReg\_:reg::liftReg\_DOWN\_DELAY - Lift down delay time

---

States of heads On/Off buttons of revolver mode – sequence 1. Low and high words of double word register. Fields for 8 sequences

- `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU1\_H
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU1\_L
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU2\_H
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU2\_L
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU3\_H
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU3\_L
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU4\_H
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU4\_L
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU5\_H
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU5\_L
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU6\_H
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU6\_L
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU7\_H
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU7\_L
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU8\_H
  - `uint16_t` Register::LiftReg\_:reg::liftReg\_SEQU8\_L
- 

`uint16_t` Register::LiftReg\_:reg::REG\_GET\_ZERO\_OFF\_CONTACT - Lift gear ratio

`uint16_t` Register::LiftReg\_:reg::REG\_PCB35\_ERR\_DEV - Number of head with error

`uint16_t` Register::LiftReg\_:reg::REG\_PCB35\_ERR\_MESSAGE - Number of the error message

`uint16_t` Register::LiftReg\_:reg::REG\_PCB35\_HEAT - States of flashes

`uint16_t` Register::LiftReg\_:reg::REG\_PCB35\_MACHINE\_TYPE - Type of machine

`uint16_t` Register::LiftReg\_:reg::REG\_PCB35\_ROZ - Commands for heads

`uint16_t` Register::LiftReg\_:reg::REG\_PCB35\_SELECT - Select which head must execute the command

`uint16_t` Register::LiftReg\_:reg::REG\_PCB35\_STAN - States of heads On/Off

`uint16_t` Register::LiftReg\_:reg::REG\_SKIPC\_H - Value of skipped shirts counter. High Word of Double Word Register

`uint16_t` Register::LiftReg\_:reg::REG\_SKIPC\_L - Value of skipped shirts counter. Low Word of Double Word Register

`uint16_t` Register::LiftReg\_:reg::REG\_TEMP\_UNIT - Temperature units of flash with IR sensor

- bit 0: 0 = Celsius, 1 = Fahrenheit
- bit 1: 0 = sensor not active, 1 = sensor active

The documentation for this struct was generated from the following file:

- [settings.h](#)
-

## 7.47 Register Class Reference

```
#include <settings.h>
```

## Classes

- union [HeadReg\\_](#)
- union [IndexerReg\\_](#)
- union [LiftReg\\_](#)
- union [MasterReg\\_](#)

## Public Types

- enum [MasterRegNom](#) {
  - [masterReg\\_DEV\\_INF\\_H](#),
  - [masterReg\\_SIZE](#),
  - [\\_res](#),
  - [masterReg\\_PRZ](#),
  - [masterReg\\_STA](#),
  - [masterheadReg](#),
  - [masterReg\\_LAM](#),
  - [masterReg\\_TOTALH](#),
  - [masterReg\\_TOTALL](#),
  - [masterheadReg1](#),
  - [masterReg\\_paletStLow](#),
  - [masterReg\\_INPUT](#),
  - [masterReg\\_REMAINH](#),
  - [masterReg\\_REMAINL](#),
  - [masterReg\\_SPEED](#),
  - [masterReg\\_PRINTEDH](#),
  - [masterReg\\_PRINTEDL](#),
  - [masterReg\\_MACHINE\\_TYPE](#),
  - [masterReg\\_paletStHigh](#),
  - [masterReg\\_EKR](#),
  - [masterReg\\_ACTIVHEAD\\_L](#),
  - [masterReg\\_ACTIVHEAD\\_H](#),
  - [masterReg\\_DEVERR](#),
  - [masterReg\\_ERR](#),
  - [masterReg\\_KODH](#),
  - [masterReg\\_KODL](#),
  - [masterReg\\_DATH](#),
  - [masterReg\\_DATL](#),
  - [masterReg\\_KOD\\_ON](#),
  - [masterReg\\_DAT](#),
  - [REG\\_KOD\\_WPISZ](#),
  - [masterReg\\_HRW](#),
  - [masterReg\\_HRW1](#),
  - [masterReg\\_KOD\\_ON2](#),
  - [masterReg\\_KOD\\_ON3](#),
  - [masterReg\\_ERROR\\_MESSAGE](#),
  - [masterReg\\_DEV\\_INF\\_L](#)
- }
- enum [IndexerRegNom](#) {
  - [indexerReg\\_HOME\\_OFFSET](#) = 0x01,
  - [indexerReg\\_DIST\\_OFFSET](#),
  - [indexerReg\\_SPEED](#),
  - [indexerReg\\_DIR](#),

```

- _res0,
- indexerReg_CYCLE_DWELL,
- indexerliftReg_UP_DELAY,
- indexerReg_DIST,
- indexerReg_TM,
- indexerReg_ACC,
- indexerReg_RACC,
- _res1,
- indexerReg_RSPEED,
- indexerReg_WARN,
- _res2,
- _res3,
- _res4,
- _res5,
- _res6,
- _res7,
- liftReg_HOME_OFFSET,
- liftReg_DIST,
- liftReg_SPEED,
- liftReg_ACC,
- indexerReg_HRW_TIME_1,
- indexerReg_HRW_TIME_2,
- indexerReg_HRW_TIME_3,
- indexerReg_HRW_TIME_4,
- indexerReg_MODE,
- _res8,
- indexerReg_LOAD,
- indexerReg_WARM_CYCLES,
- indexerReg_WARM_TEMP,
- indexerReg_WARM_TIME,
- indexerReg_SKOK_SR,
- indexerReg_DEV_INF_L
}
• enum LiftRegNom {
- liftReg_SEQU1_L = 0x06,
- liftReg_SEQU1_H,
- liftReg_SEQU2_L,
- liftReg_SEQU2_H,
- liftReg_SEQU3_L,
- liftReg_SEQU3_H,
- liftReg_SEQU4_L,
- liftReg_SEQU4_H,
- liftReg_SEQU5_L,
- liftReg_SEQU5_H,
- liftReg_SEQU6_L,
- liftReg_SEQU6_H,
- liftReg_SEQU7_L,
- liftReg_SEQU7_H,
- liftReg_SEQU8_L,
- liftReg_SEQU8_H,
- liftReg_DIST_PULSE_L,
- liftReg_DIST_PULSE_H,
- REG_TEMP_UNIT,
- REG_GET_ZERO_OFF_CONTACT,
- REG_SKIPC_H,
- REG_SKIPC_L,
- liftReg_DOWN_DELAY,
- REG_PCB35_STAN,
- REG_PCB35_ROZ,
- REG_PCB35_SELECT,
- REG_PCB35_HEAT,
- REG_PCB35_ERR_DEV,
- REG_PCB35_MACHINE_TYPE,

```

```

    - REG_PCB35_ERR_MESSAGE
}
• enum HeadRegNom {
    - headReg_REG_DEV_INF_H,
    - headReg_ON,
    - headReg_SPD_REAR,
    - headReg_SPD_FRONT,
    - headReg_FLDWE,
    - headReg_SQDWE,
    - headReg_STR_COUNT,
    - headReg_RW1_TIME,
    - headReg_RW2_TIME,
    - headReg_RW3_TIME,
    - REG_INP,
    - headReg_ROZ,
    - headReg_STAN,
    - REG_KEY,
    - headReg_StBk_STR_COUNT,
    - REG_LED,
    - headRegT,
    - headReg_R,
    - headReg_G,
    - headReg_B,
    - headReg_SERVO_HOLD,
    - REG_SHUTTLE_REAR_POS,
    - headReg_RANGE_1,
    - headReg_RANGE_2,
    - REG_TEMP_SET,
    - REG_SENSOR_TIME,
    - REG_STANDBY_POWER,
    - REG_STANDBY_TIME,
    - headReg_CONFIG,
    - REG_RW_POWER,
    - REG_TEMP_MEAS,
    - headReg_WARM_PAL_TIME,
    - headReg_WARM_FLASH_TIME,
    - headReg_MACHINE_TYPE,
    - headReg_PRESSURE_1,
    - headReg_PRESSURE_2,
    - headReg_PRESSURE_3,
    - headReg_PRESSURE_4,
    - headReg_PRESSURE_5,
    - headReg_ERROR_MESSAGE,
    - REG_HMI_DATA,
    - headReg_REVOLVER_STR_L,
    - headReg_REVOLVER_STR_H,
    - headReg_REG_DEV_INF_L
}
• typedef union Register::MasterReg_ MasterReg
• typedef union Register::IndexerReg_ IndexerReg
• typedef union Register::LiftReg_ LiftReg
• typedef union Register::HeadReg_ HeadReg

```

#### Public Member Functions

- [Register](#) (uint16\_t headCount)
- void [writeReg](#) (uint8\_t dev, uint8\_t place, uint16\_t data)
- uint16\_t [readReg](#) (uint8\_t dev, uint8\_t place)
- void [setMasterReg](#) (MachineSettings mSett)
- void [setHeadReg](#) (int index, HeadSetting hSett)
- void [setIndexLiftReg](#) (IndexerLiftSettings iLSett)

## Static Public Member Functions

- static `uint32_t calcLiftPulse (uint16_t gearRatio, uint16_t liftDist=118)`

## Private Attributes

- `MasterReg masterReg`
- `IndexerReg indexerReg`
- `LiftReg liftReg`
- `QList< HeadReg > headRegList`
- `uint16_t * masterRegPtr`
- `uint16_t * indexerRegPtr`
- `uint16_t * liftRegPtr`
- `QList< uint16_t * > headRegPtrList`
- `HeadSetting headSettings`
- `IndexerLiftSettings indexerLiftSettings`
- `MachineSettings machineSettings`

### 7.47.1 Member Typedef Documentation

```
typedef union Register::HeadReg_ Register::HeadReg
```

```
typedef union Register::IndexerReg_ Register::IndexerReg
```

```
typedef union Register::LiftReg_ Register::LiftReg
```

```
typedef union Register::MasterReg_ Register::MasterReg
```

### 7.47.2 Member Enumeration Documentation

```
enum Register::HeadRegNom
```

Enumerator

<code>headReg_REG_DEV_INF_H</code>	<code>0x00</code>
<code>headReg_ON</code>	<code>0x01</code>
<code>headReg_SPD_REAR</code>	<code>0x02</code>
<code>headReg_SPD_FRONT</code>	<code>0x03</code>
<code>headReg_FLDWE</code>	<code>0x04</code>
<code>headReg_SQDWE</code>	<code>0x05</code>
<code>headReg_STR_COUNT</code>	<code>0x06</code>
<code>headReg_RW1_TIME</code>	<code>0x07</code>
<code>headReg_RW2_TIME</code>	<code>0x08</code>
<code>headReg_RW3_TIME</code>	<code>0x09</code>
<code>REG_INP</code>	<code>0x0A</code>
<code>headReg_ROZ</code>	<code>0x0B</code>
<code>headReg_STAN</code>	<code>0x0C</code>
<code>REG_KEY</code>	<code>0x0D</code>
<code>headReg_StBk_STR_COUNT</code>	<code>0x0E</code>
<code>REG_LED</code>	<code>0x0F</code>
<code>headRegT</code>	<code>0x10</code>
<code>headReg_R</code>	<code>0x11</code>
<code>headReg_G</code>	<code>0x12</code>
<code>headReg_B</code>	<code>0x13</code>

Enumerator

headReg_SERVO_HOLD	0x14
REG_SHUTTLE_REAR_POS	0x15
headReg_RANGE_1	0x16
headReg_RANGE_2	0x17
REG_TEMP_SET	0x18
REG_SENSOR_TIME	0x19
REG_STANDBY_POWER	0x1A
REG_STANDBY_TIME	0x1B
headReg_CONFIG	0x1C
REG_RW_POWER	0x1D
REG_TEMP_MEAS	0x1E
headReg_WARM_PAL_TIME	0x1F
headReg_WARM_FLASH_TIME	0x20
headReg_MACHINE_TYPE	0x21
headReg_PRESSURE_1	0x22
headReg_PRESSURE_2	0x23
headReg_PRESSURE_3	0x24
headReg_PRESSURE_4	0x25
headReg_PRESSURE_5	0x26
headReg_ERROR_MESSAGE	0x27
REG_HMI_DATA	0x28
headReg_REVOLVER_STR_L	0x29
headReg_REVOLVER_STR_H	0x2A
headReg_REG_DEV_INF_L	0x2B

```
enum Register::IndexerRegNom
```

Enumerator

indexerReg_HOME_OFFSET	0x01
indexerReg_DIST_OFFSET	0x02
indexerReg_SPEED	0x03
indexerReg_DIR	0x04
_res0	0x05
indexerReg_CYCLE_DWELL	0x06
indexerliftReg_UP_DELAY	0x07
indexerReg_DIST	0x08
indexerReg_TM	0x09
indexerReg_ACC	0x0A
indexerReg_RACC	0x0B
_res1	0x0C
indexerReg_RSPEED	0x0D
indexerReg_WARN	0x0E
_res2	0x0F
_res3	0x10
_res4	0x11
_res5	0x12
_res6	0x13
_res7	0x14
liftReg_HOME_OFFSET	0x15
liftReg_DIST	0x16
liftReg_SPEED	0x17
liftReg_ACC	0x18
indexerReg_HRW_TIME_1	0x19
indexerReg_HRW_TIME_2	0x1A



Enumerator

indexerReg_HRW_TIME_3	0x1B
indexerReg_HRW_TIME_4	0x1C
indexerReg_MODE	0x1D
_res8	0x1E
indexerReg_LOAD	0x1F
indexerReg_WARM_CYCLES	0x20
indexerReg_WARM_TEMP	0x21
indexerReg_WARM_TIME	0x22
indexerReg_SKOK_SR	0x23
indexerReg_DEV_INF_L	0x24

enum `Register::LiftRegNom`

Enumerator

liftReg_SEQU1_L	0x06
liftReg_SEQU1_H	0x07
liftReg_SEQU2_L	0x08
liftReg_SEQU2_H	0x09
liftReg_SEQU3_L	0x0A
liftReg_SEQU3_H	0x0B
liftReg_SEQU4_L	0x0C
liftReg_SEQU4_H	0x0D
liftReg_SEQU5_L	0x0E
liftReg_SEQU5_H	0x0F
liftReg_SEQU6_L	0x10
liftReg_SEQU6_H	0x11
liftReg_SEQU7_L	0x12
liftReg_SEQU7_H	0x13
liftReg_SEQU8_L	0x14
liftReg_SEQU8_H	0x15
liftReg_DIST_PULSE_L	0x16
liftReg_DIST_PULSE_H	0x17
REG_TEMP_UNIT	0x18
REG_GET_ZERO_OFF_CONTACT	0x19
REG_SKIPC_H	0x1A
REG_SKIPC_L	0x1B
liftReg_DOWN_DELAY	0x1C
REG_PCB35_STAN	0x1D
REG_PCB35_ROZ	0x1E
REG_PCB35_SELECT	0x1F
REG_PCB35_HEAT	0x20
REG_PCB35_ERR_DEV	0x21
REG_PCB35_MACHINE_TYPE	0x22
REG_PCB35_ERR_MESSAGE	0x23

enum `Register::MasterRegNom`

Enumerator

masterReg_DEV_INF_H	0x00
masterReg_SIZE	0x01
_res	0x02
masterReg_PRZ	0x03
masterReg_STA	0x04

Enumerator

masterheadReg	0x05
masterReg_LAM	0x06
masterReg_TOTALH	0x07
masterReg_TOTALL	0x08
masterheadReg1	0x09
masterReg_paletStLow	0x0A
masterReg_INPUT	0x0B
masterReg_REMAINH	0x0C
masterReg_REMAINL	0x0D
masterReg_SPEED	0x0E
masterReg_PRINTEDH	0x0F
masterReg_PRINTEDL	0x10
masterReg_MACHINE_TYPE	0x11
masterReg_paletStHigh	0x12
masterReg_EKR	0x13
masterReg_ACTIVHEAD_L	0x14
masterReg_ACTIVHEAD_H	0x15
masterReg_DEVERR	0x16
masterReg_ERR	0x17
masterReg_KODH	0x18
masterReg_KODL	0x19
masterReg_DATH	0x1A
masterReg_DATL	0x1B
masterReg_KOD_ON	0x1C
masterReg_DAT	0x1D
REG_KOD_WPISZ	0x1E
masterReg_HRW	0x1F
masterReg_HRW1	0x20
masterReg_KOD_ON2	0x21
masterReg_KOD_ON3	0x22
masterReg_ERROR_MESSAGE	0x23
masterReg_DEV_INF_L	0x24

### 7.47.3 Constructor & Destructor Documentation

`Register::Register (   
     uint16\_t headCount )` - class constructor. Function used to append

- [Register::MasterReg\\_::memBeg](#) to [masterRegPtr](#),
- [Register::IndexerReg\\_::memBeg](#) to [indexerRegPtr](#),
- [Register::LiftReg\\_::memBeg](#) to [liftRegPtr](#)
- [headCount](#) of [Register::HeadReg\\_::memBeg](#) to [headRegPtrList](#)

and allocate memory for all structures and pointers.

### 7.47.4 Member Function Documentation

`uint32\_t Register::calcLiftPulse (   
     uint16\_t gearRatio,   
     uint16\_t liftDist = 118 )` [static] - static function to calculate (from given parameters) 32 bit value to put it into [liftReg\\_DIST\\_PULSE\\_L](#) and [liftReg\\_DIST\\_PULSE\\_H](#) registers.

`uint16\_t Register::readReg (   
     uint8\_t dev,   
     uint8\_t place )` - function to read register at dev and place.

```

void Register::setHeadReg (
    int index,
    HeadSetting hSett ) - function to fill headRegPtrList[index] with parameters
given in hSett

void Register::setIndexLiftReg (
    IndexerLiftSettings iLSett ) - function to fill indexerReg and liftReg with parameters
given in iLSett

void Register::setMasterReg (
    MachineSettings mSett ) - function to fill masterReg with parameters given in
mSett

void Register::writeReg (
    uint8\_t dev,
    uint8\_t place,
    uint16\_t data ) - function to write data to register at dev and place.

```

#### 7.47.5 Member Data Documentation

```

QList<HeadReg> Register::headRegList [private]

QList<uint16\_t*> Register::headRegPtrList [private]

HeadSetting Register::headSettings [private]

IndexerLiftSettings Register::indexerLiftSettings [private]

IndexerReg Register::indexerReg [private]

uint16\_t* Register::indexerRegPtr [private]

LiftReg Register::liftReg [private]

uint16\_t* Register::liftRegPtr [private]

MachineSettings Register::machineSettings [private]

MasterReg Register::masterReg [private]

uint16\_t* Register::masterRegPtr [private]

```

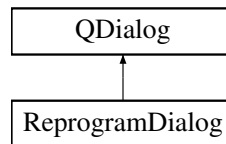
The documentation for this class was generated from the following files:

- [settings.h](#)
- [settings.cpp](#)

## 7.48 ReprogramDialog Class Reference

```
#include <reprogramdialog.h>
```

Inheritance diagram for ReprogramDialog:



### Public Types

- enum `BoardType_` {
  - `Master_ATXmega,`
  - `Heads_ATXmega,`
  - `Master_ATmega,`
  - `Heads_ATmega`
- typedef enum `ReprogramDialog::BoardType_` `BoardType`

### Public Slots

- void `setProgress` (int progress)

### Signals

- void `programArrReady` (`ReprogramDialog::BoardType` type, `QByteArray` progArr)

### Public Member Functions

- `ReprogramDialog` (`QWidget *parent=0`)
- `~ReprogramDialog` ()

### Private Slots

- void `on_pushButtonOpenFile_clicked` ()
- void `on_pushButtonErase_clicked` ()
- void `on_pushButtonWrite_clicked` ()

### Private Attributes

- `Ui::ReprogramDialog * ui`
- `QFile inFile`
- `QString inFileName`
- `QByteArray programArr`

#### 7.48.1 Member Typedef Documentation

typedef enum `ReprogramDialog::BoardType_` `ReprogramDialog::BoardType` - describe type of processor and type of device which will be reprogrammed by this module.

#### 7.48.2 Member Enumeration Documentation

```
enum ReprogramDialog::BoardType_
```

Enumerator

Master_ATXmega	0x00
Heads_ATXmega	0x01
Master_ATmega	0x02
Heads_ATmega	0x03

### 7.48.3 Constructor & Destructor Documentation

ReprogramDialog::ReprogramDialog (   
 QWidget \* *parent* = 0 ) [explicit] - standard QWidget constructor.

ReprogramDialog::~ReprogramDialog ( ) - standard QWidget destructor.

### 7.48.4 Member Function Documentation

void ReprogramDialog::on\_pushButtonErase\_clicked ( ) [private], [slot] - function to emit `programArrReady(...)` with `progArr` filled with zeros.

void ReprogramDialog::on\_pushButtonOpenFile\_clicked ( ) [private], [slot] - function to open text-HEX file with program in Intel-hex format, set value of `inFileName` and set file name of `inFile`.

void ReprogramDialog::on\_pushButtonWrite\_clicked ( ) [private], [slot] - function which used for open `inFile`, read text from it and fill `programArr` with hex data to write it into devices. After that function emit `programArrReady(...)` signal with appropriate parameters.

void ReprogramDialog::programArrReady (   
 ReprogramDialog::BoardType type,   
 QByteArray *progArr* ) [signal] - signal which emitted by `on_pushButtonErase_↵`   
 `clicked(...)` and `on_pushButtonWrite_clicked(...)` functions and handle in parent object.

void ReprogramDialog::setProgress (   
 int *progress* ) [slot] - function to set value of programming process at dialog.

### 7.48.5 Member Data Documentation

QFile ReprogramDialog::inFile [private]

QString ReprogramDialog::inFileName [private]

QByteArray ReprogramDialog::programArr [private]

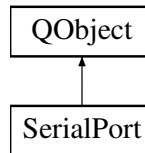
The documentation for this class was generated from the following files:

- `reprogramdialog.h`
- `reprogramdialog.cpp`

## 7.49 SerialPort Class Reference

```
#include <serialport.h>
```

Inheritance diagram for SerialPort:



### Public Slots

- void `openSerialPort` ()
- void `openSerialPort` (ComSettings cSett)
- void `closeSerialPort` ()
- void `sendProgram` (ReprogramDialog::BoardType type, QByteArray programArr)
- void `sendData` (QByteArray data, bool send=false, bool halfByte=false)
- void `sendModData` (uint8\_t dev, uint8\_t place, uint16\_t data)
- void `sendReg` (uint8\_t dev, uint8\_t place)
- void `setupPort` ()
- void `setComParams` (ComSettings sett)
- void `setStyleSheet` (QString stSheet)

### Signals

- void `dataReady` (QByteArray data)
- void `dataReady` (ModData modData)
- void `serialSettingAccepted` (ComSettings settings)
- void `working` ()
- void `programProgress` (int progress)

### Public Member Functions

- `SerialPort` (QObject \*parent=0)
- `SerialPort` (ComSettings settings, QObject \*parent=0)
- void `setRegisterPointer` (Register \*regPtr)

### Private Slots

- void `readData` ()
- void `handleError` (QSerialPort::SerialPortError error)
- void `showStatusMessage` (const QString &message)
- void `getSerialSetting` (ComSettings setting)

### Private Member Functions

- QByteArray `dataTransform` (QByteArray data)

### Private Attributes

- `SerialSettingsDialog` \* `settingsComDialog`
- QSerialPort \* `serial`
- Register \* `registers`
- ModData `modData8`
- QByteArray `dataToSendBuff`
- int `replyCnt`
- bool `initApp`

### 7.49.1 Constructor & Destructor Documentation

`SerialPort::SerialPort[1/2] (`  
     `QObject * parent = 0 ) [explicit]` - first variant of polymorphic constructor of class. Create `QSerialPort` class object and `SerialSettingsDialog` class object with default parameters.

`SerialPort::SerialPort() [2/2] (`  
     `ComSettings settings,`  
     `QObject * parent = 0 ) [explicit]` - second variant of polymorphic constructor of class. Create `QSerialPort` class object and `SerialSettingsDialog` class object with given at `settings` parameters.

### 7.49.2 Member Function Documentation

`void SerialPort::closeSerialPort ( ) [slot]` - function to close `serial` port if it's open.

`void SerialPort::dataReady[1/2] (`  
     `QByteArray data ) [signal]` - signal which emits by `readData(...)` function. Signal contain byte array with data after CRC check.

`void SerialPort::dataReady[2/2] (`  
     `ModData modData ) [signal]` - signal which emits by `readData(...)` function. Signal contain `modData` after CRC check.

`QByteArray SerialPort::dataTransform (`  
     `QByteArray data ) [private]` - function to transform data from half-byte to full-byte array.

`void SerialPort::getSerialSetting (`  
     `ComSettings setting ) [private], [slot]` - function to emit `serialSettingAccepted(...)` signal to save serial settings to hard drive.

`void SerialPort::handleError (`  
     `QSerialPort::SerialPortError error ) [private], [slot]` - serial port errors handler. Used to show `QMessageBox` with error text.

`void SerialPort::openSerialPort[1/2] ( ) [slot]` - function to open serial port with `settings←ComDialog→currentSettings`

`void SerialPort::openSerialPort[2/2] (`  
     `ComSettings cSett ) [slot]` - function to open serial port with settings given in `cSett` variable.

`void SerialPort::proramProgres (`  
     `int progres ) [signal]` - signal to send process of reprogramming PCB's.

`void SerialPort::readData ( ) [private], [slot]` - function to get data from serial port, analyze that data and create reply to send that to master PCB.

`void SerialPort::sendData (`  
     `QByteArray data,`  
     `bool send = false,`  
     `bool halfByte = false ) [slot]` - function to send data to serial port. Function take `data` to send it to `serial` port. Parameter `send` describe way of data send: if `true` than data sends immediately, if `false` than data adding to `dataToSendBuff` and send in it's turn. Parameter `halfByte` specifies requirement of data transformation - if `true` data transform to half byte variant and sends after that, if `false` data sends as it is.

`void SerialPort::sendModData (`  
     `uint8_t dev,`  
     `uint8_t place,`  
     `uint16_t data ) [slot]` - function to send mod data to serial port. At function data gather to `QByteArray` and sends to serial port.

`void SerialPort::sendProgram (`  
     `ReprogramDialog::BoardType type,`

QByteArray *programArr* ) [slot] - function to send *programArr* to PCB. Parameter type describe which commands will be sends to start reprogramming and size of memory page.

void SerialPort::sendReg (   
     uint8\_t *dev*,   
     uint8\_t *place* ) [slot] - function to send data from *registers* at *dev* and *place*.  
 At function data gather to QByteArray and sends to serial port.

void SerialPort::serialSettingAccepted (   
     ComSettings *seittngs* ) [signal] - signal which handle in parent and used to save serial settings to hard drive.

void SerialPort::setComParams (   
     ComSettings *sett* ) [slot] - function to set parameters of *serial*. If serial port is open it will be closed before setting up parameters.

void SerialPort::setRegisterPointer (   
     Register \* *regPtr* ) - function to set pointer to *Register* class object. In a whole program used only one sample of class and pointer (*registers*) sets from parent object.

void SerialPort::setStyleSheet (   
     QString *stSheet* ) [slot] - function to set style sheet to *settingsComDialog*.

void SerialPort::setupPort ( ) [slot] - function to configure *serial* with *settingsComDialog*. If serial port is open it will be closed before setting up parameters. After setting serial port it will be open again.

void SerialPort::showStatusMessage (   
     const QString & *message* ) [private], [slot] - function to send status messages from serial port to debug text thread.

void SerialPort::working ( ) [signal] - signal to update watchDog timer at parent object.

### 7.49.3 Member Data Documentation

QByteArray SerialPort::dataToSendBuff [private] - FIFO buffer variable, which contain data to send that to serial port.

bool SerialPort::initApp [private] - variable which contain info about state of program. If variable is true at *readData(...)* function execute specific code, and after that variable sets to *false*.

ModData SerialPort::modData8 [private]

Register\* SerialPort::registers [private]

int SerialPort::replyCnt [private] - counter for specific code.

QSerialPort\* SerialPort::serial [private]

SerialSettingsDialog\* SerialPort::settingsComDialog [private]

The documentation for this class was generated from the following files:

- [serialport.h](#)
- [serialport.cpp](#)

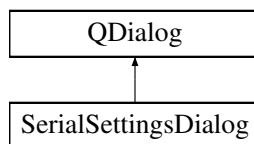


## 7.50 SerialSettingsDialog Class Reference

The class is taken from standard Qt examples.

```
#include <serialsettingsdialog.h>
```

Inheritance diagram for SerialSettingsDialog:



### Signals

- void `serialSettingAccepted` (`ComSettings` comSett)

### Public Member Functions

- `SerialSettingsDialog` (`QWidget` \*parent=0)
- `SerialSettingsDialog` (`ComSettings` nSett, `QWidget` \*parent=0)
- `~SerialSettingsDialog` ()
- `ComSettings settings` () const
- void `setSettings` (`ComSettings` nSett)

### Protected Member Functions

- void `changeEvent` (`QEvent` \*event)

### Private Slots

- void `showPortInfo` (int idx)
- void `apply` ()
- void `reject` ()
- void `checkCustomBaudRatePolicy` (int idx)
- void `checkCustomDevicePathPolicy` (int idx)
- void `fillPortsParameters` ()
- void `fillPortsInfo` ()
- void `updateSettings` ()

### Private Attributes

- `Ui::SerialSettingsDialog` \* `ui`
- bool `acceptOnDeactilationEn`
- `ComSettings` `currentSettings`
- `QIntValidator` \* `intValidator`

#### 7.50.1 Constructor & Destructor Documentation

```
SerialSettingsDialog::SerialSettingsDialog() [1/2] (
    QWidget * parent = 0 ) [explicit]
```

```
SerialSettingsDialog::SerialSettingsDialog [2/2] (
    ComSettings nSett,
    QWidget * parent = 0 ) [explicit]
```

```
SerialSettingsDialog::~~SerialSettingsDialog ( )
```

### 7.50.2 Member Function Documentation

```
void SerialSettingsDialog::apply ( ) [private], [slot]

void SerialSettingsDialog::changeEvent (
    QEvent * event ) [protected]

void SerialSettingsDialog::checkCustomBaudRatePolicy (
    int idx ) [private], [slot]

void SerialSettingsDialog::checkCustomDevicePathPolicy (
    int idx ) [private], [slot]

void SerialSettingsDialog::fillPortsInfo ( ) [private], [slot]

void SerialSettingsDialog::fillPortsParameters ( ) [private], [slot]

void SerialSettingsDialog::reject ( ) [private], [slot]

void SerialSettingsDialog::serialSettingAccepted (
    ComSettings comSett ) [signal]

void SerialSettingsDialog::setSettings (
    ComSettings nSett )

ComSettings SerialSettingsDialog::settings ( ) const

void SerialSettingsDialog::showPortInfo (
    int idx ) [private], [slot]

void SerialSettingsDialog::updateSettings ( ) [private], [slot]
```

### 7.50.3 Member Data Documentation

```
bool SerialSettingsDialog::acceptOnDeactilationEn [private]

ComSettings SerialSettingsDialog::currentSettings [private]

QIntValidator* SerialSettingsDialog::intValidator [private]
```

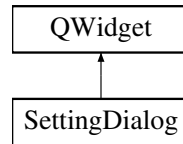
The documentation for this class was generated from the following files:

- [serialsettingsdialog.h](#)
- [serialsettingsdialog.cpp](#)

## 7.51 SettingDialog Class Reference

```
#include <headsettingdialog.h>
```

Inheritance diagram for SettingDialog:



### Signals

- void `accept` (int `index`, QByteArray `hParamArr`)
- void `changeNumber` (int `newIndex`)
- void `sendCommand` (int `index`, QByteArray `command`)
- void `setParamsToAll` (int `index`, QByteArray `hParamArr`)
- void `colorChanged` (int `index`, QColor `col`)

### Public Member Functions

- `SettingDialog` (HeadSetting `hSttg`, int `index`=0, QWidget `*parent`=0)
- `~SettingDialog` ()
- void `setRegisters` (Register `*reg`)
- void `setHeadParams` (int `index`=0, bool `disconnect`=true)
- void `setHeadParams` (HeadSetting `hSttg`, int `index`=0, bool `disconnect`=true)
- void `setIconFolder` (QString `path`)

### Protected Member Functions

- bool `event` (QEvent `*e`)
- bool `eventFilter` (QObject `*watched`, QEvent `*event`)
- void `showEvent` (QShowEvent `*ev`)
- void `changeEvent` (QEvent `*event`)

### Private Slots

- void `accept` ()
- void `reject` ()
- void `pButtonIncClkd` ()
- void `pButtonDecClkd` ()
- void `connectAll` ()
- void `disconnectAll` ()
- void `eventFilterSetup` ()
- void `temperatureSensoreChanged` (bool `tempSens`)
- void `on_toolButtonPlast_clicked` ()
- void `on_toolButtonFL_clicked` ()
- void `on_toolButtonMoveRear_clicked` ()
- void `on_toolButtonFL_SQup_clicked` ()
- void `on_toolButtonMoveFront_clicked` ()
- void `on_toolButtonMTPMove_clicked` ()
- void `on_toolButtonSQ_clicked` ()
- void `on_toolButtonMoveTest_clicked` ()
- void `on_toolButtonPressure_clicked` ()
- void `on_toolButtonHoldOn_clicked` ()
- void `on_toolButtonFL_SQ_clicked` ()
- void `on_toolButtonStepBack_clicked` ()
- void `on_toolButtonIndexHere_clicked` ()
- void `on_toolButtonInkColor_clicked` ()
- void `on_toolButtonPressureAir_clicked` ()
- void `on_pButtonHeadOnOff_clicked` ()
- void `on_toolButtonQuartzPreheat_clicked` ()
- void `on_toolButtonQuartzTest_clicked` ()
- void `on_toolButtonQuartzStepBack_clicked` ()
- void `on_toolButtonQuartzWarming_clicked` ()
- void `on_pushButtonCopyToAll_clicked` ()
- void `tabWidget_currentChanged` (int `index`)

- void `spinBoxRearSpeed_valueChanged` (double arg1)
- void `dSpinBoxRearRange_valueChanged` (double arg1)
- void `spinBoxFrontSpeed_valueChanged` (double arg1)
- void `dSpinBoxFrontRange_valueChanged` (double arg1)
- void `spinBoxStrokCount_valueChanged` (double arg1)
- void `spinBoxSBStrokCount_valueChanged` (double arg1)
- void `dSpinBoxHeatTime1IR_valueChanged` (double arg1)
- void `dSpinBoxHeatTime2IR_valueChanged` (double arg1)
- void `dSpinBoxDryingRangeIR_valueChanged` (double arg1)
- void `dSpinBoxFlDwellTime_valueChanged` (double arg1)
- void `dSpinBoxSqDwellTime_valueChanged` (double arg1)
- void `dSpinBoxHeatTime1Q_valueChanged` (double arg1)
- void `dSpinBoxHeatTime2Q_valueChanged` (double arg1)
- void `spinBoxDryPowerQ_valueChanged` (double arg1)
- void `dSpinBoxStepbackDryTimeQ_valueChanged` (double arg1)
- void `dSpinBoxTemperatureSetQ_valueChanged` (double arg1)
- void `dSpinBoxDryTimeQ_valueChanged` (double arg1)
- void `spinBoxStandbyPowerQ_valueChanged` (double arg1)
- void `dSpinBoxStandbyTimeQ_valueChanged` (double arg1)
- void `dSpinBoxWarmFlashTimeQ_valueChanged` (double arg1)
- void `rButtonTime1_clicked` ()

### Private Attributes

- `Ui::SettingDialog * ui`
- bool `withTemperatureSensor`
- int `index`
- bool `acceptOnDeactilationEn`
- bool `acceptEnable`
- `HeadSetting headSettings`
- `Register * registers`

#### 7.51.1 Constructor & Destructor Documentation

`SettingDialog::SettingDialog (`  
     `HeadSetting hSttg,`  
     `int index = 0,`  
     `QWidget * parent = 0 )` [explicit] - class constructor. At constructor fields of dialog fill with parameters given at `hSttg` and call `eventFilterSetup(...)` function.

`SettingDialog::~~SettingDialog ( )` - standard destructor.

#### 7.51.2 Member Function Documentation

`void SettingDialog::accept (`  
     `int index,`  
     `QByteArray hParamArr )` [signal] - signal which emitted by `accept(...)` function and handled in parent object. Parameter `index` contain information about head number, and `hParamArr` contain all head parameters. Parameters are save to hard drive at parent object.

`void SettingDialog::accept ( )` [private], [slot] - function to collect head parameters, put them to appropriate places at array and emit `accept(...)` signal.

`void SettingDialog::changeEvent (`  
     `QEvent * event )` [protected] - reimplementation of default event of `QDialog` to enable user interface translation. Function called automatically.

`void SettingDialog::changeNumber (`  
     `int newIndex )` [signal] - signal which handle at parent object and used for fast change of head number. Signal emits by `pButtonIncClkd(...)` and `pButtonDecClkd(...)` functions with appropriate value of parameter.

`void SettingDialog::colorChanged (`  
     `int index,`  
     `QColor col )` [signal] - signal which emits by `on_toolButtonInkColor_clicked(...)` function. Signal contain information about rag color on head.

`void SettingDialog::connectAll ( )` [private], [slot] - function to connect all fields of dialog to appropriate functions.

`void SettingDialog::disconnectAll ( )` [private], [slot] - function to disconnect all fields of dialog to appropriate functions.

---

All functions in next block used to send parameters to master PCB to configure machine. Functions used to get values from fields with head parameters, gather them into byte array and emit `sendCommand(...)` signal with appropriate data.

```

    • void SettingDialog::dSpinBoxDryingRangeIR_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxDryTimeQ_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxFlDwellTime_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxFrontRange_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxHeatTime1IR_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxHeatTime1Q_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxHeatTime2IR_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxHeatTime2Q_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxRearRange_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxSqDwellTime_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxStandbyTimeQ_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxStepbackDryTimeQ_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxTemperatureSetQ_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::dSpinBoxWarmFlashTimeQ_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::spinBoxDryPowerQ_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::spinBoxFrontSpeed_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::spinBoxRearSpeed_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::spinBoxSBStrokCount_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::spinBoxStandbyPowerQ_valueChanged (
        double arg1 ) [private], [slot]
    • void SettingDialog::spinBoxStrokCount_valueChanged (
        double arg1 ) [private], [slot]

```

---

`bool SettingDialog::event (`  
 `QEvent * e ) [protected]` - reimplementation of standard function to handle `QEvent::WindowDeactivation` or `QEvent::Leave` for automatic invoke of `accept(...)` function on window deactivation.

`bool SettingDialog::eventFilter (`  
 `QObject * watched,`  
 `QEvent * event ) [protected]` - reimplementation of standard function to handle `QEvent::MouseButtonDblClick` or `QEvent::MouseButtonRelease` events to call `NumpadDialog` or `KeyboardDialog` to enter data to appropriate widgets. Widgets which will call this function are defined in `eventFilterSetup(...)` function.

`void SettingDialog::eventFilterSetup ( ) [private], [slot]` - function to configure widgets on dialog to use `eventFilter(...)`

`void SettingDialog::on_pButtonHeadOnOff_clicked ( ) [private], [slot]` - function to handle `clicked()` signal from `pButtonHeadOnOff` button. Used to send command to turn on/off print head.

`void SettingDialog::on_pushButtonCopyToAll_clicked ( ) [private], [slot]` - function to emit `setParamsToAll` signal. Before emitting signal function collect parameters and send data in signal parameters.

All functions in next block used to send button codes to master PCB.  
Function emit `sendCommand(...)` signal with appropriate data.

```

    • void SettingDialog::on_toolButtonFL_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonFL_SQ_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonFL_SQup_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonHoldOn_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonIndexHere_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonInkColor_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonMoveFront_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonMoveRear_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonMoveTest_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonMTPMove_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonPlast_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonPressure_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonPressureAir_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonQuartzPreheat_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonQuartzStepBack_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonQuartzTest_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonQuartzWarming_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonSQ_clicked ( ) [private], [slot]
    • void SettingDialog::on_toolButtonStepBack_clicked ( ) [private], [slot]

```

```

void SettingDialog::pButtonDecClkd ( ) [private], [slot] - function to emit changeNumber(...)
signal with index decreased by 1.

```

```

void SettingDialog::pButtonIncClkd ( ) [private], [slot] - function to emit changeNumber(...)
signal with index increased by 1.

```

```

void SettingDialog::rButtonTime1_clicked ( ) [private], [slot]

```

```

void SettingDialog::reject ( ) [private], [slot]

```

```

void SettingDialog::sendCommand (
    int index,
    QByteArray command ) [signal]

```

```

void SettingDialog::setHeadParams [1/2] (
    int index = 0,
    bool disconnect = true )

```

```

void SettingDialog::setHeadParams [2/2] (
    HeadSetting hSttg,
    int index = 0,
    bool disconnect = true )

```

```

void SettingDialog::setIconFolder (
    QString path )

```

```

void SettingDialog::setParamsToAll (
    int index,
    QByteArray hParamArr ) [signal]

```

```

void SettingDialog::setRegisters (
    Register * reg )

```

```

void SettingDialog::showEvent (
    QShowEvent * ev ) [protected]

```

```

void SettingDialog::tabWidget_currentChanged (
    int index ) [private], [slot]

```

```

void SettingDialog::temperatureSensoreChanged (
    bool tempSens ) [private], [slot]

```

### 7.51.3 Member Data Documentation

```

bool SettingDialog::acceptEnable [private]
bool SettingDialog::acceptOnDeactilationEn [private]
HeadSetting SettingDialog::headSettings [private]
int SettingDialog::index [private]
Register* SettingDialog::registers [private]
bool SettingDialog::withTemperatureSensor [private]

```

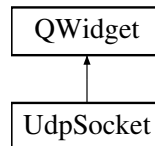
The documentation for this class was generated from the following files:

- `headsettingdialog.h`
- `headsettingdialog.cpp`

## 7.52 UdpSocket Class Reference

```
#include <udpsocket.h>
```

Inheritance diagram for UdpSocket:



### Signals

- void `exitReq` ()
- void `dataReady` (QByteArray data)

### Public Member Functions

- `UdpSocket` (QWidget \*parent=0)
- void `startUdp` (bool `isMaster`=true)
- void `stopUdp` ()
- void `sendData` (QByteArray bArr)

### Private Slots

- void `connectToSocket` ()
- void `disconnectFromSocket` ()
- void `startTransmit` (QByteArray data)
- void `recvData` ()

### Private Attributes

- QUdpSocket \* `udp`
- QHostAddress `udpAdr`
- quint16 `udpPort`
- bool `search`
- bool `isMaster`
- QHostAddress `myIP`

### 7.52.1 Constructor & Destructor Documentation

#### 7.52.1.1 UdpSocket()

```
UdpSocket::UdpSocket (
    QWidget * parent = 0 ) [explicit]
```

### 7.52.2 Member Function Documentation

#### 7.52.2.1 connectToSocket

```
void UdpSocket::connectToSocket ( ) [private], [slot]
```

### 7.52.2.2 dataReady

```
void UdpSocket::dataReady (
    QByteArray data ) [signal]
```

### 7.52.2.3 disconnectFromSocket

```
void UdpSocket::disconnectFromSocket ( ) [private], [slot]
```

### 7.52.2.4 exitReq

```
void UdpSocket::exitReq ( ) [signal]
```

### 7.52.2.5 reciveData

```
void UdpSocket::reciveData ( ) [private], [slot]
```

### 7.52.2.6 sendData()

```
void UdpSocket::sendData (
    QByteArray bArr )
```

### 7.52.2.7 startTransmit

```
void UdpSocket::startTransmit (
    QByteArray data ) [private], [slot]
```

### 7.52.2.8 startUdp()

```
void UdpSocket::startUdp (
    bool isMaster = true )
```

### 7.52.2.9 stopUdp()

```
void UdpSocket::stopUdp ( )
```

## 7.52.3 Member Data Documentation



### 7.52.3.1 isMaster

```
bool UdpSocket::isMaster [private]
```

### 7.52.3.2 myIP

```
QHostAddress UdpSocket::myIP [private]
```

### 7.52.3.3 search

```
bool UdpSocket::search [private]
```

### 7.52.3.4 udp

```
QUdpSocket* UdpSocket::udp [private]
```

### 7.52.3.5 udpAdr

```
QHostAddress UdpSocket::udpAdr [private]
```

### 7.52.3.6 udpPort

```
quint16 UdpSocket::udpPort [private]
```

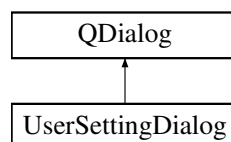
The documentation for this class was generated from the following files:

- [udpsocket.h](#)
- [udpsocket.cpp](#)

## 7.53 UserSettingDialog Class Reference

```
#include <usersetting.h>
```

Inheritance diagram for UserSettingDialog:



### Public Member Functions

- [UserSettingDialog](#) (QWidget \*parent=0)
- [~UserSettingDialog](#) ()
- bool [isUser](#) (QString userName, QString userPassw)
- QStringList [getUserNames](#) ()
- bool [getLoginWindowEnable](#) ()

### Protected Member Functions

- void `showEvent` (QShowEvent \*ev)
- void `changeEvent` (QEvent \*event)

### Private Slots

- void `tableCellActivated` (int row, int col)
- void `addUser` ()
- void `removeUsers` ()
- void `removeAllUsers` ()
- void `acceptSlot` ()
- void `rejectSlot` ()
- void `loginDialogEnable` ()

### Private Attributes

- Ui::UserSettingDialog \* `ui`
- int `tableRowSelected`
- int `tableColnumSelected`
- QSettings \* `usersData`

## 7.53.1 Constructor & Destructor Documentation

### 7.53.1.1 UserSettingDialog()

```
UserSettingDialog::UserSettingDialog (  
    QWidget * parent = 0 ) [explicit]
```

### 7.53.1.2 ~UserSettingDialog()

```
UserSettingDialog::~UserSettingDialog ( )
```

## 7.53.2 Member Function Documentation

### 7.53.2.1 acceptSlot

```
void UserSettingDialog::acceptSlot ( ) [private], [slot]
```

### 7.53.2.2 addUser

```
void UserSettingDialog::addUser ( ) [private], [slot]
```

### 7.53.2.3 changeEvent()

```
void UserSettingDialog::changeEvent (  
    QEvent * event ) [protected]
```

#### 7.53.2.4 getLoginWindowEnable()

```
bool UserSettingDialog::getLoginWindowEnable ( )
```

#### 7.53.2.5 getUserNames()

```
QStringList UserSettingDialog::getUserNames ( )
```

#### 7.53.2.6 isUser()

```
bool UserSettingDialog::isUser (
    QString userName,
    QString userPassw )
```

#### 7.53.2.7 loginDialogEnable

```
void UserSettingDialog::loginDialogEnable ( ) [private], [slot]
```

#### 7.53.2.8 rejectSlot

```
void UserSettingDialog::rejectSlot ( ) [private], [slot]
```

#### 7.53.2.9 removeAllUsers

```
void UserSettingDialog::removeAllUsers ( ) [private], [slot]
```

#### 7.53.2.10 removeUsers

```
void UserSettingDialog::removeUsers ( ) [private], [slot]
```

#### 7.53.2.11 showEvent()

```
void UserSettingDialog::showEvent (
    QShowEvent * ev ) [protected]
```

#### 7.53.2.12 tableCellActivated

```
void UserSettingDialog::tableCellActivated (
    int row,
    int col ) [private], [slot]
```

## 7.53.3 Member Data Documentation

## 7.53.3.1 tableColnumSelected

```
int UserSettingDialog::tableColnumSelected [private]
```

## 7.53.3.2 tableRowSelected

```
int UserSettingDialog::tableRowSelected [private]
```

## 7.53.3.3 ui

```
Ui::UserSettingDialog* UserSettingDialog::ui [private]
```

## 7.53.3.4 usersData

```
QSettings* UserSettingDialog::usersData [private]
```

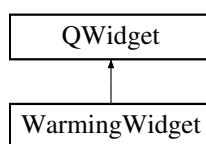
The documentation for this class was generated from the following files:

- [usersetting.h](#)
- [usersetting.cpp](#)

## 7.54 WarmingWidget Class Reference

```
#include <warmingwidget.h>
```

Inheritance diagram for WarmingWidget:



## Signals

- void [sendCommand](#) (QByteArray)

## Public Member Functions

- [WarmingWidget](#) (QWidget \*parent=0)
- [~WarmingWidget](#) ()
- void [setIconFolder](#) (QString path)
- void [setRegisters](#) (Register \*reg)
- void [resetWidget](#) ()

## Protected Member Functions

- void [showEvent](#) (QShowEvent \*ev)

## Private Slots

- void [on\\_toolButton\\_clicked](#) ()
- void [on\\_dSpinBoxTime\\_valueChanged](#) (double arg1)
- void [on\\_dSpinBoxCycles\\_valueChanged](#) (double arg1)
- void [on\\_dSpinBoxTemperature\\_valueChanged](#) (double arg1)

## Private Attributes

- Ui::WarmingWidget \* [ui](#)
- QString [pathIcon](#)
- Register \* [registers](#)

## 7.54.1 Constructor & Destructor Documentation

7.54.1.1 [WarmingWidget::WarmingWidget](#) (  
                   QWidget \* *parent* = 0 ) [explicit]

7.54.1.2 [WarmingWidget::~~WarmingWidget](#) ( )

## 7.54.2 Member Function Documentation

7.54.2.1 [void WarmingWidget::on\\_dSpinBoxCycles\\_valueChanged](#) (  
                   double *arg1* ) [private],[slot]

10

7.54.2.2 [void WarmingWidget::on\\_dSpinBoxTemperature\\_valueChanged](#) (  
                   double *arg1* ) [private],[slot]

10

7.54.2.3 [void WarmingWidget::on\\_dSpinBoxTime\\_valueChanged](#) (  
                   double *arg1* ) [private],[slot]

7.54.2.4 [void WarmingWidget::on\\_toolButton\\_clicked](#) ( ) [private],[slot]

7.54.2.5 [void WarmingWidget::resetWidget](#) ( )

7.54.2.6 [void WarmingWidget::sendCommand](#) (  
                   QByteArray ) [signal]

7.54.2.7 [void WarmingWidget::setIconFolder](#) (  
                   QString *path* )

7.54.2.8 [void WarmingWidget::setRegisters](#) (  
                   Register \* *reg* )

7.54.2.9 [void WarmingWidget::showEvent](#) (  
                   QShowEvent \* *ev* ) [protected]

## 7.54.3 Member Data Documentation

7.54.3.1 [QString WarmingWidget::pathIcon](#) [private]

7.54.3.2 [Register\\* WarmingWidget::registers](#) [private]

7.54.3.3 [Ui::WarmingWidget\\* WarmingWidget::ui](#) [private]

The documentation for this class was generated from the following files:

- [warmingwidget.h](#)
- [warmingwidget.cpp](#)

## 8 File Documentation

### 8.1 countersdialog.cpp File Reference

```
#include "countersdialog.h"  
#include "ui_countersdialog.h"
```

### 8.2 countersdialog.h File Reference

```
#include <QDialog>  
#include <QShowEvent>  
#include "numpaddialog.h"
```

#### Classes

- class `CountersDialog`

#### Namespaces

- `Ui`

### 8.3 crc16.h File Reference

```
#include <stdio.h>  
#include <stdint.h>  
#include <QByteArray>  
#include <QDebug>
```

#### Classes

- class `CrcCalc`

#### Variables

- const `uint16_t crctable` [256]

#### 8.3.1 Variable Documentation

##### 8.3.1.1 crctable

```
const uint16_t crctable[256]
```

### 8.4 cyclesdialog.cpp File Reference

```
#include "cyclesdialog.h"  
#include "ui_cyclesdialog.h"
```

## 8.5 cyclesdialog.h File Reference

```
#include <QDialog>
#include <QDoubleSpinBox>
#include <QLabel>
#include <QLineEdit>
#include <QList>
#include <QShowEvent>
#include <QFocusEvent>
#include <QEvent>
#include <QSettings>
#include <QByteArray>
#include "settings.h"
#include "numpaddialog.h"
#include "crc16.h"
#include <QDebug>
```

### Classes

- class [CyclesDialog](#)

### Namespaces

- [Ui](#)

## 8.6 exitdialog.cpp File Reference

```
#include "exitdialog.h"
#include "ui_exitdialog.h"
```

## 8.7 exitdialog.h File Reference

```
#include <QDialog>
#include <QShowEvent>
#include "settings.h"
```

### Classes

- class [ExitDialog](#)

### Namespaces

- [Ui](#)

## 8.8 generalsettingdialog.cpp File Reference

```
#include "generalsettingdialog.h"
#include "ui_generalsettingdialog.h"
#include "crc16.h"
```

## 8.9 generalsettingdialog.h File Reference

```
#include <QDialog>
#include <QString>
#include <QDebug>
#include <QByteArray>
#include <QMessageBox>
#include <QInputDialog>
#include <QShowEvent>
#include "settings.h"
#include "numpaddialog.h"
#include "keyboarddialog.h"
#include "serialsettingsdialog.h"
```

### Classes

- struct [EmailSettings](#)
- class [GeneralSettingDialog](#)

### Namespaces

- [Ui](#)

### Functions

- [QDataStream & operator<<](#) ([QDataStream](#) &out, const [EmailSettings](#) &st)
- [QDataStream & operator>>](#) ([QDataStream](#) &in, [EmailSettings](#) &st)

### 8.9.1 Function Documentation

#### 8.9.1.1 [operator<<\(\)](#)

```
QDataStream& operator<< (
    QDataStream & out,
    const EmailSettings & st ) [inline]
```

#### 8.9.1.2 [operator>>\(\)](#)

```
QDataStream& operator>> (
    QDataStream & in,
    EmailSettings & st ) [inline]
```

## 8.10 headactivationdialog.cpp File Reference

```
#include "headactivationdialog.h"
#include "ui_headactivationdialog.h"
```



## 8.11 headactivationdialog.h File Reference

```
#include <QDialog>
#include <QCheckBox>
#include <QList>
#include <QShowEvent>
#include <QDebug>
#include "settings.h"
#include "crc16.h"
```

### Classes

- class [CheckBoxIndexed](#)
- class [HeadActivationDialog](#)

### Namespaces

- [Ui](#)

## 8.12 headform.cpp File Reference

```
#include "headform.h"
#include "ui_headform.h"
#include <QBitmap>
#include <QPalette>
```

## 8.13 headform.h File Reference

```
#include <QWidget>
#include <QPixmap>
#include <QPushButton>
#include <QLabel>
#include <QMouseEvent>
#include <QDebug>
#include <QGraphicsEffect>
#include <QGraphicsScene>
#include <QGraphicsRectItem>
#include <QGraphicsPixmapItem>
```

### Classes

- class [HeadSettingButton](#)
- class [HeadForm](#)

### Namespaces

- [Ui](#)

### Typedefs

- typedef enum [LoadState\\_](#) [LoadState](#)

## Enumerations

- enum `LoadState_` { `LoadClean` = 0x00, `LoadOne` = 0x01, `LoadAuto` = 0x02 }

### 8.13.1 Typedef Documentation

#### 8.13.1.1 LoadState

```
typedef enum LoadState_ LoadState
```

### 8.13.2 Enumeration Type Documentation

#### 8.13.2.1 LoadState\_

```
enum LoadState_
```

Enumerator

LoadClean	
LoadOne	
LoadAuto	

## 8.14 headsettingdialog.cpp File Reference

```
#include "headsettingdialog.h"
#include "ui_headsettingdialog.h"
```

## 8.15 headsettingdialog.h File Reference

```
#include <QDialog>
#include <QWidget>
#include <QFocusEvent>
#include <QEvent>
#include <QApplicationStateChangeEvent>
#include <QDebug>
#include <QMessageBox>
#include <QByteArray>
#include <QLineEdit>
#include <QColorDialog>
#include "settings.h"
#include "numpaddialog.h"
#include "crc16.h"
```

## Classes

- class `SettingDialog`

## Namespaces

- [Ui](#)

## 8.16 indexersettingdialog.cpp File Reference

```
#include "indexersettingdialog.h"  
#include "ui_indexersettingdialog.h"
```

## 8.17 indexersettingdialog.h File Reference

```
#include <QWidget>  
#include <QByteArray>  
#include <QEvent>  
#include <QShowEvent>  
#include <QLineEdit>  
#include "settings.h"  
#include "numpaddialog.h"  
#include "crc16.h"
```

## Classes

- class [IndexerSettingDialog](#)

## Namespaces

- [Ui](#)

## 8.18 indexerwidget.cpp File Reference

```
#include "indexerwidget.h"  
#include "ui_indexerwidget.h"
```

## 8.19 indexerwidget.h File Reference

```
#include <QWidget>  
#include <QResizeEvent>  
#include <QPushButton>  
#include <QDebug>  
#include <QThread>  
#include "settings.h"  
#include "crc16.h"
```

## Classes

- class [IndexerWidget](#)

## Namespaces

- [Ui](#)

## 8.20 infowidget.cpp File Reference

```
#include "infowidget.h"  
#include "ui_infowidget.h"
```

## 8.21 infowidget.h File Reference

```
#include <QFrame>  
#include <QImage>  
#include <QBitmap>  
#include <QGraphicsEffect>  
#include <QDebug>  
#include <QTime>  
#include <QSettings>  
#include "settings.h"
```

### Classes

- class [InfoWidget](#)

### Namespaces

- [Ui](#)

## 8.22 keyboarddialog.cpp File Reference

```
#include "keyboarddialog.h"  
#include "ui_keyboarddialog.h"
```

## 8.23 keyboarddialog.h File Reference

```
#include <QDialog>  
#include <QPushButton>  
#include <QDebug>  
#include <QApplication>
```

### Classes

- class [KeyboardButton](#)
- class [KeyboardDialog](#)

### Namespaces

- [Ui](#)

## 8.24 logindialog.cpp File Reference

```
#include "logindialog.h"  
#include "ui_logindialog.h"
```

## 8.25 logindialog.h File Reference

```
#include <QDialog>
#include <QEvent>
#include "numpaddialog.h"
#include "keyboarddialog.h"
```

### Classes

- class [LoginDialog](#)

### Namespaces

- [Ui](#)

## 8.26 logodialog.cpp File Reference

```
#include "logodialog.h"
#include "ui_logodialog.h"
```

## 8.27 logodialog.h File Reference

```
#include <QDialog>
#include <QShowEvent>
#include <QTimer>
#include <QMovie>
```

### Classes

- class [LogoDialog](#)

### Namespaces

- [Ui](#)

## 8.28 mailsender.cpp File Reference

```
#include "mailsender.h"
#include <QDebug>
```

## 8.29 mailsender.h File Reference

```
#include <QObject>
#include <QDebug>
#include <QMessageBox>
#include "mailSrc/SmtPMime"
```

## Classes

- class `MailSender`

## 8.30 main.cpp File Reference

```
#include "mainwindow.h"
#include <QApplication>
#include <QDebug>
#include "logodialog.h"
#include <QProcess>
```

## Functions

- int `main` (int argc, char \*argv[])

### 8.30.1 Function Documentation

#### 8.30.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

## 8.31 maintancedialog.cpp File Reference

```
#include "maintancedialog.h"
#include "ui_maintancedialog.h"
```

## 8.32 maintancedialog.h File Reference

```
#include <QDialog>
#include <QIcon>
#include <QDebug>
#include <QSettings>
#include "maintancewidget.h"
```

## Classes

- class `MaintanceDialog`

## Namespaces

- `Ui`

## 8.33 maintancewidget.cpp File Reference

```
#include "maintancewidget.h"
#include "ui_maintancewidget.h"
```

## 8.34 maintancewidget.h File Reference

```
#include <QDialog>
```

### Classes

- class `MaintanceElement`
- class `MaintanceWidget`

### Namespaces

- `Ui`

### Functions

- `QDataStream & operator<<` (`QDataStream &out`, const `MaintanceElement` &st)
- `QDataStream & operator>>` (`QDataStream &in`, `MaintanceElement` &st)

### 8.34.1 Function Documentation

#### 8.34.1.1 `operator<<()`

```
QDataStream& operator<< (  
    QDataStream & out,  
    const MaintanceElement & st ) [inline]
```

#### 8.34.1.2 `operator>>()`

```
QDataStream& operator>> (  
    QDataStream & in,  
    MaintanceElement & st ) [inline]
```

## 8.35 mainwindow.cpp File Reference

```
#include "mainwindow.h"  
#include "ui_mainwindow.h"  
#include "math.h"  
#include "crc16.h"
```

## 8.36 mainWindow.h File Reference

```
#include <QMainWindow>
#include <QPushButton>
#include <QResizeEvent>
#include <QShowEvent>
#include <QDebug>
#include <QPixmap>
#include <QImage>
#include <QSettings>
#include <QPalette>
#include <QGradient>
#include <QStyle>
#include <QInputDialog>
#include <QByteArray>
#include <QFileDialog>
#include <QEvent>
#include <QMessageBox>
#include <QProgressDialog>
#include <QTime>
#include <QTimer>
#include <QDate>
#include <QStringList>
#include <QTranslator>
#include "headform.h"
#include "headsettingdialog.h"
#include "indexerwidget.h"
#include "indexerssettingdialog.h"
#include "generalsettingdialog.h"
#include "numpaddialog.h"
#include "keyboarddialog.h"
#include "serialport.h"
#include "mailsender.h"
#include "serialsettingsdialog.h"
#include "infowidget.h"
#include "usersetting.h"
#include "logindialog.h"
#include "maintancedialog.h"
#include "exitdialog.h"
#include "cyclesdialog.h"
#include "udpsocket.h"
#include "headactivationdialog.h"
#include "reprogramdiallog.h"
#include "settings.h"
```

### Classes

- class [MainWindow](#)

### Namespaces

- [Ui](#)

## 8.37 numpaddialog.cpp File Reference

```
#include "numpaddialog.h"
#include "ui_numpaddialog.h"
```

## 8.38 numpaddialog.h File Reference

```
#include <QDialog>
#include <QPushButton>
#include <QDebug>
#include <QAction>
#include <QKeySequence>
```



## Classes

- class [NumpadButton](#)
- class [NumpadDialog](#)

## Namespaces

- [Ui](#)

## 8.39 README.md File Reference

## 8.40 reprogramdialog.cpp File Reference

```
#include "reprogramdialog.h"  
#include "ui_reprogramdialog.h"
```

## 8.41 reprogramdialog.h File Reference

```
#include <QDialog>  
#include <QFile>  
#include <QFileDialog>  
#include <QByteArray>  
#include <QMessageBox>  
#include <QDebug>
```

## Classes

- class [ReprogramDialog](#)

## Namespaces

- [Ui](#)

## 8.42 serialport.cpp File Reference

```
#include "serialport.h"
```

## 8.43 serialport.h File Reference

```
#include <QObject>  
#include <QByteArray>  
#include <QDebug>  
#include <QtSerialPort/QSerialPort>  
#include <QMessageBox>  
#include <QThread>  
#include <QProgressDialog>  
#include "serialsettingsdialog.h"  
#include "settings.h"  
#include "crc16.h"  
#include "reprogramdialog.h"
```

## Classes

- union `ModData_`
- class `SerialPort`

## Typedefs

- typedef union `ModData_ ModData`

### 8.43.1 Typedef Documentation

#### 8.43.1.1 ModData

```
typedef union ModData_ ModData
```

## 8.44 serialsettingsdialog.cpp File Reference

```
#include "serialsettingsdialog.h"
#include "ui_serialsettingsdialog.h"
#include "generalsettingdialog.h"
#include <QtSerialPort/QSerialPortInfo>
#include <QIntValidator>
#include <QLineEdit>
```

## Variables

- static QT\_USE\_NAMESPACE const char `blankString` [] = QT\_TRANSLATE\_NOOP("SettingsDialog", "N/A")

### 8.44.1 Variable Documentation

#### 8.44.1.1 blankString

```
QT_USE_NAMESPACE const char blankString[] = QT_TRANSLATE_NOOP("SettingsDialog", "N/A") [static]
```

## 8.45 serialsettingsdialog.h File Reference

```
#include <QDialog>
#include <QtSerialPort/QSerialPort>
#include <QDataStream>
#include <QEvent>
#include <QMetaType>
#include <QStringList>
```

## Classes

- struct `ComSettings`
- class `SerialSettingsDialog`

## Namespaces

- [Ui](#)

## Functions

- [Q\\_DECLARE\\_METATYPE](#) ([ComSettings](#))
- [QDataStream & operator<<](#) ([QDataStream](#) &out, const [ComSettings](#) &st)
- [QDataStream & operator>>](#) ([QDataStream](#) &in, [ComSettings](#) &st)

### 8.45.1 Function Documentation

#### 8.45.1.1 [operator<<\(\)](#)

```
QDataStream& operator<< (  
    QDataStream & out,  
    const ComSettings & st ) [inline]
```

#### 8.45.1.2 [operator>>\(\)](#)

```
QDataStream& operator>> (  
    QDataStream & in,  
    ComSettings & st ) [inline]
```

#### 8.45.1.3 [Q\\_DECLARE\\_METATYPE\(\)](#)

```
Q_DECLARE_METATYPE (  
    ComSettings )
```

### 8.46 [settings.cpp](#) File Reference

```
#include "settings.h"  
#include <QDebug>
```

### 8.47 [settings.h](#) File Reference

```
#include <QByteArray>  
#include <QStringList>  
#include <QList>  
#include <QDate>  
#include "math.h"
```

## Classes

- class `HeadSetting`
- struct `HeadSetting::HeadParameters_`
- struct `HeadSetting::HeadComands_`
- class `MachineSettings`
- union `MachineSettings::MachineState_`
- union `MachineSettings::MachineHeadType_`
- union `MachineSettings::MachineIndexLiftType_`
- union `MachineSettings::LastRevolverWarm_`
- struct `MachineSettings::MachineParameters_`
- class `IndexerLiftSettings`
- struct `IndexerLiftSettings::LiftParameters_`
- struct `IndexerLiftSettings::IndexParameters_`
- class `Register`
- union `Register::MasterReg_`
- struct `Register::MasterReg_::reg`
- union `Register::IndexerReg_`
- struct `Register::IndexerReg_::reg`
- union `Register::LiftReg_`
- struct `Register::LiftReg_::reg`
- union `Register::HeadReg_`
- struct `Register::HeadReg_::reg`

## Typedefs

- typedef `u_int32_t` `uint32_t`
- typedef `u_int16_t` `uint16_t`
- typedef `u_int8_t` `uint8_t`

### 8.47.1 Typedef Documentation

#### 8.47.1.1 `uint16_t`

```
typedef u_int16_t uint16_t
```

#### 8.47.1.2 `uint32_t`

```
typedef u_int32_t uint32_t
```

#### 8.47.1.3 `uint8_t`

```
typedef u_int8_t uint8_t
```

## 8.48 udpsocket.cpp File Reference

```
#include "udpsocket.h"
```

## 8.49 udpsocket.h File Reference

```
#include <QUdpSocket>
#include <QByteArray>
#include <QDebug>
#include <QWidget>
#include <QNetworkInterface>
#include <QThread>
#include <QMessageBox>
```

## Classes

- class `UdpSocket`

## 8.50 usersetting.cpp File Reference

```
#include "usersetting.h"  
#include "ui_usersetting.h"
```

## 8.51 usersetting.h File Reference

```
#include <QDialog>  
#include <QDebug>  
#include <QShowEvent>  
#include <QCheckBox>  
#include <QSettings>  
#include "keyboarddialog.h"
```

## Classes

- class `UserSettingDialog`

## Namespaces

- `Ui`

## 8.52 warmingwidget.cpp File Reference

```
#include "warmingwidget.h"  
#include "ui_warmingwidget.h"
```

## 8.53 warmingwidget.h File Reference

```
#include <QWidget>  
#include <QByteArray>  
#include <QShowEvent>  
#include "settings.h"  
#include "crc16.h"
```

## Classes

- class `WarmingWidget`

## Namespaces

- `Ui`