

Battery Discharger

Generated by Doxygen 1.9.4

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 batteryPack Class Reference	5
3.1.1 Constructor & Destructor Documentation	6
3.1.1.1 batteryPack()	6
3.1.2 Member Function Documentation	6
3.1.2.1 chemistry()	6
3.1.2.2 clear()	6
3.1.2.3 isPack()	6
3.1.2.4 printDetails()	7
3.1.2.5 readXml()	7
3.1.3 Member Data Documentation	7
3.1.3.1 batt_chem	7
3.1.3.2 capacity	7
3.1.3.3 cells	7
3.1.3.4 current	7
3.1.3.5 endVoltage	7
3.1.3.6 id	8
3.1.3.7 nominalVoltage	8
3.2 eventLoop Class Reference	8
3.2.1 Constructor & Destructor Documentation	10
3.2.1.1 eventLoop()	10
3.2.2 Member Function Documentation	10
3.2.2.1 applySettings()	10
3.2.2.2 closeHardware()	10
3.2.2.3 listHardware()	11
3.2.2.4 listPackIDs()	11
3.2.2.5 listPrinters()	11
3.2.2.6 loadSetup()	11
3.2.2.7 openLog()	11
3.2.2.8 openPort()	11
3.2.2.9 operate	11
3.2.2.10 produceQALabel()	11
3.2.2.11 readPhysical	12
3.2.2.12 runGnuplot	12
3.2.2.13 setActiveChannel()	12
3.2.2.14 startDischarge()	12
3.2.2.15 stdOutValues	12

3.2.3 Member Data Documentation	12
3.2.3.1 current	12
3.2.3.2 currentPack	12
3.2.3.3 dischargeTimeSecs	13
3.2.3.4 endVoltage	13
3.2.3.5 energy	13
3.2.3.6 fileNameStem	13
3.2.3.7 ioStream	13
3.2.3.8 lastPower	13
3.2.3.9 logFile	13
3.2.3.10 m_activeChannel	13
3.2.3.11 m_progLoad	14
3.2.3.12 opTimer	14
3.2.3.13 packs	14
3.2.3.14 plotTimer	14
3.2.3.15 power	14
3.2.3.16 QALabelPrinter	14
3.2.3.17 startTime	14
3.2.3.18 stdOutTimer	14
3.2.3.19 voltage	15
3.3 loadHardware Class Reference	15
3.3.1 Detailed Description	16
3.3.2 Constructor & Destructor Documentation	17
3.3.2.1 loadHardware()	17
3.3.2.2 ~loadHardware()	17
3.3.3 Member Function Documentation	17
3.3.3.1 current()	17
3.3.3.2 enable()	17
3.3.3.3 factory()	18
3.3.3.4 factorySetup()	18
3.3.3.5 initialise()	18
3.3.3.6 isOfType()	18
3.3.3.7 name()	18
3.3.3.8 power()	18
3.3.3.9 processSettings()	19
3.3.3.10 readSettings()	19
3.3.3.11 setCurrent()	19
3.3.3.12 shutdown()	19
3.3.3.13 typesAvailable()	19
3.3.3.14 voltage()	20
3.3.3.15 xmlType()	20
3.3.4 Member Data Documentation	20

3.3.4.1 m_hardwareTag	20
3.3.4.2 m_hwname	20
3.4 PiDischarger Class Reference	21
3.4.1 Detailed Description	22
3.4.2 Constructor & Destructor Documentation	22
3.4.2.1 PiDischarger()	22
3.4.3 Member Function Documentation	22
3.4.3.1 current()	22
3.4.3.2 enable()	23
3.4.3.3 initialise()	23
3.4.3.4 power()	23
3.4.3.5 processSettings()	23
3.4.3.6 shutdown()	23
3.4.3.7 voltage()	23
3.4.4 Member Data Documentation	24
3.4.4.1 m_activeChannel	24
3.4.4.2 m_adc	24
3.4.4.3 m_adcBits	24
3.4.4.4 m_digloPort	24
3.4.4.5 m_disChan	24
3.5 piDischargerChannel Class Reference	25
3.5.1 Constructor & Destructor Documentation	25
3.5.1.1 piDischargerChannel()	26
3.5.1.2 ~piDischargerChannel()	26
3.5.2 Member Function Documentation	26
3.5.2.1 configure()	26
3.5.2.2 current()	26
3.5.2.3 enable()	26
3.5.2.4 power()	26
3.5.2.5 registerHardware()	27
3.5.2.6 voltage()	27
3.5.3 Member Data Documentation	27
3.5.3.1 m_adc	27
3.5.3.2 m_adcChannel	27
3.5.3.3 m_digloPort	27
3.5.3.4 m_dioChannel	27
3.5.3.5 m_lastVolts	27
3.5.3.6 m_offset	28
3.5.3.7 m_resistance	28
3.5.3.8 m_slope	28
3.6 QObject Class Reference	28
3.7 Tenma7213210Load Class Reference	29

3.7.1 Detailed Description	30
3.7.2 Constructor & Destructor Documentation	30
3.7.2.1 Tenma7213210Load()	30
3.7.2.2 ~Tenma7213210Load()	30
3.7.3 Member Function Documentation	30
3.7.3.1 current()	31
3.7.3.2 enable()	31
3.7.3.3 initialise()	31
3.7.3.4 power()	31
3.7.3.5 processSettings()	31
3.7.3.6 readReply()	31
3.7.3.7 setCurrent()	32
3.7.3.8 shutdown()	32
3.7.3.9 voltage()	32
3.7.4 Member Data Documentation	32
3.7.4.1 commPort	32
3.7.4.2 ioStream	32
Index	33

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

batteryPack	5
loadHardware	15
PiDischarger	21
Tenma7213210Load	29
piDischargerChannel	25
QObject	28
eventLoop	8

Chapter 2

Class Index

2.1 Class List

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

batteryPack	5
eventLoop	8
loadHardware	15
PiDischarger	21
piDischargerChannel	25
QObject	28
Tenma7213210Load	29

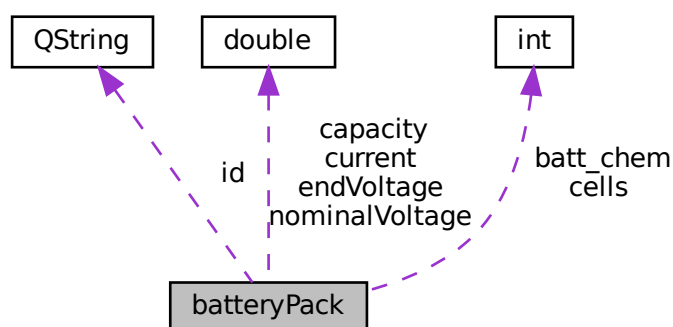
Chapter 3

Class Documentation

3.1 batteryPack Class Reference

```
#include <batteryPack.h>
```

Collaboration diagram for batteryPack:



Public Member Functions

- `batteryPack ()`
- `QString chemistry ()`
- `void clear ()`
- `bool isPack (QString packId)`
- `void printDetails (FILE *stream)`
- `void readXml (QDomElement xml)`

Public Attributes

- double [capacity](#)
- int [cells](#)
- double [current](#)
- double [endVoltage](#)
- QString [id](#)
- double [nominalVoltage](#)

Private Attributes

- int [batt_chem](#) =1

3.1.1 Constructor & Destructor Documentation

3.1.1.1 batteryPack()

```
batteryPack::batteryPack ( )
```

3.1.2 Member Function Documentation

3.1.2.1 chemistry()

```
QString batteryPack::chemistry ( )
```

3.1.2.2 clear()

```
void batteryPack::clear ( )
```

3.1.2.3 isPack()

```
bool batteryPack::isPack (
    QString packId )
```

3.1.2.4 printDetails()

```
void batteryPack::printDetails (
    FILE * stream )
```

3.1.2.5 readXml()

```
void batteryPack::readXml (
    QDomElement xml )
```

3.1.3 Member Data Documentation

3.1.3.1 batt_chem

```
int batteryPack::batt_chem =1 [private]
```

3.1.3.2 capacity

```
double batteryPack::capacity
```

3.1.3.3 cells

```
int batteryPack::cells
```

3.1.3.4 current

```
double batteryPack::current
```

3.1.3.5 endVoltage

```
double batteryPack::endVoltage
```

3.1.3.6 id

```
QString batteryPack::id
```

3.1.3.7 nominalVoltage

```
double batteryPack::nominalVoltage
```

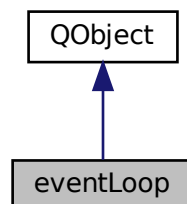
The documentation for this class was generated from the following files:

- src/batteryPack.h
- src/batteryPack.cpp

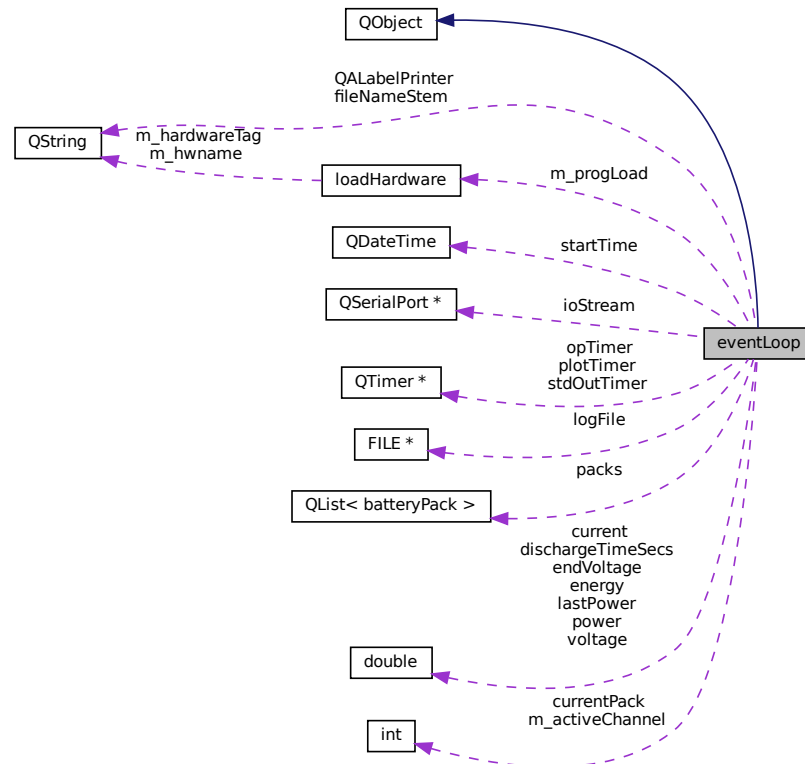
3.2 eventLoop Class Reference

```
#include <eventLoop.h>
```

Inheritance diagram for eventLoop:



Collaboration diagram for eventLoop:



Public Member Functions

- `eventLoop` (`QObject *parent=0`)
- `bool applySettings` (`QString packID`)
- `void closeHardware` ()
- `void listHardware` ()
- `void listPackIDs` ()
- `void listPrinters` ()
- `bool loadSetup` ()
- `bool openLog` ()
- `bool openPort` ()
- `void produceQALabel` ()
- `void setActiveChannel` (`int chan`)
- `void startDischarge` ()

Public Attributes

- `QString QALabelPrinter`

Private Slots

- `void operate` ()
- `void readPhysical` ()
- `void runGnuplot` ()
- `void stdOutValues` ()

Private Attributes

- double `current`
- int `currentPack`
- double `dischargeTimeSecs`
- double `endVoltage`
- double `energy`
- QString `fileNameStem`
- QSerialPort * `ioStream`
- double `lastPower`
- FILE * `logFile`
- int `m_activeChannel`
- `loadHardware` * `m_progLoad`
- QTimer * `opTimer`
- QList< `batteryPack` > `packs`
- QTimer * `plotTimer`
- double `power`
- QDateTime `startTime`
- QTimer * `stdOutTimer`
- double `voltage`

3.2.1 Constructor & Destructor Documentation

3.2.1.1 `eventLoop()`

```
eventLoop::eventLoop (
    QObject * parent = 0 )
```

3.2.2 Member Function Documentation

3.2.2.1 `applySettings()`

```
bool eventLoop::applySettings (
    QString packID )
```

3.2.2.2 `closeHardware()`

```
void eventLoop::closeHardware ( )
```


3.2.2.3 listHardware()

```
void eventLoop::listHardware ( )
```

3.2.2.4 listPackIDs()

```
void eventLoop::listPackIDs ( )
```

3.2.2.5 listPrinters()

```
void eventLoop::listPrinters ( )
```

3.2.2.6 loadSetup()

```
bool eventLoop::loadSetup ( )
```

3.2.2.7 openLog()

```
bool eventLoop::openLog ( )
```

3.2.2.8 openPort()

```
bool eventLoop::openPort ( )
```

3.2.2.9 operate

```
void eventLoop::operate ( ) [private], [slot]
```

3.2.2.10 produceQALabel()

```
void eventLoop::produceQALabel ( )
```

3.2.2.11 readPhysical

```
void eventLoop::readPhysical ( ) [private], [slot]
```

3.2.2.12 runGnuplot

```
void eventLoop::runGnuplot ( ) [private], [slot]
```

3.2.2.13 setActiveChannel()

```
void eventLoop::setActiveChannel (
    int chan )
```

3.2.2.14 startDischarge()

```
void eventLoop::startDischarge ( )
```

3.2.2.15 stdOutValues

```
void eventLoop::stdOutValues ( ) [private], [slot]
```

3.2.3 Member Data Documentation

3.2.3.1 current

```
double eventLoop::current [private]
```

3.2.3.2 currentPack

```
int eventLoop::currentPack [private]
```

3.2.3.3 dischargeTimeSecs

```
double eventLoop::dischargeTimeSecs [private]
```

3.2.3.4 endVoltage

```
double eventLoop::endVoltage [private]
```

3.2.3.5 energy

```
double eventLoop::energy [private]
```

3.2.3.6 fileNameStem

```
QString eventLoop::fileNameStem [private]
```

3.2.3.7 ioStream

```
QSerialPort* eventLoop::ioStream [private]
```

3.2.3.8 lastPower

```
double eventLoop::lastPower [private]
```

3.2.3.9 logFile

```
FILE* eventLoop::logFile [private]
```

3.2.3.10 m_activeChannel

```
int eventLoop::m_activeChannel [private]
```

3.2.3.11 m_progLoad

```
loadHardware* eventLoop::m_progLoad [private]
```

3.2.3.12 opTimer

```
QTimer* eventLoop::opTimer [private]
```

3.2.3.13 packs

```
QList<batteryPack> eventLoop::packs [private]
```

3.2.3.14 plotTimer

```
QTimer * eventLoop::plotTimer [private]
```

3.2.3.15 power

```
double eventLoop::power [private]
```

3.2.3.16 QALabelPrinter

```
QString eventLoop::QALabelPrinter
```

3.2.3.17 startTime

```
QDateTime eventLoop::startTime [private]
```

3.2.3.18 stdOutTimer

```
QTimer * eventLoop::stdOutTimer [private]
```

3.2.3.19 voltage

```
double eventLoop::voltage [private]
```

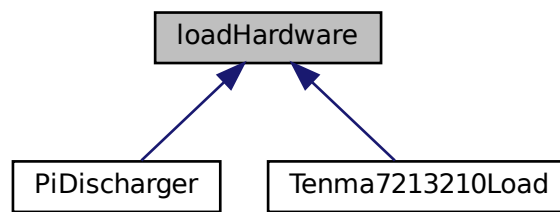
The documentation for this class was generated from the following files:

- src/eventLoop.h
- src/eventLoop.cpp

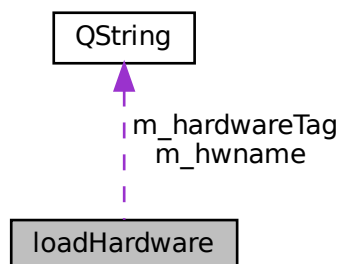
3.3 loadHardware Class Reference

```
#include <loadHardware.h>
```

Inheritance diagram for loadHardware:



Collaboration diagram for loadHardware:



Public Member Functions

- [loadHardware](#) (QString [xmlType](#), QString hwName)
- virtual [~loadHardware](#) ()
- virtual bool [current](#) (double *amps)
- virtual void [enable](#) (bool enable)
- virtual bool [initialise](#) (int channel)
- QString [name](#) ()
- virtual bool [power](#) (double *watts)
- void [readSettings](#) (QDomElement element)
- virtual void [setCurrent](#) (double preset)
- virtual bool [shutdown](#) ()
- virtual bool [voltage](#) (double *volts)
- QString [xmlType](#) ()

Static Public Member Functions

- static [loadHardware](#) * [factory](#) (QDomElement element)
- static QStringList [typesAvailable](#) ()

Protected Member Functions

- virtual void [processSettings](#) (QDomElement element)

Private Member Functions

- bool [isOfType](#) (QString [xmlType](#))

Static Private Member Functions

- static bool [factorySetup](#) (QDomElement element, [loadHardware](#) *temp)

Private Attributes

- QString [m_hardwareTag](#)
- QString [m_hwname](#)

3.3.1 Detailed Description

Base class for programmable load hardware. Don't instantiate this class directly, subclass it and add your own functionality as required.

The call sequence should be:

- [readSettings](#)
- [initialise\(\)](#)
- [operations](#)

3.3.2 Constructor & Destructor Documentation

3.3.2.1 loadHardware()

```
loadHardware::loadHardware (
    QString xmlType,
    QString hwName )
```

Constructor to allow subclasses to do a once-only assignment of the XML type for thier hardware. This constructor *MUST* be called, otherwise the XML parser will not recognise the hardware and the factory system will not instantiate your subclass.

3.3.2.2 ~loadHardware()

```
loadHardware::~~loadHardware ( ) [virtual]
```

Destructor.

3.3.3 Member Function Documentation

3.3.3.1 current()

```
bool loadHardware::current (
    double * amps ) [virtual]
```

Read current from hardware. Do not change the value passed in unless the reading was successful. The default implementation returns false and leaves amps unchanged.

Reimplemented in [PiDischarger](#), and [Tenma7213210Load](#).

3.3.3.2 enable()

```
void loadHardware::enable (
    bool enable ) [virtual]
```

Enable load function. Default implementation does nothing. Subclasses should enable the load hardware when passed "true".

Reimplemented in [PiDischarger](#), and [Tenma7213210Load](#).

3.3.3.3 factory()

```
loadHardware * loadHardware::factory (
    QDomElement element ) [static]
```

This function produces a specific hardware instance from the XML element provided. If adding new hardware your hardware drive must be instantiated here. To avoid a memory leak the temporary instance gets deleted in factorySetup.

3.3.3.4 factorySetup()

```
bool loadHardware::factorySetup (
    QDomElement element,
    loadHardware * temp ) [static], [private]
```

This function handles type checking of the temporary hardware instance passed in. If it is of the required type then the hardware is configured and the function returns true. If it is not, the instance is deleted and the function returns false.

3.3.3.5 initialise()

```
bool loadHardware::initialise (
    int channel ) [virtual]
```

Default initialise routine. Do nothing and return failed. Use this to open connections to hardware etc.

Reimplemented in [PiDischarger](#), and [Tenma7213210Load](#).

3.3.3.6 isOfType()

```
bool loadHardware::isOfType (
    QString xmlType ) [private]
```

Used in the factory system to produce hardware interfaces without making the type publicly known.

3.3.3.7 name()

```
QString loadHardware::name ( )
```

Return hardware name.

3.3.3.8 power()

```
bool loadHardware::power (
    double * watts ) [virtual]
```

Read power from hardware. Do not change the value passed in unless the reading was successful. The default implementation returns false and leaves watts unchanged.

Reimplemented in [PiDischarger](#), and [Tenma7213210Load](#).

3.3.3.9 processSettings()

```
void loadHardware::processSettings (
    QDomElement element ) [protected], [virtual]
```

Allows processing of configuration options in sub-class. Default implementation does nothing.

Reimplemented in [PiDischarger](#), and [Tenma7213210Load](#).

3.3.3.10 readSettings()

```
void loadHardware::readSettings (
    QDomElement element )
```

If the passed XML element is "hardware" loop over all sub-elements and pass them to processSettings for handling in subclass.

3.3.3.11 setCurrent()

```
void loadHardware::setCurrent (
    double preset ) [virtual]
```

Set current function. Default implementation does nothing.

Reimplemented in [Tenma7213210Load](#).

3.3.3.12 shutdown()

```
bool loadHardware::shutdown ( ) [virtual]
```

Default shutdown routine. Do nothing and return failed. Use this in your subclass to close open connections to hardware etc.

Reimplemented in [PiDischarger](#), and [Tenma7213210Load](#).

3.3.3.13 typesAvailable()

```
QStringList loadHardware::typesAvailable ( ) [static]
```

This function provides a list of all the available hardware names. If adding new hardware your hardware drive must be instantiated here.

3.3.3.14 voltage()

```
bool loadHardware::voltage (
    double * volts ) [virtual]
```

Read voltage from hardware. Do not change the value passed in unless the reading was successful. The default implementation returns false and leaves volts unchanged.

Reimplemented in [PiDischarger](#), and [Tenma7213210Load](#).

3.3.3.15 xmlType()

```
QString loadHardware::xmlType ( )
```

Return xml type attribute.

3.3.4 Member Data Documentation

3.3.4.1 m_hardwareTag

```
QString loadHardware::m_hardwareTag [private]
```

3.3.4.2 m_hwname

```
QString loadHardware::m_hwname [private]
```

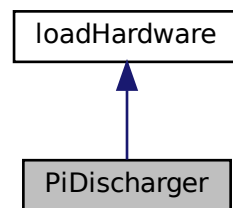
The documentation for this class was generated from the following files:

- src/loadHardware.h
- src/loadHardware.cpp

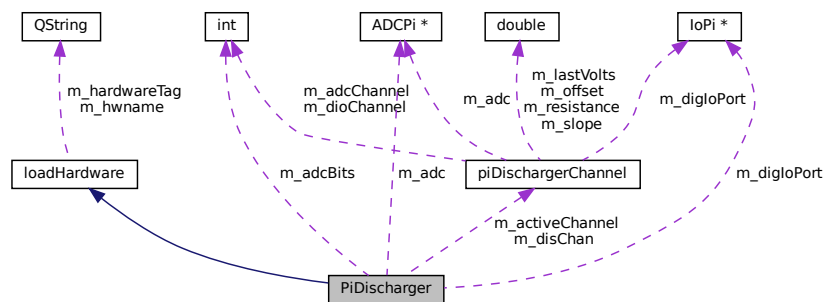
3.4 PiDischarger Class Reference

```
#include <PiDischarger.h>
```

Inheritance diagram for PiDischarger:



Collaboration diagram for PiDischarger:



Public Member Functions

- `PiDischarger` ()
- `bool current` (double *amps)
- `void enable` (bool enable)
- `bool initialise` (int channel)
- `bool power` (double *watts)
- `bool shutdown` ()
- `bool voltage` (double *volts)

Protected Member Functions

- `void processSettings` (QDomElement element)

Private Attributes

- [piDischargerChannel](#) * [m_activeChannel](#)
- ADCPi * [m_adc](#)
- int [m_adcBits](#)
User selectable bit-width for ADC.
- IoPi * [m_digIoPort](#)
- [piDischargerChannel](#) [m_disChan](#) [8]

Additional Inherited Members

3.4.1 Detailed Description

Base class for programmable load hardware. Don't instantiate this class directly, subclass it and add your own functionality as required.

The call sequence should be:

- [readSettings](#)
- [initialise\(\)](#)
- [operations](#)

3.4.2 Constructor & Destructor Documentation

3.4.2.1 PiDischarger()

```
PiDischarger::PiDischarger ( )
```

3.4.3 Member Function Documentation

3.4.3.1 current()

```
bool PiDischarger::current (
    double * amps ) [virtual]
```

Reimplemented from [loadHardware](#).

3.4.3.2 enable()

```
void PiDischarger::enable (
    bool enable ) [virtual]
```

Reimplemented from [loadHardware](#).

3.4.3.3 initialise()

```
bool PiDischarger::initialise (
    int channel ) [virtual]
```

Reimplemented from [loadHardware](#).

3.4.3.4 power()

```
bool PiDischarger::power (
    double * watts ) [virtual]
```

Reimplemented from [loadHardware](#).

3.4.3.5 processSettings()

```
void PiDischarger::processSettings (
    QDomElement element ) [protected], [virtual]
```

Reimplemented from [loadHardware](#).

3.4.3.6 shutdown()

```
bool PiDischarger::shutdown ( ) [virtual]
```

Reimplemented from [loadHardware](#).

3.4.3.7 voltage()

```
bool PiDischarger::voltage (
    double * volts ) [virtual]
```

Reimplemented from [loadHardware](#).

3.4.4 Member Data Documentation

3.4.4.1 m_activeChannel

```
piDischargerChannel* PiDischarger::m_activeChannel [private]
```

3.4.4.2 m_adc

```
ADCPi* PiDischarger::m_adc [private]
```

3.4.4.3 m_adcBits

```
int PiDischarger::m_adcBits [private]
```

3.4.4.4 m_digIoPort

```
IoPi* PiDischarger::m_digIoPort [private]
```

3.4.4.5 m_disChan

```
piDischargerChannel PiDischarger::m_disChan[8] [private]
```

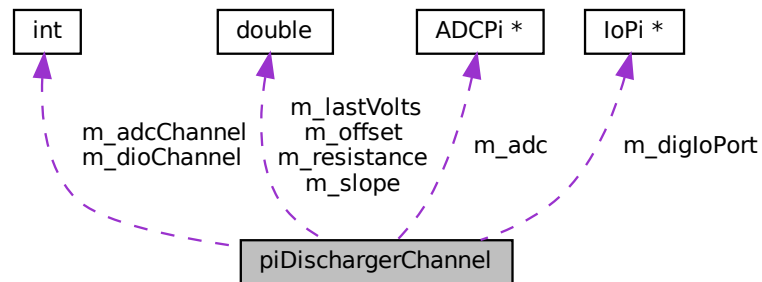
The documentation for this class was generated from the following files:

- src/PiDischarger.h
- src/PiDischarger.cpp

3.5 piDischargerChannel Class Reference

```
#include <PiDischarger.h>
```

Collaboration diagram for piDischargerChannel:



Public Member Functions

- `piDischargerChannel ()`
- `~piDischargerChannel ()`
- `void configure (QDomElement cfgElement)`
- `bool current (double *amps)`
- `void enable (bool enable)`
- `bool power (double *watts)`
- `void registerHardware (ADCPi *adc, IoPi *diglo)`
- `bool voltage (double *volts)`

Private Attributes

- `ADCPi * m_adc`
- `int m_adcChannel`
- `IoPi * m_digloPort`
- `int m_dioChannel`
- `double m_lastVolts`
- `double m_offset`
- `double m_resistance`
- `double m_slope`

3.5.1 Constructor & Destructor Documentation

3.5.1.1 piDischargerChannel()

```
piDischargerChannel::piDischargerChannel ( )
```

3.5.1.2 ~piDischargerChannel()

```
piDischargerChannel::~~piDischargerChannel ( )
```

3.5.2 Member Function Documentation

3.5.2.1 configure()

```
void piDischargerChannel::configure (
    QDomElement cfgElement )
```

3.5.2.2 current()

```
bool piDischargerChannel::current (
    double * amps )
```

3.5.2.3 enable()

```
void piDischargerChannel::enable (
    bool enable )
```

3.5.2.4 power()

```
bool piDischargerChannel::power (
    double * watts )
```


3.5.2.5 registerHardware()

```
void piDischargerChannel::registerHardware (
    ADCPi * adc,
    IoPi * digIo )
```

3.5.2.6 voltage()

```
bool piDischargerChannel::voltage (
    double * volts )
```

3.5.3 Member Data Documentation

3.5.3.1 m_adc

```
ADCPi* piDischargerChannel::m_adc [private]
```

3.5.3.2 m_adcChannel

```
int piDischargerChannel::m_adcChannel [private]
```

3.5.3.3 m_digIoPort

```
IoPi* piDischargerChannel::m_digIoPort [private]
```

3.5.3.4 m_dioChannel

```
int piDischargerChannel::m_dioChannel [private]
```

3.5.3.5 m_lastVolts

```
double piDischargerChannel::m_lastVolts [private]
```

3.5.3.6 m_offset

```
double piDischargerChannel::m_offset [private]
```

3.5.3.7 m_resistance

```
double piDischargerChannel::m_resistance [private]
```

3.5.3.8 m_slope

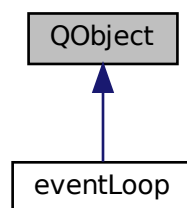
```
double piDischargerChannel::m_slope [private]
```

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

- src/PiDischarger.h
- src/PiDischarger.cpp

3.6 QObject Class Reference

Inheritance diagram for QObject:



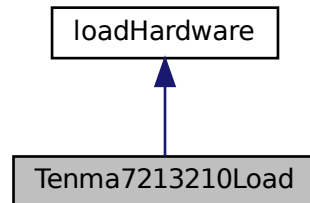
The documentation for this class was generated from the following file:

- src/eventLoop.h

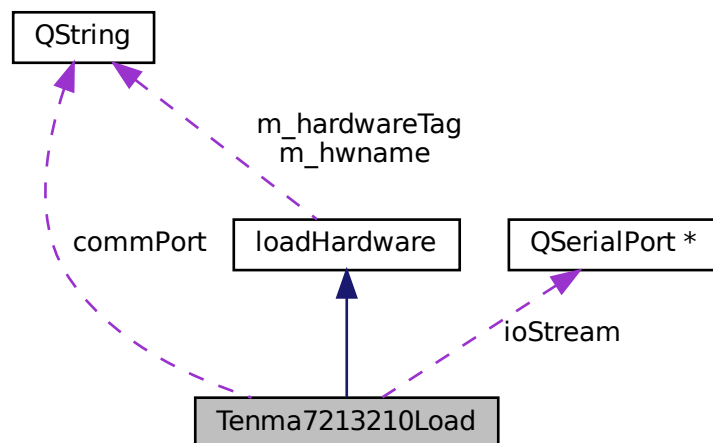
3.7 Tenma7213210Load Class Reference

```
#include <Tenma7213210Load.h>
```

Inheritance diagram for Tenma7213210Load:



Collaboration diagram for Tenma7213210Load:



Public Member Functions

- `Tenma7213210Load ()`
- `~Tenma7213210Load ()`
- `bool current (double *amps)`
- `void enable (bool enable)`
- `bool initialise (int channel)`
- `bool power (double *watts)`
- `void setCurrent (double preset)`
- `bool shutdown ()`
- `bool voltage (double *volts)`

Protected Member Functions

- void [processSettings](#) (QDomElement element)

Private Member Functions

- QByteArray [readReply](#) ()

Private Attributes

- QString [commPort](#)
- QSerialPort * [ioStream](#)

Additional Inherited Members

3.7.1 Detailed Description

Base class for programmable load hardware. Don't instantiate this class directly, subclass it and add your own functionality as required.

The call sequence should be:

- readSettings
- [initialise\(\)](#)
- operations

3.7.2 Constructor & Destructor Documentation

3.7.2.1 Tenma7213210Load()

```
Tenma7213210Load::Tenma7213210Load ( )
```

3.7.2.2 ~Tenma7213210Load()

```
Tenma7213210Load::~~Tenma7213210Load ( )
```

3.7.3 Member Function Documentation

3.7.3.1 current()

```
bool Tenma7213210Load::current (
    double * amps ) [virtual]
```

Reimplemented from [loadHardware](#).

3.7.3.2 enable()

```
void Tenma7213210Load::enable (
    bool enable ) [virtual]
```

Reimplemented from [loadHardware](#).

3.7.3.3 initialise()

```
bool Tenma7213210Load::initialise (
    int channel ) [virtual]
```

Reimplemented from [loadHardware](#).

3.7.3.4 power()

```
bool Tenma7213210Load::power (
    double * watts ) [virtual]
```

Reimplemented from [loadHardware](#).

3.7.3.5 processSettings()

```
void Tenma7213210Load::processSettings (
    QDomElement element ) [protected], [virtual]
```

Reimplemented from [loadHardware](#).

3.7.3.6 readReply()

```
QByteArray Tenma7213210Load::readReply ( ) [private]
```

3.7.3.7 setCurrent()

```
void Tenma7213210Load::setCurrent (
    double preset ) [virtual]
```

Reimplemented from [loadHardware](#).

3.7.3.8 shutdown()

```
bool Tenma7213210Load::shutdown ( ) [virtual]
```

Reimplemented from [loadHardware](#).

3.7.3.9 voltage()

```
bool Tenma7213210Load::voltage (
    double * volts ) [virtual]
```

Reimplemented from [loadHardware](#).

3.7.4 Member Data Documentation

3.7.4.1 commPort

```
QString Tenma7213210Load::commPort [private]
```

3.7.4.2 ioStream

```
QSerialPort* Tenma7213210Load::ioStream [private]
```

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

- src/Tenma7213210Load.h
- src/Tenma7213210Load.cpp

Index

- ~Tenma7213210Load
 - Tenma7213210Load, [30](#)
- ~loadHardware
 - loadHardware, [17](#)
- ~piDischargerChannel
 - piDischargerChannel, [26](#)
- applySettings
 - eventLoop, [10](#)
- batt_chem
 - batteryPack, [7](#)
- batteryPack, [5](#)
 - batt_chem, [7](#)
 - batteryPack, [6](#)
 - capacity, [7](#)
 - cells, [7](#)
 - chemistry, [6](#)
 - clear, [6](#)
 - current, [7](#)
 - endVoltage, [7](#)
 - id, [7](#)
 - isPack, [6](#)
 - nominalVoltage, [8](#)
 - printDetails, [6](#)
 - readXml, [7](#)
- capacity
 - batteryPack, [7](#)
- cells
 - batteryPack, [7](#)
- chemistry
 - batteryPack, [6](#)
- clear
 - batteryPack, [6](#)
- closeHardware
 - eventLoop, [10](#)
- commPort
 - Tenma7213210Load, [32](#)
- configure
 - piDischargerChannel, [26](#)
- current
 - batteryPack, [7](#)
 - eventLoop, [12](#)
 - loadHardware, [17](#)
 - PiDischarger, [22](#)
 - piDischargerChannel, [26](#)
 - Tenma7213210Load, [30](#)
- currentPack
 - eventLoop, [12](#)
- dischargeTimeSecs
 - eventLoop, [12](#)
- enable
 - loadHardware, [17](#)
 - PiDischarger, [22](#)
 - piDischargerChannel, [26](#)
 - Tenma7213210Load, [31](#)
- endVoltage
 - batteryPack, [7](#)
 - eventLoop, [13](#)
- energy
 - eventLoop, [13](#)
- eventLoop, [8](#)
 - applySettings, [10](#)
 - closeHardware, [10](#)
 - current, [12](#)
 - currentPack, [12](#)
 - dischargeTimeSecs, [12](#)
 - endVoltage, [13](#)
 - energy, [13](#)
 - eventLoop, [10](#)
 - fileNameStem, [13](#)
 - ioStream, [13](#)
 - lastPower, [13](#)
 - listHardware, [10](#)
 - listPackIDs, [11](#)
 - listPrinters, [11](#)
 - loadSetup, [11](#)
 - logFile, [13](#)
 - m_activeChannel, [13](#)
 - m_progLoad, [13](#)
 - openLog, [11](#)
 - openPort, [11](#)
 - operate, [11](#)
 - opTimer, [14](#)
 - packs, [14](#)
 - plotTimer, [14](#)
 - power, [14](#)
 - produceQALabel, [11](#)
 - QALabelPrinter, [14](#)
 - readPhysical, [11](#)
 - runGnuplot, [12](#)
 - setActiveChannel, [12](#)
 - startDischarge, [12](#)
 - startTime, [14](#)
 - stdOutTimer, [14](#)
 - stdOutValues, [12](#)
 - voltage, [14](#)

- factory
 - loadHardware, 17
- factorySetup
 - loadHardware, 18
- fileNameStem
 - eventLoop, 13
- id
 - batteryPack, 7
- initialise
 - loadHardware, 18
 - PiDischarger, 23
 - Tenma7213210Load, 31
- ioStream
 - eventLoop, 13
 - Tenma7213210Load, 32
- isOfType
 - loadHardware, 18
- isPack
 - batteryPack, 6
- lastPower
 - eventLoop, 13
- listHardware
 - eventLoop, 10
- listPackIDs
 - eventLoop, 11
- listPrinters
 - eventLoop, 11
- loadHardware, 15
 - ~loadHardware, 17
 - current, 17
 - enable, 17
 - factory, 17
 - factorySetup, 18
 - initialise, 18
 - isOfType, 18
 - loadHardware, 17
 - m_hardwareTag, 20
 - m_hwname, 20
 - name, 18
 - power, 18
 - processSettings, 18
 - readSettings, 19
 - setCurrent, 19
 - shutdown, 19
 - typesAvailable, 19
 - voltage, 19
 - xmlType, 20
- loadSetup
 - eventLoop, 11
- logFile
 - eventLoop, 13
- m_activeChannel
 - eventLoop, 13
 - PiDischarger, 24
- m_adc
 - PiDischarger, 24
 - piDischargerChannel, 27
- m_adcBits
 - PiDischarger, 24
- m_adcChannel
 - piDischargerChannel, 27
- m_digloPort
 - PiDischarger, 24
 - piDischargerChannel, 27
- m_dioChannel
 - piDischargerChannel, 27
- m_disChan
 - PiDischarger, 24
- m_hardwareTag
 - loadHardware, 20
- m_hwname
 - loadHardware, 20
- m_lastVolts
 - piDischargerChannel, 27
- m_offset
 - piDischargerChannel, 27
- m_progLoad
 - eventLoop, 13
- m_resistance
 - piDischargerChannel, 28
- m_slope
 - piDischargerChannel, 28
- name
 - loadHardware, 18
- nominalVoltage
 - batteryPack, 8
- openLog
 - eventLoop, 11
- openPort
 - eventLoop, 11
- operate
 - eventLoop, 11
- opTimer
 - eventLoop, 14
- packs
 - eventLoop, 14
- PiDischarger, 21
 - current, 22
 - enable, 22
 - initialise, 23
 - m_activeChannel, 24
 - m_adc, 24
 - m_adcBits, 24
 - m_digloPort, 24
 - m_disChan, 24
 - PiDischarger, 22
 - power, 23
 - processSettings, 23
 - shutdown, 23
 - voltage, 23
- piDischargerChannel, 25
 - ~piDischargerChannel, 26

- configure, [26](#)
- current, [26](#)
- enable, [26](#)
- m_adc, [27](#)
- m_adcChannel, [27](#)
- m_digloPort, [27](#)
- m_dioChannel, [27](#)
- m_lastVolts, [27](#)
- m_offset, [27](#)
- m_resistance, [28](#)
- m_slope, [28](#)
- piDischargerChannel, [25](#)
- power, [26](#)
- registerHardware, [26](#)
- voltage, [27](#)
- plotTimer
 - eventLoop, [14](#)
- power
 - eventLoop, [14](#)
 - loadHardware, [18](#)
 - PiDischarger, [23](#)
 - piDischargerChannel, [26](#)
 - Tenma7213210Load, [31](#)
- printDetails
 - batteryPack, [6](#)
- processSettings
 - loadHardware, [18](#)
 - PiDischarger, [23](#)
 - Tenma7213210Load, [31](#)
- produceQALabel
 - eventLoop, [11](#)
- QALabelPrinter
 - eventLoop, [14](#)
- QObject, [28](#)
- readPhysical
 - eventLoop, [11](#)
- readReply
 - Tenma7213210Load, [31](#)
- readSettings
 - loadHardware, [19](#)
- readXml
 - batteryPack, [7](#)
- registerHardware
 - piDischargerChannel, [26](#)
- runGnuplot
 - eventLoop, [12](#)
- setActiveChannel
 - eventLoop, [12](#)
- setCurrent
 - loadHardware, [19](#)
 - Tenma7213210Load, [31](#)
- shutdown
 - loadHardware, [19](#)
 - PiDischarger, [23](#)
 - Tenma7213210Load, [32](#)
- startDischarge
 - eventLoop, [12](#)
- startTime
 - eventLoop, [14](#)
- stdOutTimer
 - eventLoop, [14](#)
- stdOutValues
 - eventLoop, [12](#)
- Tenma7213210Load, [29](#)
 - ~Tenma7213210Load, [30](#)
 - commPort, [32](#)
 - current, [30](#)
 - enable, [31](#)
 - initialise, [31](#)
 - ioStream, [32](#)
 - power, [31](#)
 - processSettings, [31](#)
 - readReply, [31](#)
 - setCurrent, [31](#)
 - shutdown, [32](#)
 - Tenma7213210Load, [30](#)
 - voltage, [32](#)
- typesAvailable
 - loadHardware, [19](#)
- voltage
 - eventLoop, [14](#)
 - loadHardware, [19](#)
 - PiDischarger, [23](#)
 - piDischargerChannel, [27](#)
 - Tenma7213210Load, [32](#)
- xmlType
 - loadHardware, [20](#)