Electric Tiger DAQ

0.0.1

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

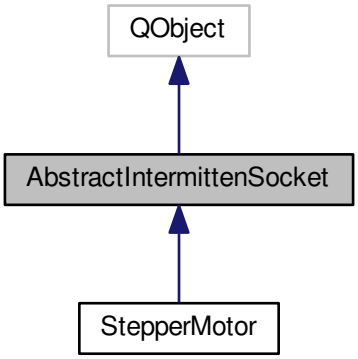
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

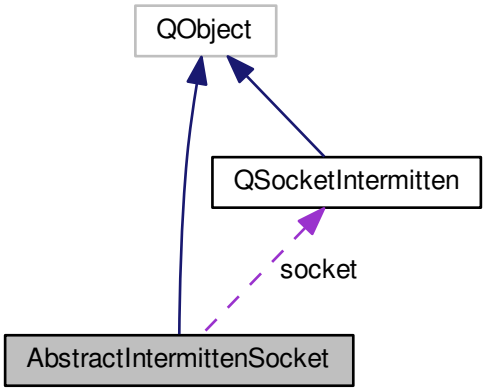
Class Documentation

3.1 AbstractIntermittenSocket Class Reference

Inheritance diagram for AbstractIntermittenSocket:



Collaboration diagram for AbstractIntermittenSocket:



Public Member Functions

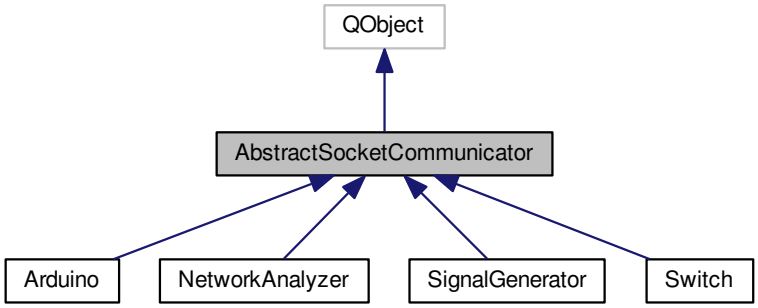
- **AbstractIntermittenSocket** (std::string ip_addr, uint port_number, QObject *parent=0)
- **AbstractIntermittenSocket** (const TCPSocketParam socket_param, QObject *parent=0)
- AbstractIntermittenSocket & **operator=** (const AbstractIntermittenSocket &)=delete

Protected Attributes

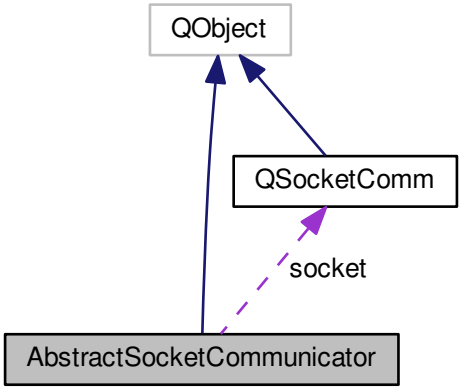
- QSocketIntermitten * **socket**

3.2 AbstractSocketCommunicator Class Reference

Inheritance diagram for AbstractSocketCommunicator:



Collaboration diagram for AbstractSocketCommunicator:



Public Member Functions

- **AbstractSocketCommunicator** (std::string ip_addr, uint port_number, QObject *parent=0)
- **AbstractSocketCommunicator** (const TCPSocketParam socket_param, QObject *parent=0)
- AbstractSocketCommunicator & **operator=** (const AbstractSocketCommunicator &)=delete

Protected Attributes

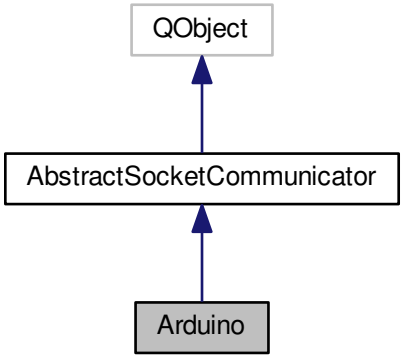
- QSocketComm * **socket**

3.3 Arduino Class Reference

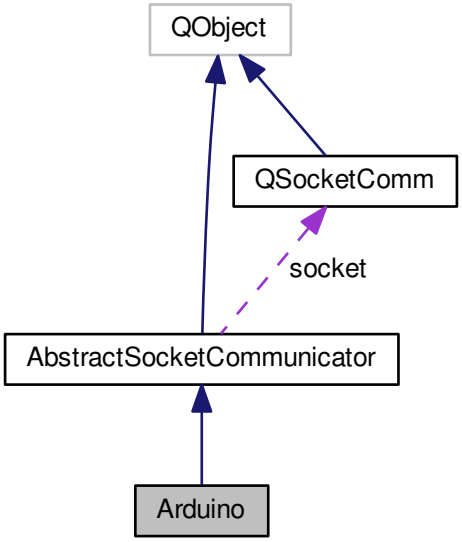
Object to send and receive commands from an Arduino Uno (R3)

```
#include <arduino.h>
```

Inheritance diagram for Arduino:



Collaboration diagram for Arduino:



Public Member Functions

- **Arduino** (std::string ip_addr, uint port_number, QObject *parent=0)
- Arduino & **operator=** (const Arduino &)=delete
- double GetCavityLength ()

  Get the current cavity length from the Arduino.

Additional Inherited Members

3.3.1 Detailed Description

Object to send and receive commands from an Arduino Uno (R3)

Arduino is expected to be equipped with an Ethernet Shield and string potentiometer.

3.3.2 Member Function Documentation

3.3.2.1 double Arduino::GetCavityLength ( )

Get the current cavity length from the Arduino.

This function will poll the Arduino until a non-empty string is returned, guaranteeing that the return value will be valid.

Note that this does not elminate the problem of getting 'doubled' responses, e.g. "7.5\r\n7.5"

Returns

Current length of the cavity, in inches

3.4 ConfigProcessor Class Reference

Public Member Functions

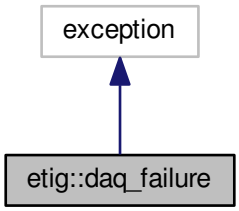
- **ConfigProcessor** (std::string file_path)
- template<typename T >
  T **GetValue** (std::string param_name)

Public Attributes

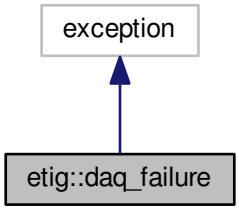
- std::vector< PlainDataParam< double > > **data_list**
- std::vector< PlainDataParam< std::string > > **string_list**
- std::vector< TCPSocketParam > **socket_info_list**

3.5 etig::daq_failure Class Reference

Inheritance diagram for etig::daq_failure:



Collaboration diagram for etig::daq_failure:



Public Member Functions

- **daq_failure** (const char ∗message)
- const char ∗ **what** () const throw ()

3.6 data_triple< T > Struct Template Reference

Public Member Functions

- **data_triple** (T cav_len, T freq, T power)
- data_triple & **operator=** (const data_triple &triple)

Public Attributes

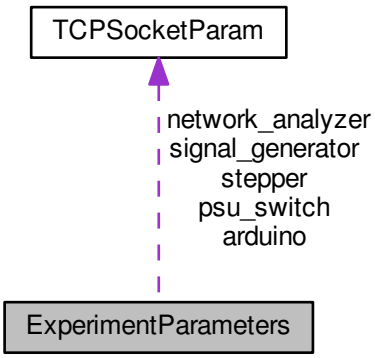
- T **cavity_length**
- T **frequency_MHz**
- T **power_dBm**

Friends

- std::ostream & **operator**<< (std::ostream &stream, const data_triple &triple)
- std::ofstream & **operator**<< (std::ofstream &stream, const data_triple &triple)

3.7 ExperimentParameters Class Reference

Collaboration diagram for ExperimentParameters:



Public Attributes

- const std::string **save_file_path** = "/home/bephillips2/workspace/Electric_Tiger_Control_Code/data/"
- const double **length_of_tune** = 3.0
- const double **revs_per_iterations** = 2.5
- const double **start_length** = 7.0
- const double **nwa_span_MHz** = 400.0
- const uint **nwa_points** = 401
- const double **nwa_power_dBm** = -15.0
- const double **freq_window_MHz** = 100.0
- const double **digitizer_rate_MHz** = 50.0
- const uint **num_averages** = 10000
- const TCPSocketParam **psu_switch** = TCPSocketParam("Switch", "10.95.100.174", 9221)
- const TCPSocketParam **network_analyzer** = TCPSocketParam("NetworkAnalyzer", "10.95.100.176", 1234)
- const TCPSocketParam **stepper** = TCPSocketParam("Stepper", "10.95.100.177", 7776)
- const TCPSocketParam **signal_generator** = TCPSocketParam("SignalGenerator", "10.95.100.175", 5025)
- const TCPSocketParam **arduino** = TCPSocketParam("Arduino", ";10.66.192.41", 23)

3.8 FlatFileSaver Class Reference

Public Member Functions

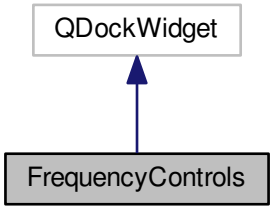
- **FlatFileSaver** (std::string save_file_path)
- void **Save** (std::vector< data_triple< double > > data_values, std::string header)

Protected Member Functions

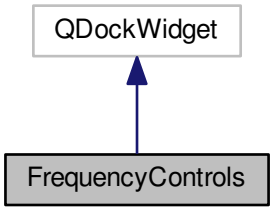
- std::string **GenerateSaveFileName** (uint index)

3.9 FrequencyControls Class Reference

Inheritance diagram for FrequencyControls:



Collaboration diagram for FrequencyControls:



Public Slots

- void **SetFreqSpan** (int span)
- void **SetMinMax** (std::pair< int, int > vals)

Signals

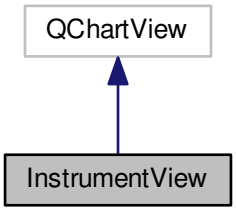
- void **MinSet** (double min_val)
- void **MaxSet** (double max_val)
- void **UnitSelected** (QString units)
- void **SpanSet** (int span_val)
- void **CenterSet** (int cent_val)

Public Member Functions

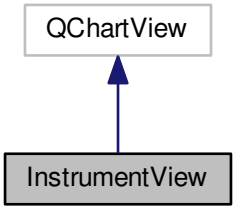
- **FrequencyControls** (QWidget ∗parent=0)

3.10 InstrumentView Class Reference

Inheritance diagram for InstrumentView:



Collaboration diagram for InstrumentView:



Public Slots

- void **SetFrequencyMin** (double min_frequency)
- void **SetPowerMin** (double min_power)
- void **SetFrequencyMax** (double max_frequency)
- void **SetPowerMax** (double max_power)
- void **UpdateSignal** (std::vector< double > data, double freq_span)

Signals

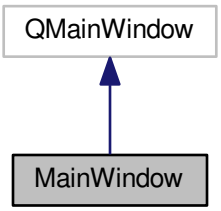
- void **SignalChanged** ()

Public Member Functions

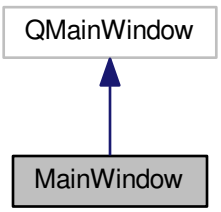
- **InstrumentView** (QString title, QWidget ∗parent=0)
- template<class T , typename F >
  void **PlotAutoScale** (const T &y_signal_elements, F x_frequency_range)
- template<class T >
  void **Plot** (const T &y_signal_elements, double x_frequency_range)

3.11 MainWindow Class Reference

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:

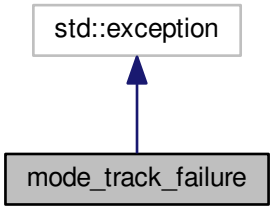


Public Member Functions

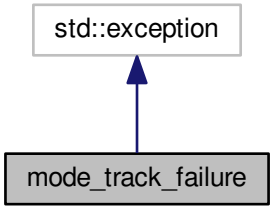
- **MainWindow** (QWidget ∗parent=0)
- void **SetSpectrumAnalyzerView** (SpectrumAnalyzer ∗spec_analyzer)
- void **SetNetworkAnalyzerView** (InstrumentView ∗network_analyzer)

3.12 mode_track_failure Class Reference

Inheritance diagram for mode_track_failure:



Collaboration diagram for mode_track_failure:



Public Member Functions

- **mode_track_failure** (const char ∗message)
- const char ∗ **what** () const throw ()

3.13 ModeTrack Class Reference

Base Class for mode tracking algorithims; designed to be wrapped with Swig and called from Python module.

`#include <modetrack.h>`

Public Member Functions

- void SetBackground (const std::vector< data_triple< double > > &background_list)

  Set background data which will be subtracted from each measurement.
- double GetPeaksGauss (const std::vector< data_triple< double > > &power_list, int mod_number)

  Identify minima peaks in a list of power data using Gaussian filtering.
- double GetPeaksBiLat (const std::vector< data_triple< double > > &power_list, int mod_number)

  Identify minima peaks in a list of power data using Bilateral filtering.
- double GetMaxPeak (const std::vector< data_triple< double > > &power_list)

  Find a local maximum in a list of data.
- void **SetLowerBound** (double frequency)
- void **SetUpperBound** (double frequency)

3.13.1 Detailed Description

Base Class for mode tracking algorithims; designed to be wrapped with Swig and called from Python module.

3.13.2 Member Function Documentation

3.13.2.1 double ModeTrack::GetMaxPeak (const std::vector< data_triple< double > > & *power_list*)

Find a local maximum in a list of data.

This method applies the same Gaussian Blur/Derivative filter combination that 'GetPeaks' uses, but does not make reference to the estimated peak positions. If multiple peaks are identified take the one with the highest overall value. This function is designed to be called when identifying peaks when looking at a transmission measurement.

Returns

Frequency of maxima, if one is found. Otherwise return value will be 0.

3.13.2.2 double ModeTrack::GetPeaksBiLat (const std::vector< data_triple< double > > & *power_list*, int *mod_number*)

Identify minima peaks in a list of power data using Bilateral filtering.

This function is very similar to GetPeaksGauss() except for the method that is used to filter data. This function makes use of a Bilateral filter for data pre-processing.

See https://users.cs.duke.edu/~tomasi/papers/tomasi/tomasiIccv98.pdf for more details.

Parameters

data_str	string containing power data that should be searched through. Needs to be in the a list of values seperated by commas, eg $\{p_1, p_2, ..., p_n\}$
mode_number	Identify which mode should be tracked. Choices are 0,1,2 and 3.

Returns

The frequency of the requested mode in MHz. If the requested mode was not found a value of 0 will be returned.

3.13.2.3 double ModeTrack::GetPeaksGauss (const std::vector< data_triple< double > > & *power_list,* int *mod_number*)

Identify minima peaks in a list of power data using Gaussian filtering.

This function is designed to be called by the main control code during data taking. The main control program will collect reflection measurements and call this function to identify the position of the mode of desire.

Parameters

data_str	string containing power data that should be searched through. Needs to be in the a list of values seperated by commas, eg $\{p_1, p_2, ..., p_n\}$
mode_number	Identify which mode should be tracked. Choices are 0,1,2 and 3.

Returns

The frequency of the requested mode in MHz. If the requested mode was not found a value of 0 will be returned.

3.13.2.4 void ModeTrack::SetBackground (const std::vector< data_triple< double > > & *background_list*)

Set background data which will be subtracted from each measurement.

Parameters

background_str	string of power values seperated by commas, eg $\{p_1, p_2, ..., p_n\}$

3.14 ModeTraits Class Reference

Public Member Functions

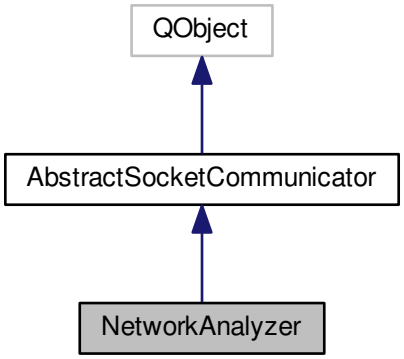
- **ModeTraits** (std::vector< data_triple< double > > data_values)
- double **Q** ()
- double **f0** ()

3.15 NetworkAnalyzer Class Reference

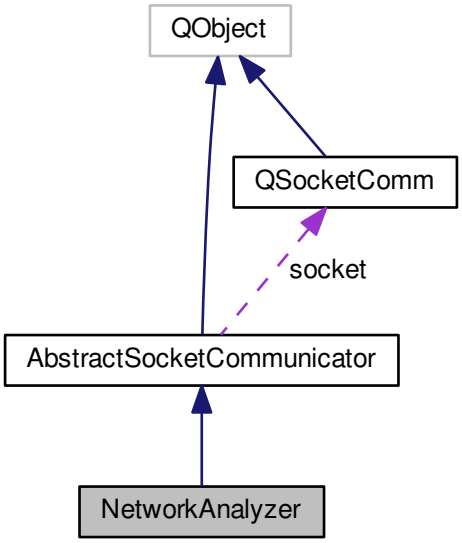
Object to communicate with the HP8757 C Network Analyzer.

```
#include <network_analyzer.h>
```

Inheritance diagram for NetworkAnalyzer:



Collaboration diagram for NetworkAnalyzer:



Public Member Functions

- **NetworkAnalyzer** (std::string ip_addr, uint port_number, uint points, double span, double power, QObject *parent=0)
- NetworkAnalyzer & **operator=** (const NetworkAnalyzer &)=delete
- std::vector< double > **TakeDataMultiple** ()
- std::vector< double > TakeDataSingle ()

Collect a single power spectrum from the Network Analyzer.

- void **SetFrequencyWindow** (double frequency, double frequency_span)
- void **SetFrequencySpan** (double frequency_span)
- void TurnOnRFSource ()

  Turn the RF source on, at whatever power it was set to most recently.
- void TurnOffRFSource ()

  Turn the RF source off.

Additional Inherited Members

3.15.1 Detailed Description

Object to communicate with the HP8757 C Network Analyzer.

3.15.2 Member Function Documentation

3.15.2.1 std::vector< double > NetworkAnalyzer::TakeDataSingle ( )

Collect a single power spectrum from the Network Analyzer.

Settings will be whatever the Network Analyzer was set to when this function is called.

Returns

3.16 PlainDataParam< T > Struct Template Reference

Public Member Functions

- **PlainDataParam** (std::string name, T val)

Public Attributes

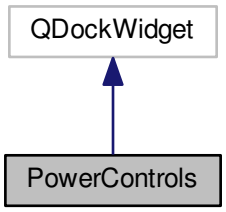
- const std::string **data_name**
- const T **data_value**

Friends

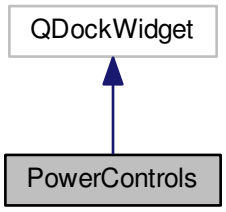
- std::ostream & **operator<<** (std::ostream &stream, PlainDataParam &param)

3.17 PowerControls Class Reference

Inheritance diagram for PowerControls:



Collaboration diagram for PowerControls:



Public Slots

- void **SetFreqSpan** (int span)
- void **SetMinMax** (std::pair< int, int > vals)

Signals

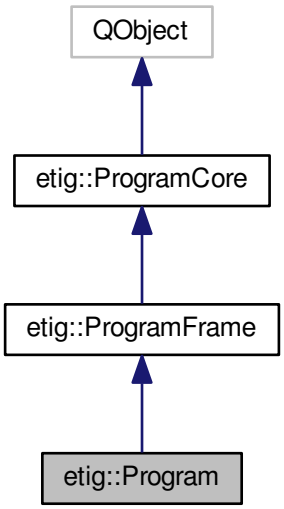
- void **MinSet** (double min_val)
- void **MaxSet** (double max_val)
- void **UnitSelected** (QString units)
- void **SpanSet** (int span_val)
- void **CenterSet** (int cent_val)
- void **SelectedVolts** ()
- void **SelecteddBm** ()

Public Member Functions

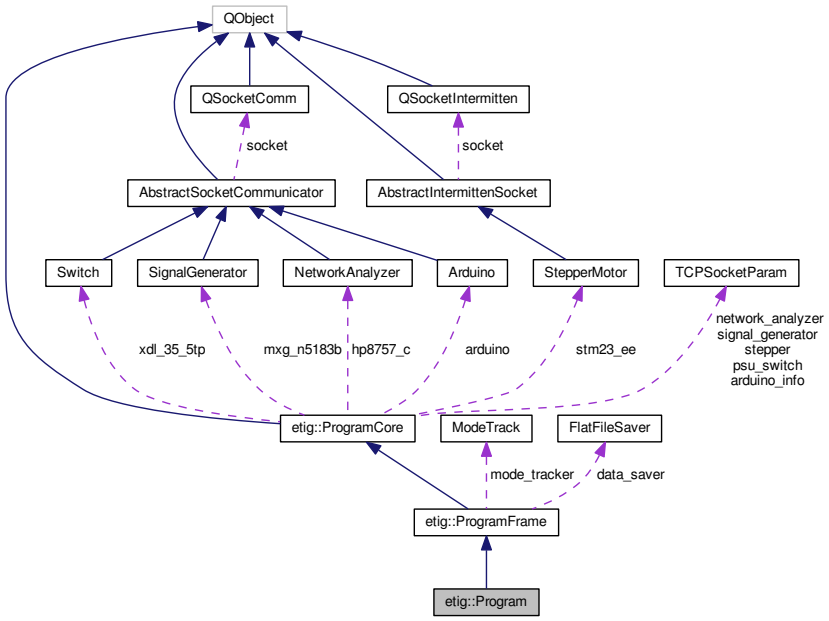
- **PowerControls** (QWidget ∗parent=0)

3.18 etig::Program Class Reference

Inheritance diagram for etig::Program:



Collaboration diagram for etig::Program:



Public Slots

- void **Run** ()
- void **Stop** ()

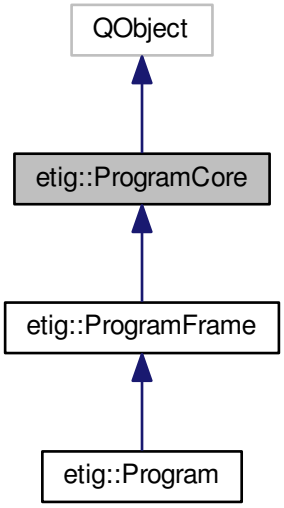
Public Member Functions

- **Program** (QObject ∗parent=0)
- double **FindModeReflection** ()
- double **FindModeTransmission** (double mode_frequency)
- data_list **TakeData** (double mode_frequency)
- void **SavePowerSpectrum** (data_list scan)
- void **PanicCleanUp** ()

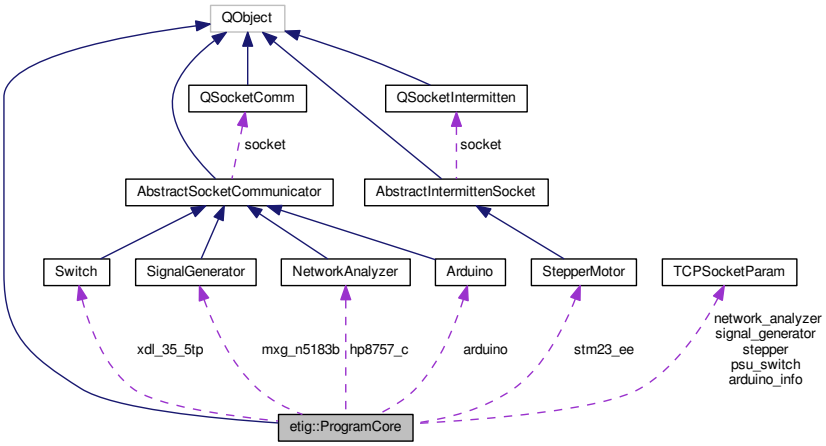
Additional Inherited Members

3.19 etig::ProgramCore Class Reference

Inheritance diagram for etig::ProgramCore:



Collaboration diagram for etig::ProgramCore:



Public Member Functions

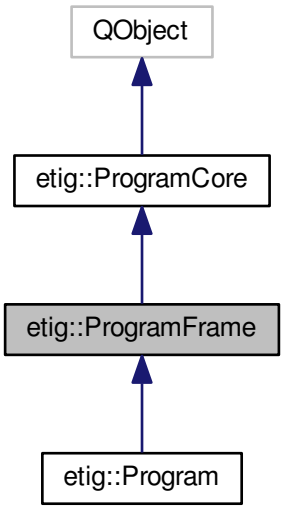
- **ProgramCore** (QObject ∗parent=0)
- void **RetractCavity** ()
- void **RapidTraverse** ()
- void **PrequelTransmission** ()
- void **PrequelReflection** ()
- void **NextIteration** ()

Protected Attributes

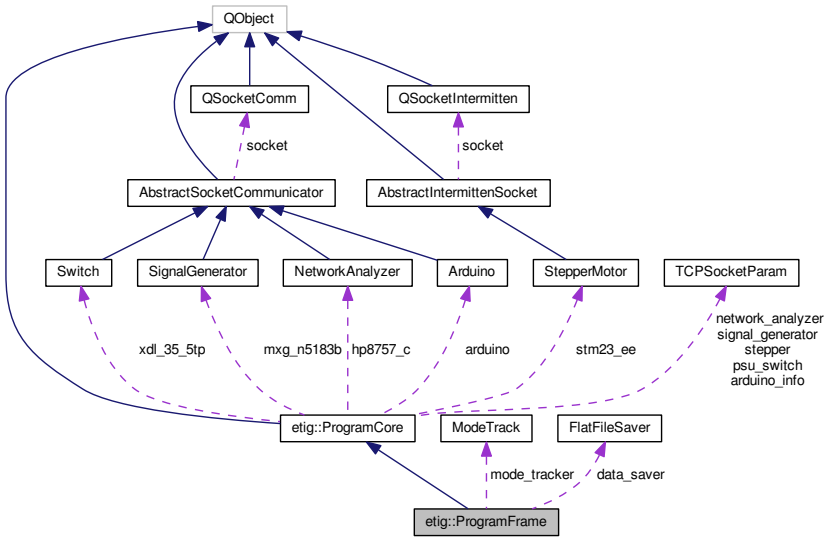
- const std::string **save_file_path** = "/home/admx/Electric_Tiger_Data/"
- const double **length_of_tune** = 2.0
- const double **revs_per_iterations** = 2.5
- const double **start_length** = 7.0
- const double **nwa_span_MHz** = 400.0
- const uint **nwa_points** = 401
- const double **nwa_power_dBm** = -15.0
- const double **signal_generator_power_dBm** = 15.0
- const double **freq_window_MHz** = 100.0
- const double **digitizer_rate_MHz** = 0.5
- const double **na_min_freq** = 3000.0
- const double **na_max_freq** = 4600.0
- const uint **num_averages** = 10000
- const TCPSocketParam **psu_switch** = TCPSocketParam("Switch", "10.95.100.174", 9221)
- const TCPSocketParam **network_analyzer** = TCPSocketParam("NetworkAnalyzer", "10.95.100.176", 1234)
- const TCPSocketParam **stepper** = TCPSocketParam("Stepper", "10.95.100.177", 7776)
- const TCPSocketParam **signal_generator** = TCPSocketParam("SignalGenerator", "10.95.100.175", 5025)
- const TCPSocketParam **arduino_info** = TCPSocketParam("Arduino", "10.95.100.173", 23)
- std::shared_ptr< ATS9462Engine > **ats9462**
- Arduino * **arduino**
- NetworkAnalyzer * **hp8757_c**
- SignalGenerator * **mxg_n5183b**
- StepperMotor * **stm23_ee**
- Switch * **xdl_35_5tp**
- double **number_of_iterations** = 0.0
- uint **iteration** = 0

3.20 etig::ProgramFrame Class Reference

Inheritance diagram for etig::ProgramFrame:



Collaboration diagram for etig::ProgramFrame:



Signals

- void **UpdateNA** (std::vector< double > na_data, double na_span)
- void **UpdateSpec** (std::vector< float > spec_data, uint digi_rate)

Public Member Functions

- **ProgramFrame** (QObject *parent)
- void **Prequel** ()
- void **ShiftFrequencyWindow** (double center_frequency)
- void **SetBackground** ()
- double **FindMinimaPeak** (data_list formatted_points)

Protected Member Functions

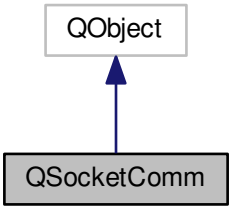
- data_list **power_to_data_list** (std::vector< float > power_list, float min_freq, float max_freq)
- data_list **power_to_data_list** (std::vector< double > power_list, double min_freq, double max_freq)
- template<typename T >
  std::vector< T > **data_list_to_power** (std::vector< data_triple< T > > data)
- double **CheckPeak** (double possible_mode_position)
- std::string **BuildHeader** ()

Protected Attributes

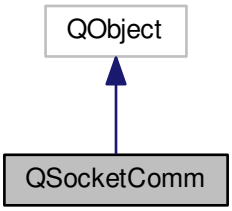
- ModeTrack **mode_tracker**
- FlatFileSaver **data_saver** { save_file_path }

3.21 QSocketComm Class Reference

Inheritance diagram for QSocketComm:



Collaboration diagram for QSocketComm:

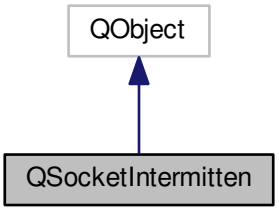


Public Member Functions

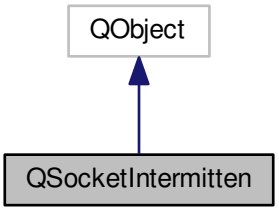
- **QSocketComm** (std::string host_name, uint port_number, QObject ∗parent=0)
- void **Send** (std::string command, std::string terminator="\n")
- void **SendScl** (std::string command)
- std::string **Receive** ()
- std::string **ReceiveSafe** ()

3.22 QSocketIntermitten Class Reference

Inheritance diagram for QSocketIntermitten:



Collaboration diagram for QSocketIntermitten:



Public Member Functions

- **QSocketIntermitten** (std::string host_name, uint port_number, QObject ∗parent=0)
- void **OpenConnection** ()
- void **CloseConnection** ()
- void **Send** (std::string command, std::string terminator="\n")
- void **SendScl** (std::string command)
- std::string **Receive** ()
- std::string **ReceiveSafe** ()

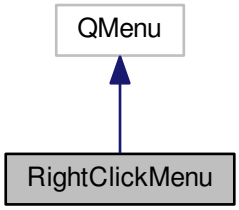
3.23 RetrieveVal Struct Reference

Public Member Functions

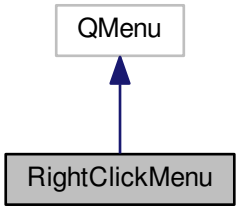
- template<typename T >
  T::first_type **operator()** (T keyValuePair) const

3.24 RightClickMenu Class Reference

Inheritance diagram for RightClickMenu:



Collaboration diagram for RightClickMenu:



Signals

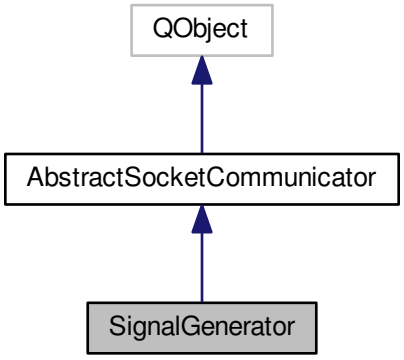
- void **Scaling** ()

Public Member Functions

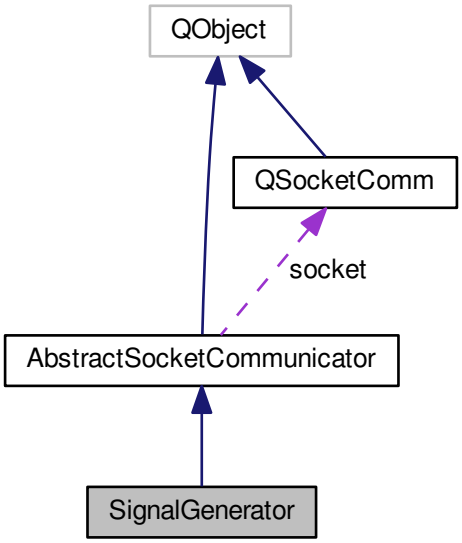
- **RightClickMenu** (QWidget ∗parent)

3.25 SignalGenerator Class Reference

Inheritance diagram for SignalGenerator:



Collaboration diagram for SignalGenerator:



Public Member Functions

- **SignalGenerator** (std::string ip_addr, uint port_number, QObject *parent=0)
- SignalGenerator & **operator=** (const SignalGenerator &)=delete

- void **RFOn** ()
- void **RFOff** ()
- void **SetFrequency** (double freq_MHz)
- void **SetPower** (double power_dBm)

Additional Inherited Members

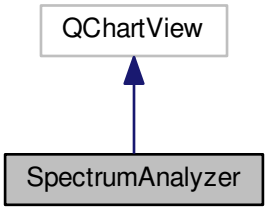
3.26 SocketComm Class Reference

Public Member Functions

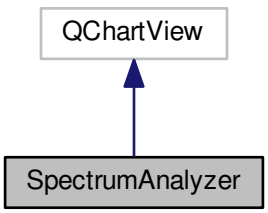
- **SocketComm** (std::string host_name, uint port_number)
- void **Send** (std::string command, std::string terminator="\n")
- void **SendScl** (std::string command)
- std::string **Receive** ()
- std::string **ReceiveSafe** ()

3.27 SpectrumAnalyzer Class Reference

Inheritance diagram for SpectrumAnalyzer:



Collaboration diagram for SpectrumAnalyzer:



Public Slots

- void **UpdateSignal** (std::vector< float > time_series, uint sample_rate)
- void **SetFrequencyMin** (double min_frequency)
- void **SetPowerMin** (double min_power)
- void **SetFrequencyMax** (double max_frequency)
- void **SetPowerMax** (double max_power)
- void **ChangeToVolts** ()
- void **ChangeTodBm** ()

Signals

- void **SignalChanged** ()

Public Member Functions

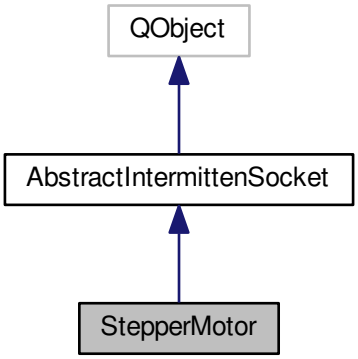
- **SpectrumAnalyzer** (QWidget *parent=0)
- template<class T , typename F >
  void **PlotAutoScale** (const T &y_signal_elements, F x_frequency_range)
- template<class T >
  void **Plot** (const T &y_signal_elements, double x_frequency_range)

3.28 StepperMotor Class Reference

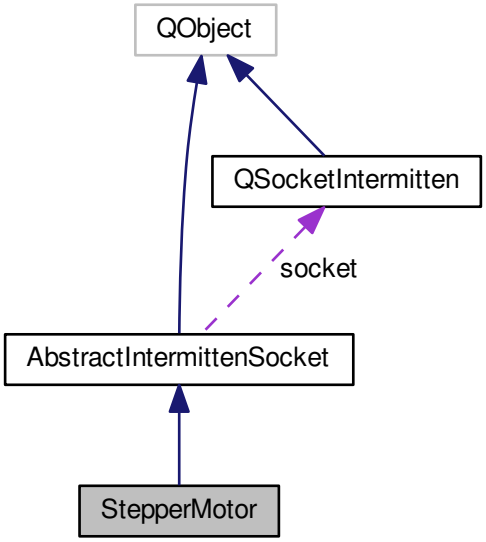
Object to sends commands to an Applied Motion products stepper motor.

`#include <stepper_motor.h>`

Inheritance diagram for StepperMotor:



Collaboration diagram for StepperMotor:



Public Member Functions

- **StepperMotor** (std::string ip_addr, uint port_number, QObject *parent=0)
- StepperMotor & **operator=** (const StepperMotor &)=delete
- void **SetToInitialLength** (double initial_length, double current_length)
- void **TuneCavity** (double length_of_tune)
- void **PanicResetCavity** (uint iteration, double revs_per_iter)
- void **TuningLoop** (double len_of_tune, double revs, uint iters)

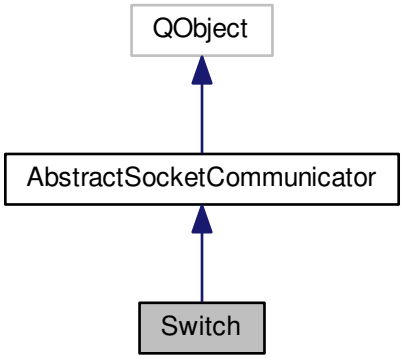
Additional Inherited Members

3.28.1 Detailed Description

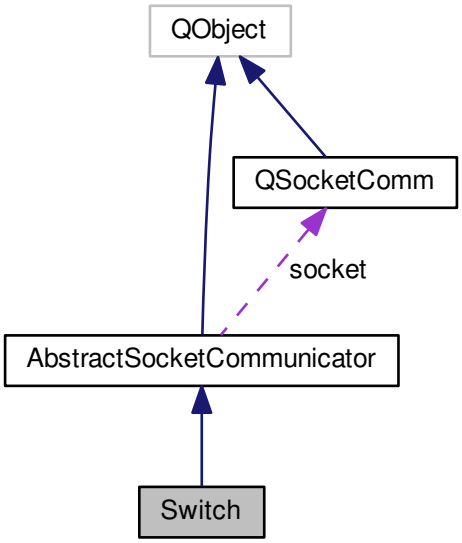
Object to sends commands to an Applied Motion products stepper motor.

3.29 Switch Class Reference

Inheritance diagram for Switch:



Collaboration diagram for Switch:



Public Member Functions

- **Switch** (std::string ip_addr, uint port_number, QObject *parent=0)
- Switch & **operator=** (const Switch &)=delete

- void **SwitchToNetworkAnalyzer** ()
- void **SwitchToDigitizer** ()
- void **SwitchToTransmission** ()
- void **SwitchToReflection** ()

Additional Inherited Members

3.30 TCPSocketParam Struct Reference

Public Member Functions

- **TCPSocketParam** (std::string name, std::string addr, uint port)
- **TCPSocketParam** (const std::string &name, const std::string &addr, uint port)
- TCPSocketParam & **operator=** (const TCPSocketParam &sock_param)

Public Attributes

- const std::string **inst_name**
- const std::string **ip_addr**
- const uint **port_addr**
- std::string **inst_name**
- std::string **ip_addr**
- uint **port_addr**

Friends

- std::ostream & **operator**<< (std::ostream &stream, TCPSocketParam &param)
- std::ostream & **operator**<< (std::ostream &stream, TCPSocketParam &param)

3.31 TestConfigProcessor Class Reference

3.32 etig::test::TestInstrumentView Class Reference

Public Member Functions

- void **Test** ()

3.33 etig::test::TestPanels Class Reference

Public Member Functions

- void **Test** ()

3.34 etig::test::TestSpectrumAnalyzer Class Reference

Public Member Functions

- void **Test** ()

3.35 etig::test::VoltsSqrTodBm Struct Reference

Public Member Functions

- void **operator()** (data_triple< double > &data) const

Index