SD3K Quick Start Guide

*Smart grid Synchrophasor Software Development Toolkit*

Repository

<https://github.com/SmarTS-Lab-Parapluie/S3DK>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **User** | **Comment** |
| 1 | 206-07-14 | L. Vanfretti | First Version – General Installation from Source is Missing |

# Introduction

## Background

NOTE: Usage of the PRL will require some knowledge of Labview. This documentation is NOT an introduction to Labview.

## Vocabulary

|  |  |
| --- | --- |
| Client | The client application that connects to the data queue, or reads data from the Access Buffer.  The software bundle contains an example Client that also starts/stops the PRL remotely. |
| C37 Device | The device (PMU or PDC) that sends data on the C37.118 protocol. |

# Installation

Instructions for installation from source to be added.

Quick Install: Go to the ”Releases”: <https://github.com/SmarTS-Lab-Parapluie/S3DK/releases> and install the software using the VI package manager in LabView.

# User Manual

## Prerequisites

The user should have some familiarity with Labview, or go through the first five modules here:

<http://www.ni.com/white-paper/7466/en>

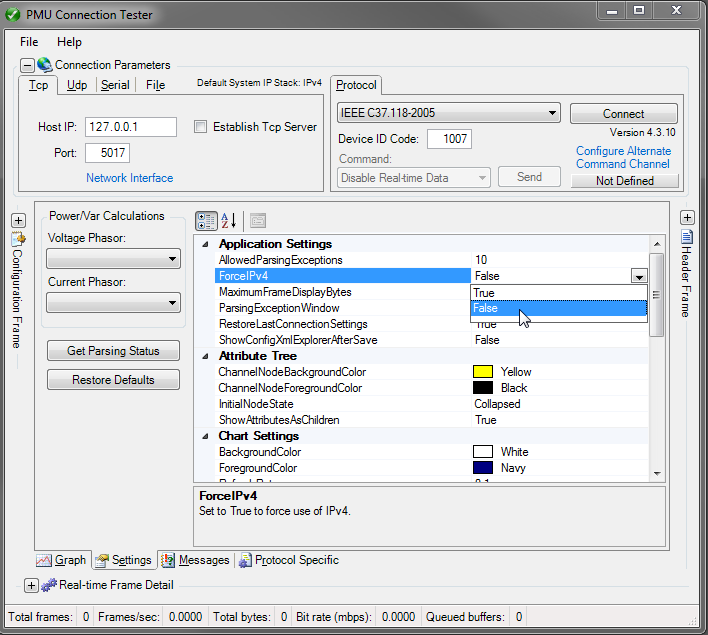
## Useful Software

### PMU Connection Tester

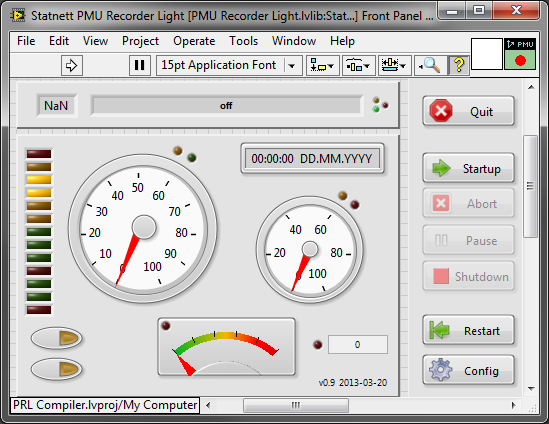
As the name indicates it can be used to test the connection to a PMU or a PDC, to see that all the signals are arriving as they are supposed to. It can be downloaded from this site:

<http://pmuconnectiontester.codeplex.com/>

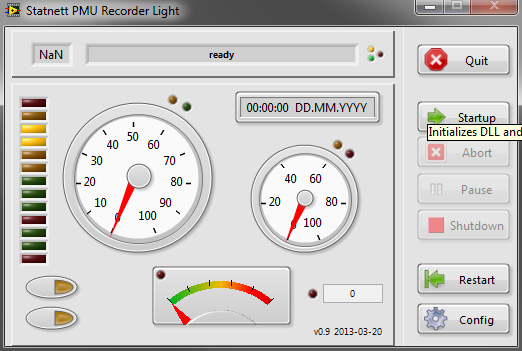
Before using the connection tester, you will need to set it to use Ipv4, in the Settings tab:



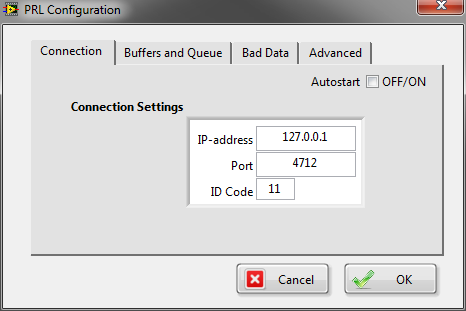
## Quick Start



The main window is depicted above. Press "run" (the arrow to the upper left) to start the PRL, which will now look like this:



Press the Config button to open the Configuration Window.



The Connection tab always shows first. Enter the IP-address, Port and PDCID of the connection, and press OK.

Now press Startup to activate the connection. The PRL will retry the connection until it comes alive. If it does not work, press Abort to cancel or press Quit directly.

NOTE: If the IP-address and/or the port is invalid, this will cause Labview to hang for a while. Problem is solved by waiting a bit, or restart.

If there is a problem with the connection, use the PMU Connection tester to verify that the connection settings are correct. The signals should arrive fine there before you try to connect with the PRL.

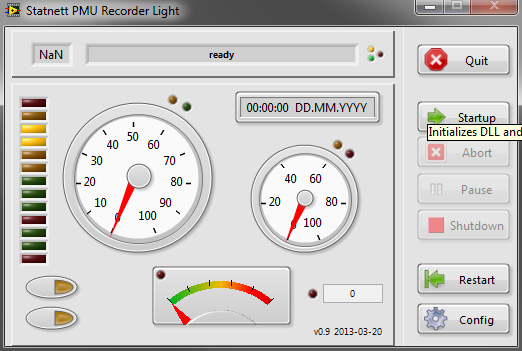
When the connection is active, the Pause and Shutdown buttons are activated and the indicators will show activity. This means data is available and can be retrieved from the Access Buffer.

Hold the mouse pointer over the different indicators and buttons to see what they do, if this is unclear, or look at the detailed description below.

Press Shutdown to stop the data stream and Quit to exit the application.

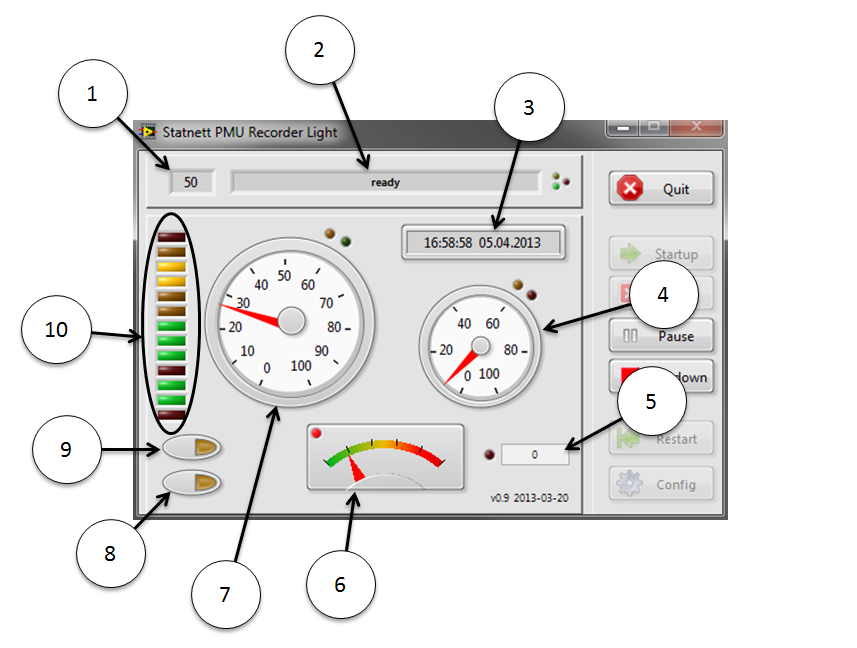
## Main Window

### Buttons



|  |  |
| --- | --- |
| Quit | Stops data collection and exits the program |
| Startup | Connects to the C37 device using the connection settings set in Config. |
| Abort | If connection fails, the software will wait a given amount of time (as set in Config) before retrying. This wait may be aborted with this button. |
| Pause | Pauses/continues data collection. The data will be buffered for a given amount of time in the DLL. |
| Shutdown | Disconnects from the C37 device. |
| Restart | Reinitializes the software according to current settings. |
| Config | Opens the configuration window. |
| Raw Data Indicator (9 below) | Opens a panel showing all incoming channels and raw data. |
| Channel Selector (8 below) | Opens a panel where you can select which signals should be the output of the Snapshooter. |

### Indicators



|  |  |  |
| --- | --- | --- |
| **1** | Data Rate | The number of data frames received per second. |
| **2** | Software Activity | Indicates the current operation, if any. Ready otherwise. |
| **3** | Time Stamp | The time stamp of the latest received data frame. |
| **4** | Access Buffer Write Rate | Indicates how much time the software spends on writing data to the access buffer. |
| **5** | Bad dt Counter | Counts the number of dt above sampling period. Zero means no data has been lost. |
| **6** | Data Quality Indicator | Shows the percentage of missing data. The maximum value of the indicator is 50%.  NOTE: If the "Remove Bad Data" selection in Config is unchecked, bad data will not be tagged, and this indicator will always show zero. |
| **7** | Data Frame Processing Rate | Indicates the amount of time that the software spends on processing data frames. 0 means no data is coming and hence that the data stream has stopped. 100 means data processing is continuous, and hence that there is an incoming data overload. |
| **8** | Channel Selector | Open/close the Channel Selector window |
| **9** | Raw Data Indicator | Open/close the raw data display |
| **10** | Status Indicators | Indicates the status of the connection and of the software. Hold the mouse pointer over the different lights to see what they mean. |

# 

|  |  |  |
| --- | --- | --- |
| **1** | Incoming Data | Blinks when data is received |
| **2** | Data to Buffer | Blinks when data is transferred from live buffer to access buffer |
| **3** | System Activity | System Activity |
| **4** | System Exit | System Exit |
| **5** | Processing | Processing |
| **6** | Data Lost | Data lost on the way to the access buffer |
| **7** | Data Received | Data received in the access buffer |

# Configuration

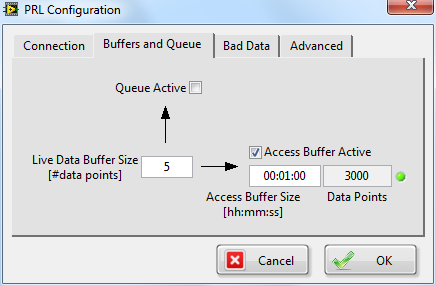
## Connection

## C:\Users\vemunda\Dropbox\1 Project\06 Strongrid\PRL\PRL\PRL Config 1.png

**Connection Settings** contain the IP-address, Port and PDC ID of the PMU/PDC. These are the same as would be set in the PMU Connection Tester.

The **Autostart** feature determines whether the connection will be made automatically on software startup (this means that when calling the PRL remotely, it is sufficient to send the "Run" command. The "Startup" command automatically follows).

## Buffer and Queue



NOTE: The connection to the PMU/PDC must first be tested in the PRL, since the buffer size depends on sampling rate. This is when the PRL connects to the PMU/PDC.

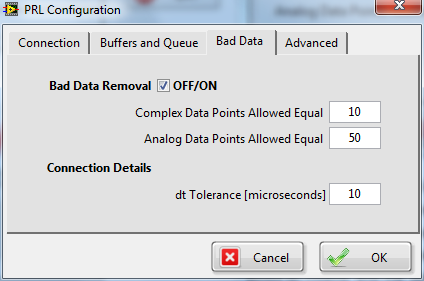
The **Live Data Buffer** **Size** sets the number of data frames that will be read from the incoming data before the data is passed on to the Access Buffer and/or the output queue. This may be set to 1, which means the data will go directly to the buffer/queue. Using values above 1 may however prove appropriate for data efficiency purposes. The size of this buffer will determine the maximum usable refresh rate when reading data from the Access Buffer.

**Access Buffer Size** determines the amount of data made available to the client application.

**Queue Active** and **Access Buffer Active** switches off/on the respective data streams.

**Data Points** indicates the actual amount of data frames contained in the Access Buffer, and the LED to the right indicates whether the combination Live Data Buffer Size, Access Buffer Size and PMU/PDC data rate gives an integer amount of data points. In other words it indicates if the length of the data buffer, in time, is in fact the length stated in Access Buffer Size.

## Bad Data



When a signal is missing, the phasors coming from a PMU is typically 0, but may also be 1. Sometimes the signal is maxed out. The same goes for the analog frequency signal. When the signal is missing, it might give an output of 50 Hz flat instead of 0 Hz. It is therefore difficult to determine if the signal is valid based on limits. Instead the software is set to compare succeeding data points, and if enough of them are equal the signal is treated as invalid, and the data is replaced with NaN's until the values come alive.

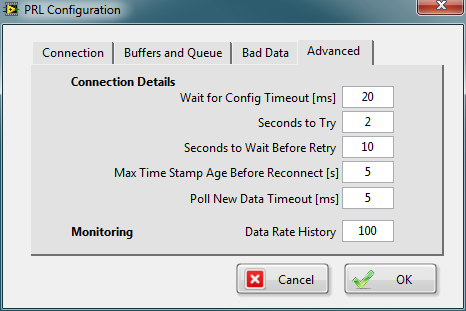
A certain type of PMU’s gave some problems with this solution. Since the resolution of the signal was quite low, steps were visible when looking closely at the signal. This resulted in the software somtimes producing invalid frequency signals—even when the number of allowed data points was set to 50.

When setting these values, care should therefore be taken with the frequency signal. Make sure the signal is floating or set the number of allowed data points high.

NOTE: When using a test signal, this signal may be a steady flat value. For testing purposes it may therefore be best to switch off the Bad Data Removal alltogether.

**dt Tolerance** is the accepted slide in time between two data frames. The GPS clocks that PMUs use to synchronize has an accuracy of one microsecond. The default ten microseconds is therefore a good start.

## Advanced



**Wait for Config Timeout** decides for how long the software should wait for the PDC/PMU to supply a configuration file, and **Seconds to Try** is the time it will continue to look for the configuration file. If no configuration is received, **Seconds to Wait if Config Fails** determines how long the software will wait before retrying.

**Max Time Stamp Age Before Reconnect** is the the maximum allowed lag of the incoming PMU-data to the system clock of the computer running the PRL. The background for this is that if the PDC restarts, the PRL wil not automatically continue reading the data stream. Instead it requires a new connection.

NOTE: If the configuration has changed after a reconnect, this is not detected by this version of the PRL.

**Poll New Data Timeout** is how long each check for a newly arrived data frame waits. This number affects the Data Frame Processing Rate.

**Data Rate History** is the number of data points used to determine how busy the software is (Data Frame Processing Rate). The timeout in the procedure waiting for new data to arrive is 5 ms. If more than 5 ms passes before the software receives new data, this counts as a no data period. The history of data/no data gives a percentage of time used on processing data, and a low value will mean the data stream has stopped, while a high value will mean too much processing power is used on processing data. This may also indicate that the computer is overloaded with other tasks.

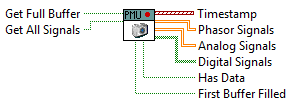
If the Data Frame Processing Rate peaks at 100, this means all power is used on processing data and it indicates that data may be piling up in the DLL buffer and that there is a risk of eventually loosing data. If this happens, it could for example indicate that the Client is using too much resources on processing. The Data Frame Processing Rate is not the same as CPU Usage. It is an artificial, yet good, indicator of how stressed the PRL is. A good number is 30 to 60.

# Access Components

The following is a list of the components that may be used to access the PRL—both for reading data and starting/stopping it without opening the window.

NOTE: Usage of analog signals is "user defined" by the protocol, and has therefore not been fully tested.

### PRL Read Buffer



The Snapshooter reads data from the Access Buffer. There are two main ways of usage:

1. Inserting the VI in a timed loop will read data from the Access Buffer at regular intervals. By keeping the input and output time stamps the VI will return only new data. The amount of data received each time may vary, since at a data rate of 50, there is only 20 ms between each new data frame.
2. By activating the **Get Full Buffer** option, the VI will return the full buffer each time it is used. The length of this buffer is set in the PRL configuration. This is useful for making a sliding FFT or similar operations that require a fixed length buffer at regular intervals.

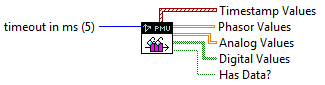
### Inputs

|  |  |
| --- | --- |
| Get Full Buffer | Ignores the input time stamp and returns the full available buffer |
| Get All Signals | Skips the signal filtering made by the Channel Selector, and sends all avaliable signals to the data outputs |

### Outputs

|  |  |
| --- | --- |
| Time Stamp | Time stamp array |
| Phasor Signals | Phasor data array |
| Analog Signals | Analog data array |
| Digital Signals | Digital data array |
| Has Data | Indicates if new data has been found |
| First Buffer Filled | Gives false until the buffer has been filled. When the buffer is filled, it will remain fillled since it is a rotating buffer. |

## PRL Read Queue



This component reads data from a queue directly from the Live Buffer. The size of this buffer is set in the PRL Configuration, and is entered as number of data points. The minimum value is 1.

The Queue Reader is designed for applications that need a steady flow of data sets of equal length, or in general faster access to the data.

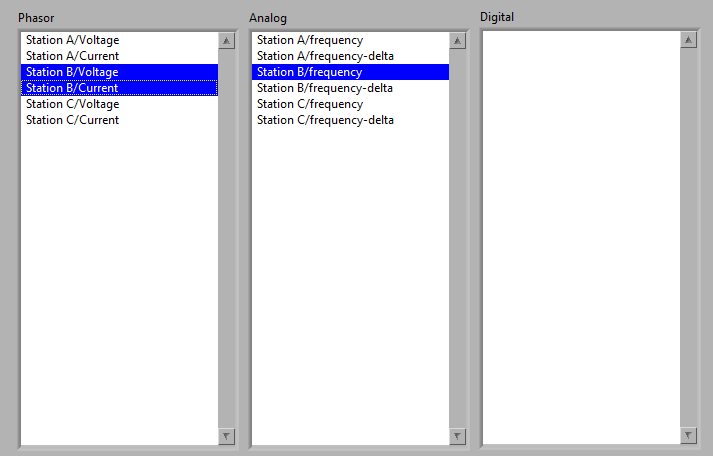
### Inputs

|  |  |
| --- | --- |
| Timeout in ms | Decides how long the VI will wait for new data to arrive before exiting. Setting depends on the design of the client software. The default value is 5 milliseconds. Entering -1 will cause the VI never to time out. This is not recommended, but is one way of doing it. |

### Outputs

|  |  |
| --- | --- |
| Timestamp Values | Time Stamp Array |
| Phasor Values | Phasor Data Array |
| Analog Values | Analog Data Array |
| Digital Values | Digital Data Array |
| Has Data? | Indicates if new data has arrived before timeout. If not, read again until it does. |

### PRL Channel Selector



The Channel Selector is used to select which channels are supplied in the output. The selected channels will appear in order in the output of the Read Buffer or Read Queue functions.

### PRL Channel Selector Configuration (Advanced)



|  |  |
| --- | --- |
| Action | Executes one of the following actions:   * Get All Signal Names delivers all available signal names to the Signal Names Out node * Select Signals on Name will select all signals in the Signal Names In node * Select Frequency Only will select all analog signals that are frequency * Get Selected Signal Names returns the selected signal names to the Signal Names Out node |
| Get All Signals | Overrides the Channel Selector and delivers all available signals to the outputs. |
| Signal Names In | Input for a cluster of phasor, analog and digital signal names |
| Signal Names Out | Output for a cluster of signal names |

### PRL Check Bad dt



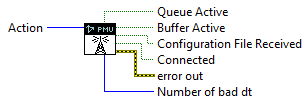
Returns true if the number of detected bad dt since last time the function was called, is higher than the input Allowed Increment. The default is 0.

### PRL PMU Status



Reads the latest PMU Status from the PRL.

### PRL Remote (Advanced Users)



The PRL Remote allows a Client application to programmatically send commands to the PRL. The following commands are available:

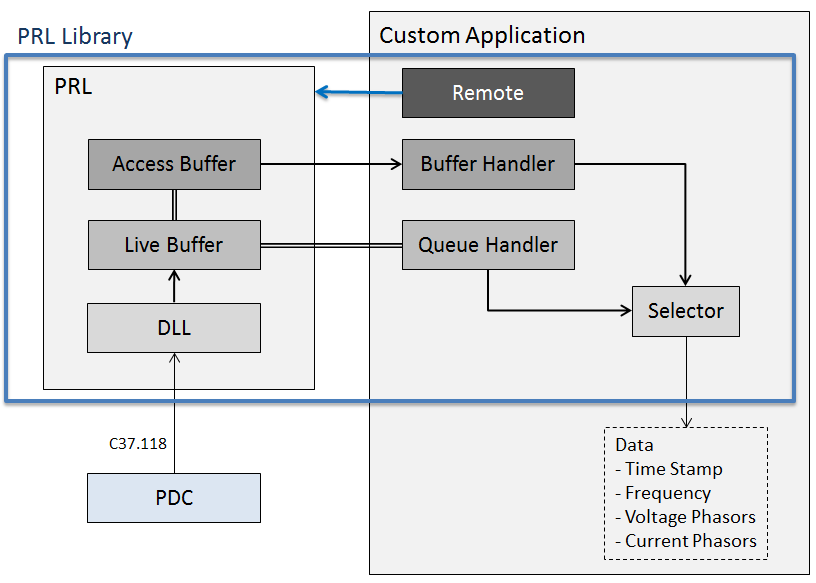
|  |  |
| --- | --- |
| **Run** | Starts the PRL |
| **Startup** | Corresponds to pressing the Startup button. If Autostart is switched on, the startup will automatically follow the Run command. |
| **Get Status** | Returns the following parameters indicating the status of the PRL:   * (Access) Queue Active * (Access) Buffer Active * Configuration File Received * Connected (to C37 Device) * Number of bad dt (returns the number of bad dt that has been detected since startup of the PRL)   NOTE: All these values are returned as false (and are not valid) if Action is not set to Get Status. |
| **Pause** | Corresponds to pressing the Pause button of the PRL. |
| **Continue** | Corresponds to pressing the Continue button of the PRL. |
| **Shutdown** | Corresponds to pressing the Shutdown button of the PRL. |
| **Abort** | Corresponds to pressing the Abort button of the PRL. |
| **Open Panel** | Opens the PRL panel. |
| **Close Panel** | Closes the PRL panel. |

# Software Design

The PRL is a packed Labview library. The code is not open to the user. Usage is meant to be through remote or manual control of the PRL, and using the Access Buffer or Access Queue remote components to read data, as well as using the Channel Selector to select which data is retrieved.

### PRL Library Contents

The PRL Library supports a Client application by taking care of connection to the C37 Device and handle PMU-data buffering. The PRL component connects to the device, reads data and puts these in an Access Buffer and/or queue. The Client may access these data through separate components that communicate with the PRL remotely.



The library has two parts:

* The **PRL** is a separate component which has to be started and stopped remotely or manually. It reads data and makes them accessible from a buffer and a queue.
* The **Access Components** are used to start/stop the PRL, as well as access the data in different ways. These components are not self-supportive and merely represent an entrance point for the Client application to the PRL.

#### PMU Recorder Light (PRL)

* Uses a DLL[[1]](#footnote-2) to connect to the PDC stream via the C37.118 protocol.
* Receives configuration (channel names and scaling) from the PDC
* Stores incoming data in the Live Buffer
* Sends data on a queue to the Queue Handler and to the Access Buffer
* Has remote controls for start/stop
* Reconnects when signal is lost

#### Access Components

##### Queue Handler

* Used in the Custom Application to read data from the queue

##### Buffer Handler

* Used in the Custom Application to read data from the Access Buffer
* Has variable sample rate

##### Selector

* Provides a tool to select channels from the Custom Application

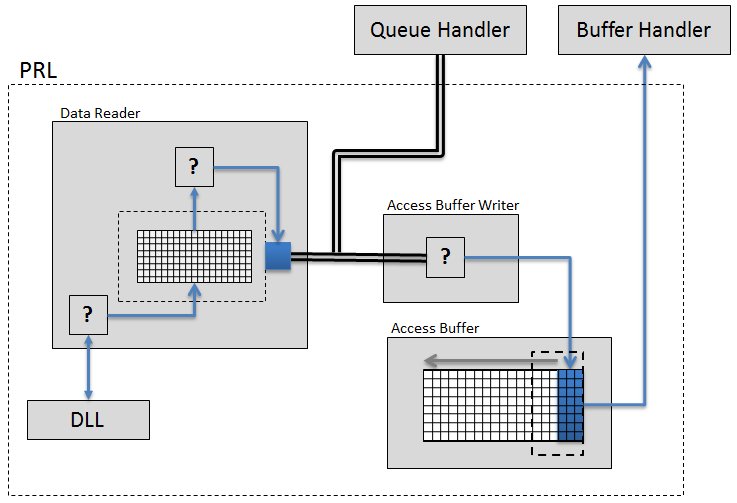
##### Remote

* Allows the user to send commands to the PRL from the Custom Application

### PMU Recorder Light (PRL)

The PRL shall make PMU-data readily available to the Custom Application, and take care of the full connection with the PDC. The purpose being that the researcher will not have to deal with complicated data handling.

Streaming data must be buffered to handle variations in the stream. The PRL has the data flow depicted below.



The Data Reader asks the DLL regularly for new data. New data is then put in the Data Buffer, and when the buffer is full the data is put in a queue that can be received by the Custom Application using the Queue Handler, and also sends the data to the Access Buffer via the Access Buffer Writer.

The Access Buffer size decides the amount of data history in memory, and data can be read from the buffer in the Custom Application via the Buffer Handler.

### Custom Application

Due to the design of the PRL and the library, the Custom Application will need a particular design to use the components:

* The PRL must be started with the Remote before data can be read. Also, the PRL must be stopped by the Custom Application after use. Otherwise it will continue running in memory—although this will not cause memory overflow.
* If using the Queue Handler, the data must be read at regular intervals—otherwise data will be lost
* The same goes for the Buffer Handler, and when using this component there must be an input for update rate (how often the data will be read)
* The Selector is a separate window, and in order to use it, the Custom Application must have logic for opening/closing this window

The delivery will contain a sample project, which uses the PRL Library with all its features. This can be modified to suit the user’s specific needs. The delivery will also contain a detailed user manual and an installation package for the Windows 7 operating system and Labview 2011 or later.

1. A DLL will be developed in a different project, but in the first phase an old DLL will be used. [↑](#footnote-ref-2)