Serial PlugIn
How to integrate your own device to IPEmotion

Felix Ottofueling 20.12.2012

Contents

Introduction

- The serial PlugIn is a very flexible PlugIn which supports data communication to many serial devices.
- Anybody can integrate his own device by developing the device specific “acquisition.dll”.
- Supported interface of this PlugIn version is based on RS232.
- The Visual Studio 2010 solution template for the C++ interface to handle the data communication is available.
- The PlugIn supports only to read data. You cannot write or send values from IPEmotion to the serial device.

Overview of the architecture

IPEMOTION

Test & Measurement Software

IPEmotion

Serial PlugIn

+

Device 1.Acquisition.dll ← RS232 reading

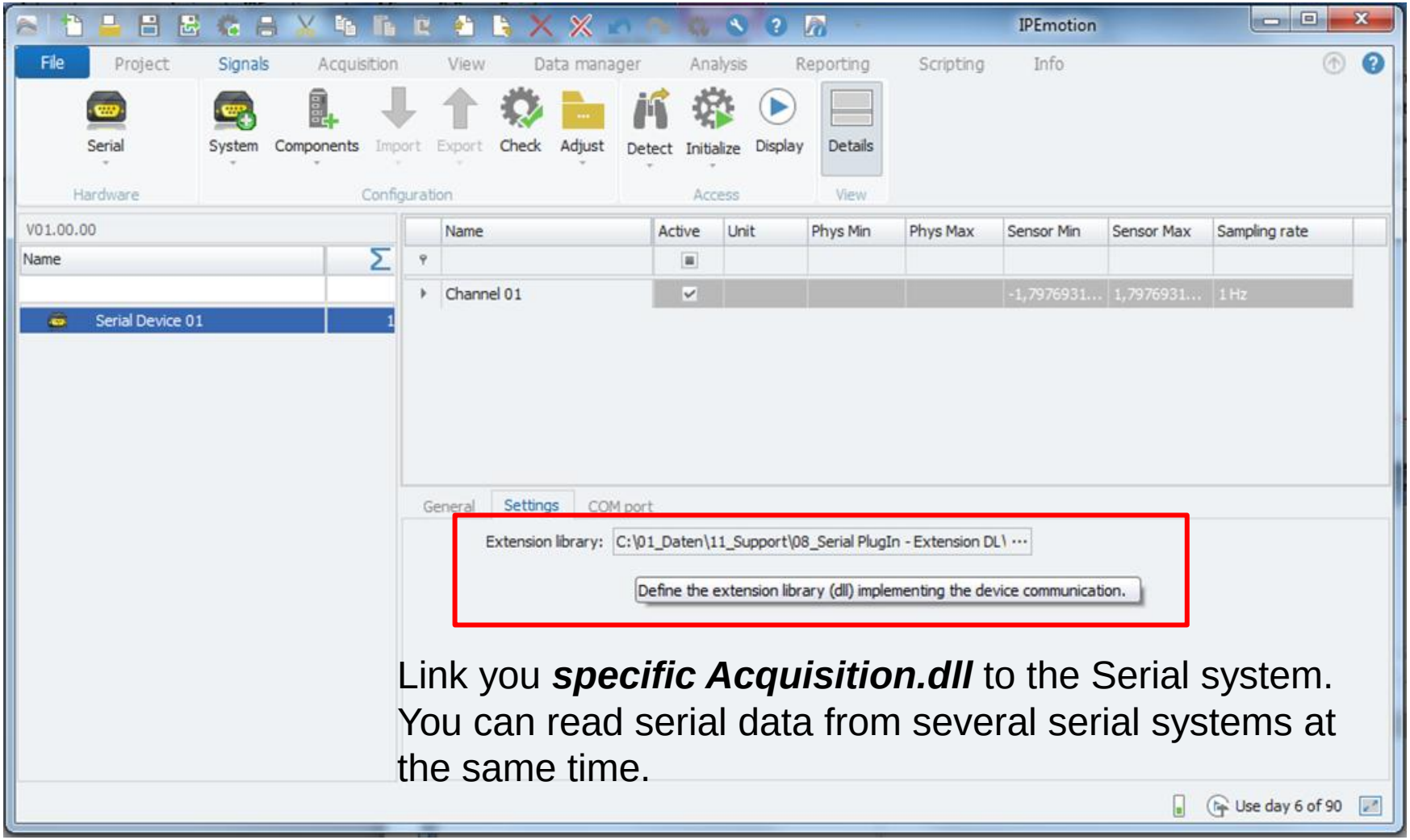
+

Device 2.Acquisition.dll ← RS232 reading

+

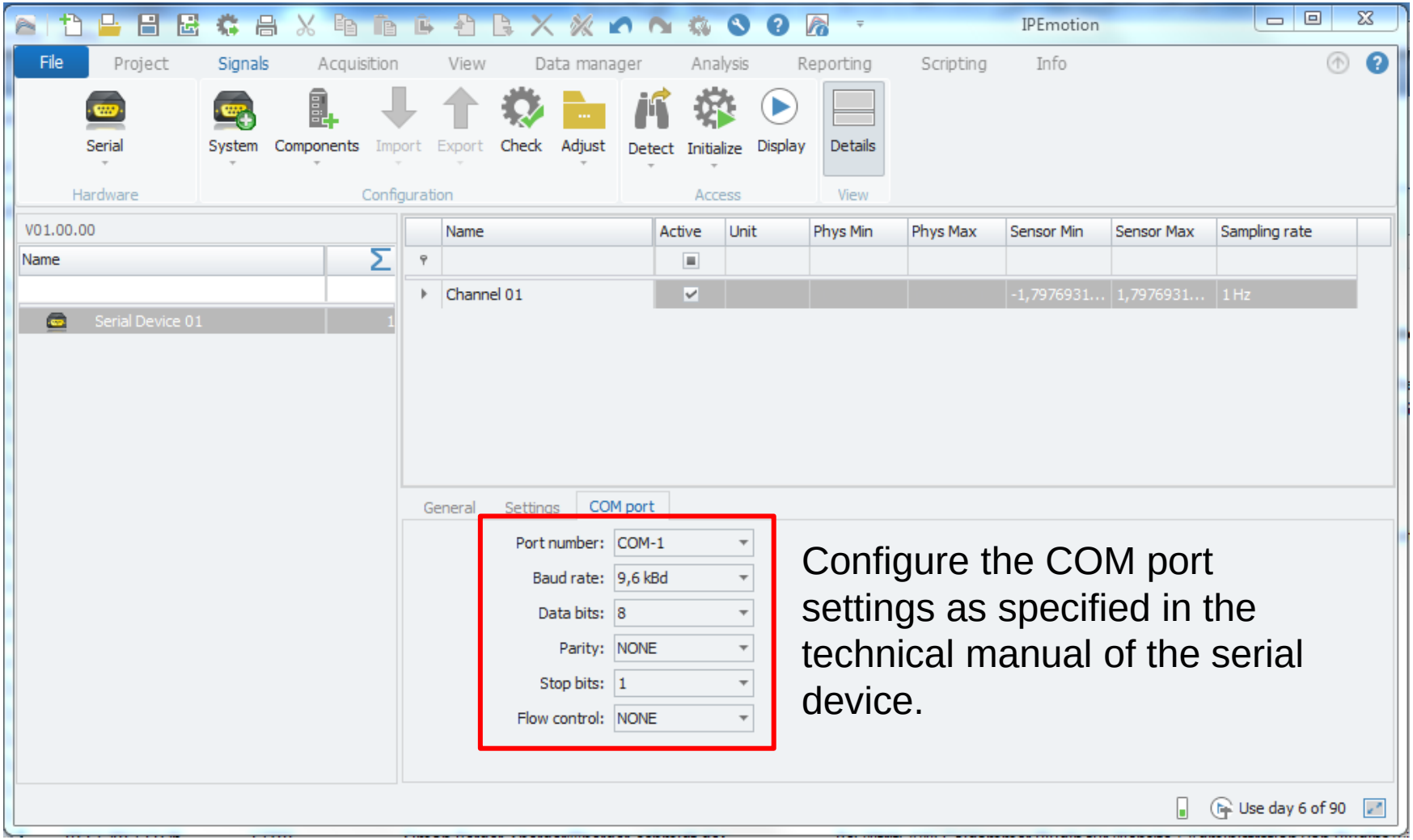
... n other device ← RS232 reading

IPEmotion interface 1



Link you ***specific Acquisition.dll*** to the Serial system. You can read serial data from several serial systems at the same time.

IPEmotion interface 2



Configure the COM port settings as specified in the technical manual of the serial device.

Visual Studio template of Acquisition.dll

```
#pragma region Copyright 2012 IPETRONIK GmbH & Co.KG
//
// All rights are reserved. Reproduction or transmission in whole or in part, in
// any form or by any means, electronic, mechanical or otherwise, is prohibited
// without the prior written consent of the copyright owner.
//
// KGI
//
#pragma endregion

////////////////////////////////////////////////////////////////////////////////
/// <summary>
/// include
/// </summary>
#include "Extension.h"

////////////////////////////////////////////////////////////////////////////////
/// <summary>
/// lib
/// </summary>
#pragma comment(lib, "Version.lib")

////////////////////////////////////////////////////////////////////////////////
/// <summary>
/// Global member: callback functions
/// </summary>
PFN_EXT_SET_PACKET_DATA_FLOAT64_FUNC g_pfnDataFloat64 = NULL;      // transfer packet data of double type to the plugin
PFN_EXT_SEND_DATA_FUNC               g_pfnSendData = NULL;         // send data via the systems COM port
PFN_EXT_READ_INPUT_BUFFER            g_pfnReadInputBuffer = NULL;  // read new data received via the
                                                                   // systems COM port since last call of g_pfnReadInputBuffer
PFN_EXT_RESET_INPUT_BUFFER           g_pfnResetInputBuffer = NULL; // reset the input buffer of the systems COM port

PFN_EXT_TRACE_MESSAGE                g_pfnTraceInformation = NULL; // Set a information message to the trace log. Message is
                                                                   // a null terminated WCHAR string with maximum length of MAX_WCHAR (currently 256)
PFN_EXT_TRACE_MESSAGE                g_pfnTraceWarning = NULL;     // Set a warning message to the trace log. Message is
                                                                   // a null terminated WCHAR string with maximum length of MAX_WCHAR (currently 256)
PFN_EXT_TRACE_MESSAGE                g_pfnTraceError = NULL;       // Set a error message to the trace log. Message is
                                                                   // a null terminated WCHAR string with maximum length of MAX_WCHAR (currently 256)


#ifdef __cplusplus
extern "C" {
#endif

////////////////////////////////////////////////////////////////////////////////
/// <summary>
/// Called when the driver is being initialized. This function is called once at the end of the PlugIn function "InitDriver".
/// InitDriver is called directly after SetCallback.
/// </summary>
/// <param name="pwszErrorMessage">
///   reserved for future use -
///   pointer to buffer for an error message, propagated to
///   IPEmotion when function returns E_FAIL
/// </param>
```

- The Visual Studio 2010 C++ template is available.
- The integration of new devices is therefore more easy.
- The Acquisistion.dll includes the C code to handle the device specific commands to read the data.

For any questions get in touch with the IPEmotion SW development team!

Examples

1. Metrix Multimeter MX 556

   Is reading the display value via RS232