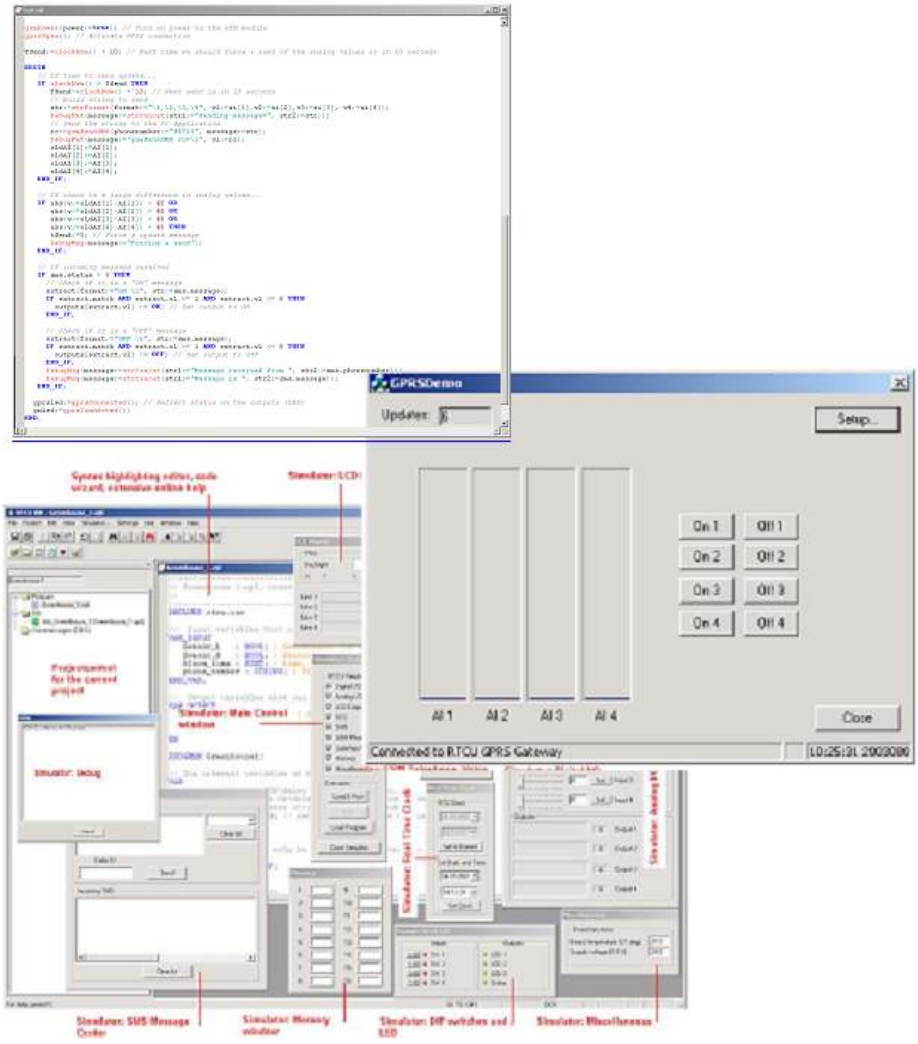
The Logic IO

GPRS Gateway Deployment Package

Version 1.20



Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Table of Content

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Introduction

This document describes the details of a practical GPRS application, based on the RTCU products from Logic IO, the RTCU GPRS Gateway, and an example application.
If you follow this document, you will be able to install and test the Demo application, and see for yourself, that the deployment of GPRS technology easy, when using Logic IO GPRS Technology!

The package contains both a RTCU application and a simple PC application.
The RTCU application runs on one of the GPRS enabled RTCU products, and the PC application runs on a standard Windows 2000/XP/VISTA PC.

The Demo transfers the value of the up to 4 analog inputs of the RTCU units to the PC application every 10 seconds (or if one of the analog inputs changes significantly). The PC application can switch one of 4 digital outputs either on or off on the RTCU unit.
If there are fewer than 4 Analog inputs on the unit, the remaining inputs will stay at 0 (zero).
All the communication is transferred with the help of the GPRS Gateway, which simplifies the implementation dramatically! Messages between the PC application and the RTCU unit, is transferred using VSMS (Virtual SMS) messages. VSMS messages can be routed either by normal SMS messages, GPRS, datacall (CSD) or via a cable connection to the RTCU unit.

What's new in version 1.20:

- Support for encryption and compression using the GPRS Gateway Professional.
- Support for "GPRS Send Packet" and "GPRS Receive Packet" transactions for large packet communication over the GPRS Gateway.
- A new function to initialize GPRS Gateway connection with the advanced options (encryption) is added.
- New functions for registering the callback functions for SMS, PDU and GPRS data packet are added.
- It is now possible to have separate user argument variables for each message type (SMS, PDU and GPRS data packet).

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Contents of package

The package this document is part of, contains the following:

"\GPRS Gateway Deployment Package.pdf"	This document
"\PC App\"	The complete PC application
"\RTCU App\"	The complete RTCU application
"\VSMSGW\"	The DLL containing library functions for the GPRS Gateway communication.
"\Library\"	The .H, .LIB and .DLL file for the VSMSGW library

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

How to install and run the Demo

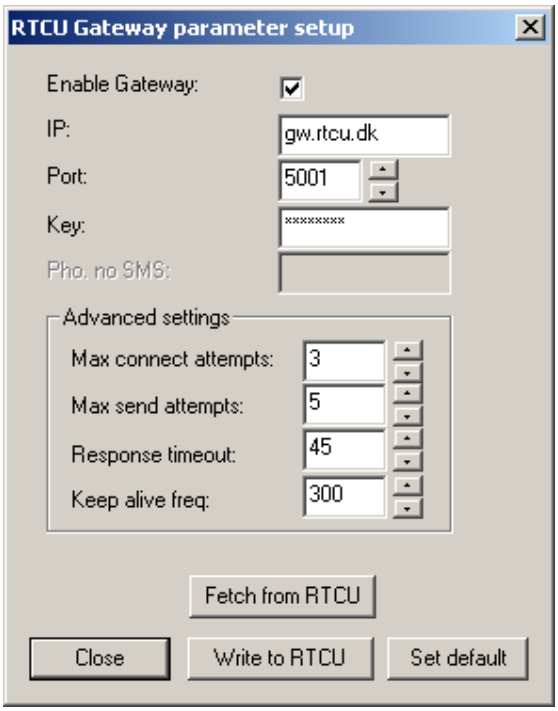
Below, you will find a step-by-step instruction how to install and make the demo application run:

1) Unpack the contents of the “GPRS Gateway Deployment Package.ZIP” file.

2) Start the RTCU-IDE program, and connect to a GPRS enabled RTCU Unit

3) Configure the TCP/IP settings for the RTCU Unit with the correct settings for the actual GPRS enabled SIM card you have installed in the unit.

4) Configure the Gateway settings for the RTCU Unit with the following parameters:



5) Using these parameters, the unit will connect to the test GPRS Gateway that is running at Logic IO. This removes the need for you to install the GPRS Gateway yourself, and just use the GPRS Gateway installed at Logic IO. If you decide to install the GPRS Gateway at your own place, it can be downloaded for free from www.logicio.com. You must then adjust the parameters above to the correct ones, which is set in the GPRS Gateway setup window (primarily the IP, Port and Key)

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

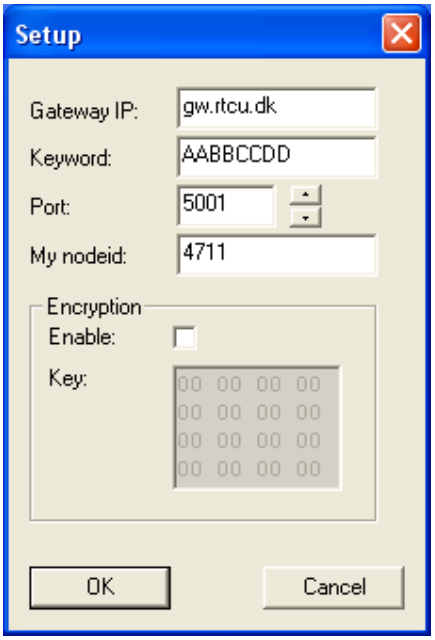
6) Load the “RTCU App” project in the RTCU-IDE

7) Upload the project to the RTCU Unit

8) After some tenths of seconds, it should connect itself to the GPRS Gateway.

9) Then start the PC Application located in the “PC App\Release” directory.

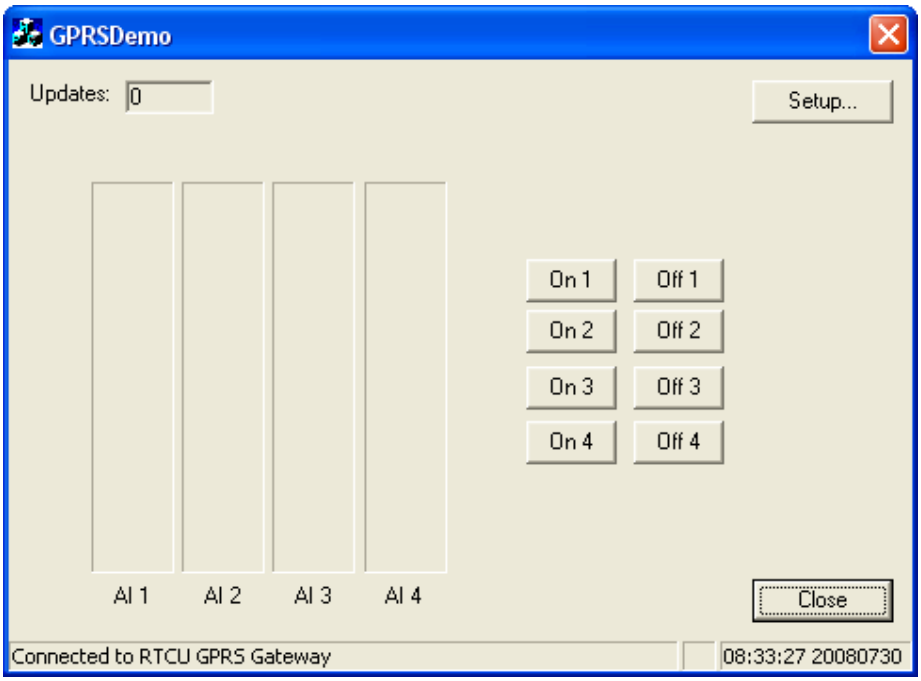
10) The default settings in the “Setup” dialog should be sufficient for the program to connect to the GPRS Gateway at Logic IO, however, should you decide to install and run your own GPRS Gateway, you will have to change the parameters in the Setup dialog. Factory default for the parameters is:



Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

11) After some seconds, the PC application should look something like this:



The field “Updates” shows how many messages that has been sent from the Unit, and the 4 bargraph shows graphically the values of the 4 analog inputs. By pressing one of the 8 buttons (On 1..4 and Off 1..4) it is possible to control the 4 digital outputs on the RTCU Unit.

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Short introduction to the Logic IO GPRS Gateway

The Logic IO GPRS Gateway is a software solution that allows easy communication and access to remote units connected using the revolutionary GPRS technology. The product is especially suited in cases where the GSM operator does not offer a fixed and/or global IP-address. With the Logic IO GPRS GATEWAY product the remote units can easily be accessed without any need for expensive IP-tunneling solutions provided by the specific GSM-operators.

The GPRS Gateway, acts as a kind of “switch-board”, each “client” can connect to the switchboard, and then send messages to other “participants”.

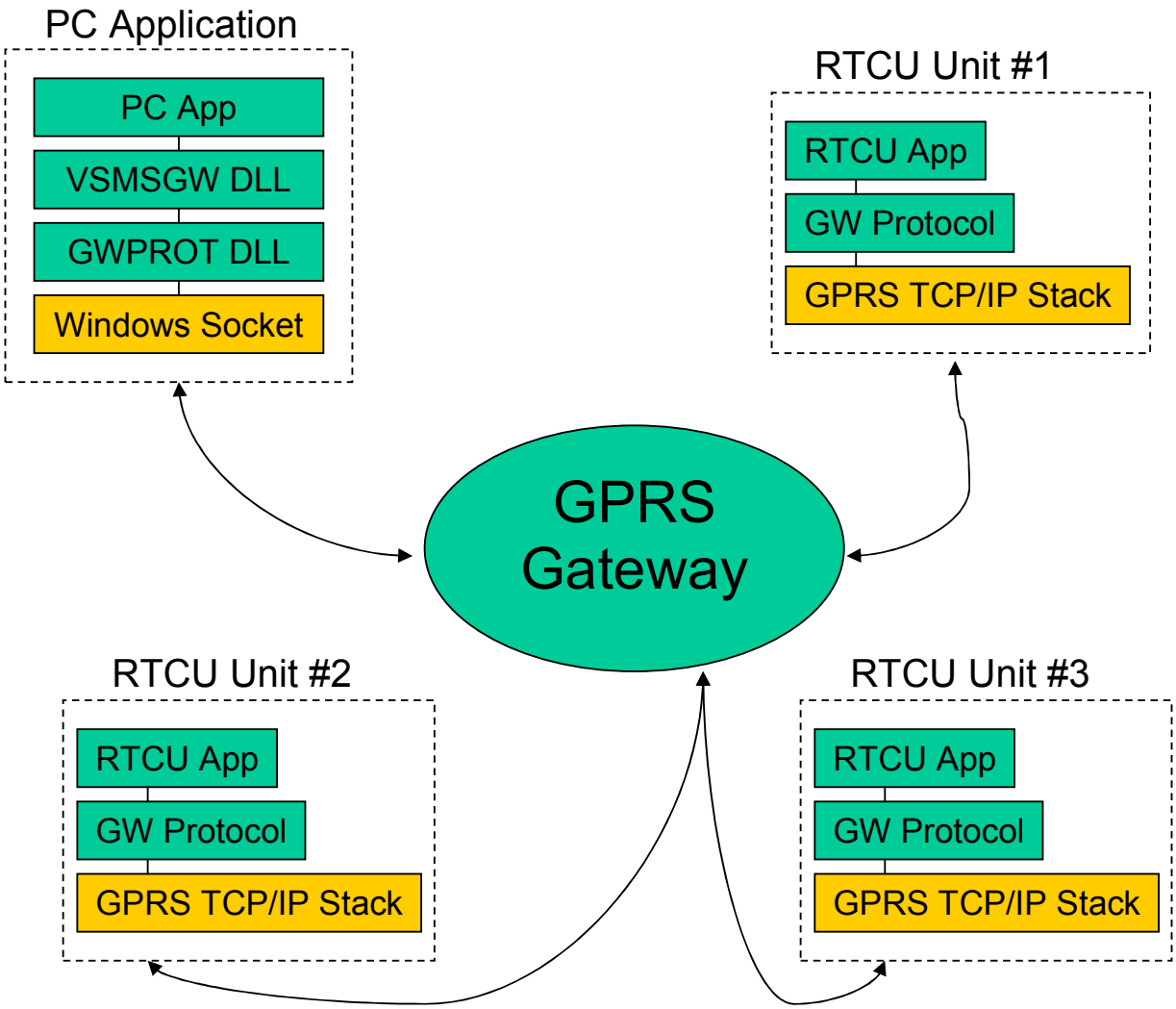
When seen from the GPRS Gateway, all connecting parties, either RTCU units or “PC” applications, are seen as “Clients”. The GPRS Gateway can see no difference between them; they are all “just clients of the switchboard”. It is a Client’s nodeid that makes it unique to the Gateway. A nodeid can be one of two things: if the Client is a RTCU unit, the nodeid of the unit is simply the serial number of the unit. If the Client is a “PC” application, the nodeid is a number the application selects (it can be “0”, in that case the Gateway assigns it a unique nodeid).

The GWPROT DLL forms the low-level protocol towards the GPRS Gateway. It handles all issues regarding connect/disconnect GPRS connection, monitoring of line status, and it is responsible for bringing the connection up again, should it be down etc. The VSMSGW Library, uses the GWPROT DLL for the GPRS Gateway connection, and essentially just encapsulates some of the functions of the GWPROT DLL, and makes sending and receiving VSMS messages more easy and straightforward.

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Schematic presentation of the architecture:



Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Introduction to the Library

The VSMSGW library is a small library that encapsulates some of the transactions that is possible to make against a RTCU unit.
For a description of the complete Gateway protocol, please consult the appropriate document from Logic IO.
The VSMSGW library is written in Microsoft Visual Studio 2005, and was created using the wizard in Visual Studio for creation of a DLL. The VSMSGW project is part of the workspace for the PC Application.

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Functions in the VSMSGW Library

The Library is a collection of the following functions described in this section.

Return Codes:

Symbolic name	Value	Description
GWRC_OK	0	Operation successful
GWRC_ERROR	1	Unspecified error
GWRC_NOT_CON	2	Not connected
GWRC_TIMEOUT	3	A timeout occurred
GWRC_INV_LEN	5	Invalid length
GWRC_IS_OPEN	7	Is already open
GWRC_NOT_OPEN	8	Is not open
GWRC_INV_PARM	9	Invalid parameters
GWRC_DST_UNREACH	10	Destination node is unreachable

cbfuncText()

Synopsis
typedef int(_cdecl *cbfuncText)(int SenderNodeID, char str[161], void *arg);

Description
Definition of callback function for receiving Text SMS messages.

Input :

SenderNodeID	The nodeid of the sender (the serial number of the RTCU that sent the message)
str	The text string sent by the RTCU. Max size is 160 characters + a null-terminator.
arg	A user supplied 32 bit variable that was set when vsmsgwInit() was called

Output:

rc	This is the return code that should be returned to the RTCU Unit, 0 if OK, else error

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

cbfuncPDU()

Synopsis
typedef int(_cdecl *cbfuncPDU)(int SenderNodeID, unsigned char data[140], int length, void *arg);

Description
Definition of callback function for receiving PDU SMS messages.

Input :

SenderNodeID	The nodeid of the sender (the serial number of the RTCU that sent the message)
data	The data sent by the RTCU. Max size is 140 bytes.
length	The length of data sent by the RTCU
arg	A user supplied 32 bit variable that was set when vsmsgwInit() was called

Output:

rc	This is the return code that should be returned to the RTCU Unit, 0 if OK, else error

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

cbfuncPackage()

Synopsis
typedef int(_cdecl * cbfuncPackage)(int SenderNodeID, unsigned char data[480], int length, void *arg);

Description
Definition of callback function for receiving GPRS data packages.

Input :

SenderNodeID	The nodeid of the sender (the serial number of the RTCU that sent the message)
data	The data sent by the RTCU. Max size is 480 bytes.
length	The length of data sent by the RTCU
arg	A user supplied 32 bit variable that was set when vsmsgwSetDataPackageCB() was called

Output:

rc	This is the return code that should be returned to the RTCU Unit, 0 if OK, else error

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

vsmsgwInit()

Synopsis
VSMSGW_API int _cdecl
vsmsgwInit(int MyNodeID, const char *GWIP, int GWPort, const char GWKey[9], cbfuncText SMSText, cbfuncPDU SMSPDU, void *arg);

Description
Initialize the connection to the GPRS Gateway

Input :

MyNodeID	The nodeid for the PC application. If set to 0, it will be assigned by the GPRS Gateway
GWIP	The IP address (or symbolic name) of the GPRS Gateway
GWPort	The portnumber the GPRS Gateway listens on
GWKey	The key value (an 8 character password) used to access the GPRS Gateway. Must be null terminated,
SMSText	A callback function that will be called whenever an Text SMS is received
SMSPDU	A callback function that will be called whenever an PDU SMS is received

Returns 0 if OK, else error (Look at the GWRC_xx codes)

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

vsmsgwInitAdv()

Synopsis
VSMSGW_API int _cdecl
vsmsgwInitAdv(int MyNodeID, const char *GWIP, int GWPort, const char GWKey[9], unsigned char CryptKey[16]);

Description
Initialize the connection to the GPRS Gateway

Input :

MyNodeID	The nodeid for the PC application. If set to 0, it will be assigned by the GPRS Gateway
GWIP	The IP address (or symbolic name) of the GPRS Gateway
GWPort	The portnumber the GPRS Gateway listens on
GWKey	The key value (an 8 character password) used to access the GPRS Gateway. Must be null terminated,
CryptKey	The encryption key used to access the GPRS Gateway.

Returns 0 if OK, else error (Look at the GWRC_xx codes)

vsmsgwSetSMSTextCB()

Synopsis
VSMSGW_API void _cdecl
vsmsgwSetSMSTextCB(cbfuncText SMSText, void *arg);

Description
Set callback function that will be called whenever a Text SMS is received.

Input :

SMSText	A callback function that will be called whenever a Text SMS is received
arg	A user supplied 32 bit variable that will be supplied to the callback function when it gets called

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

vsmsgwSetSMSPDUCB()

Synopsis
VSMSGW_API void _cdecl
vsmsgwSetSMSPDUCB(cbfuncPDU SMSPDU, void *arg);

Description
Set callback function that will be called whenever a PDU SMS is received.

Input :

SMSPDU	A callback function that will be called whenever a PDU SMS is received
arg	A user supplied 32 bit variable that will be supplied to the callback function when it gets called

vsmsgwSetDataPackageCB()

Synopsis
VSMSGW_API void _cdecl
vsmsgwSetDataPackageCB(cbfuncPackage PACKAGE, void *arg);

Description
Set callback function that will be called whenever a data package is received.

Input :

PACKAGE	A callback function that will be called whenever a data package is received
arg	A user supplied 32 bit variable that will be supplied to the callback function when it gets called

vsmsgwIsConnected()

Synopsis
VSMSGW_API bool _cdecl vsmsgwIsConnected(void);

Description
Returns true if connected to the GPRS Gateway

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

vsmsgwGetMyNodeID()

Synopsis
VSMSGW_API int _cdecl **vsmsgwGetMyNodeID**(unsigned long* MyNodeID);

Description
This function will return this nodes nodeid. This function is used especially when a dynamic node-id is requested (by setting MyNodeID to 0 in the vsmsgwInit() function)

vsmsgwSendSMS()

Synopsis
VSMSGW_API int _cdecl vsmsgwSendSMS(unsigned int HisNodeID,
const char str[161], int *rc);

Description
Send a Text SMS message to the specified NodeID, the returncode from the RTCU unit, will be contained in rc

Input :

HisNodeID	The nodenumber (serialnumber) of the receiving RTCU unit
str	The string to send to the receiving RTCU unit. Maximum size is 160 characters + NULL terminator.

Output :

rc	The return code from the receiving RTCU unit, 0 if OK, else error

Returns 0 if OK, else error (Look at GWRC_xx codes)

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

vsmsgwSendPDU()

Synopsis
VSMSGW_API int _cdecl vsmsgwSendPDU(unsigned int HisNodeID,
const unsigned char data[140], int length, int *rc);

Description
Send a PDU SMS message to the specified NodeID, the returncode from the RTCU unit, will be contained in rc

Input :

DstNodeID	The nodenumber (serialnumber) of the receiving RTCU unit
data	The data to send to the receiving RTCU unit Maximum size is 140 bytes.
length	The length of data to send

Output :

rc	The return code from the receiving RTCU unit, 0 if OK, else error

Returns 0 if OK, else error (Look at the GWRC_xx codes)

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

vsmsgwSendDataPackage()

Synopsis
VSMSGW_API int _cdecl vsmsgwSendDataPackage(unsigned int HisNodeID,
const unsigned char data[480], int length, int *rc);

Description
Send a data package to the specified NodeID, the returncode from the RTCU unit, will be contained in rc.

Input :

DstNodeID	The nodenumber (serialnumber) of the receiving RTCU unit
data	The data to send to the receiving RTCU unit Maximum size is 480 bytes.
length	The length of data to send

Output :

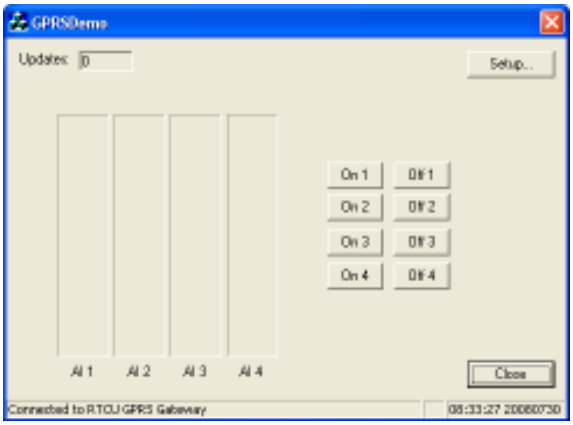
rc	The return code from the receiving RTCU unit, 0 if OK, else error

Returns 0 if OK, else error (Look at the GWRC_xx codes)

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Description of the Demo project



The Demo project consists of a PC application and a RTCU application (see Appendix A). The RTCU application sends periodically the values of its 4 analog inputs to the PC application, and the PC application can send commands to the RTCU Unit that will command one of the four outputs on the RTCU Unit either on or off.

The PC application uses the VSMSGW Library for sending/receiving information from the RTCU Unit. When started, the application tries to connect to the GPRS Gateway, using information set in the “Setup” dialog. In the status bar of the program, you can see if the application is connected or not to the GPRS Gateway.

Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

Appendix A: RTCU Application

Below you will see a listing of the RTCU Application, developed in the RTCU-IDE development environment (available for free at www.logicio.com)

```
//------------------------------------------------------------------------------
// test.vpl, created 2003-05-07 10:32
//
// RTCU Application for the GPRS Deployment Package
// The application sends the values of it's 4 analog inputs every 10 seconds to
// the PC application. The Application can receive commands (ON and OFF) that will
// activate one of it's 4 digital outputs. The PC Application's nodeid is set below
// to 4711, this number must be set in the "Setup" dialog in the PC Application
//
//------------------------------------------------------------------------------
INCLUDE rtcu.inc
INCLUDE tcpip.inc

VAR_INPUT
  ai : ARRAY[1..4] OF INT; | The analog inputs we monitor
END_VAR;

VAR_OUTPUT
   gwled   : bool; | Indicates when there is a connection to the GPRS Gateway
   gprsled : bool; | Indicates when the RTCU Unit is connected to the GPRS network
   outputs : ARRAY[1..4] OF BOOL; | The 4 digital outputs we control
END_VAR;

VAR
  TSend : DINT; // Keeps track of when to send the next update to the PC App
  rc    : INT;  // Return code from various functions
  str   : STRING; // Used for building the message in
  oldAI : ARRAY[1..4] OF INT; // Shadow value of the 4 analog inputs
  sms    : gsmIncomingSMS; // Receives incoming SMS messages
  extract: strGetValues; // Matching of strings (match ON and OFF)
END_VAR;


//------------------------------------------------------------------------------
// Return the absolute value of a number
//------------------------------------------------------------------------------
function abs:int;
var_input
   v:int;
end_var
  if v<0 then
     abs:=-v;
  else
     abs:=v;
   end_if;
end_function;
```


Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk

```
//------------------------------------------------------------------------------
// The main program
//------------------------------------------------------------------------------
PROGRAM test;

gsmPower(power:=true); // Turn on power to the GSM module
gprsOpen(); // Activate GPRS connection

TSend:=clockNow() + 10; // Next time we should force a send of the analog values is in 10 seconds

BEGIN
   // If time to send update...
   IF clockNow() > TSend THEN
      TSend:=clockNow() + 10; // Next send is in 10 seconds
      // Build string to send
      str:=strFormat(format:="\1,\2,\3,\4", v1:=ai[1],v2:=ai[2],v3:=ai[3], v4:=ai[4]);
      DebugFmt(message:=strConcat(str1:="Sending message=", str2:=str));
      // Send the string to the PC Application
      rc:=gsmSendSMS(phonenumber:="@4711", message:=str);
      DebugFmt(message:="gsmSendSMS rc=\1", v1:=rc);
      oldAI[1]:=AI[1];
      oldAI[2]:=AI[2];
      oldAI[3]:=AI[3];
      oldAI[4]:=AI[4];
   END_IF;

   // If there is a large difference in analog values...
   IF abs(v:=oldAI[1]-AI[1]) > 40 OR
      abs(v:=oldAI[2]-AI[2]) > 40 OR
      abs(v:=oldAI[3]-AI[3]) > 40 OR
      abs(v:=oldAI[4]-AI[4]) > 40 THEN
      tSend:=0; // Force a update message
      DebugMsg(message:="Forcing a send");
   END_IF;

   // If incoming message received
   IF sms.status > 0 THEN
     // Check if it is a "ON" message
     extract(format:="ON \1", str:=sms.message);
     IF extract.match AND extract.v1 >= 1 AND extract.v1 <= 8 THEN
       outputs[extract.v1] := ON; // Set output to ON
     END_IF;

     // Check if it is a "OFF" message
     extract(format:="OFF \1", str:=sms.message);
     IF extract.match AND extract.v1 >= 1 AND extract.v1 <= 8 THEN
       outputs[extract.v1] := OFF; // Set output to OFF
     END_IF;
     DebugMsg(message:=strConcat(str1:="Message received from ", str2:=sms.phonenumber));
     DebugMsg(message:=strConcat(str1:="Message is ", str2:=sms.message));
   END_IF;

  gprsled:=gprsConnected(); // Reflect status on the outputs (LED)
  gwled:=gprsConnected();
END;

END_PROGRAM;
```


Logic IO Aps
Holmboes alle 14
8700 Horsens
Denmark

Ph: (+45) 7625 0210
Fax: (+45) 7625 0211
Email: info@logicio.com
www.logicio.com / www.rtcu.dk