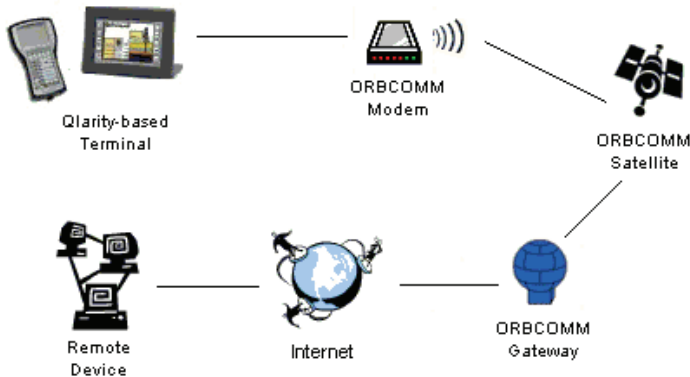


Introduction

ORBCOMM is a global wireless data and messaging system. The ORBCOMM System uses satellites to provide tracking, monitoring and messaging capabilities to and from anywhere in the world. With the OrbComm library for Qlarity-based terminals users are able to communicate with an ORBCOMM-approved modem which supports the “ORBCOMM serial interface specification” Rev E or F.

Using a Qlarity-based terminal to communicate to a remote device with an ORBCOMM modem.



It is helpful to have a basic knowledge of Qlarity Foundry® for this tutorial. A free copy of Qlarity Foundry, Qlarity tutorials, documentation and other resources are available at www.qlarity.com.

The OrbComm Object

Launch Qlarity Foundry and start a new workspace. When the “Select the Type of Workspace to Create” window appears, select the “Standard workspace” option. The OrbComm library will need to be added. To add a library go to the Edit menu, select Edit Libraries and select “Add existing library”. After adding the OrbComm library, select *OrbComm* from the *Object Palette* and create an instance of this object. Rename the new object “OrbComm”. Configure the properties in the *Serial* category to match the modem’s serial port settings. Before running the program in Simulation View select a PC serial port under the Tools->Settings ->Simulation Tab.

Testing Communication

The OrbComm Demo program included with Qlarity Foundry can be used to test communication. This workspace is available under Help->Sample Workspaces ->OrbComm Samples from Qlarity Foundry. When the program starts a blinking label with the text “Waiting for connection with modem...” will be displayed. When this label goes away, communication with the modem is successful. In the lower right corner of the running demo program is the icon of a satellite. This icon will display a different image depending on if the satellite is in view. To add this feature to a workspace create the *OrbSatelliteStatus* object.

Notification when a Connection is Established to the Modem

When a connection is established with the modem an *orbcomm_sc_ready* message will be sent to the globals section of the workspace. A function can be created to handle this message in the format described in section 3.5.3 of the “Qlarity-based Terminal Programmer’s Reference” manual.

Handler Format:

```

func <name> (<intdata> as integer) returns boolean
  handles msg_name
  [...]
endfunc
  
```

Example:

```

`In the Globals section
func OrbCommReady(dmy as integer) returns boolean
  handles orbcomm_sc_ready
  `code here
  return true
endfunc
  
```

Notification when a Connection is Lost

When communication with a modem is lost an *orbcomm_noconnection* message will be sent to the globals section of the workspace. This can only occur if connection has already been established. An example situation that would cause this message would be disconnecting the serial cable.

Example:

```
`In the Globals section
func OrbCommConnectionClosed(dmy as integer) returns boolean
    handles orbcomm_noconnection
    `code here
    return true
endfunc
```

Receiving New Mail

Whenever new mail comes in from the ORBCOMM network an *orbcomm_newmail* message is sent to globals. A function should be created to handle this message.

Example:

```
`In the Globals section
func NewMailEvent(dmy as integer) returns boolean
    handles orbcomm_newmail
    `put your code here to check mail.
    return true
endfunc
```

The OrbComm object has many different functions for getting mail information. These are documented under Help->Show Object Documentation (F1) in Qlarity Foundry. Some useful functions for handling new mail are: *GetInboxMessageContent*, *DeleteInbox*, and *GetInboxRecordCount*.

Example:

```
func NewMailEvent(dmy as integer) returns boolean
    handles orbcomm_newmail
    dim i as integer
    dim maildata[] as byte
    for i = 0 to orbcomm.GetInboxRecordCount()-1
        maildata = orbcomm.GetInBoxMessageContent(i)
        `your code here to process the maildata bye array
```

```
next
orbcomm.DeleteInbox()
return true
endfunc
```

There is an object that can be used to manage received mail called *OrbInbox*. Be aware that calling the function *DeleteInbox()* it will also remove mail from the *OrbInbox* object.

Sending Mail

The function *WriteMessagePacket* can be used to send mail. Many of the parameters of this function are in the format of a string plus integer array. This is what can be used for arrays of strings in Qlarity. To manipulate these type of string arrays, built in functions are available such as: *_AppendString*, *_GetString*, *_DeleteString*, and *_SetString* (see Help for documentation).

Example:

```
`code sample in a Button's Click event
func click()
    dim toList, ccList, bccList as string
    dim toListI[], ccListI[], bccListI[] as integer
    dim subject, outdata as string

    _appendString("test@test.com", toList, toListI)
    subject = "test"
    outdata = "message text"
    OrbComm.WriteMessagePacket(OrbComm.DefaultACKlevel, ->
    OrbComm.DefaultPriority, orbComm.DefaultMsgBodyType, ->
    toList, toListI, ccList, ccListI, bccList, bccListI, subject, outdata)
endfunc
```

The sent mail message will be uploaded to the modem immediately but may take time to travel from the ORBCOMM network to the target mail address.

Determining if a Satellite is in View

The OrbComm object stores a number of different parameter values that are sent or received from the modem. Some parameters such as the Satellite view are updated periodically from the modem. Documentation, including a list of parameters, on the data type *ScParameterName* can be found under Help->Object Documentation from Qlarity Foundry. The function

GetScParameter can be used to retrieve the value of a parameter. To check if a satellite is in view the parameter `_SC_SatelliteIDInView` can be used.

Example:

```
`in a Timerv2 object
func timeout()
    if orbcomm.GetScParameter(_SC_SatelliteIDInView) > 0 then
        `satellite in view
    else
        `satellite not in view
    endif
endfunc
```

Further Reading

Qlarity® FAQ: Can string arrays be used in Qlarity?

<http://www.qsiforums.com/viewtopic.php?t=38>

Links

Press Release

http://www.qsicorp.com/press/2005/sep-10_orbcomm.php

ORBCOMM

<http://www.orbcomm.com/>

Stellar Satellite Communications

<http://www.stellar-sat.com>