

## ZylGPSReceiver 3.74



ZylGPSReceiver is a Delphi & C++Builder component collection that communicates with a GPS receiver (Global Positioning System).

It returns latitude, longitude, altitude, speed, heading and many other useful parameters of the current position and the parameters of the satellites in view. The component is extended to calculate distances and make conversions between different measurement units.

This component works with any NMEA 0183 compliant GPS receiver connected to one of the serial ports. NMEA 0183 (or NMEA for short) is a combined electrical and data specification for communication between marine electronic devices such as echo sounder, sonars, Anemometer (winds speed and direction), gyrocompass, autopilot, GPS receivers and many other types of instruments. It has been defined by, and is controlled by, the US-based National Marine Electronics Association.

You can use it also with USB, IrDA and Bluetooth devices, because these devices have a driver that redirects the input from the USB, IrDA or Bluetooth port to a virtual serial port (you can check it in System/Device Manager/Ports). If your USB device is not provided with such a driver, then use a USB controller whose vendor provides a virtual serial port driver, such as FTDI or use a USB/RS-232 adapter. For Garmin receivers you have to install Spanner software.

The demo version is fully functional in Delphi and C++Builder IDE, but it displays a nag dialog (the licensed version will, of course, not have a nag dialog and will not be limited to the IDE). The package includes demo programs for Delphi and C++Builder and a help file with the description of the component.

### **Supported Operating Systems:** Windows

95/98/Me/NT/2000/XP/Serv2003/Vista/Serv2008/7/8/Serv2012/10

**Available for:** Delphi 10.1 Berlin (Win32 & Win64), Delphi 10 Seattle (Win32 & Win64), Delphi XE8 (Win32 & Win64), Delphi XE7 (Win32 & Win64), Delphi XE6 (Win32 & Win64), Delphi XE5 (Win32 & Win64), Delphi XE4 (Win32 & Win64), Delphi XE3 (Win32 & Win64), Delphi XE2 (Win32 & Win64), Delphi XE, Delphi 2010, Delphi 2009, Delphi 2007, Delphi 2006, Delphi 2005, Delphi 7, Delphi 6, Delphi 5, C++Builder 10.1 (Win32 & Win64), C++Builder 10 (Win32 & Win64), C++Builder XE8 (Win32 & Win64), C++Builder XE7, C++Builder XE6, C++Builder XE5, C++Builder XE4, C++Builder XE3, C++Builder XE2, C++Builder XE, C++Builder 2010, C++Builder 2009, C++Builder 2007, C++Builder 2006, C++Builder 6, C++Builder 5, Turbo Delphi, Turbo C++

### **Remarks:**

- The Delphi 2006 version is fully compatible with Turbo Delphi
- The C++Builder 2006 version is fully compatible with Turbo C++

### **Installation:**

If you have a previous version of the component installed, you must remove it completely before installing this version. To remove a previous installation, proceed as follows:

- Start the IDE, open the packages page by selecting Component - Install Packages
- Select ZylGPSRecPack package in the list and click the Remove button
- Open Tools - Environment Options - Library and remove the library path pointing to ZylGPSReceiver folder
- Close the IDE
- Browse to the folder where your bpl and dcp files are located (default is \$(DELPHI)\Projects\Bpl for Delphi, \$(BCB)\Projects\Bpl for C++ Builder). -Delete all of the files related to ZylGPSReceiver
- Delete or rename the top folder where ZylGPSReceiver is installed
- Start regedit (click Start - Run, type "regedit.exe" and hit Enter). Open the key HKEY\_CURRENT\_USER\Software\Borland\\\Palette and delete all name/value items in the list related to ZylGPSReceiver. (<compiler> is either "Delphi" or "C++Builder", <version> is the IDE version you have installed)

- Unzip the zip file and open the ZylGPSRecPack.dpk file in Delphi (ZylSerialPortPack.bpk or ZylSerialPortPack.cbproj file in C++Builder), compile and install it and add to Tools/Environment Options/Library (in older Delphi/C++Builder menu) or Tools/Options/Delphi Options/Library/Library Path (in newer Delphi menu) or Tools/Options/C++ Options/Paths and Directories/Library Path & Include Path (in newer C++Builder menu, in C++Builder 10 or later, set them also for the classic compiler) the path of the installation (where the ZylGPSReceiver.dcu file is located). The component will be added to the "Zyl Soft" tab of the component palette. After you have the component on your component palette, you can drag and drop it to any form, where you can set its properties by the Object Inspector and you can write event handlers selecting the Events tab of the Object Inspector and double clicking the preferred event. If you still have problems in C++Builder, running an application, which contains the component, then open the project and in C++Builder menu, Project/Options/Packages and uncheck "Build with runtime packages".
- another possible problem with C++Builder: Go to Project options, C++ Linker, and uncheck Link with dynamic RTL.

-It is indicated to use this component with "Stop on Delphi exception" option deactivated. You can do this from Delphi / C++Builder menu, "Tools/Debugger Options/Language Exceptions/Stop on Delphi exceptions" in older versions or Tools/Options/Debugger Options/Embarcadero Debuggers/Language Exceptions/Notify on language exceptions in newer versions, otherwise you will have a break at all the handled exceptions.

64-bit platform:

Delphi/C++Builder 64-bit support is only for runtime, so you have to use it in the following way: Install the 32-bit version of the component as it described above and add to Tools/Options/Delphi Options/Library/Library Path, selected platform: 64-bit Windows the path of the Win64 subfolder of the component.

Before compiling the host application for 64-bit Windows, right click on Target Platforms, Add Platform and add 64-bit Windows (Make the selected platform active). If you compile the application in this way, it will be a native 64-bit application.

### Constants:

RADIUS\_EARTH = 6378.14

### Types:

**TCardinalPoint** = (cpNorth, cpSouth, cpEast, cpWest);

**TDirection** = (dirForward, dirLeft, dirRight);

**TNMEACommands** = set of (GPAAM, GPBWC, GPGGA, GPGLL, GPMSS, GPRMB, GPRMC, GPGSA, GPGSV, GPVTG, GPZDA,

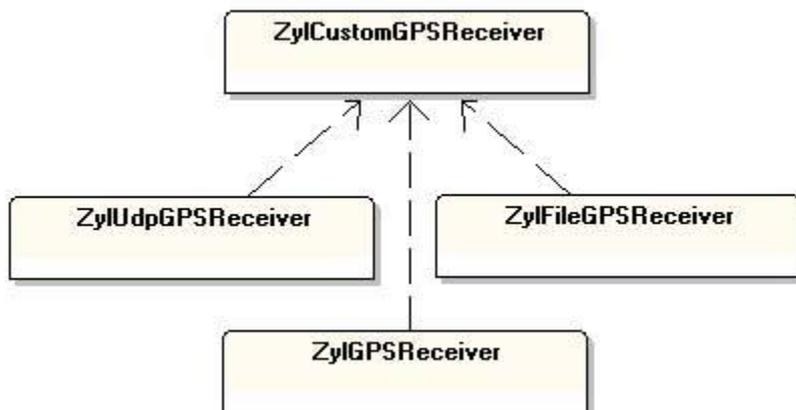
GPWPL, GPRTE, GPXTE, GPHDT, GPHDM, AIIINMEA);

**TCommPort** = (spNone, spCOM1, spCOM2, spCOM3, spCOM4, spCOM5, spCOM6, spCOM7, spCOM8, spCOM9, spCOM10, spCOM11, spCOM12, spCOM13, spCOM14, spCOM15, spCOM16, spCOM17, spCOM18, spCOM19, spCOM20, spCOM21, spCOM22, spCOM23, spCOM24, spCOM25, spCOM26, spCOM27, spCOM28, spCOM29, spCOM30, spCOM31, spCOM32, spCOM33, spCOM34, spCOM35, spCOM36, spCOM37, spCOM38, spCOM39, spCOM40, spCOM41, spCOM42, spCOM43, spCOM44, spCOM45, spCOM46, spCOM47, spCOM48, spCOM49, spCOM50, spCOM51, spCOM52, spCOM53, spCOM54, spCOM55, spCOM56, spCOM57, spCOM58, spCOM59, spCOM60, spCOM61, spCOM62, spCOM63, spCOM64, spCOM65, spCOM66, spCOM67, spCOM68, spCOM69, spCOM70, spCOM71, spCOM72, spCOM73, spCOM74, spCOM75, spCOM76, spCOM77, spCOM78, spCOM79, spCOM80, spCOM81, spCOM82, spCOM83, spCOM84, spCOM85, spCOM86, spCOM87, spCOM88, spCOM89, spCOM90, spCOM91, spCOM92, spCOM93, spCOM94, spCOM95, spCOM96, spCOM97, spCOM98, spCOM99, spCOM100, spCOM101, spCOM102, spCOM103, spCOM104, spCOM105, spCOM106, spCOM107, spCOM108, spCOM109, spCOM110, spCOM111, spCOM112, spCOM113, spCOM114, spCOM115, spCOM116, spCOM117, spCOM118, spCOM119, spCOM120, spCOM121, spCOM122, spCOM123, spCOM124, spCOM125, spCOM126, spCOM127, spCOM128, spCOM129, spCOM130, spCOM131, spCOM132, spCOM133, spCOM134, spCOM135, spCOM136, spCOM137, spCOM138, spCOM139, spCOM140, spCOM141, spCOM142, spCOM143, spCOM144, spCOM145, spCOM146, spCOM147, spCOM148, spCOM149, spCOM150, spCOM151, spCOM152, spCOM153, spCOM154, spCOM155, spCOM156, spCOM157, spCOM158, spCOM159, spCOM160, spCOM161, spCOM162, spCOM163, spCOM164, spCOM165, spCOM166, spCOM167, spCOM168, spCOM169, spCOM170, spCOM171, spCOM172, spCOM173, spCOM174, spCOM175, spCOM176, spCOM177, spCOM178, spCOM179, spCOM180, spCOM181, spCOM182, spCOM183, spCOM184, spCOM185, spCOM186, spCOM187, spCOM188, spCOM189, spCOM190, spCOM191, spCOM192, spCOM193, spCOM194, spCOM195, spCOM196, spCOM197, spCOM198, spCOM199, spCOM200, spCOM201, spCOM202, spCOM203, spCOM204, spCOM205, spCOM206, spCOM207, spCOM208, spCOM209, spCOM210, spCOM211, spCOM212, spCOM213, spCOM214, spCOM215, spCOM216, spCOM217, spCOM218, spCOM219, spCOM220, spCOM221, spCOM222, spCOM223, spCOM224, spCOM225, spCOM226, spCOM227, spCOM228, spCOM229, spCOM230, spCOM231, spCOM232, spCOM233, spCOM234, spCOM235, spCOM236, spCOM237, spCOM238, spCOM239, spCOM240, spCOM241, spCOM242, spCOM243, spCOM244, spCOM245, spCOM246, spCOM247, spCOM248, spCOM249, spCOM250, spCOM251, spCOM252, spCOM253, spCOM254, spCOM255);

**TCommPortSet** = set of TCommPort;

**TBaudRate** = (br000075, br000110, br000134, br000150, br000300, br000600, br001200, br001800,

br002400, br004800, br007200, br009600, br014400, br019200, br038400, br057600, br115200, br128000, br230400, br256000, br460800, br921600, brCustom);  
**TStopBits** = (sb1Bit, sb1\_5Bits, sb2Bits);  
**TDataWidth** = (dw5Bits, dw6Bits, dw7Bits, dw8Bits);  
**TParityBits** = (pbNone, pbOdd, pbEven, pbMark, pbSpace);  
**THwFlowControl** = (hfNONE, hfDTRDTS, hfRTSCTS);  
**TSwFlowControl** = (sfNONE, sfXONXOFF);  
**TArrivalEvent** = procedure(Sender: TObject; const WayPoint: TWayPoint; const CircleRadius: Extended) of object;  
**TConnectEvent** = procedure(Sender: TObject; Port: TCommPort) of object;  
**TSendReceiveEvent** = procedure(Sender: TObject; Buffer: AnsiString) of object;  
**TParamChangeEvent** = procedure(Sender: TObject; Value: Extended) of object;  
**TPositionChangeEvent** = procedure(Sender: TObject; Degree, Minute: Integer; Second: Extended; Direction: TCardinalPoint) of object;  
**TShortPositionChangeEvent** = procedure(Sender: TObject; Latitude, Longitude: Extended) of object;  
**TShortPosition3DChangeEvent** = procedure(Sender: TObject; Latitude, Longitude, Altitude: Extended) of object;  
**TSatelliteReceiveEvent** = procedure(Sender: TObject; Satellite: TSatellite) of object;  
**TDetectEvent** = procedure(Sender: TObject; Port: TCommPort; BaudRate: TBaudRate; var Cancel: Boolean) of object;  
**TCommandEvent** = procedure(Sender: TObject; Command: AnsiString) of object;  
**TWayPointReceiveEvent** = procedure(Sender: TObject; const WayPoint: TWayPoint) of object;  
**EZylGPSReceiverException** = class(Exception); //custom exception class  
**TSatelliteTypes** = set of (AllSatellites, GpsSatellites, GlonassSatellites, BeidouSatellites)



[Buy Now!](#)

Copyright by Zyl Soft 2003 - 2016

<http://www.zylsoft.com>

[info@zylsoft.com](mailto:info@zylsoft.com)

