

---

## 3DM-G Adapter

### **Overview**

This document describes a 3DM-G adapter written in C that is open-source and cross-platform capable. The adapter supports the functions described in the document “3DM-G Data Communication Protocol, rev 2.10”.

### **Platforms**

The driver can be supplied in layers, with the lowest layer being platform specific. The operating system platforms to be supported are:

Microsoft Windows (98, NT, 2000, XP)

Linux (RedHat 7.x)

Sun Solaris 8 (SPARC)

### **Driver Layers**

#### **Layer 0 – Serial Port Interface**

A set of functions written in C that control communications with the serial port. Each platform will require slightly different source code for this layer.

#### **Layer 1 – 3DM-G Device Functionality**

This comprises a set of functions that abstracts the capability of the 3DMG-device. A single set of functions, which can be used across all platforms.

### **Test Programs – Layer 3**

#### **m3dmg - Console Program Basic Function Test**

A C program run from the console is provided to test all basic functions. This can be used by embedded for testing in environments that do not support a graphics user interface (gui). Compilation instructions for Win32 and Linux platforms are provided.

#### **m3dmgloop – Repetitive data access to monitor throughput.**

A simple console application is provided to retrieve the orientation matrix from the sensor multiple times. The program lists total time and average time for data access.

#### **m3dmgbaud – Utility to change baud rate from the console.**

This program allows the user to change the sensor from an old baud rate to a different baud rate. Supports rates of 19200, 38400 and 115200.

### **3DM-G Adapter**

#### Testing

The d3.c program was written to exercise all sensor functions\* contained in d3mgAdpater.c. In addition, tests were performed against non-existent serial ports and ports without a sensor attached.

All functions executed correctly in all modes with the following exceptions:

- a. The getEEPROM and setEEPROM functions do not work on the Solaris (Unix) platform. These particular functions require that multiple bytes be sent to the sensor device. It is possible that there is hardware flow control problem that is affecting how the 2<sup>nd</sup> through last command bytes are received by the sensor.
- b. Communications at 115200 do not work on the Linux/Dell platform. Both 19200 and 38400 work on this platform, but openPort fails at 115200. Some further testing is required to determine whether this is a timing issues or a fault with the hardware port.

At 38400 baud, the adapter is able to retrieve an orientation matrix in 0.02 seconds. This provides an average throughput rate of 50 frames per second. This speed was consistent on both the Linux and Win32 platform.

The software was tested in the following environments:

- a. Windows 2000 on a Dell Inspiron 7000 Laptop, using COM1 serial port. 350MHz P-3 processor, 128 MB RAM.
- b. Redhat Linux 7.3 (Kernel 2.4.18-3 on an i686) on a Dell 2300 desktop
- c. Solaris 8.0 on a Sun SPARC Ultra -5 workstation.

Each of the modules was tested for normal operation on the appropriate platform. This included exercising each of the supported functions to observe correct operation

The functions for continuous mode (on & disable) have not been implemented. These functions are very simple to add, but will not be useful until a mechanism is provided to store the continuous data and make it available through the interface. This will require either multiple threads or a call back function.

---

## 3DM-G Adapter

### Source Code Files

The software for this adapter is provided in two layers for ease of portability. Only the software in layer 0 need be modified for each platform. The bulk of the code, found in Layer 1, is written in to compile without change across platforms. This code has been checked for ANSI C compliance using the `gcc` compiler with the `-pedantic` flag.

Layer 2:           Application layer. Some test code is provided for this layer.

Layer 1:           Sensor Functions (cross platform)

Layer 0:           Serial I/O Functions (platform specific)

`m3dmgAdpater.c` :     a set of functions in 'C', each of which corresponds to the equivalent D3M-G sensor function.

`m3dmgUtils.c` :       general utility functions for performing type conversion and retrieval of error codes.

`m3dmgSerialWin32.c`:   serial I/O functions in 'C' that are platform specific. There is a version for Win32 and another for Linux. The Linux code is POSIX compliant but there have been some issues with full functionality under Solaris. Please refer to Testing section earlier in this document.

`m3dmg.c`                A console (non-gui) program which exercises all of the functions in `m3dmgAdapter.c`.

`m3dmgloop.c`:          A program with retrieves the orientation matrix multiple times, in a loop.

`m3dmgbaud.c`           A program which allows the user to quickly change baud rates. The sensor must still be powered down, then up for the rate change to take effect.

---

**3DM-G Adapter****Source Code Functions**

The table below lists all functions contained in the sensor adapter software found in the file m3dmgAdapter.c. Please refer to the source code for documentation of function arguments and return values. Notice that most of the function names begin with m3dmg\_. This was done to avoid name space conflicts when this code is compiled and linked with application code developed by sensor users.

Function name	Purpose
m3dmg_mapDevice	Map a device number to a serial port. This allows the calling software to refer to devices rather than attached ports. This anticipates the use of RS-485 where multiple devices may share a single port.
m3dmg_getGyroScale	Retrieve the value of the gyro gain scale from the sensor device EEPROM. This factor is used in the calculation of angular rate X,Y,Z.
m3dmg_sendCommand	Send a command to the sensor and wait for the appropriate response, if any.
m3dmg_closeDevice	Close a device and the associated port.
m3dmg_openPort	Open a port. In this version ports refer to serial ports, but future revisions that use different port types can be accommodated here.

  

m3dmg_getRawSensorOutput	Retrieve the raw sensor data for the accelerometer, magnetometer and the gyroscope.
m3dmg_getVectors	Retrieve the sensor data for acceleration, magnetic field, and angular rate of change. The values have been scaled and corrected for mechanical misalignment. Options are provided to retrieve instantaneous or gyro-stabilized results
m3dmg_getQuaternions	Retrieve the four component quaternion data from the sensor. Options are provided to retrieve instantaneous or gyro-stabilized results
m3dmg_captureGyroBias	Instructs the sensor to manually zero the gyro bias.
m3dmg_getTemperature	Retrieve the internal temperature of the sensor device in degrees Celsius.
m3dmg_getEEPROMValue	Retrieve the contents of an EEPROM location by address.
m3dmg_setEEPROMValue	Set the contents of an EEPROM location by address.
m3dmg_getOrientMatrix	Retrieve the coordinate transformation matrix from the sensor. Options are provided to retrieve instantaneous or gyro-stabilized results.
m3dmg_getGyroStabQuatVectors	Retrieve the quaternion, acceleration, magnetic field, and angular rate of change data from the sensor. All results are gyro-stabilized.
m3dmg_getEulerAngles	Retrieve the Euler Angles (pitch, roll, yaw) from the sensor.
m3dmg_getFirmwareVersion	Retrieve the firmware version number of the sensor device.
m3dmg_getSerialNumber	Retrieve the serial number of the sensor device.
m3dmg_getTimerSeconds	Retrieve the number of timer ticks converted to seconds since the last timer rollover.

---

### 3DM-G Adapter

#### ***Coding Conventions and Notes***

The code in this adapter has been written to conform to ANSI C standards. Furthermore, whenever possible clarity is chosen over compactness. A few things worth mentioning:

- a. Functions return an error code. Data is returned through the calling arguments, which the user supplies by way of a pointer. When using the adapter, special care must be taken to allocate (and later, free) sufficient memory for the passed arguments. Examples of this are contained in `d3.c`, which exercises all the adapter functions.
- b. Magic numbers are defined using pre-compile directives. You will find these in the header files (`.h`). In some instances, a magic number is defined within the function. The exceptions to this are the values for baud rate, char size, parity and stop bits. It seems clearer just to use literals here.
- c. There are a few `printf` calls left in the code, disabled by a `DEBUG` flag. Hopefully, these will be of help if the developer runs into problems.
- d. The serial I/O functions for Win32 and Linux are contained in completely separate files. The code is different enough that it is clearer to separate the code rather than confuse things with a large number of `#ifdef` statements.
- e. The serial functions for Linux use the `termios` functions for `tcsetattr`, `tcflush`. The ports are opened in non-canonical mode. The values for `VMIN` and `VTIME` are not used as timeouts; these do not seem to be reliable across UNIX platforms. Instead, a `select()` call is made to wait for port response and then wait for the appropriate number of bytes to arrive, one-by-one.
- f. When creating additional error conditions in the `d3mgErrors.h` file, please be sure to adjust the value for `LAST_ERROR`. This is used to return error text.
- g. On UNIX platforms, please ensure that you have the read-write (`rw`) permissions set for accessing the serial port (`/dev/ttyS0` for port 1 on Linux). You will need root access to modify the port permissions.

#### ***Error Messages***

The source header file `d3mgErrors.h` contains definitions of the errors that are detected by either the serial I/O functions or the sensor functions. English text messages for these errors are provided in the file `d3mgUtils.h`.

#### ***Compile Instructions***

Two additional files have been provided to assist in compiling the source on Win32 and Linux platforms.

`compileWin.bat` – A batch file which can be used to compile and link the source.

`compileLinux` – a shell script file to compile under Linux.

Under Linux, you must supply a flag to indicate the type of operating system. This code is really intended for use across UNIX platforms and is POSIX compliant. This tells the compiler which source to include in the file `d3mgSerialPosix.c`. The only difference in these files is the names used to refer to the serial ports on various platforms.

On Win32 or Linux, you can add the flag `-DDEBUG`. This will tell the compiler to include some of the `printf` statements that could help in debugging on your specific platform.

### **3DM-G Adapter**

#### ***Limitations***

The software described in this document is intended for a single sensor device connected to a single serial port. Design consideration has been given multiple devices across multiple ports for future expansion.

The communications adapter at layer 0 is developed for serial port use. Some small consideration has been given to implementation of a network adapter that allows for communication with a sensor device across a TCP/IP protocol network. The communications functions have been isolated and it should be possible to change the type of port (socket vs. serial) with only slight change to code in d3mgAdpater.c.