IMR File

Inertial Explorer File Formats
- **IMR File**
- DMR File
- HMR File
- MMR File
- PVA File

Waypoint converts all custom IMU raw binary formats into a generic format (IMR), which is read from Inertial Explorer following the decoding process in *IMU Data Converter*. See Raw IMU Data Converter for more details.

Because it contains vital information for reading and decoding the data, the first 512 bytes of the generic IMU data format is a header which must be filled in, read and interpreted. In a C/C++ structure definition, the generic format header has the following fields:

IMR Header Struct Definition

Word	Size (bytes)	Type	Description
szHeader	8	char[8]	"$IMURAW\0" – NULL terminated ASCII string
bIsIntelOrMotorola	1	int8_t	0 = Intel (Little Endian), default 1 = Motorola (Big Endian)
dVersionNumber	8	double	Inertial Explorer program version number (e.g. 8.80)
bDeltaTheta	4	int32_t	0 = Data to follow will be read as scaled angular rates 1 = (default), data to follow will be read as delta thetas, meaning angular increments (i.e. scale and multiply by *dDataRateHz* to get degrees/second)
bDeltaVelocity	4	int32_t	0 = Data to follow will be read as scaled accelerations 1 = (default), data to follow will be read as delta velocities, meaning velocity increments (i.e. scale and multiply by *dDataRateHz* to get m/s^2)
dDataRateHz	8	double	The data rate of the IMU in Hz. e.g. 0.01 second data rate is 100 Hz
dGyroScaleFactor	8	double	If *bDeltaTheta* == 0, multiply the gyro measurements by this to get degrees/second If *bDeltaTheta* == 1, multiply the gyro measurements by this to get degrees, then multiply by *dDataRateHz* to get degrees/second
dAccelScaleFactor	8	double	If *bDeltaVelocity* == 0, multiply the accel measurements by this to get m/s^2 If *bDeltaVelocity* == 1, multiply the accel measurements by this to get m/s, then multiply by *dDataRateHz* to get m/s^2
iUtcOrGpsTime	4	int32_t	Defines the time tags as GPS or UTC seconds of the week 0 = Unknown, will default to GPS 1 = Time tags are UTC seconds of week 2 = Time tags are GPS seconds of week
iRcvTimeOrCorrTime	4	int32_t	Defines whether the time tags are on the nominal top of the second or are corrected for receiver time bias 0 = Unknown, will default to corrected time 1 = Time tags are top of the second 2 = Time tags are corrected for receiver clock bias
dTimeTagBias	8	double	If you have a known bias between your GPS and IMU time tags enter it here
szImuName	32	char[32]	Name of the IMU being used
reserved1	4	uint8_t[4]	Reserved for future use
szProgramName	32	char[32]	Name of calling program
tCreate	12	time_type	Creation time of file
bLeverArmValid	1	bool	True if lever arms from IMU to primary GNSS antenna are stored in this header
lXoffset	4	int32_t	X value of the lever arm, in millimeters
lYoffset	4	int32_t	Y value of the lever arm, in millimeters
lZoffset	4	int32_t	Z value of the lever arm, in millimeters
Reserved[354]	354	int8_t[354]	Reserved for future use

The single header, which is a total of 512 bytes long, is followed by a structure of the following type for each IMU measurement epoch:

IMR Record Struct Definition

Word	Size	Type	Description
Time	8	double	Time of the current measurement
gx	4	int32_t	Scaled gyro measurement about the IMU X-axis
gy	4	int32_t	Scaled gyro measurement about the IMU Y-axis
gz	4	int32_t	Scaled gyro measurement about the IMU Z-axis
ax	4	int32_t	Scaled accel measurement about the IMU X-axis
ay	4	int32_t	Scaled accel measurement about the IMU Y-axis
az	4	int32_t	Scaled accel measurement about the IMU Z-axis

> The angular increments (or angular rates) are signed integers. The scale factor to obtain a double precision word must be supplied by the *dGyroScaleFactor* variable in the IMR header. Similarly, the accelerations (or velocity increments) are signed integers and must be scaled by the *dAccelScaleFactor* variable in the IMR header.