

MICROBOTICS INC.

FAQ

Instructions: Use Adobe's search feature to make queries to this sheet

MIDGII INS (INERTIAL NAVIGATION SYSTEM) with GPS and Dead Reckoning

Questions:

1. What does the MIDG report for position during GPS outages, or degradation of the GPS signal to 1,2, or 3-satellite solutions?

The MIDG continues to provide position and velocity updates during GPS outages for a period of about 30 seconds. After that, the MIDG reverts to a vertical gyro mode in which only the attitude, rates, and accelerations are provided.

2. Are the angular rate data the raw outputs from the rate gyros, or are they outputs of a full-state filter (or something else)?

They are the outputs of the state filter.

3. Are the accelerations the raw outputs from the accelerometer, or are they outputs of a full-state filter?

The accelerations are raw outputs.

4. Are GPS pseudo ranges available as an output?

Not currently.

5. If a full-state filter is being used, are the rate-gyro and accelerometer biases available as outputs? How about the magnetometer biases?

Currently the biases are not available as outputs.

Note that the sys status word of the 0xC4 message has changed.

The bits are now defined as follows:

bits	description
----	-----
31-16	reserved
15-8	GPS HDOP (divide by 5 for actual value)
7	Nonvolatile configuration valid
6	ECEF to NED transform update (~1 minute intervals)
5	GPS update from GPS receiver
4	reserved
3-0	Mode
	0 = VG (vertical gyro mode) Initialization
	1 = VG fast slew rate
	2 = VG medium slew rate

3 = VG slow slew rate
4 = VG slow, eligible for INS mode
5 = INS mode

Note that you can now save mag biases in nonvolatile memory on the MIDG.
The new display utility supports this operation. However, since the new filter does not use the magnetometer, the relevance of this new functionality is reduced. It would be handy in the case that GPS is lost and the unit must run in vertical gyro mode slaved to the magnetometer for heading.

The relevant links for the most recent MIDG firmware and support software are:

Latest Firmware

http://microboticsinc.com/downloads/ins_v1_x_x.bin

Windows Flash Utility

http://microboticsinc.com/downloads/MIDGFlash_vx_x_x.exe

Message Spec (updated for the latest firmware)

http://microboticsinc.com/downloads/MIDG_Message_Spec.pdf

MIDG Display Utility

http://microboticsinc.com/downloads/midgdisp_vx_x_x.exe

NOTES:

You cannot render the MIDG inoperable by upgrading the Firmware. The MIDG firmware upgrade mechanism is independent of the operational firmware. Using the Windows Flash Utility to upgrade the MIDG operational firmware does not alter/compromise the internal firmware upgrade mechanism.

The system status word in the output stream has changed. Please see the new message specification document for details.

The MIDG message specification now includes magnetometer data in the output stream at 10Hz. Also, the new display utility allows for calibrating the magnetometer for biases after installation. Use the 'Options | MIDG Config' menu item to access this feature. The mag biases are stored in non-volatile memory and, consequently, are persistent across power cycles. A good magnetometer calibration improves performance when the MIDG is operating in Vertical Gyro mode (i.e., GPS is not available).

To verify the current firmware revision in your MIDG

simply connect it to a terminal program (115200,8,n,1) and capture the first text that comes out on power up. It contains lots of useful information, including the current firmware version.

You cannot render the MIDG inoperable by upgrading the Firmware.

The MIDG firmware upgrade mechanism is independent of the operational firmware. Using the Windows Flash Utility to upgrade the MIDG operational firmware does not alter/compromise the internal firmware upgrade mechanism.

The system status word in the output stream has changed. Please see the new message specification document for details.

The MIDG message specification now includes magnetometer data in the output stream at 10Hz. Also, the new display utility allows for calibrating the magnetometer for biases after installation. Use the 'Options | MIDG Config' menu item to access this feature. The mag biases are stored in non-volatile memory and, consequently, are persistent across power cycles. A good magnetometer calibration improves performance when the MIDG is operating in Vertical Gyro mode (i.e., GPS is not available).

Transmit outputs of the Serial Converter boards

should be connected to the MIDG receivers, not to the MIDG transmitters.

SC		MIDG
--		----
T1+	->	Rb
T1-	->	Ra
R1-	<-	Ta
R1+	<-	Tb

Also, the SC board requires +4VDC to +32VDC to operate. I believe the MIDG requires +9VDC to +32VDC to operate. Typically, when we make cables for customers, we provide power from a power supply to the SC board and then from the SC board to the MIDG as follows.

Banana plug +VDC -----> SC Vs+ SC Vs+ -----> MIDG pin 4
Banana plug GND <----- SC GND SC GND <----- MIDG pin 9,5

Input Voltages

1. Since both SC and MIDG using one power supply, what voltage should set to?

I typically use 24V here. Also, I just checked the datasheet for the serial converter. The max input for it is 26V, not 36V as I originally suspected. So, the MIDG/SC should be OK on 12V, 24V, or anywhere in between.

2. There is 'r', reference voltage (1.65V) on SC, do we need another input voltage for this?

You will not need this. This reference voltage is generated by the serial converter to allow communication with TTL devices instead of 422 devices.

3. Do we need connect pin 5 (9 pin serial connector) to Banana plug GND, or leave it alone?

Please tie it to GND at the SC.

MIDG Accuracy

I expect ~1 degree (1 sigma) accuracy in pitch and roll, and about ~2.5 degrees (1 sigma) accuracy in heading based on the tests I did here. However, if FAP-9 was running the old code, you were not getting true CMIGITS heading. The old code saves magnetic heading as calculated using INS pitch and roll along with the three axis magnetometer data. The new code saves true heading as calculated by the INS. I would expect significant difference between these two, especially mounted on a car and driving among buildings, etc.

The system status word in the output stream has changed.

Please see the new message specification document for details. The MIDG message specification now includes magnetometer data in the output stream at 10Hz. Also, the new display utility allows for calibrating the magnetometer for biases after installation. Use the 'Options | MIDG Config' menu item to access this feature. The mag biases are stored in non-volatile memory and, consequently, are persistent across power cycles. A good magnetometer calibration improves performance when the MIDG is operating in Vertical Gyro mode (i.e., GPS is not available).

The midg parser

On the CD that gets shipped with the MIDG. There is a text file that describes how to use the parser.

Curvature formula

In your ecef2lla.m, line 31 calculate N as $N = \text{SemiMaj} / \sqrt{1.0 - \text{elsqr} * \sin(\text{lat}) * \sin(\text{lat})}$; Should this be $N = \text{SemiMaj} * \text{SemiMaj} / \sqrt{1.0 - \text{elsqr} * \sin(\text{lat}) * \sin(\text{lat})}$? which is the MIDG Message Specification page 7. Which one is correct?

N represents the radius of curvature. Both methods of calculating it are correct. I did not realize that I had used two different methods (one on the documentation and another in the Matlab source), but either one should give the same result.

ECEF to Lat/Long

I would like to plot latitude, longitude from ICEFALL, what should I type from Matlab?

The components returned from ecef2lla are Lat,Lon,Alt, in that order, so that lla(1) is latitude. Also note that ecef2lla does not operate on an array of ecef values as does ecef2enu. It will convert a single ecef point into a single lla point.

North East Down

It would help to know why you need to use NED. In the simplest scenario, you can simply reassign the information. You have both position and velocity in ENU. If you want NED, then:

$$N = N$$

$$E = E$$

$$D = -U$$

or using vector notation:

$$\text{NED}(1) = \text{ENU}(2)$$

$$\text{NED}(2) = \text{ENU}(1)$$

$$\text{NED}(3) = -\text{ENU}(3)$$

Questions and answers

1. Do you have sample test data or specifications for expected test results for navigation solution (i.e. CEP with sensor on bench)?

Expected position accuracy is about 5m CEP.

2. Do you have detail explanation of method used to obtain euler angles & navigation solution?

- Is it a true inertial navigation solution aided by GPS?
- If so, how is it initialized?
- If not, what is solution process?

No details are available about how the unit works. However, when in INS mode, it is a true inertial navigation solution aided by GPS.

3. Do you have paper or report that describes the methods employed and their limitations?

No, that is not available. However, if you have a particular scenario that concerns you, perhaps I can address it.

4. The data which I sent you last week, we drove a car in a parking lot. Everything looks fine except the start and end position. The car stop normally, we started to record, and drove car, ... stopped car, stopped recorded.

It is always difficult to get a general idea of performance from a single test. However, the data does not look bad to me. I believe the start and stop are within the 5m CEP that is typical for GPS.

5. My next step is that I need to program serial port myself. I know you have sample codes in SourceCode directory. Do you have a stand alone code to run under Windows? Do you have a batch file come with it? Thanks again for your help.

Along with the source that is provided with the MIDG (use to drive a serial port under windows), that should be all you need to get started.

Decode a packet

Given the packet below:

r = angular rates (p,q,r)

p = 0xFFE4 signed decimal = -28 (1e-2 deg/s)

q = 0x002E signed decimal = 46 (1e-2 deg/s)

r = 0xFFEF signed decimal = -17 (1e-2 deg/s)

a = accelerations (ax,ay,az)

ax = 0x0012 signed decimal = 18 (1e-3 g)

ay = 0x0012 signed decimal = 18 (1e-3 g)

az = 0xFC18 signed decimal = -1000 (1e-3 g)

All of the other parameters are decoded similarly. Look at mMIDG.c for an example of how to use the mHex code to do this work for you. After getting a valid AsciiHex packet that passes checksum using a call to mHexGetPacket, then that packet is parsed by a loop that looks for the various data tags in the message and extracts them using calls to the mHex routines. For example, the angular rates are extracted by the

section:

```
case 'r':
```

```

if(mHexToShort(&p,&h))  mMIDG.pqr[0] = h;
if(*p != '/') break;
++p;
if(mHexToShort(&p,&h))  mMIDG.pqr[1] = h;
if(*p != '/') break;
++p;
if(mHexToShort(&p,&h))  mMIDG.pqr[2] = h;
break;

```

Basically, if the character we have just pulled from the incoming data is 'r', then we begin trying to convert the AsciiHex number that follows into an angular rate. The angular rate is only stored into the MIDG structure if the conversion is successful. After the first and second conversions, we make sure that a subfield designator is present ('/').

Again, we would recommend that you use the provided source code to aid you parsing the stream from the MIDG. The only thing you would need to provide is a replacement for the function 'some_read_function' in mMIDGParse.c that reads data from your serial device into 'buffer'.

Read to a buffer

Am I looking from packet start byte ~ to checksum tag | and two bytes of checksum for a complete packet is received?

Correct, but if you make a read function to replace some_read_function, you don't need to look for the packet framing at all. Just write whatever bytes have been received from the serial port into the queue (as shown in mMIDGParse.c), and the MIDG parser will look for valid packets for you. If it finds valid packets it will parse and store the contents into the global structure mMIDG. At the end of mMIDGParse.c, there is a loop:

```

while(mMIDGParse())
{
}

```

When mMIDGParse is called, it returns non-zero if a packet has been extracted and stored in the global structure mMIDG. So, as part of this loop, you could do anything that needs to be done each time a packet is received. When no more data is available in the queue, this function will return zero, causing the loop to not be executed. Code after this loop would have access to the most recent MIDG data stored in mMIDG.

Heading issues with the BEI/Boeing CMIGITS

Upon power-up, it does not assume a heading; rather it requires the user to provide one. Just as a pilot has to set his directional gyro to runway heading when he taxis onto the runway, the operator would have to provide initial heading to the MIDG. Currently, we rely on the magnetometer for initial heading. I would expect the heading to correct itself after a few seconds of acceleration, but that is no help when you are trying to roll down the runway for takeoff. If we set the heading before takeoff, I would expect the velocity errors on takeoff roll to be greatly diminished.

Another possible solution would be to provide for mag calibration in your aircraft. This is already possible (as of the latest code) with our native protocol, but not practical in the ZNBin protocol. To

correct this, we would have to add a packet that outputs the magnetometer values (10Hz probably). This would allow calibration on the ground as well as the ability to calculate magnetic heading during operation. The latest MIDG Display Utility provides a way to set the Mag Bias values in the MIDG non-volatile memory. Actually, the best solution might be to implement both of these.

1. I used Borland C version 4.0. The compiler warning `if(pkt=mHexGetPacket(Pktzr))` 'Possibly incorrect assignment in function > mMIDGParse()', any command?

No, that is OK. It is just telling you that usually, it expects to see `==` in conditional locations, but we actually want `'=`' here.

2. Anyway, after I called `while(mMIDGParse())`

```
{
  what should I do here, may I print results on screen yet??
}
```

yes, you can print to screen from the mMIDG structure.

What's difference between Position and Position Measurement, Velocity and Velocity Measurement?

I can tell from Table 1, the unit frequency is different. I guess ECEF_X is `mMIDG.P[0]`, and CECF_Vx is `mMIDG.V[0]`. Am I right?

Position and velocity are predictions from the INS based on the inertial data (rates and accels) and previous updates from GPS. That is why you can get them at high rates. The position and velocity measurements are the samples that the GPS receiver is providing to the MIDG internally. These are only available to the MIDG (and thus, to the user) at 1Hz. You are correct P,V are position and velocity; g,v are the measurements. See the message spec for format. I believe V is actually in ENU coordinates and v is in ECEF coordinates. Both are in cm/s.

Reset MIDG

A couple of things. First, a string always uses double quotes. So, your definition line should read:

```
char *ResetMIDG[ ] = {"~r1310655|D1", };
```

Further, unless you have a need for an array of strings, I'd recommend using:

```
char ResetMIDG[] = "~r1310655|D1";
```

...

```
mQueueWriteString(hwMIDG, ResetMIDG, 13);
```

Using 13 instead of 12 sends the terminating null. I think the closing null is optional for the MIDG, but it is historically part of the AsciiHex protocol.

MIDG Firmware not displayed when booting

Well, if I don't see "MIDG Firmware...", either my reset command failed, or...??, After I hit a function key (call reset MIDG), I still see the data stream keep coming in, how do I check whether my reset command success?

Q: In the document for channel assignment, the first serial downlink item has conflicting information. I assume you want the euler angles (yaw,pitch,roll), which are not related to the magnetometer, are designated by tages (Y,P,R), not (X,Y,Z). You want the Euler angles, right?

Next, I wanted to explain something about our measurement of the engine and pinwheel RPM values. Those values will be calculated by measuring the time between rising edges from the sensors. This has a couple of implications when the RPM is low. Lets say that a pinwheel is providing a 10Hz pulse signal (600 RPM). Rising edges will be 0.1 seconds apart. This means that at best we can provide new information at 10Hz (as opposed to the 50Hz specified in the document). As the signal gets faster, it will provide results more quickly so that by 3000 RPM, I will have fresh data at 50Hz.

One side effect of this is that we have to choose a value at which we MUST have fresh data. Below this threshold, the RPM will read zero. The reasons for this: My counters can't count to infinity, and as the time between rising edges goes to infinity, so does the time it takes to get information about the RPM. So, initially we have chosen the dropout point at 1Hz (60 RPM). Below this value, the timer will timeout and the RPM will be calculated as zero. However, you will always get feedback at a minimum of 1Hz. If this lower threshold needs to be higher.

A couple of questions about the accelerometer outputs from the MIDG.

1) Are you using the PWM duty cycle output and counting the pulse length with a microprocessor (or something similar) or are you buffering the analog outputs from the ADXL210 and sampling those? From previous info you sent me, it sounds like you're sampling the analog output; if so, how many bits of sampling does your ADC have?

Analog. It is 12 bit. We sample at 200Hz and digital filter down to 50Hz with a cutoff near 20Hz.

2) In the accelerometer outputs, has 0-g offset and sensitivity calibration been incorporated? Or is that left to the user?

Both have been calibrated. There are still small residual errors as a function of temperature that are not captured in our calibration. Zero offset is assumed a linear function of temperature (it is actually slightly non-linear). Sensitivity is assumed constant.

Magnetic heading

1. I understand that when not moving the MIDG outputs the magnetic heading and when moving (and GPS available) the heading is true. Which bits (or combination) indicate which type of heading is being output (mag or true).

Actually, the heading is referenced to magnetic heading when the MIDG is in VG Mode (see the description of the various modes below). On startup, magnetic heading is used. When the first position fix becomes available from the GPS receiver, the output becomes estimated true heading (mag heading corrected for local mag variation based on a global, low order magnetic model that is good to within a degree in most places). Once in INS mode, the output heading becomes true heading as estimated by the MIDG filter. Occasional maneuvering is required to keep a good estimate of heading while in INS mode, but with most vehicles that is generally not an issue. However, if the MIDG remains motionless (parked on a taxi-way) for several minutes in INS mode, the estimated heading can drift away from truth and will be wrong until an acceleration occurs that can provide enough information for the correct heading to be recovered. We are considering adding a command to the MIDG spec that would allow the

user to set the heading (the way a pilot would set his directional gyro before takeoff). Then, if desired, the INS heading could be slaved to the magnetometer output (or any other reference the user might have) when the MIDG is stationary for several minutes.

2) Am I right in thinking that the heading output is aircraft (or more accurately MIDG) heading and not the track of the GPS? Obviously with a side wind the two could be very different.

The output in INS mode is estimated true heading.

3) Could you tell me a bit more about the meaning of the bits in the system status field (message S), I understand what the GPS DOP is, its just the rest I am not clear about.

I'll assume you are looking at the documentation that matches the latest firmware in your unit: http://microboticsinc.com/downloads/MIDG_Message_Spec.pdf So, the items represented are: Mode, New GPS Data, Transform Update, NV Valid, and GPS DOP.

Mode: This describes the operational mode of the MIDG. It will sequence through the listed modes at startup in an effort to get to INS mode, which is the normal operating mode. The earlier modes are VG (vertical gyro) modes in which the MIDG acts like a vertical gyro that is slaved to magnetic heading. Once GPS becomes available and the MIDG is in "VG Mode, slow, eligible for INS Mode" (100b), then the transition to INS mode will take place. Currently, the transition through the VG modes is based only on time. If the angular rate range is exceeded at any time, the MIDG will drop back to "VG Mode, medium slew rate" and begin its transition back to INS Mode.

New GPS Data: This is a flag that indicates when the GPS receiver has produced a position/velocity/time fix that is usable by the MIDG filter.

Transform Update: This is a flag that indicates an update of the ECEF to locally level frame transformation has been recomputed. It is probably of little use to the user, but has been left in to aid development work.

NV Valid: Currently, the only non-volatile information that can be stored in the MIDG are the magnetic bias corrections. You can use the MIDG display utility (http://microboticsinc.com/downloads/midgdisp_v1_7_0.exe) to set these parameters. (See the menu 'Options, MIDG Config').

GPS DOP: For the MIDG, this is GPS HDOP scaled by 5.

Euler angles:

I am confused about the use of the magnetometers. I want the Euler angles from a separate instrument not an integration of the rate gyro signal. Please give me the details on what the X,Y,Z means on the IMU message and what Y, P, R are on the INS message is.

The magnetometer (X,Y,Z on IMU port) simply measures the 3D magnetic field to which the INS is subjected. Ideally, this would be earth-field alone, and the measured vector would represent magnetic north (of course the vector points into the ground, about 60 degrees down locally). Anyway, the magnetometer is a very error-prone and hard-to-calibrate instrument. It is used as a guess for magnetic heading in the INS.

Yaw, pitch, and roll (Y,P,R on INS port) are the estimated Euler angles. These are estimated given a time history of the angular rates and measured accelerations. Also, the magnetometer is used as a heading reference. Internally, two locally level reference frames are maintained: one slaved to magnetic north, and the other to true north. These frames are separated by the magnetic variation as calculated by a low order magnetic model of the earth.

Concerning pulses

The dropout point at 1Hz (60 RPM) is good. Please verify with me that you will be giving the data at 50Hz regardless. I.e. if you don't have "fresh data" you, will be sending the last known value?

That is correct. The output rate will be 50Hz.

Ashtech GPS (current midg2 uses Ublox GPS)

1. What is GPS receiver (inside MIDG) manufacture (is)?

The receiver is based on the SiRF 2 chipset.

2. What is minimum request messages in order to make differential mode work? (Because one of your customer only use 2,3, and 9 to make it success.)

RTCM messages are defined as:

1. DGPS corrections (pseudorange)
2. Delta DGPS corrections
3. Reference station parameters
9. High rate DGPS corrections.

It should be possible to use type 1 or 9 alone to achieve corrections, however if the receivers on different ends are using different sets of ephemeris data for the satellites in view, then message type 2 becomes necessary. I believe type 3 is optional. As far as I can tell, type 1 is obsoleted by type 9. However, I have not worked with DGPS very much so you might need to address your specific questions elsewhere.

Initial Altitude measurements Midg2

We would like to know how the initial attitude estimates are obtained (i.e. magnetic compass for the yaw, and digital pendulum for pitch and roll, or a three axis magnetometer), and if there is some way to turn them off and on through the interfacing protocol.

Initial attitude estimates are obtained using a digital pendulum for pitch and roll and the magnetometer for heading. Until GPS is available, the yaw reported by the MIDG II represents magnetic heading. When GPS becomes available and position is known, then yaw becomes an estimate of true heading using a world magnetic model to correct for magnetic variation.

When the MIDG II enters INS mode, the attitude solution is driven by a kalman filter. Like any INS, if no significant maneuvering occurs, the heading cannot be determined without some independent measurement. The magnetometer can be used to generate measurements in this case. It will begin to provide measurements when the heading uncertainty reaches some threshold (about 12 degrees in the original firmware v2.0.1 that shipped in the MIDG II). The use of magnetic measurements in INS mode can be disabled.

In the absence of maneuvering, the MIDG II will typically resort to the magnetometer for heading information. (Sufficient maneuvering includes any acceleration that results in a change in velocity and/or

position that can be measured by the GPS receiver so that the correlation between the acceleration and the INS displacement is high.)

Regarding the yaw errors, it is suspected that magnetometer biases are the culprit. If you look at the magnetometer data (enable the IMU_MAG message) while rotating the MIDG II, the values should represent the components of a magnetic vector whose origin is at zero. For example, if the MIDG II is yawed 360 degrees, the x and y magnetometer values should make a sine and cosine wave centered about zero. There is a plot dialog in the MIDG II Display utility (Menu -> Options|Plot) that would be useful for detecting magnetometer bias. If you find that magnetometer biases exist, they can be removed using the magnetometer page of the configuration dialog in the display utility.

About the mating cable for the MIDGII

The documentation says Power Return and Shield are connected to Digital Ground. Does that mean I should just splice all three wires together when hooking it to the power supply?

Internally, they are all connected. The digital ground is intended to be wired with the communication lines in the event that the comm lines are long. If everything is close together and you don't have serious concerns about EMI or RFI, you can probably just leave the digital ground and shield pins unconnected.

Are the transmit and receive pins assigned with respect to the MIDG?
The MIDG Tx pin is a data source. The MIDG Rx pin is a data sink.

Loss of GPS

We are interested in designing a self-contained, computerized test system which would be flown in a military aircraft and monitor and log measurements of a given device under test. In it, I would like to incorporate a MIDG-II sensor to allow the monitoring and logging computer to measure the accelerations and angular orientation of the aircraft. Since the test system is self-contained and sealed in a metal housing inside the aircraft, I can not give it access to a GPS antenna and so would forgo the GPS information.

Question: would the MIDG-II still function in every other respect? Will the lack of a GPS signal impair the other sensing capabilities in any way?

Without GPS, the MIDG II will continue to provide attitude. Attitude accuracy will not meet the 0.4 degrees specified for pitch and roll. Our most recent beta software provides 2-3 degree accuracy under moderate maneuvering. If GPS is not available, the position and velocity estimates will not be valid. Attitude, angular rate, acceleration, and compass measurements are provided even without GPS.

Describe the MIDG II sensor bias stability

MIDG II inertial navigation system as a sensor option as you continue to develop your platform. We offer a complete INS solution at a weight of under 2 ounces, a volume of just over 2 cubic inches, and a power consumption of approximately 1 Watt. Complete details can be found at: <http://microboticsinc.com/midg.html>

Typical values:

- 1) Bias Stability of Gyros is 150 deg/hr
- 2) Bias Stability of Accels is 5e-4 g

- 3) Total Bias of Gyros < 2 deg/s (corrected by INS filter)
- 4) Total Bias of Accels < 10mg
- 5) Static Attitude Accuracy (pitch,roll = 0.2 deg, yaw = 0.5 deg)
- 6) Dynamic Attitude Accuracy (pitch,roll = 0.4 deg, yaw = 2 deg)

Are MIDG II data values in big or little endian format?
Big endian

Timestamps question

(1) Can you please verify what I've learned so far?

- Timestamps start counting from zero at power-on, and when GPS data becomes valid they jump to the GPS time-of-week and continue counting from there
- Timestamps have a 13-second offset from UTC time-of-day due to leap-seconds correct. The GPS epoch and UTC epoch offsets must be derived from various other message parameters correct. The MIDG II also provides UTC time via the TIM_UTC message. Rather than roll over each week like GPS time-of-week does, the message timestamps keep on incrementing above 604800 seconds. Actually, the MIDG II timestamp should roll over with GPS time. If GPS signal is lost, the timestamps keep counting up, with the "time is GPS time" bit still set (but with timestamp accuracy >possibly drifting, until GPS data becomes valid, when timestamps will jump to sync up again) It will only jump to catch up if the time error when GPS is acquired is greater than about 2 seconds. My guess is that GPS would have to be unavailable for days to reach the 2 second threshold.

(2) What happens when the 4-byte timestamp rolls over after about 7 weeks? Will my code have to remember to add 0x100000000 milliseconds for each rollover, or does the MIDG-II mod the timestamps by weeks at some point before then?

Rollover should occur at end of week. Please let me know if you have experienced different behavior.

(3) Any other timestamp insights or caveats?

The GPS receiver continues to provide the GPS_PV message (which has GPS time and week) and the TIM_UTC message (which counts to the year 2099) when GPS is lost. I would guess that the GPS on-board clock is slightly better than the processor clock on the MIDG II.

Autopilot and Baro system (barometer)

The specs for the upcoming autopilot lists the 12 analog inputs. What is the maximum sample rate per channel for these inputs?

Actually, those analog inputs will be handled by the FPGA so that the processor does not have to deal with the overhead. As a result, we can acquire the samples at very high speed. As most applications don't require high speed analog, our initial plan is to sample the channels at a few hundred Hz and square window filter them in the FPGA to get them at 50Hz.

Auto-pilot
Describe the Autopilot feature data storage.

We have the SD memory card on our system for data recording, we can support up to 1GB of data on an SD card; transmitting the data to the ground in real time is not a reasonable option. Our autopilot is capable of recording its 12 channels of 12 bit (which equates to 0.025% accuracy) analog at 4kHz. A 1GB SD card gives more than 15 minutes of continuous recording.

Midg2 general questions

Recovery time of the data after temporary loss of power during flight ?

IMU data (rates and accels) will be available immediately after power returns. Attitude will be available, but its accuracy will depend on maneuvering. Full accuracy will be achieved a few seconds after GPS is available (typically within one minute of power on).

Is there a need to calibrate the unit before every flight or just once during installation ?

Correction for mounting misalignment and magnetometer biases are only required at installation. No other calibration is necessary.

What happens if the unit is operated in temp lower than -20 degrees C ?

The unit will operate below -20 degrees C, but the specified accuracy values are not guaranteed.

What are the differences between the mLibC and mLibMCORE libraries, specifically related the CMIGITS functions? The mLibMCORE implementation defines a function mCMIGParse, which seems to have been replaced by packetizer functions defined in file mCMIG.h. I have tried to used the mLibMCORE version, however the mCMIGParse return parses the message queue, but always returns zero, indicating no message (as per the comment in the mCMIGITS.h file). However, the message queue does contain data. This routine is contained in the libmDev.a file. Does this indicate that this function should not be used in production code? Should I use the implementation in mLibC?

Answer: [interface to a BEI CMIGITS] The mLibC and mLibMCORE routines function a little differently, and I'll be happy to help you along with those.

The writing on the box says "MIDG series INS/IMU", so I should use the mMIDG or mMIDG2 modules? These are new in version 1.3. I did not have these in the version 1.1 that we go last year, which only had the MIDGITS library. What is the difference between the MIDG and MIDG2? Which is more suited to the MIDG that we have?

Answer: If the part number is SIS90031, it is a MIDG I, and MIDG should be used. If the part number is SIS90032, it is a MIDG II, and mMIDG2 should be used. As a separate check, the MIDG II serial numbers are ≥ 100 . From the interface point of

view, the primary difference between the MIDG I and MIDG II is the native protocol. MIDGI natively speaks AsciiHex. MIDG II speaks our mBin protocol.

How is velocity calculated? ('V' tag) in mMIDG.h)

Answer: Velocity is provided in ENU.

This question relates to clock functions in the 3.4.1 libC library. When I try to use any of the clock functions that relies on a time of day clock, the processor crashes. Is this a bug, or is there not TOD clock on the MMC2107 board?

Answer: There is no TOD clock on the board. There are two PIT timers in the MCore and routines in mLibMCore to access them. Those might be the best place to start. Set a PIT timer to run at 100Hz and use it to increment your time estimate between GPS time samples.

What is the maximum angular rate for the MIDG?

Answer: The MIDG II can sense angular rates up to 300 degrees per second. Angular rates above this are clipped and result in uncertainty in the attitude estimate. To recover, the MIDG II drops back into VG_MED.

What are the accuracy specifications of the MIDG in Vertical Gyro Mode?

Answer: In VG mode, the level of accuracy depends on the amount of maneuvering. Less maneuvering will result in a better attitude estimate. We have beta firmware (2.1.x series available with the latest 2.0.x firmware on our website) that has much better VG mode performance than the original code. It should typically provide better than 2-3 degree pitch/roll accuracy.

When GPS signals become available again, how much time does it take to switch to INS mode again.

Answer: When GPS is lost the MIDG II will drop into VG_SLOW mode. After approximately 20 seconds, it will transition into VG_SE mode. In VG_SE mode, the transition to INS mode occurs immediately when GPS is reacquired.

What is the attitude and accuracy of the MIDG2?

Answer: The MIDG II provides attitude accuracy of 0.4 degrees in pitch and roll, and 1 degree in heading. Position accuracy can be as good as 2m CEP when WAAS (or EGNOS) is available.

MIDG II messages can be configured to transmit at regular intervals up to 50Hz, or they can be polled when needed by the user. The MIDG II does not provide for internal recording of data samples upon external stimulus (for example, when the camera is activated.) An external computer would be necessary to capture or actuate the camera, associate the image capture with current MIDG II data, and store the data for post-processing.

Connecting the MIDGII to a photographic camera.

When taking the picture from an airplane I want to obtain the parameters of attitude and geographical position. Later, I need download de database with attitude and position information for post-processing the aero-photos. Is possible to make this? Is it possible to take the parameters of attitude and position in an intermittent way and when the operator wants it by means of the photographic camera? What is the error in the angles and in the position?

Answer: Yes, the MIDG2 provides time stamps and vehicle attitude within 2-3 degrees pitch-roll accuracy without GPS and .4 degrees with GPS. The MIDG can be mounted on the camera gimbals, orientated with the lenses. This will provide 3 axis GPS location (ECEF) and the tilt angle of the camera.

In the message spec it states that "The elements of the quaternion must be multiplied by 2^{-30} to get a unit quaternion". This would imply that the magnitude of the quaternion supplied by the MIDG is 2^{30} . However, I'm not seeing this to be the case. Am I misunderstanding the quaternion data supplied by the MIDG?

Answer: Take, for example, this example of a packet of midg_gui1.log:

```
81 a1 0a 27 18 c3 a1 f0 00 07 ff f6 ff cd fe b1 ff a9 fc 5c 00 05 f8 6d
02 50 3f 00 24 00 03 43 74 bc f5 3b 87 00 00 95 b3 55 c0 bd 29
```

The quaternion elements are:

$0x3F002400 = 1,056,973,824$

$0x034374BC = 54,752,444$

$0xF53B8700 = -180,648,192$

$0x0095B355 = 9,810,773$

The magnitude is the square root of the sum of squares. If 2^{30} is not removed before computing the magnitude, the result is very close to 2^{30} . If 2^{30} is removed before the operation, the result is very close to 1.0.

eg: $\text{sqrt}(1,056,973,824^2 + 54,752,444^2 + 180,648,192^2 + 9,810,773^2)$
 $= \text{sqrt}(1.15292151528E18) = 1073741828.97$ (i.e. $\sim 2^{30}$)

or: $1,056,973,824 / 2^{30} = 0.984383583069$

$54,752,444 / 2^{30} = 0.050992187113$

$180,648,192 / 2^{30} = 0.168241739273$

$9,810,773 / 2^{30} = 0.009136994369$

$\text{sqrt}(\text{sum of squares}) \approx 1.00000000463$

I am interested in your MIDG II INS for use in a prototype marine positioning and guidance system I am developing. I have read what information you have on your website and would like some further information if possible. In reference to position accuracy; what is the position accuracy when there is no WAAS or EGNOS available, as in NZ? Also there is no land based differential system available in NZ that I know of, so I assume that I may have to implement my own differential system locally; do you have any information on what level of position accuracy can be achieved with such systems?

Answer: Position accuracy without WAAS/EGNOS is 5-7m CEP. With local RTCM differential corrections, position accuracy can be as good as 2m CEP.

Is the GPS engine upgradeable? I have noticed that a lot of GPS manufacturers are offering dual frequency receivers that offer high accuracy position outputs. Is this something that Microbotics offers or will be offering in the future?

Answer: While the GPS engine in the MIDG II is not upgradable for dual frequency support, we are considering position enhancing technologies for future versions of our inertial navigation system.

What is the level of integration between the IMU and GPS? Would you consider it tightly or loosely coupled? I understand that integrating the GPS and INS can offer better position accuracy than either alone, so without any differential GPS correction, what position accuracy may be available. And with this GPS configuration, would the attitude accuracy be as stated? Does the attitude accuracy stated (tilt) refer to pitch, roll and yaw?

Answer: The MIDG II is a loosely coupled system. The position accuracy of the combined GPS/INS solution in the MIDG II is slightly better (roughly 5-10%) than GPS alone. The big advantage of MIDG II is the ability to get the long-term accuracy of GPS with the higher frequency dynamic response of an IMU. Tilt accuracy (pitch/roll) is approximately 0.4 degrees. Heading accuracy is currently specified as 2 degrees. Some customers have reported heading accuracies of better than 1 degree in airborne camera pointing applications.

What is the drift in the position when GPS signal is lost. Can he rely on the position data for 10 minutes after GPS signal is lost ?

Answer: Position accuracy degrades according to:

$$HPacc = 0.1 * T^2 + 2$$

where:

T is in seconds

HPacc is in meters

The HPacc equation represents a very basic curve fit of typical MIDG II accuracy estimate (1 sigma, conservative) based on collected data from several trials in which GPS was lost and the INS continued to estimate position without position measurements from GPS.

What is the GPS chipset in the MIDG-II?

Answer: The receiver in the MIDG II is based on the ANTARIS chipset from Atmel. It is L1 only. CAPCOM limits apply.

Can the MIDG2 accept input from external GPS sources?

Answer: The MIDG II does not currently accept GPS measurements from an external source.

What could cause Pitch, Roll measurements up to 5 degrees inaccurate?

Answer: It depends on the circumstances. If GPS was not available at the time, then it is quite possible that pitch/roll (tilt) could be off by 5 degrees during maneuvering, especially in the 2.0.x series firmware, or this could happen if GPS were intermittent. Although it is possible to have such an excursion during normal INS operation (with good GPS), it is highly unlikely and should be of very short duration.

Is there any sample data of the MIDG2 output?

Answer: As for sample data, perhaps the best source would be flight data, of which we have two sets. The first is a complete set of data from a flight on board a Cessna 172. http://microboticsinc.com/downloads/c172_flight.exe The second is a reduced set of data (so that it will fit in MS Excel) that shows the performance of the MIDG II vs a CMIGITS II on board a Cessna 182. http://microboticsinc.com/downloads/M2_vs_CM_xls.zip

Is there sample source code for parsing MIDG2 data?

Answer: This code can be found on the MIDG2 support CDROM disk that came with your MIDG2

In the data sheet there is a position accuracy of 2m CEP. I was first wondering what this really means and how that error probability is distributed. Second is this error in the INS section or an error in the GPS calculation.

Answer: CEP is "circular error probable" as defined by most GPS receiving equipment. In our case, the GPS receiver we're using specs 2m CEP as its accuracy with WAAS. That assumes a good GPS constellation configuration so as to have a low dilution of precision (DOP). For the MIDG II, the knowledge of absolute position in Earth coordinates is limited to the accuracy of the GPS solution. As a result, we spec the GPS position accuracy as the MIDG II accuracy.

What operating system is the software for MIDG II written for? Have you ported it to Linux?

Answer: Currently, all of our MIDG II support software (display/configuration program and data parsers) are compiled for Windows. C source code is supplied that is capable of handling the MIDG II messages. We can make Linux binaries available for the data parsers if you require them.

We are getting the 1PPS pulse from the Midg II, but we don't know which time is associated with it. Or where this time message might be so that we could see it.

Answer: You need to enable the TIM_PPS message (ID=27). This message is sent prior to each pulse and indicates the GPS time of the upcoming pulse. For convenience, the latest message spec document is at:
http://microboticsinc.com/midg_files/2Message1b.pdf

Does the MIDG give Latitude data in geocentric or geodetic format.

Answer: Geodetic

The question of accuracy of some of the payload data. For instance the position data comes in 4 byte packets in the payload (if I am understanding this right). Can I assume that this means the number has 32 bits of accuracy?

Answer: ECEF coordinates are presented in centimeters, and there are no 'missing' centimeters in the solution space. LLA is a little more complicated. (Also, the answer that follows applies to firmware v2.0.5 and later, as a related bug was corrected in v2.0.5). The LLA value is presented as an integer with scaling such that the least significant bit represents LE^{-7} degrees. Altitude is presented in cm. Internally, position is maintained and propagated in ECEF

coordinates. The conversion from ECEF (cm) to LLA is done using double precision. So, at best, the LLA format will be able to specify cm level accuracy. Get the latest version of the MIDG II firmware available to be certain you have all the latest fixes. These latest firmware and utilities needed to update the MIDG II can be downloaded from the MIDG II product page at:
<http://microboticsinc.com/midg.html>

Land Vehicle Application of INS MIDG2

How would the MIDG2 unit respond to magnetometer inputs affected by high voltage DC power lines? How might we deal with this?

Answer: The best solution is to mount the MIDG II as far away from the magnetic disturbance as possible. Also, the MIDG II accepts heading measurements from an external source. If the MIDG II must be at the vehicle center of gravity, a separate magnetometer could be mounted far from the magnetic disturbance and used to provide heading measurements. When GPS data is available, it is possible to use the ground track as a heading measurement for most ground vehicles.

Can we use wheel odometers with your product?

Answer: Wheel speed sensing is not currently supported, but we are interested in adding support if enough interest exists.

What protocols, data rates, and measurement update rates can we expect with your product? (e.g. RS-422, 115Kbd, 50Hz data update rates)

Answer: All data is available at up to 50Hz. Individual message rates are configurable. The physical interface is a single RS-422 asynchronous serial port. Baud rate is configurable.

Can we electronically reset the equipment autonomously in the field? What would the consequences of such a restart be?

Answer: There is a reset message, which will command the MIDG II to reset. With good satellite visibility, the MIDG II transitions into full INS mode within approximately 1 minute after reset.

What is the price of your unit as it might be configured for our purpose? What is the expected delivery time from order?

Answer: Special pricing is available for educational research

In message id 15 (NAV_ACC) in byte 14, bit 7, a flag named "Content valid" is contained. Can you please specify to which "content" is it referred to?

Answer: Bit 7 indicates that the content of the NAV_ACC message is valid. It is only valid if the kalman filter is running and the MIDG is in INS mode. For example, the NAV_ACC contents have no meaning if the MIDG is in vertical gyro (VG) mode. Bit 7 is a combination of all of the internal conditions required in order for the NAV_ACC values to be meaningful. It is included in this message so that you don't have to do the tests on the contents of other messages externally (STATUS:Current Mode, etc) to conclude whether the data is meaningful.

1. Could you expand on this new interface, and is it just taking advantage of the QADC?

Answer: The new system uses a separate 12 bit ADC, instead of the 10 bit QADC in the processor.

2. If we rewired one of the connectors would it be possible to enable this function on the current LMCO board?

Answer: The QADC analog resources are not available on the 37 pin connector of the flight computer. All of the pins of the Power/Temp Sensor connector are already assigned. It would be a bad idea to use the spare pins of the debug connectors because RS-232 signals from a PC serial port can go well beyond 5V. But, there is another option. There are 8 analog channels that can be enabled (via a relatively simple hardware mod) on the 65 pin safety switch connector. We can modify the safety switch software to provide an analog message that reports the values of these channels.

3. If we can't do the previous, there are 8 general purpose digital inputs that are available to us. Is there a way by reconfiguring the FPGA to have these be outputs? If we could do this we could use an analog mux external to the board and use the general purpose outputs to select which analog input we want and use the current 10-Bit ADC on the board. This could be a relatively trivial design until your new autopilot becomes available.

Answer: Unfortunately, we can't turn those 4 channels around without significant hardware modification.

4. If we can't turn the general purpose inputs to outputs, could you please expand on these inputs and how we might be able to use an external ADC and shift register to input our data in serial using these inputs.

Answer: #2 seems a better approach if it works for you.

1. Is the 'Yaw' information appearing on Message ID 10 'Magnetic Heading' or 'True Heading' ?

Answer: Yaw is magnetic heading on startup and will remain magnetic heading until a GPS position fix is achieved. Then, heading becomes an estimate of true heading based on an internal world magnetic model, even in vertical gyro mode. When in INS mode, and maneuvering is sufficient, the heading is true heading as calculated by the correlation between INS predicted motion and GPS measured motion (i.e., the magnetometer is ignored).

2. MIDG configured as LLA, will the 'X Axis Position' (Message ID 12) present the Longitude and 'Y Axis Position' present the Latitude ?

Answer: correct.

3. Is the MIDG taking in account the Z velocity (Inertial) in the 'Z Axis Position' (Message ID 12), or pure GPS ?

Answer: Z position/velocity is part of the inertial solution. It is not GPS alone.

4. Would you be able to define the valid data ranges of each and every parameter on the ICD?

Answer: We have not done such a documentation effort up till now. If you have specific data items you're interested in, perhaps I can provide valid data ranges for them.

Q&A

What is the time it takes your unit to provide accurate data after power up?

Answer: one minute maximum

Will Altitudes in excess of 50,000 ft affect your product?

Answer: Our GPS receiver is limited by COCOM restrictions:
Altitude <60000ft and velocity <1000 knots (515m/s).
Either limit may be exceeded but not both.

What is the resolution of your accelerometers?

Answer: The accels are speced at 1.13mg, but with a 12 bit ADC, we only get 4.88mg (before filtering/decimation).

What is the resolution of your MEMS gyros?

Answer: Rate sensors not spec'd. Using technique from Accel datasheet, 0.56 deg/s. ADC gives 0.15 deg/s.

AutoPilot Instructions

The following utilities are described below:

APFlashUtil v1.0.1

APRamExec v1.0.0

The ODP subdirectory contains the OnCE debug proxy that allows the GNU debugger

for mcore to interact with the autopilot using the autopilot's JTAG (OnCE) port. The BDM Interface cable for mcore is required to connect the PC parallel port to the autopilot's OnCE port.

In the System subdirectory are the dynamic link libraries needed for these applications. Place them somewhere in the system PATH, preferably in the system32 subdirectory of the Windows directory.

APFlashUtil

The APFlashUtil provides access to the flash storage on the Microbot AP Autopilot. Operations that can be performed include program, verify, erase, and blank check. Also, the user can query region information that is stored along with flash images.

Two regions of flash may be accessed by the flash utility: Firmware and FPGA. The Firmware region is for user code that is booted and executed by the MCORE processor of the Microbot AP. The FPGA region holds the programming for the IO peripheral processor, which should not be changed by the user.

Flash Utility Operation

In order for the flash utility to operate on the autopilot, it must first capture it. This can be accomplished as follows:

- 1) Connect a PC serial port to either Comm1 or Comm2 of the autopilot.
A simple three wire connection is adequate, connecting transmit to receive, receive to transmit, and ground to ground.
- 2) Execute the flash utility.
- 3) Select the PC COM port to which the target is connected and press

the 'Capture Target' Button.

- 4) Apply power to the target. A progress indicator will appear to indicate the progress of the capture. When the progress indicator disappears, the target is captured and ready for commands.
(see NOTE A below)

At this point, the possible flash operations will become available to the user. A programming operation on the Firmware region would be carried out as follows:

- 1) Select the 'Firmware' flash region.
- 2) Click the 'Erase' button to erase the old flash image. A progress indicator will appear and the flash utility will report success or failure.
- 3) Click the 'Load Image' button and select the S Record file that contains the image you would like to download to the target.
- 4) Click the 'Program' button to start the download. A progress indicator will appear and the flash utility will report success or failure.
- 5) If all desired operations are complete, click the 'Disconnect' button again to release the target (and the serial port that is in use).
- 6) Cycle power on the target to execute the new firmware.

NOTE A:

A 'Skip Capture' checkbox is provided in cases where the target has already been captured, but the FCU Flash Utility has been restarted. In this case, simply select 'Skip Capture' and press the 'Capture Target' button. The flash utility will assume that the capture has taken place and that the target is ready for commands.

Autopilot programming

The APS-Series Platform is supplied as a system that the user can program as desired. The Input/Output system is controlled by a hardware processor, with the needed low-level software support routines supplied on a CD. The development software (C compiler, assembler, and linker) are also supplied on the CD. A full description of the hardware operation can be found at:

<http://www.microboticsinc.com/APfiles/aps%20hdwr%20desc.pdf>

How do I program my Autopilot device?

At present, Microbotics does not supply programming of flight or control

algorithms. Export of this type of software, as well as any training in these types of algorithms, is controlled by the US Department of State under the International Trafficking in Arms Restrictions (ITAR), and requires a Department of State license before any such software is sent or training is undertaken.

What GPS antennas are recommended for the MIDG?

The MIDG II uses L1 C/A only. We usually recommend that users purchase antennas from DigiKey

How many Channels does the MIDG2 receive GPS

The GPS receiver in the MIDG II is a 16 channel receiver.

What are the cold start, warm start times for the MIDGII?

MIDG II startup is less than 1 second. It reaches GPS availability within approximately 45 seconds of power on (GPS cold start), and goes to full INS mode at about 1 minute from power on, assuming good GPS visibility.

What are the operational temperature ranges for the Midg?

Answer: -40C to 85C degrees

How many inputs are used in your Kalman Filter solution?

Our Kalman filter is proprietary, but it has more than 16 inputs

The direction (Z) of the MIDG2 changes over time when setting still with no GPS, Why?

Answer: The Midg will use the magnetic heading at boot time and switches to GPS after a solution is obtained. If GPS is lost (> 10 minutes) and there is no acceleration over time, the inherent component errors (white noise) will cause the drift.

How is the MIDG Calibrated and what are the specifications?

Answer: Each MIDG is calibrated independently in our lab for thermal response, yaw, pitch and roll, accelerations around three orthogonal axes, and three dimensional magnetometer tests. How these test are done are proprietary, but we supply a complete checklist if requested with each midg.

Does Vibration affect the MIDG output?

Answer: Vibration can affect all strap-down Inertial Systems, The appropriate technique depends on the vehicle and the vibration spectrum. If the vibration is high frequency, low magnitude, then double sided foam tape seems to work very well. For higher magnitude input, more elaborate dampening systems might be required. The real key is to prevent mechanically saturating the inertial sensors.

The MIDG II is "Differential Ready GPS" what does that mean and how would we use this feature? Additionally, there is no mention of WAAS in the "MIDG II Operating Modes" description, how (or when) is this feature activated?

The MIDG II supports both satellite based differential corrections (WAAS, EGNOS) and local RTCM corrections. If WAAS satellites are within view, their signal will be used to provide differential corrections. The MIDG II can receive RTCM corrections using the RTCM input message (message ID 30) described in the [MIDG II Message Specification](#).

Our MIDG is beginning to show errors, what can be the cause?

The MIDG has a 1200 hour operational life before re-calibration is required. This is caused by the inherent problems with electronic components, temperature tends to deteriorate their responses over time. We can recalibrate and upgrade your MIDG for a small cost. Please contact our office at 757-865-7728 or email support@microboticsinc.com.

How do we achieve synchronization with the MIDG II output?

The newer MIDG IIs (SIS90031C) have a one-pulse-per-second output signal that can be used in conjunction with the TIM_PPS message to precisely determine time. This mechanism is intended to allow the host computer to synchronize itself to GPS time just as the MIDG II is synchronized to GPS time. Also, the newer MIDG IIs have a build option that provides a pulse input pin. When a rising edge is received on the pin, the MIDG transmits a message that indicates the time that the edge occurred to within 1 millisecond. Even if you have a MIDG II that predates these hardware features, there

is a chance we can upgrade it.

Is there a way to force the message to be time stamped with GPS time no matter the state of GPS, i.e. no satellite lock, partial GPS coverage, etc.?

Answer: The TIM_TM message reports the time of a rising edge received at the pulse input. The reported time is the best estimate of GPS time from the MIDG II. When the MIDG II first starts, it does not know GPS time and cannot know GPS time until a 3D fix is first achieved. At this time, the MIDG II synchronizes its clock to GPS time, begins to calibrate its clock accordingly, and sets the 'Time is GPS Time' flag for the output messages. If GPS is lost, the MIDG II continues to propagate its best estimate of GPS time based on its own calibrated internal clock and the 'Time is GPS Time' flag is never cleared.

How can I get MIDG2 Data through the Auto-Pilot platform with the MIDG display utility?

If you want to see MIDG II data from the autopilot using the MIDG II Display and Configuration Program, then the MIDG II data must be passed through the autopilot serial ports to a PC. The MIDG II is connected to autopilot COM5. The user has external access to autopilot serial ports COM1-COM4. COM1 and COM2 are on the 44 pin APS connector, and COM3 and COM4 are on the 26 pin APX connector. There are two ways to pass data through the autopilot to one of these serial ports:

- 1) write an autopilot program to perform the pass through. See the AP_Examples\ap_tasks program on the CD for an example of how to accomplish this.
- 2) use the autopilot flash utility (AP_Uutilities\APFlashUtil.exe on the CD) to capture the autopilot and then pass the MIDG II through to the PC serial port that captured the autopilot. Then open the MIDG II Display and Configuration Utility on that same port to see the MIDG II data.