

## IMU6420 V1.00Data Logger Release Notes

Feb 22, 2013

### Features

The IMU6420 is a high performance 10 DOF Data Logger with the following features:

- Atmega1284P processor (11.0592MHz) with real time clock
- USB 2.0 interface with CDC COM serial communications
- 10DOF Sensors: ADXL345, L3GD20, HMC5883, BMP180 (or MS5611)
- MicroSD card connector with FAT16/32 file format support – 32G support
- Lithium Polymer battery charging circuit
- Interrupt driven UART1 support with putchar1, getchar1, puts1, etc.
- Support for ANT and SOC ZB1 ZigBee wireless modules
- Stk500V2 bootloader pre-loaded for Arduino bootloader support
- Optional MS5611 high performance barometer pad
- Support for new SOC ZB1 ZigBee wireless module
- AVR Studio 4.19 Data Logger V1.00 Project File Source Code included
- Extensive logging features added to support microSD
  - o start logging at future time
  - o terminate logging after time period exceeded, button press or keyboard halt
  - o start logging if threshold is exceeded on any sensor axis
  - o stores data as printable ASCII time stamped
  - o user define file name or system automatically creates file name
  - o start logging on power up
  - o automatic file name incrementing so new file is created on each power up
  - o auto restart of logging with defined restart time
  - o can combine delayed start with thresholding - thresholding only starts after a delayed start time
  - o add sensor min/max value to end of log file
- ZigBee wireless support built in – compatible with IMULINK
- Smaller (4K) bootloader supported to provide more code space
- Complementary AHRS filter implemented

Complete source code is available for download from the download page at [www.soc-robotics.com](http://www.soc-robotics.com). See the section at the end of this note for a more detailed description of logging modes.

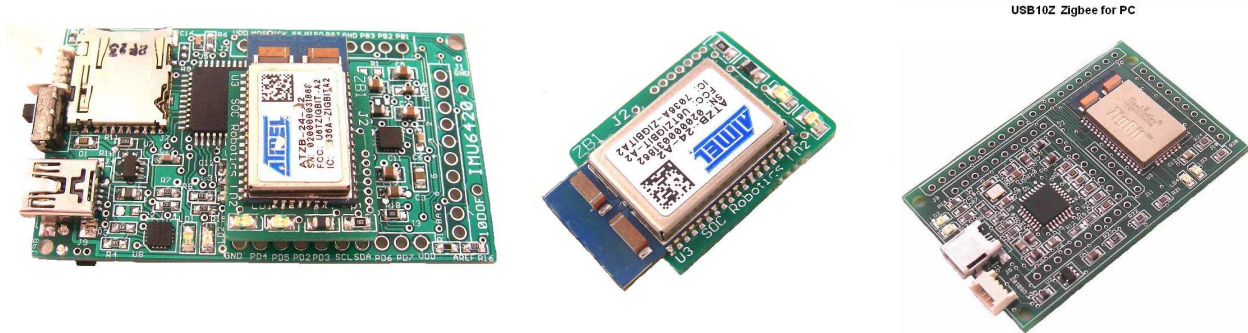
### IMU6420 Wireless Options

The IMU6420 has an ANT compatible connector that accepts ANT wireless modules or SOC ZB1 ZigBee wireless modules. The ANT connector also accepts our new ZB1 ZigBee wireless module communicating at 38,400 baud. Both wireless modules are interchangeable.



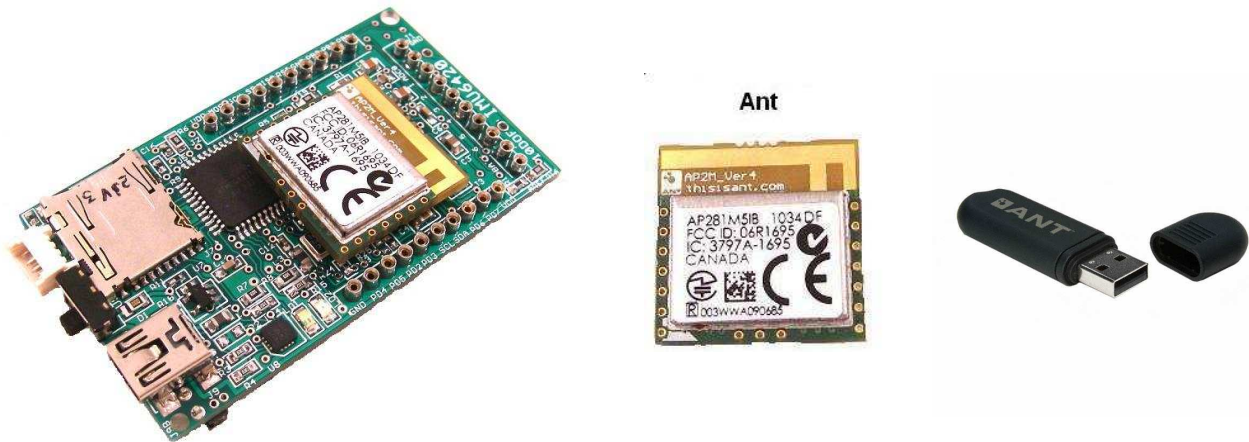
### ZB1 Wireless Module

The ZB1 module is a new SOC Robotics ZigBee compatible adapter with an Atmel ZigBit ATZB-24-2A ZigBee module attached to it. The Molex connector on the bottom ZB1 is compatible with the ANT wireless base connector on the IMU6420. The ZigBit comes preloaded with SerialNet and is ready to join a ZigBee network as a slave device. Software on the IMU6420 configures the ZigBee node for correct network operation. SerialNet supports an extensive AT command set.



### ANT Wireless Modules

ANT modules are available from several distributors and are a standard, off the shelf, component. Compatible ANT modules are the ANTAP281M5IB and ANTC782M5IB. ANT wireless devices support high speed over air operation with small packet data rates reaching 200Hz.



### USB10Z

The USB10Z is a USB ZigBee communications device that attaches to a PC and is available directly from SOC Robotics, Inc. The ZB1 and USB10Z are available now.

### New Version 1.00 Features

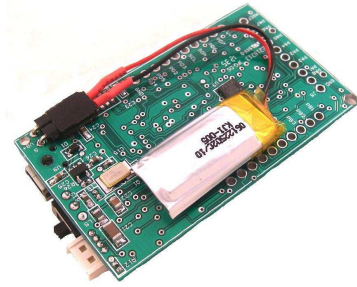
This new release provides integrated support for ZigBee wireless nodes. Control of logging start/stop, file naming, etc is now possible over the wireless link. A new beta release Java communication utility called IMULINK will be released shortly that communicates with the IMU6410 data logger in real time to manage IMU operation.

Once the new software loaded reset the EEPROM contents using the "Sr" command. The bootloader does not reset the EEPROM contents and the access procedure to this memory has been changed.

Support for ANT wireless is still under development and is scheduled for spring 2013 along with a richer set of ZigBee wireless functions.

**Lithium Polymer Battery Charging Circuit**

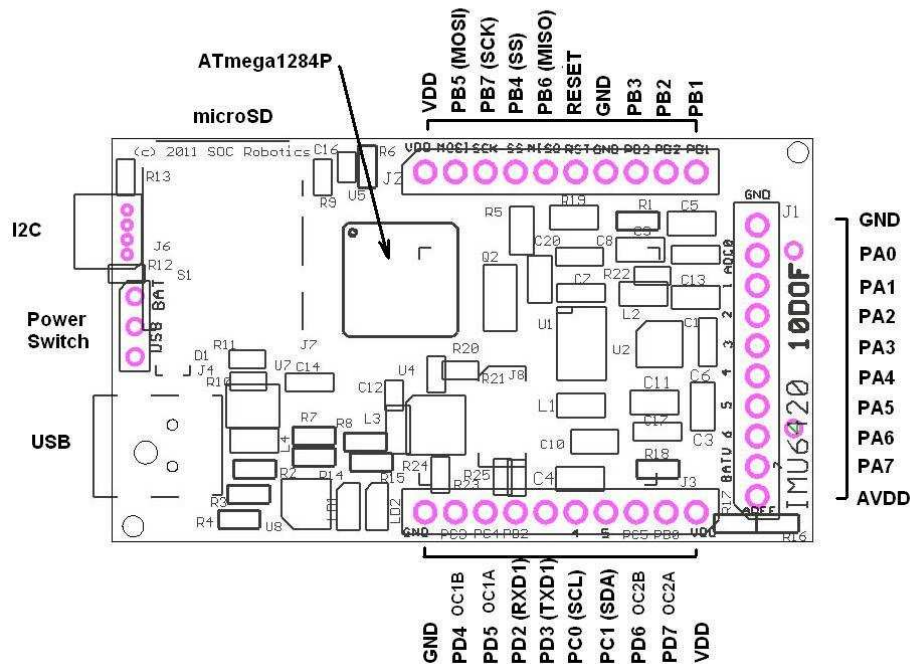
The IMU6420 has a lithium polymer battery charging circuit that charges attached batteries when ever the device is connected to a USB port. The charging accepts a range of LiPo battery capacities.



**Expansion Ports**

The IMU6420 has three expansion ports: J1, J2 and J3 and an I2C port.

**IMU6420 Connector Pin Assignment**



Hardware PWM ports have been brought out to connector J3. These pins can be used to control the motors of a UAV or other use. Port J2 has the ISP programming lines.

**stk500v2 BootLoader**

The IMU6420 has an stk500v2 compatible bootloader loading into high memory at 0x1F000. This bootloader can be used to in-situ reprogram the Atmega1284P using AVR Studio 4.19 or AVRdude without the need for a separate hardware ISP programmer. Note that programming the Atmega1284P does not change the contents of EEPROM – only the Flash is changed. The EEPROM contents must be upgraded manually by using the “Sr” command – this reloads the EEPROM with default contents.

## IMU6420 Data Logger Overview V1.00

The IMU6420 Data Logger manages the acquisition and storage of 10 DOF MEMs sensor data to microSD storage adapter.

The microSD subsystem supports both a FAT16 and FAT32 file system automatically. Files are stored by a user specified unique name or automatically using a default incremental naming convention starting with the three letters IMU followed by a file number - IMU00001.txt. The incremental naming system automatically creates the next name in sequence as below:

```
IMU00001.txt
IMU00002.txt
.
.
.
IMUnnnnn.txt
```

Logging configuration settings are selected using the “l” command at the top level command menu. As logging mode selections are entered they are automatically stored in Flash. The data logger checks if a microSD card is installed. By default, data from all four MEMs sensors is logged to the microSD card.

Several logging trigger conditions activate logging such as delayed start, keyboard start, threshold start or switch start. Delayed start allows logging to start at some time in the future. This allows the IMU6420 to be placed in a specific location, such as an instrument carriage, so the effects of sensor placement is not logged. Keyboard start is for immediate data logging start. Just enter the start logging command or the 9t, 10t commands. Threshold start depends on certain sensor values exceeding defined thresholds as a percentage plus or minus of full range at the current value. Switch start allows the press of a button to start and stop logging. The start of logging is indicated by a fast blinking Green LED.

There are also several logging stop modes. Halt logging is a keyboard command that immediately stops logging and closes the logging file. Timeout stop terminates logging after a specific time delay in seconds. Switch stop uses a switch to stop logging. Note that if switch stop and timeout are both selected then the one occurs first stops logging.

Delayed start and thresholding can be combined to allow time to place IMU6420 in position so that logging doesn't start prematurely. For example, say it takes 30 seconds to place the IMU6420 into a test jig after which movement on any axis exceeding a specific threshold is the trigger to start logging. This allows the IMU6420 to capture significant events defined by the user.

Thresholding start can be set on an individual sensor and sensor axis basis. So, for example, if the x axis acceleration exceeds +1G and the magnetometer axis falls below a specific value logging starts while changes to other sensor values does not start logging.

Automatic power up logging is also supported. This allows power cycling to start a logging event. This mode simplifies field-logging applications. By combining power on logging with delayed start, thresholding, timeout termination and incremental file naming it is possible to prepare an IMU6420 for field logging by simply cycling the power on and off.

Auto restart is a mode that closes files after a specific time period in seconds and then opens a new file to continue logging in combination with the file name increment function. The file system requires that a file be closed to save it. If power is lost and all the data is in a single file

then that file is lost. By opening and closing smaller files data loss due to power failure is significantly reduced. Up to 32,000 files can be created. The logging software also automatically snap shots logged data every 20 seconds. If power fails before the next 20 second snap shot that data is lost.

#### Logging command summary:

```

? - Logging parameters summary
p - Powerup log start mode
a - Arm/disarm switch PB0
t - Terminate mode (b,t,h)
w - Start logging at future time
j - Threshold mode setup
y - Log restart mode setup
m - Enable/Disable printing MinMax values at end of file
u - Store data in uSD or Serial Flash
v - Display logging output
x - Enter sensor range
s - Start data logging
h - Halt data logging
i - Enter Sensor ID1 or ID2 name
e - Erase chip
d - Select sensor(s) to log (a-accelerometer,g-gyro,m-mag,b-barometer)
r - Retrieve data
l - Log rate (msec)
f - Output format (raw,processed,minmax)

```

#### Example of a logging event setup:

Enter the “u” command to select logging to the microSD and select incremental logging - this resets the file number to 1.

Enter the “t” command to set terminate mode to time out and enter a time out period in seconds. If 60 seconds is entered then after 60 seconds logging will stop and the file closed (saved).

Enter the “y” command to enable automatic restart and enter the number of restarts. If 20 restarts are selected and a time out period of 60 seconds is selected this will result in 20 files starting with the name IMU00001.txt and ending in IMU00020.txt to be created.

Enter the “m” command to track sensor min/max values and attach these values to the end of the log file.

Enter the “w” command to set a future logging start time. If this time is set to 45 seconds then you have 45 seconds from power up to place the unit in position before logging to the microSD starts.

Enter the “p” command to enable automatically start logging on power up. Each time the IMU6410 is powered on it will start logging based on the other entered parameters rather than waiting for a keyboard “10t” or “9t” command.

Note that each time the unit is powered up a new file start number is used. So if 20 files are logged during the first power up then the next power up will start the file number at 21. If the u command is used again then the file numbering starts at 1 again.

Automatic restart is also designed to recover from power failures. If 20 restarts are selected and power is lost at file cycle 14 only file 14 is lost. On the next power up file 15 is stored and so on.

Thresholding is also supported as a start condition. By combining delayed start with thresholding it is possible to both place the unit in position and start logging based on a MEMS

sensor event. Thresholding is based on a percentage sensor value change from the current resting state rather than a value entered as an absolute number. This allows the unit to be placed in variety of different configurations with a change in orientation or translation triggering a logging start condition.

### Complimentary and Extended Kalman Filter

The IMU6420 processor is not fast enough to run an Extended Kalman Filter in real time so a simpler Complimentary filter is provided in the source code. This filter has not been extensively tested. The IMU8420 is capable of running an EKF in real time.

### AVR Studio 4.19 Project Files

The IMU6410 software was developed using AVR Studio 4.19. Most of the project source code files are now referred to as IMU64X0\_ as support for both the IMU6410 and IMU6420 is included in the same project. The following changes has been made:

- A new include file called IMU64X0\_Type.h determines which target to compile for by un-commenting either IMU6410 or IMU6420. Each board has slightly different pin/signal assignments that must be accounted for to ensure correct operation.
- The stk500v2 bootloader has not changed. Note, however, that updating the software using the bootloader does not change the contents of EEPROM so a reset of EEPROM is necessary using the top level "Sr" command. This must be done for correct operation.
- The new build now has the optimizer turned on to -Os, which has freed up more space for user code additions.
- ZigBee wireless is now built-in but can be disabled by commenting out the ZIGBEEWIRELESS define.
- A new ZigBee test and pass through mode is available from the top-level menu using the "z" command. With pass through direct communication with the ZigBee node using the AT command set is possible. See the Atmel ZigBit AT Command Set manual for details.
- The data logger is now listening to the USB UART0 input and ZigBee UART1 input simultaneously so commands coming from the hard connection or wireless connection are handled at the same time. The wireless connection has a new set of commands that will be documented with the release of the IMULINK application. The commands are in the top-level menu section.
- A delay was added to provide time for the ZigBee node to wake up - reducing or eliminating this delay may compromise ZigBee operation.
- The logging state machine has been slightly upgraded with a new min/max command - this command tracks the minimum and maximum value of each 9 DOF sensor and appends these readings to the end of the log file.
- The FATFs file system has several new commands reached by the lop level "F" command. Not all commands are listed - see the source code in main.c for more information.
  - o Fi 0 - mount and init file system >fi 0
  - o Fn - rename >fn imu00001.txt speed1.txt
  - o Fx - copy >fx foo.txt foonew.txt
  - o Fo - open file for reading >fo 1 foo.txt
  - o Fd - dump file contents >fd 10 (dump first 10 bytes of file)
  - o Fc - close file >fc
  - o Fl - list all files >fl
  - o Fu - delete file >fu foo.txt
- If a uSD card is installed prior to power up it is automatically initialized and mounted - the "f" command lists all files on the card