IMU8420 V1.00 Data Logger Release Notes

Feb 27, 2013

Features

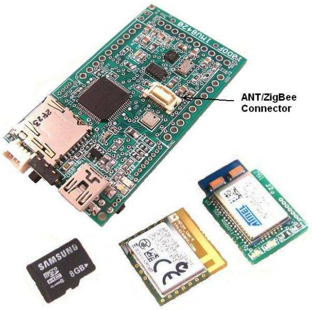
The IMU8420 is a high performance 10 DOF Data Logger with the following features:

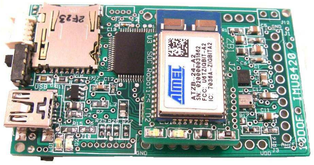
- Fast 32bit AVR32 processor AT32UC3C2512 with hardware FPU, 64MHz, 512K Flash
- USB 2.0 interface with CDC COM serial communication
- 10DOF Sensors: ADXL345, L3GD20, HMC5883, BMP180 (or MS5611)
- Real time Sensor Fusion Extended Kalman Filter (EKF) implemented (300Hz cycle rate) combining accelerometer, gyro and magnetometer to produce roll, pitch and yaw
- On chip DFU boot loader for software field upgrades
- Lithium Polymer charging circuit
- Optional MS5611 high performance barometer pad
- High voltage regulator option to support 12-24VDC input
- microSD card connector with FAT16/FAT32 support
- Support for ANT and SOC ZB1 ZigBee wireless modules
- AVR Studio 6.0 Data Logger V1.00 Project File Source Code included
- Extensive logging features added to support microSD
    - start logging at a future time
    - terminate logging after time period exceeded, button press or keyboard halt
    - start logging if threshold is exceeded on any sensor axis
    - stores data as time stamped printable ASCII
    - user defined file name or system automatically creates file name
    - start logging on power up
    - automatic file name incrementing so new file is created on each power up
    - auto restart of logging with defined restart time
    - can combine delayed start with thresholding - thresholding only starts after a delayed start time
    - append sensor min/max values to end of log file
    - optionally log Extended Kalman Filter output (roll, pitch, yaw)
- ZigBee wireless support built-in – compatible with IMULINK desktop communications application allowing wireless control of logging functions

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

IMU8420 Wireless Options

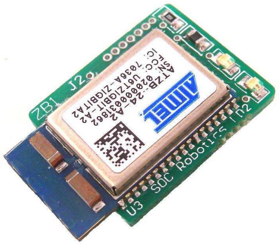
The IMU8420 has an ANT compatible connector that accepts ANT wireless modules or SOC ZB1 ZigBee wireless modules. The ANT connector also accepts our new ZB1 ZigBee wireless module communicating at 38,400 baud. Both wireless modules are interchangeable.





ZB1 Wireless Module

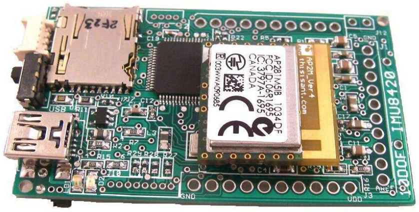
The ZB1 module is a new SOC Robotics ZigBee compatible adapter with an Atmel ZigBit ATZB-24-2A ZigBee module attached to it. The Molex connector on the bottom ZB1 is compatible with the ANT wireless base connector on the IMU8420. The ZigBit comes preloaded with SerialNet and is ready to join a ZigBee network as a slave device. Software on the IMU8420 configures the ZigBee node for correct network operation. SerialNet supports an extensive AT command set.



USB10Z Zigbee for PC

ANT Wireless Modules

ANT wireless modules are available from several distributors and is an 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.



Ant

USB10Z

The USB10Z is a USB 2.0 to 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 software 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 Java communication utility called IMULINK (Windows beta release only) will be released shortly that communicates with the IMU8420 Data logger in real time to manage IMU operation. IMULINK automatically time synchronizes one or more IMU8420, IMU6420 or IMU6410 Data Logger on power up so all files logged to uSD have the same synchronized time stamp.

The IMU8420 processor does not have EEPROM to store parameter preference settings. Preference settings are now stored in on chip Flash memory. If preferences are changed you must manually store the new settings by executing the “ss” command at the top-level menu. Due to a bug (which we are working to fix) the storing process interrupts the USB link requiring a restart of the IMU.

The IMULINK application allows you to start/stop logging, set a storage file name, set time and date. Individual ZigBee commands can also be sent. Multiple IMUs are supported by the application with the ability to set parameters for each IMU individually.

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 IMU8420 has a lithium polymer battery charging circuit that charges attached batteries whenever the device is connected to a USB port. The charging accepts a range of LiPo battery capacities.

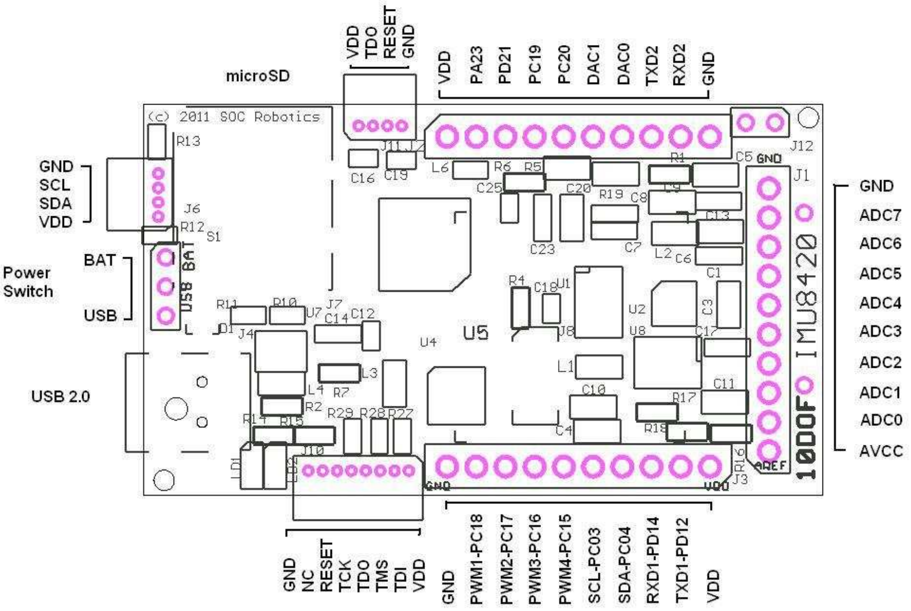
High Voltage Regulator (HVR)

The HVR is an optional circuit on the bottom of the board that converts 12-24V DC into the 3.3V needed to power the board. The HVR allows the IMU8420 to be powered in applications that use high voltage such as UAV, motion control and specialized instrumentation without the need for a separate power supply.

Expansion Ports

The IMU8420 has three expansion ports: J1, J2 and J3, two programming ports J10 and J11 and an I2C port. The programming ports do not have connectors installed, as the DFU bootloader is a more convenient way to load new software.

IMU8420 Connector Pin Assignment



The ADC inputs are capable of 1.5million samples/sec at 12bits. The DAC outputs are capable of 1million sample/sec. Each of the expansion ports can be configured in software to be an input, output or special function (PWM, UART, etc).

DFU BootLoader

The AT32UC3C2512 processor has an on chip boot loader activated by holding pin PD21 on connector J2 to ground when power is applied. This starts the DFU boot loader allowing new software to be loaded using Atmel's batchisp.exe program (available from Atmel as part of the

FLIP distribution). Instructions on how to load the appropriate software are in the IMU8420 Data Logger Project File (download from SOC Robotics web site).

IMU8420 Data Logger Overview V1.00

The IMU8420 Data Logger manages the acquisition and storage of 10 DOF MEMs sensor data to the microSD storage card. A Data Logger User Guide is in preparation that will described all the top level commands and submenus along with a description of the IMULINK application. At power up the application blinks the RED and GRN leds rapidly indicating the application is at the top-level command menu. Entering a '?' outputs a command summary. Not all commands are displayed – see the source code for a complete listing of commands.

```
Top-level Command Sumary
 ? - Output command summary
 l - Enter logging menu
 s - Enter signon/store parameters
 f - List files on uSD Card
 e - Enumerate I2C devices
 F - Enter uSD Command Menu
 t = Get current time
 T = Set time
 d - Get current date
 D - Set current date
 9s - Output 9DOF data once
 10s - Output 10DOF data once
 9t - Output 9DOF data continuously
 10t - Output 10DOF data continuously
 z - Enter ZigBee command menu
 i - Enter MEMs sensor command menu
 & - Restart the application
```

The output produced by the Extended Kalman Filter (EKF) roll, pitch and yaw can now be logged along with the MEMs data. The EKF completes a processing cycle in approximately 1.8mseconds while sensor acquisition takes approximately 0.9mseconds and uSD fsync takes 1.2mseconds every 200 records (default setting – compile time option). The fsync function snap shots an image of the data being sent to the uSD file without closing the file. If power is lost before the next fsync all data from the last fsync is lost. This means data can be logged as fast as every 2mseconds without the EKF running and 4mseconds with it running.

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

If a uSD card in installed the Data Logging application automatically recognizes it at power up. uSD cards up to 32G are supported. A file list is output by entering the "f" command. A "F" command starts the file command menu. u SD cards can be installed and removed with power on. The "F" command menu allows newly installed cards to be mounted using the "fi" command.

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 IMU8420 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 IMU8420 in position so that logging doesn't start prematurely. For example, say it takes 30 seconds to place the IMU8420 into a test jig after which movement on any axis exceeding a specific threshold is the trigger to start logging. This allows the IMU8420 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 IMU8420 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 - Store MinMax values at end of file
  u - Set Store data file name in uSD
  v - Display logging output
  x - Enter sensor range
  s - Start data logging
  h - Halt data logging
  i - Enter Sensor ID1 or ID2 name
  d - Select sensor(s) to log (a-accelerometer,g-gyro,m-mag,b-barometer,k-ekf)
  r - Retrieve data
  l - Log rate (msec)
  f - Output format (raw,processed,minmax)
  c - Display command summary
  q - Quit data logging mode
```

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 “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.

Extended Kalman Filter (beta)

The IMU8420 runs a real time sensor fusion algorithm using an Extended Kalman Filter (EKF) that outputs roll, pitch and yaw in real time. The EKF is based on the paparazzi code base modified to work with the sensors on the IMU8420. The EKF is in beta test and may require changes for optimal operation.

3-axis accelerometer -	ADXL345
3-axis gyrocompass -	L3GD20
3-axis magnetometer -	HMC5883

The EKF output is selected by entering the “kv” command at the top level menu. Entering the “k1” initializes the EKF which then waits for a “1” command to return the current roll, pitch and yaw in a query response mode.

The EKF is capable of accessing all sensor data and computing a new set of attitudes at a 300Hz rate.

EKF roll, pitch and yaw data can be logged along with MEMs sensor data by selecting EKF as a logging option using the “ld” command. By default EKF logging is not selected.

AVR Studio 6 Project Files

The IMU8420 software was developed using AVR Studio 6. The following changes has been made:

- The build 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 in the project file folder for details.
- The logging menu accepts commands from the USB interface and ZigBee wireless. When logging commands are received by the ZigBee interface the RED led on the ZB1 flashes briefly.
- When powering up the application checks to see if a ZB1 is installed and if it is turns the ZB1 GRN LED on.
- When the application is at the lop level command menu waiting for commands the RED and GRN leds flash very quickly. When actively logging data the GRN led is on and the RED led flashes slowly.
- When powering up the application checks to see if a uSD card is installed and if it is initializes it and mounts it. The “f” command lists uSD contents.
- The data logger is now listening to the USB 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.
- New logging sensor readings are appended to the end of the existing logging file if a unique file is selected.
- EKF data can be logged along with MEMs sensor data.
- Scanfu and Printf
- 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
```

- The AVR32 processor does not have EEPROM. Parameters such as log rate, logging termination mode, etc., are loaded from on chip Flash at power up. It is possible to change these parameters and store them in Flash but a bug (which hasn't been found yet) causes the USB connection to reset that breaks the USB serial link. The board must be repowered to restore the connection that then loads the new parameters. Parameters are not store automatically due to this bug. To store parameters use the "ss" command. If the code is rebuilt default parameters are automatically loaded into Flash which does not break the USB link as the save is done before the USB link comes up.
- Holding PD21 on connector J2 to ground when power is applied starts the DFU bootloader. A batch file in the debug project folder programs the IMU8420_Controller.hex file. No special programming hardware is required.
- Note that diagnostic code is included with regular commands in the source code.
- Date month and year not automatically updated.
- Reference to Serial Flash is still in the code although the IMU8420 does not have serial flash.

Please report any bugs to support@soc-robotics.com.

The IMU8420 is our high performance MEMs data logger and we plan to continually upgrade the software with new features and additional hardware options. If you have any suggestions for new features or hardware additions please let us know. Here is a list of proposed feature additions:

- Allow setting digital IO ports with logging options
- Set PWM outputs
- Support UART2 and UART3
- ANT wireless support
- ADC and DAC support

An IMU8420 Data Logger User Guide is in preparation that will document all the commands supported by the new software release and will provide an overview of the source code.