Application Report
SLAA518A–February 2012

Nine-Axis Sensor Fusion Using the Direction Cosine Matrix Algorithm on the MSP430F5xx Family

Erick Macias, Daniel Torres, Sourabh Ravindran

ABSTRACT

This application report explains the implementation of an Attitude and Heading Reference System (AHRS), using the ultra-low-power MSP430F5xx microcontroller, a magnetometer, a gyroscope, and an accelerometer on all three axes. The calibration of the sensors is key to the accuracy of the algorithm, therefore, the sensors' output must be calibrated before being input to the Direction Cosine Matrix (DCM) algorithm. The algorithm is applied to the calibrated sensor readings to calculate the Euler angles describing the orientation of a body; consisting of the yaw, roll, and pitch angles.

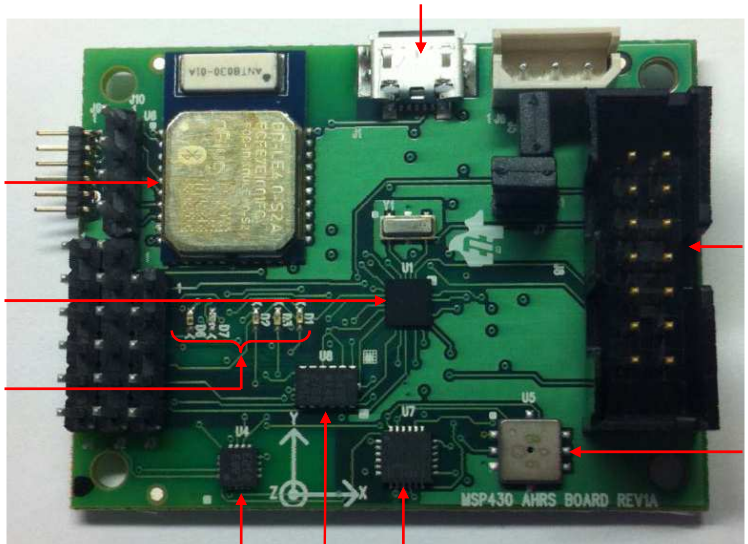
Project collateral and source code discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/slaa518.

This application report uses the *MPU-9150 MotionFit™ Wireless Developer Kit* from InvenSense (http://www.invensense.com).

Contents

List of Figures

List of Tables



1 Introduction

Modeling the orientation of a rigid body, including airplanes, RC toys, sport watches, smart phones, humans, etc. can be implemented by using the DCM algorithm. When creating an AHRS, also known as Magnetic, Angular Rate, and Gravity sensor (MARG), a magnetometer, a gyroscope, and an accelerometer are required. The calibrated sensors readings are fed to the DCM algorithm, which provides a complete measurement of the orientation, relative to the earth's magnetic field and the direction of gravity, expressed by the Euler (roll, yaw, and pitch) angles. In certain applications such as smart phones the ultra-low-power MSP430F5xx can handle all the communication with the motion sensors via I2C protocol. This leads to lower power consumption and higher CPU performance in the system, since they can request raw data or the orientation angles at any given time, meanwhile they can be in sleep mode, or they can perform other tasks that could have been delayed by the calculation of the orientation.

This document covers the following key points:

- Direction Cosine Matrix Algorithm (Section 2)
- MSP430F5xx Firmware (Section 3)
- Sensor Calibration (Section 4)

Figure 1 contains the AHRS circuit overview. The MSP430F5xx can communicate via the USB Module CDC class with two GUIs running in the computer:

- AHRS GUI (See Figure 4)
- AHRS Calibration GUI (See Figure 6)

The calibration GUI must be used when the AHRS system is used for the first time and in the case where the system's calibration values get corrupted by the presence of a constant magnetic field (hard iron effects). The AHRS GUI displays the Euler angles, as well as the visual representation of such; a horizon and a digital compass. These two GUIs require separate firmware to be downloaded to the MSP430F5xx:

- MSP430 AHRS Project AHRS Mode → AHRS GUI
- MSP430 AHRS Project Calibration Mode → Calibration GUI

There are other alternatives to send the Euler angles to the computer or other devices instead of USB; Bluetooth Low Energy (BLE) is one of them. The MSP430 communicates with BLE (BR-LE4.0) chip via UART. This application report will not cover the communication with the BR-LE4.0 chip (http://www.blueradios.com). To find more information about BLE, visit the http://www.ti.com website.

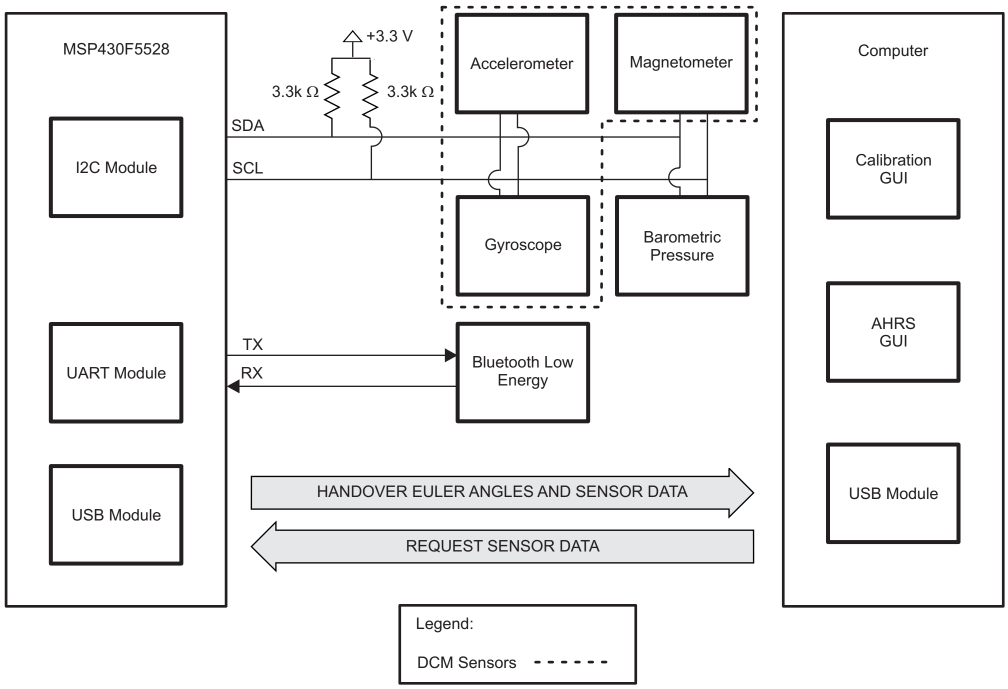


Figure 1. AHRS Circuit Overview

2 Direction Cosine Matrix Algorithm

The DCM algorithm calculates the orientation of a rigid body, in respect to the rotation of the earth by using rotation matrices. For a visual representation of the Direction Cosine Matrix Algorithm, see Figure 2. The rotation matrices are related to the Euler angles, which describe the three consecutive rotations needed to describe the orientation. The three sensors used in the algorithm are:

- The accelerometer measures earth's gravity field minus acceleration.
- The magnetometer measures earth's magnetic field.
- The gyroscope sensor measures angular velocity.

The gyroscope sensor is the primary sensor used to calculate the orientation of the system. Since the gyroscope is not affected by the gravitational or magnetic field, it requires the readings from the accelerometer and magnetometer to calculate a reference vector. Gyroscopes' readings have different offsets depending on which direction the gyroscope is facing; when these readings are integrated over time it causes the integral result to drift. The accelerometer is not affected by drift, therefore, it can be used as an orientation reference in the X and Z axis of the rigid body to compensate the roll-pitch error (gyro's offset error). The magnetometer's readings are used to calculate the heading of the rigid body. The magnetometer must be three axes to be able to calculate the heading of the system in any position of the sensor platform; to compensate yaw error. The heading of the system used as the reference vector in the Y axis (yaw error), in addition to the roll-pitch error calculated by the accelerometer, it allows the system to calculate the rotation correction matrix. Afterwards, the algorithm uses a proportional plus integral feedback controller on the correction matrix to the remove the drift from the gyro's readings.

The compensated gyroscope readings denoted as ω (omega), are then fed to the "Normalization & Kinematics" block as it can be seen in Figure 2. The rotation matrix's columns are unit vectors. Thus, before calculating the kinematics portion it must be normalized. (See *Renormalization* section in [2]). Once normalized, the gyroscope along with the previous rotation matrix are used to calculate the current rotation matrix (R Matrix) by using Equation 17 in the *Computing Direction Cosines From Gyro Signals* section in [2]. Finally, the Euler angles are calculated from the updated rotation matrix.

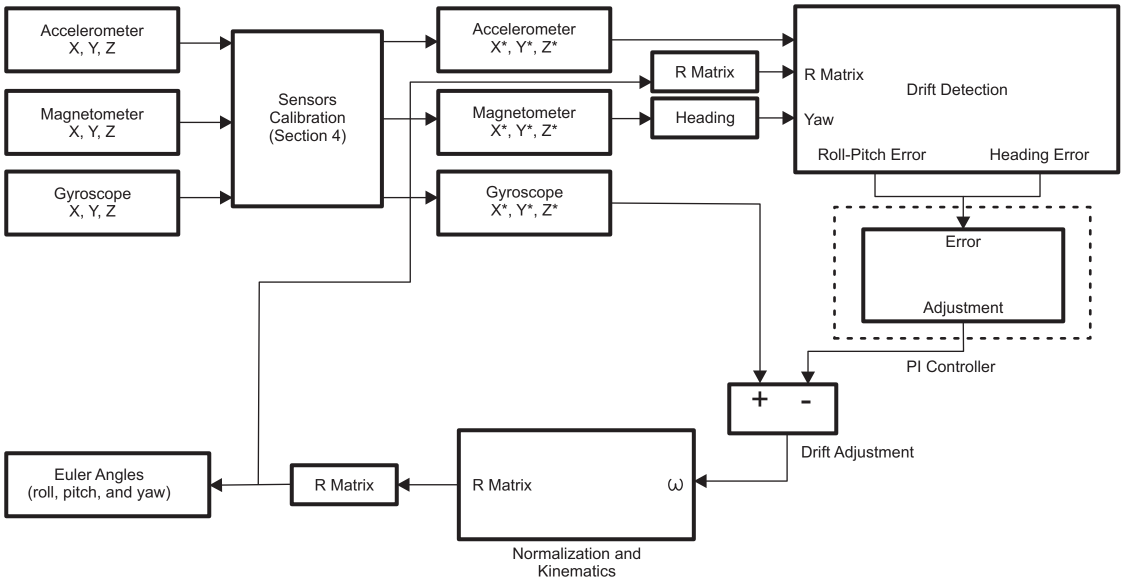


Figure 2. Direct Cosine Matrix Algorithm Overview

3 MSP430F5xx Firmware

This section covers the firmware's architecture of the AHRS. There are two modes of firmware that can be downloaded to the MSP430F5xx family: AHRS mode (see Figure 3) or the calibration mode (see Figure 4). The mode must be defined in `device.h`, where you must enable either `#define AHRS_MODE` or `#define CALIBRATION_MODE`.

Both modes initialize the MSP430F5xx by following these steps:

1. Set the main clock to 16 MHz.
2. Initialize I2C module (Master mode, Baud Rate ~ 400k Hz, 7-bit Addressing).
3. Initialize background timer (Rate ~20 mS, Disabled).
4. Initialize USB Module (CDC Class).

Afterwards the motion sensors are initialized, and the background timer is enabled. The timer wakes the MCU from low-power-mode 0 (lowest power consumption mode allowed when using USB module) at a 50 Hz (20 mS) rate. The accelerometer and gyroscope are read and calibrated at a 50 Hz (20 mS) rate. The magnetometer is read at 10 Hz (100 mS), since the heading of the system does not fluctuate as much as the gravitational field or angular velocity.

3.1 *AHRS Mode*

The magnetometer's three axes readings are soft and hard iron compensated (see Section 4), and the sensor platform's heading is calculated. The calibrated sensor readings are fed to the Direction Cosine Matrix algorithm (see Section 2) to calculate the Euler angles (roll, pitch, and yaw). The orientation angles are sent via USB to the AHRS GUI in the computer, at a 20 Hz (50 mS) rate. The GUI displays the Euler angles in a horizon (pitch & roll) and a digital compass (yaw) (see Figure 4).

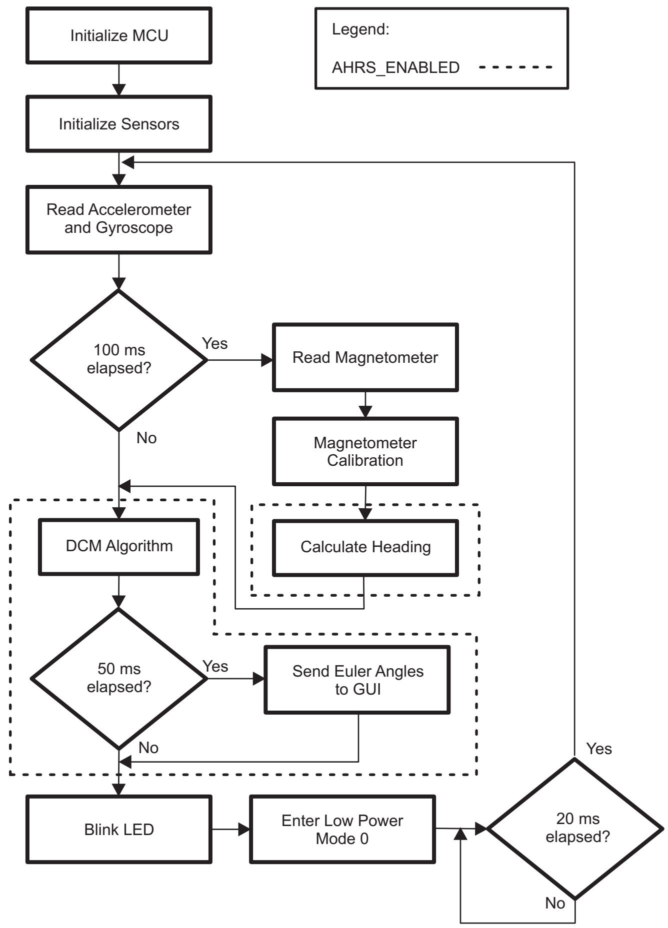


Figure 3. MSP430F5xx AHRS Firmware Overview

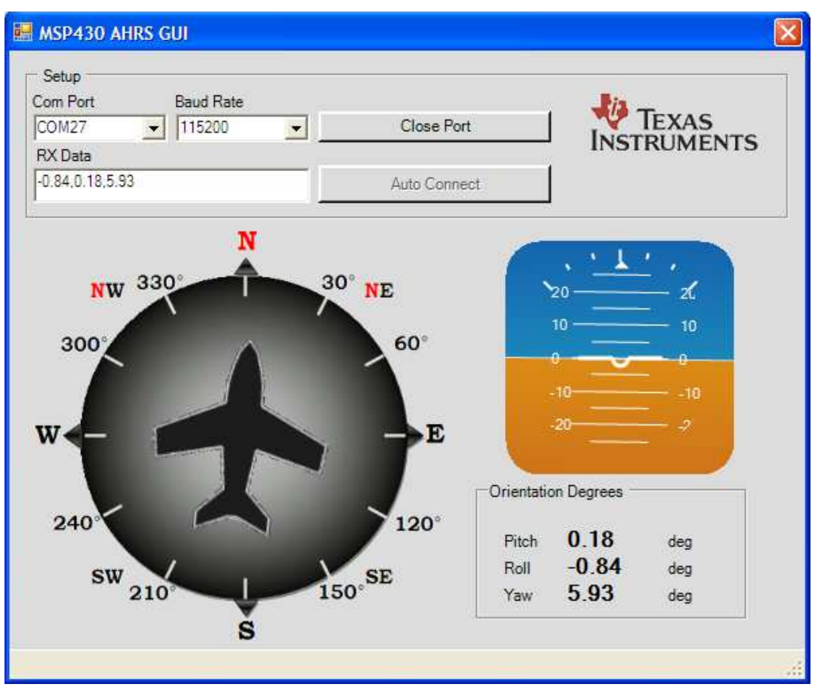


Figure 4. AHRS GUI

The background timer and the reading of the sensors can run at a faster frequency to increase the resolution of the algorithm's integration; therefore, gaining better accuracy of the orientation angles. On the other hand, when running the algorithm at a higher frequency it causes the power consumption and CPU usage to increase. Running the background timer at 50 Hz is the sweet spot for low-power consumption and orientation accuracy.

3.2 Calibration Mode

When the calibration mode (see Figure 5) is enabled instead of calculating the Euler angles, the firmware checks if a request for sensor raw data has been received or not. When the request (see Table 1) is received, the MSP430F5xx sends the GUI 500 samples of sensor data in all 3 axes via USB; this results in a 10 second calibration for the accelerometer and gyroscope, and a 50 second calibration for the magnetometer. When calibrating the magnetometer, it is very important to read the max and min value for each axis; sending more samples from the MSP430 to the GUI allows you to move the board in all the angles necessary for calibration.

Table 1. Sensor Data Request Commands

Command ID	Sensor Data Requested
0x31	Accelerometer
0x32	Gyroscope
0x33	Magnetometer

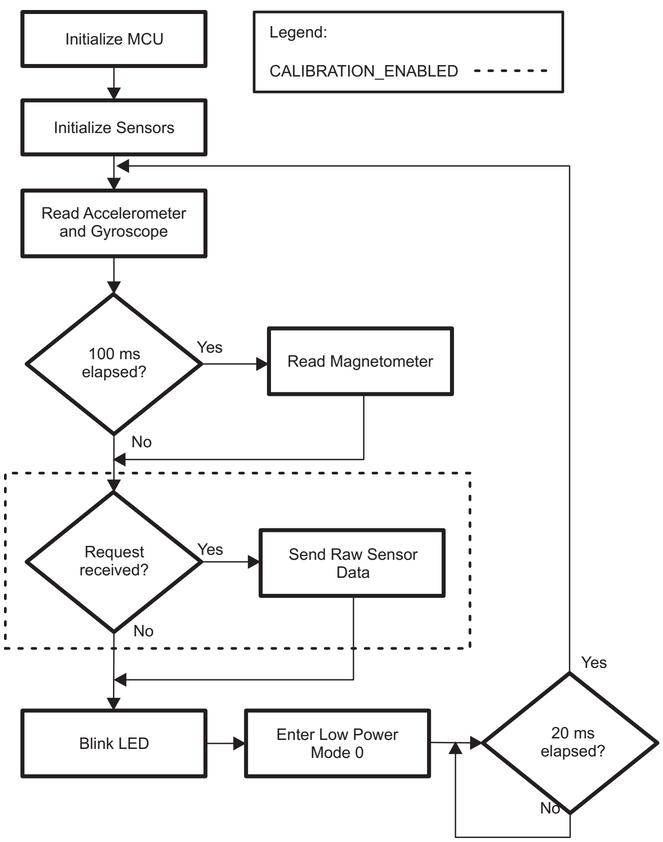


Figure 5. MSP430F5xx Calibration Firmware Overview

4 Sensors Calibration

This section covers how to calibrate the DCM sensors using the 9-Axis Sensor Fusion Calibration GUI (see Figure 6). The X, Y, and Z offsets calculated for each sensor must be updated inside `calibrationSensors()` in AppRoutines.c.

Figure 6. AHRS Calibration GUI

Follow these steps to calibrate your sensor platform:

1. Download the firmware with CALIBRATION_MODE enabled to the MSP430F5xx.
2. Install the USB drivers in the PC.
3. Open the 9-Axis Sensor Fusion Calibration GUI.
4. Click “Auto Connect”.

Now the Accelerometer, Gyroscope and Magnetometer buttons should be enabled.

4.1 Accelerometer

When calibrating the accelerometer, the GUI requests for the AHRS board to be placed in the three different positions. When the board is placed on a flat surface (parallel to earth), the only axis that should be non zero is the perpendicular axis to the surface; therefore, the readings on the two other axes will be offsets (using Equation 1). with a three axes accelerometer, readings can be taken from the accelerometer having the axis perpendicular to the flat surface and pointing to the sky being X, Y, and Z. For example, when X is pointing upwards to the sky, the offsets are calculated for Y and Z. Then when Y is pointing upwards to the sky, the offsets are calculated for X and Z. Therefore, when having gone through all three positions there are 2 offsets values for each axis, which are averaged and displayed as the X, Y, and Z offsets.

$$data_{offset} = \frac{data_{\min} + data_{\max}}{2} \quad (1)$$

4.2 Gyroscope

When calibrating the gyroscope, the AHRS board must be stationary on a flat surface where the angular velocity for all three axes should be 0. The offsets are calculated by using Equation 1 on all three axes. Once the GUI receives the requested raw gyroscope readings from the AHRS, it will display the offsets as the X, Y, and z offsets.

4.3 Magnetometer

There are two types of calibrations required for magnetometers: soft iron and hard iron calibration. Hard iron calibration is considered to remove constant magnetic field affecting the sensor platform. When graphing the output of a magnetometer in an ideal case, the output should be a perfect sphere in 3D centered at (0,0,0), but this is usually not the case. Instead, it is centered in another x,y,z location. For example, in Figure 7 the center lies in (34.5, -140.5, 46.5). These offsets were calculated by using Equation 1 on the readings of the magnetometer on all three axes after moving the board in all different angles.

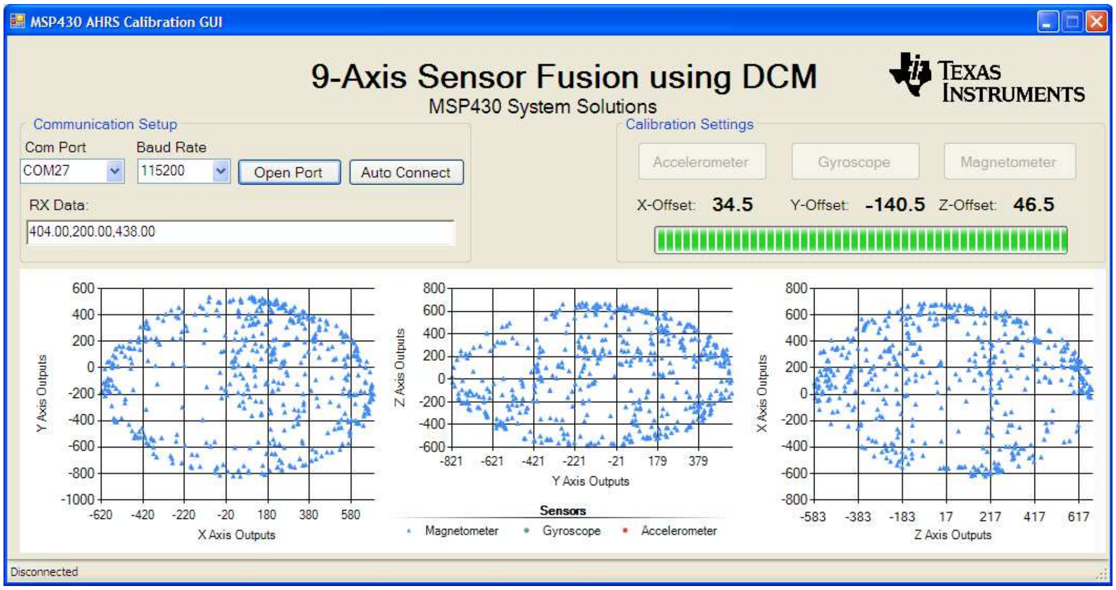


Figure 7. Magnetometer Hard Iron Calibration

When calibrating the magnetometer, it is best to rotate the board in three different rotations as displayed in Figure 8. While calibrating the magnetometer, the three graphs displayed in Figure 7 should be graphing a circle, which allows for the minimum and maximum of each axis to be taken into account when calculating the offsets. Soft iron calibration is required to eliminate the effects of electromagnetic fields, which causes the ideal sphere to become an oval shape figure. Soft iron calibration is performed in the firmware after the magnetometer values have been read and hard iron calibration has been applied.

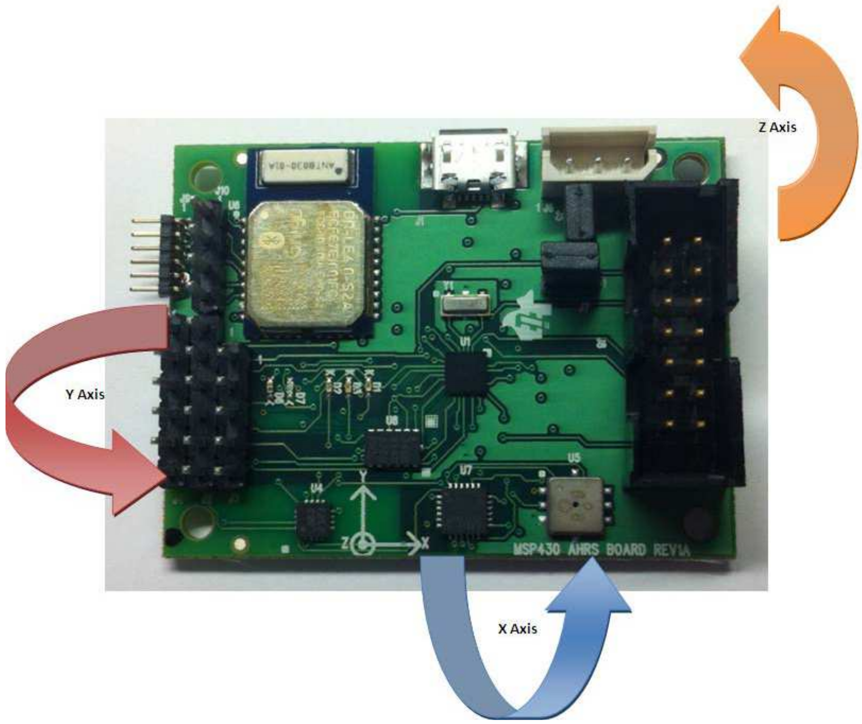


Figure 8. Rotations for Hard Iron Calibration of the Magnetometer

5 Conclusion

This section covers the power consumption of the sensor platform and the MSP430 requirements. The voltage across a shunt resistor (26.63 Ω) connected to the V_{CC} of the MSP430F5xx can be seen in Figure 9. The voltage measurements were taken with the USB module disabled and low-power-mode 3 enabled when the MSP430F5xx was neither reading the sensors nor calculating the DCM algorithm. The MSP430F5xx consumes ~ 5 µA in low-power-mode 3 and ~ 5.6 mA in active mode. The AHRS board requires being battery powered when the transmission of the Euler angles is via BLE. The cycle area of the voltage is 685.3 µVs, where each cycle is 20 ms. Therefore, the average voltage is 685.3 µVs / 20ms = 34.265 mV. Using Equation 2, the average current consumption of the MSP430F5xx is 34.265 mV/ 26.63 Ω ~ 1.28 mA.

$$V_{avg} = I_{avg} * R \xrightarrow{\Delta} I_{avg} = \frac{V_{avg}}{R} \quad (2)$$

Figure 9. MSP430 Voltage Measurement

Figure 9 shows the reading the motion sensors at different frequencies; the magnetometer (M) is read at a 10 Hz rate, while the accelerometer (A) and gyroscope (G) are read at 50 Hz. This figure shows the MSP430F5xx using ~30% of the CPU. The accuracy of the MSP430 AHRS board is TBD. Table 2 includes the current consumption measurements for the different ICs in the AHRS board.

Table 2. System Current Consumption

Integrated Circuits (ICs)	Average Current Measurement
MSP430F5xx	1.06 mA
Accelerometer	0.14 mA [(1)]
Gyroscope	6.5 mA [(1)]
Barometric Pressure Sensor	0.65 mA [(1)]
Magnetometer	0.1 mA
BLE Radio	30-mA worst case peak at 4 dBm

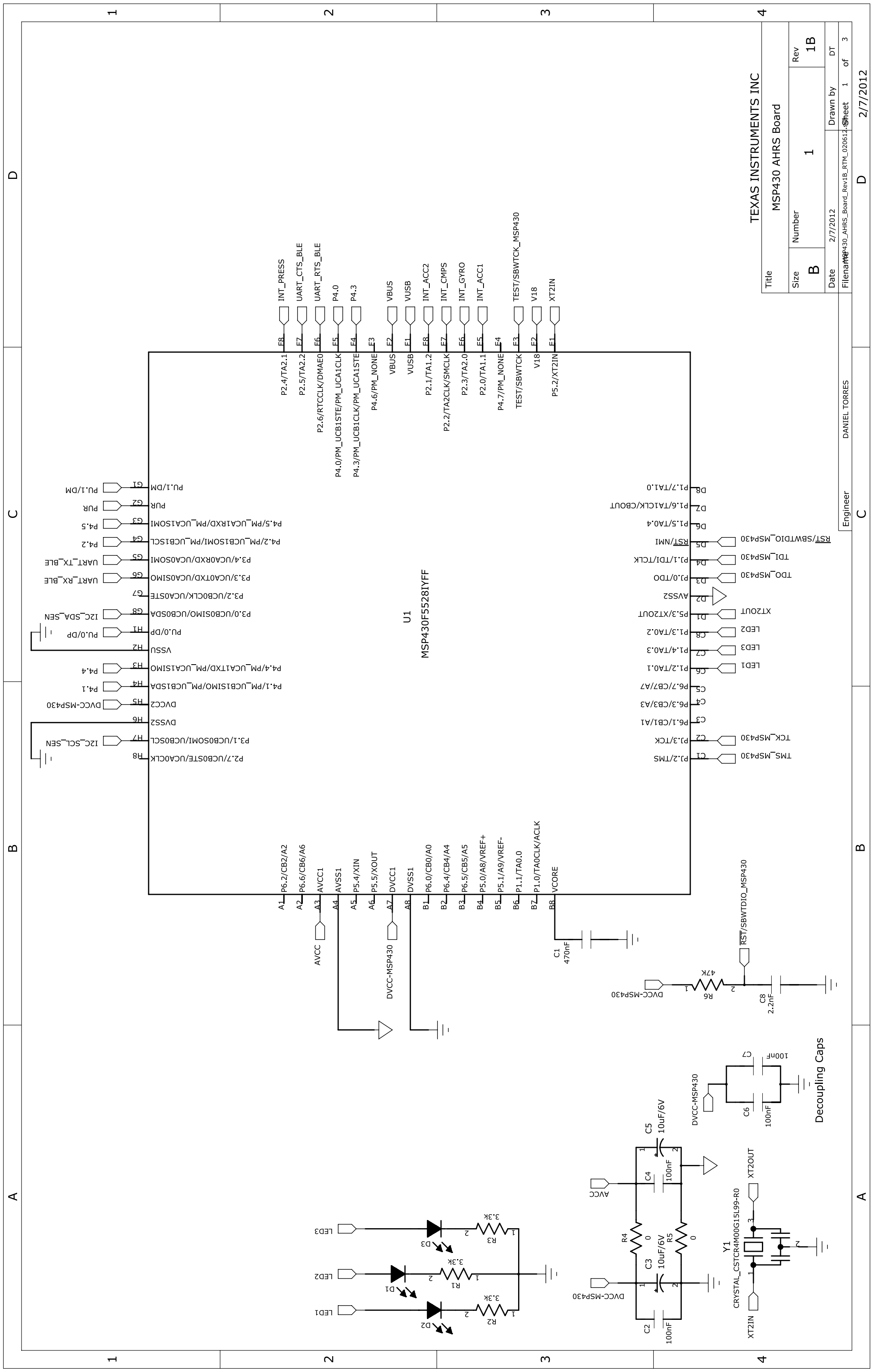
(1) The current measurement values were obtained from the sensors' datasheet.

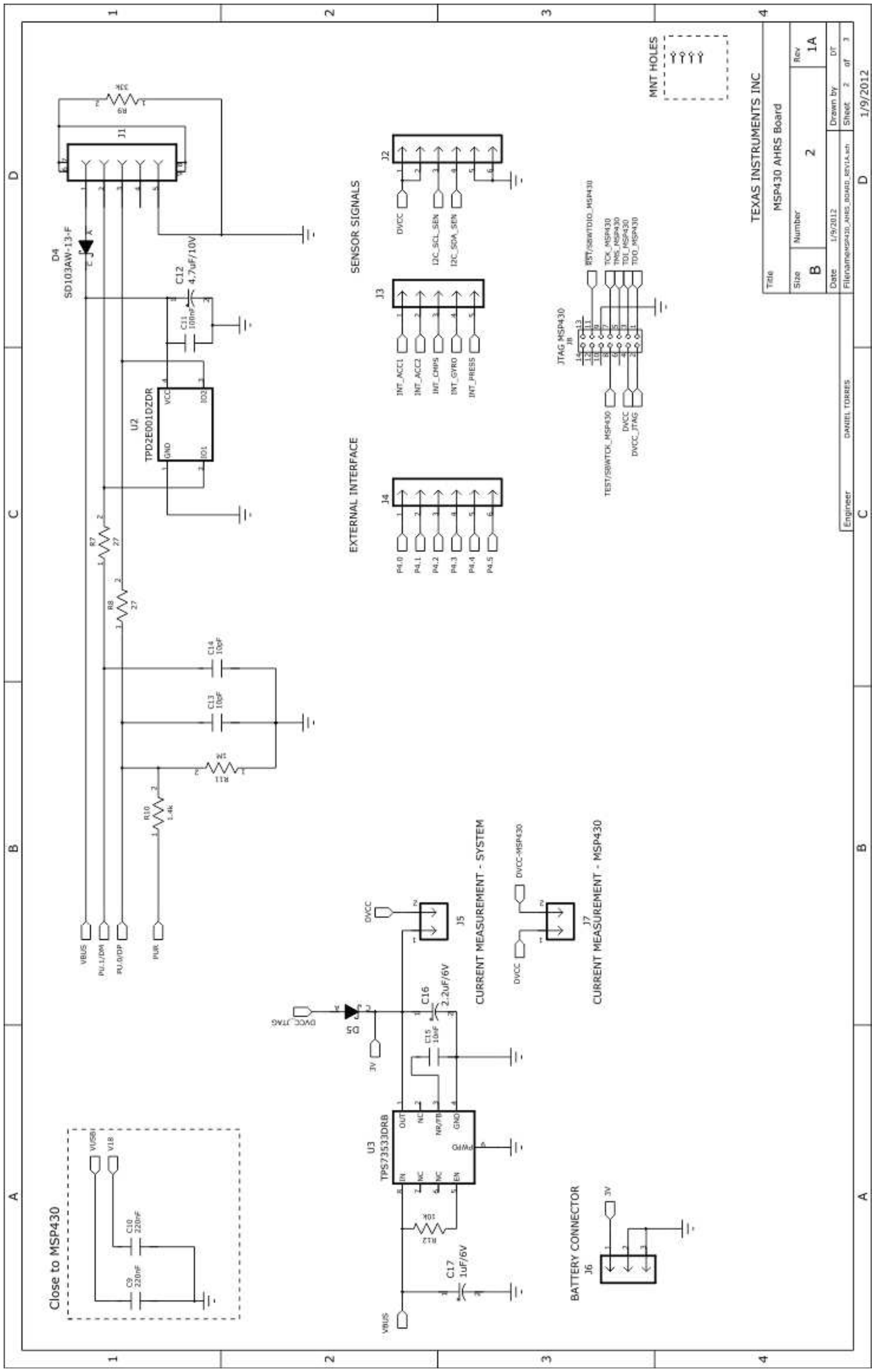
MSP430 Requirements:

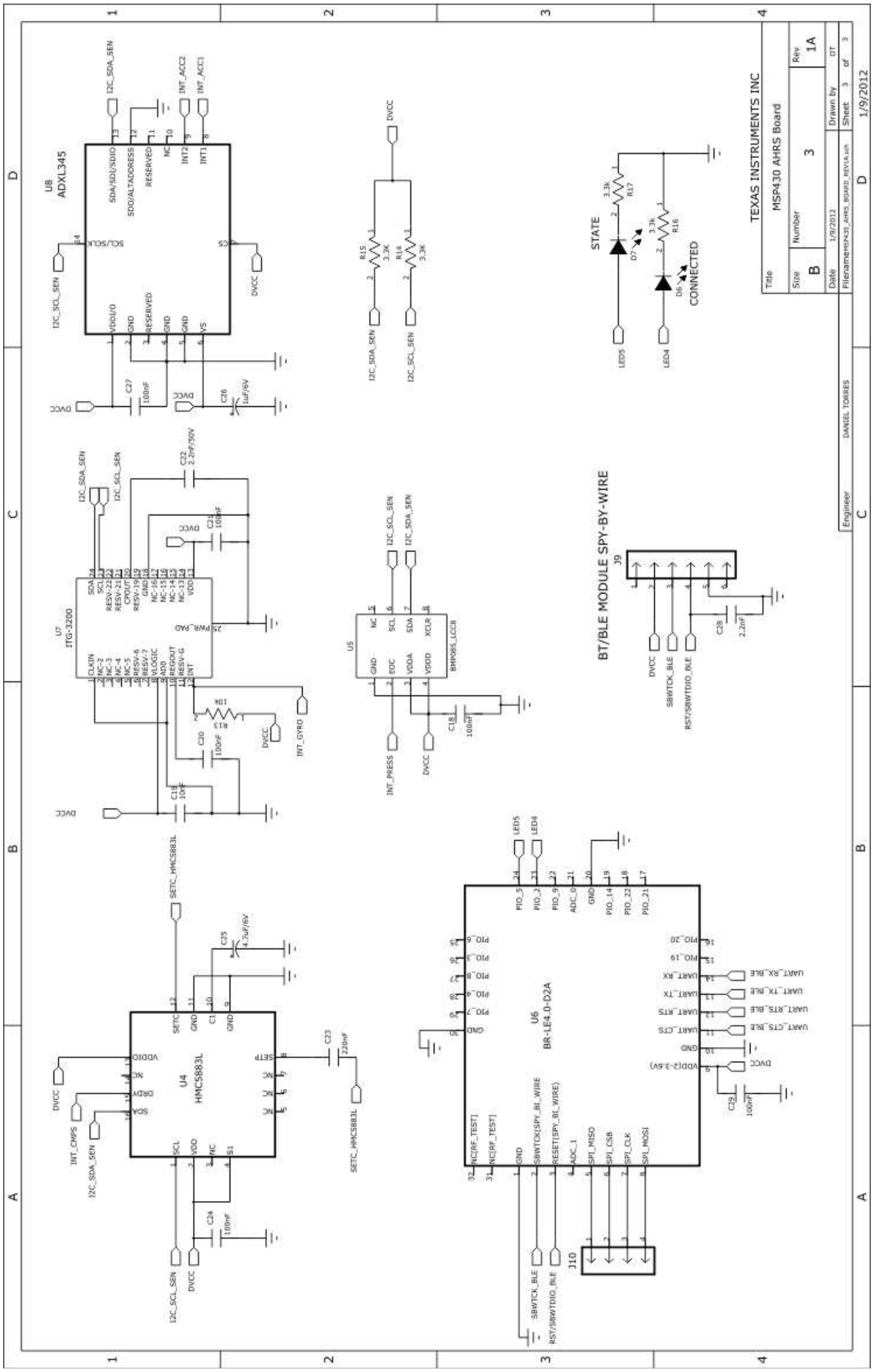
- RAM ~ 0.75 kB
- Flash ~ 11.7 kB
- CPU Usage ~ 21.34%
- SMCLK = 16 MHz, MCLK = 16 MHz, ACLK = REFO ~ 32.6 kHz

NOTE: The memory requirements include the DCM algorithm and exclude the USB Stack. IAR Embedded Workbench 5.20.1 was the IDE used to benchmark.

6 Schematics






U8
ADXL345
U7
ITG-3200
U5
BMP085_LCC8
U4
HMC5883L
U6
BR-LE4.0-D2A
BT/BLE MODULE SPY-BY-WIRE
J9
J10
STATE
CONNECTED
TEXAS INSTRUMENTS INC
MSP430 AHRS Board
DANIEL TORRES
1/9/2012

7 References

1. *MSP430x5xx/MSP430x6xx Family User's Guide* (SLAU208)
2. *Direction Cosine Matrix IMU: Theory* (http://gentlenav.googlecode.com/files/DCMDraft2.pdf)
3. *CkDevices Open Source Firmware* (http://www.ckdevices.com)
4. *CkDevices Open Source Mongoose Visualizer* (http://www.ckdevices.com)
5. *Compensating for Tilt, Hard-Iron, and Soft-Iron Effects* (http://www.sensorsmag.com/sensors/motion-velocity-displacement/compensating-tilt-hard-iron-and-soft-iron-effects-6475)

IMPORTANT NOTICE

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
		TI E2E Community Home Page	e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265