NanoSense
Fine Sun Sensor

Manual
High precision and ultra small vector sun sensor with digital interface

GomSpace A/S | Langagervej 6 | DK-9220 Aalborg East | Denmark
gomspace.com | sales@gomspace.com | T: +45 71 741 741

1 Table of Contents

2 Changelog

Date	Revision	Author	Description
18-11-2014	1.0	MB	First release
2-2-2015	1.0.1	MB	Updated command description
14-9-2015	1.1	MB	Updated correction algorithm
10-11-2015	1.2	KLK	Moved to new layout. Smaller text corrections
15-6-2017	1.3	KLK	Corrections to chapter 5.1
29-9-2017	2.0	KLK	New layout and new version of FSS. Picture chapter 5.1 changed, and added sentence with direction.
8-3-2018	2.1	KLK	Changed frontpage picture.

3 Unpacking and Handling

Warning: Please observe precautions for handling electrostatic sensitive devices.

The NanoSense Fine Sun Sensor (FSS) is an ESD sensitive device - proper precautions must be observed during the handling of the device.

Please use an ESD mat and a wrist strap as a minimum. Please wear gloves and handle only in a clean environment. If any cleaning of the parts is required prior to flight, use only ESD safe cleaning methods and a neutral, non-reactive, IPA solvent.

4 Communication

The NanoSense FSS can be communicated with as a slave I^2C device. A list of commands can be seen in Table 1.

Command Name	Command	Description
GetVersion	0x08	Get FSS firmware version
GETUUID	0x0D	Get FSS UUDI
SampleTemp	0x04	Sample FSS temperature sensor
GetTemp	0x05	Get FSS temperature
SampleSun	0x00	Sample FSS sun sensor data
GetSun	0x01	Get FSS sun sensor data
SetAdd	0x06	Set I^2C address
SaveAdd	0x07	Commit I^2C address

Table 1 List of commands

The following abbreviations are used in the command descriptions:

- S: Start condition
- RS: Repeated start
- P: Stop condition
- A+W: Write slave address plus I^2C write bit
- A+R: Write slave address plus I^2C read bit
- W: Write data to slave
- R: Read data from slave

Action	Data	Comment
S		
A+W	ADDR | W	
W	0x06	CMD_SET_I2C_ADDR
W	NEW ADDR	
W	0x00	
P		

Table 2 Command to set new device address

This command only sets a volatile address so if the device is reset the old address will be restored. To persist the address, use the command CMD_COMMIT_I2C_ADDR.

Action	Data	Comment
S		
A+W	ADDR | W	
W	0x07	CMD_COMMIT_I2C_ADDR
W	0x33	
P		

Table 3 Command to commit current device I^2C address.

Action	Data	Comment
S		
A+W	ADDR | W	
W	0x08	CMD_GET_ID
RS		
A+R	ADDR | R	
R	B1	MSB
R	B2	
R	B3	
R	B4	LSB
P		

Table 4 Command to get 32 bit device id.

Action	Data	Comment
S		
A+W	ADDR | W	
W	0x00	CMD_SUN_SAMPLE_SUNSENSOR
RS		

Table 5 Command to trigger sampling of sun sensor.

The data can be read after delay of 30 ms with the command CMD_SUN_GET_DATA. If another command is called between sample sensor and the get sensor command, the data can be destroyed.

Action	Data	Comment
S		
A+W	ADDR | W	
W	0x01	CMD_SUN_GET_DATA
R	A_MSB	A = (A_MSB << 8) + A_LSB
R	A_LSB	
R	B_MSB	B = (B_MSB << 8) + B_LSB
R	B_LSB	
R	C_MSB	C = (C_MSB << 8) + C_LSB
R	C_LSB	
R	D_MSB	D = (D_MSB << 8) + D_LSB
R	D_LSB	
P		

Table 6 Gets the sampled data from the sun sensor

Wait at least 30 ms after the sample sensor command before calling this command.

Action	Data	Comment
S		
A+W	ADDR | W	
W	0x04	CMD_SUN_SAMPLE_TEMP
P		

Table 7 Sample temperature.

Action	Data	Comment
S		
A+W	ADDR | W	
W	0x05	CMD_SUN_GET_TEMP
RS		
A+R	ADDR | R	
R	TEMP_H	Temperature MSB
R	TEMP_L	Temperature LSB
P		

Table 8 Sample temperature.

The temperature in degrees Celsius is (((int16_t)((TEMP_H << 8) + TEMP_L)) >> 2) * 0.03125.

NOTE: The NanoSense FSS does not have I^2C isolation, which means that if the NanoSense FSS is not powered, it can load the I^2C bus to a print where it is no longer usable.

5 Sun Vector Calculation

The raw output from the NanoSense FSS must be converted to a sun vector for use in an attitude determination. Then sensor is delivered with two sets of calibration data. A parametric set and a lookup table set. The parametric set is used in a set of equations to convert from raw data to a sun vector and the look-up table is used to adjust the measurement to achieve full accuracy.

Parameter	Condition	Typical value (1-sigma)
Accuracy (0°-45°)	No look-up	+/- 1°
Accuracy (0°-45°)	With look-up	+/- 0.5°
Accuracy (45°-90°)	No look-up	+/- 4°
Accuracy (45°-90°)	With look-up	+/- 2°

Table 9 Sensor precision

5.1 Parametric Calibration

The parametric calibration set is used to convert from the raw measurement (A, B, C, D) to a sun vector (R_{sun}).

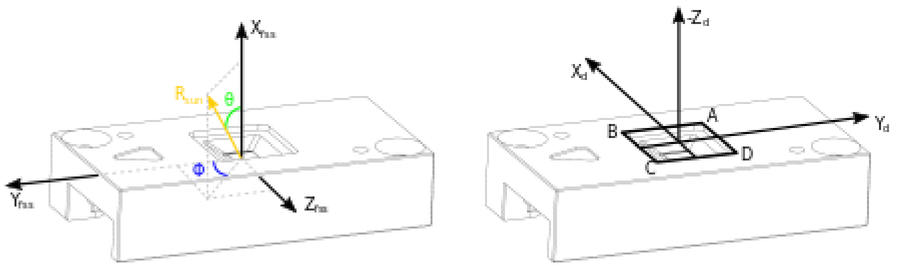

Figure 1 Two reference frames. The sun vector is shown in the FSS frame (left) and the raw measurements are given in the diode frame (d)(right)

Make sure Y_{fss} pointing the same way as connector P1 harness exits the casing.

The calibration consists of the four parameters:

$$h, x_0, y_0, \delta$$

With these parameters, a mapping from raw data to sun vector can be performed using the following four equations:

$$T = \begin{bmatrix} \cos(\delta) & \sin(\delta) \\ -\sin(\delta) & \cos(\delta) \end{bmatrix}$$

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = T \begin{bmatrix} \dfrac{(A+B)-(C+D)}{A+B+C+D} + x_0 \\ \dfrac{(A+D)-(B+C)}{A+B+C+D} + y_0 \end{bmatrix}$$

$$\phi = \operatorname{atan}\left(\frac{x_d}{y_d}\right), \qquad \theta = \operatorname{atan}\left(\frac{\sqrt{{x_d}^2 + {y_d}^2}}{h}\right)$$

$$R_{sun} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \end{bmatrix} = \begin{bmatrix} x_{fss} \\ y_{fss} \\ z_{fss} \end{bmatrix}$$

5.2 Look-up table calibration

The look-up table calibration consists of a number of points in the ϕ–θ plane as shown Figure 2

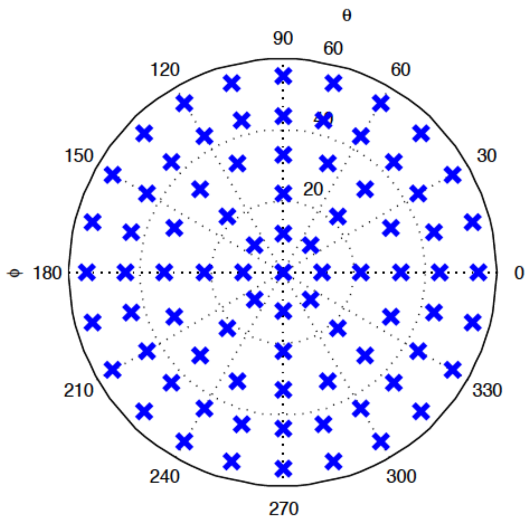

Figure 2 Loop-table sampling points

As standard 81 calibrations points are used distributed with an elevation step size of 11 degrees: θ = [0 11 22 33 44 55] degrees and the number of azimuth ϕ points distributed as [1 8 8 16 24 24]. An excerpt of a calibration table is show in Table 10

θ	ϕ	θ_{err}	ϕ_{err}
0	0	$\theta_{err}(0)$	$\phi_{err}(0)$
11	0	$\theta_{err}(1)$	$\phi_{err}(1)$
11	45	$\theta_{err}(2)$	$\phi_{err}(2)$
11	90	$\theta_{err}(3)$	$\phi_{err}(3)$
…	…	…	…
55	315	$\theta_{err}(79)$	$\phi_{err}(79)$
55	330	$\theta_{err}(80)$	$\phi_{err}(80)$
55	345	$\theta_{err}(81)$	$\phi_{err}(81)$

Table 10 Excerpt of a calibration table

An example of how to perform the correction is illustrated in the code snipped below. A look-up table (C) consisting of the last two columns of Table 10 is used to correct the angles theta and phi. The code first calculates the index of the first corresponding theta point and then calculates the correct phi index for that particular theta. The global index in the table is then found by adding all the previous theta indices. This is done for the two closest theta indices and the result is taken as a weighted average.

```
%Perform sun angle correction based on look-up table C
thetastep = 11;                      % Degrees pr. theta step
thetapoints = 6;                     % Number of theta cal. points
phipoints = [1 8 8 16 24 24];        % Number of phi cal. points
f = rem((theta*180/pi) / thetastep+1,1);

%Find index 1
itheta = ceil( (theta*180/pi) / thetastep)+1;
if itheta > theta_points
  itheta = theta_points;
end

iphi = round( (phi*180/pi) / (360/phi_points(itheta)) ) + 1;
if (iphi > phi_points(itheta));
  iphi = iphi - phi_points(itheta);
end

index = sum(phi_points(1:itheta)) - phi_points(itheta) + iphi;

%Find index 2
itheta2 = floor( (theta*180/pi) / thetastep)+1;
if itheta2 > theta_points
  itheta2 = theta_points;
end

iphi2 = round( (phi*180/pi) / (360/phi_points(itheta2)) ) + 1;
if (iphi2 > phi_points(itheta2));
 iphi2 = iphi2 - phi_points(index_theta2);
end

index2 = sum(phi_points(1:itheta2)) - phi_points(itheta2) + iphi2;


%Correct sun angles
theta = theta - (C(index,3)*(f) + C(index2,3)*(1-f))*pi/180;
phi = phi - (C(index,4)*(f)+C(index2,4)*(1-f))*pi/180;
```