Software Requirements Data

FOR

Sensor Data Acquisition and Relay System (SDARS)

Table of Contents

CHANGE HISTORY

Version	Date	Change Description	Author	Reviewers	Remarks
1.0a	12/10/20	Initial Release	Jyothesh Rao		
1.0	27/10/20	Updated based on the Oak_SDARS_SRD_Review_Report	Jyothesh Rao	IV&V	Updated the sections 3, 7 and 8

SRD COMPLIANCE MATRIX WITH RESPECT TO DO-178B

Table 1: SRD Compliance Matrix with respect to DO-178B

DO-178B section	Section Heading	Sections in SRD
11.9 a	Description of the allocation of system requirements	Section 2
11.9 b	Functional and operational requirements	Section 3
11.9 c	Performance criteria	Section 4
11.9 d	Timing requirements and constraints	Section 5
11.9 e	Memory size constraints	Section 6
11.9 f	Hardware and software interfaces	Section 7
11.9 g	Failure detection and safety monitoring requirements	Section 8
11.9 h	Partitioning requirements	Section 9

1 INTRODUCTION

This document specifies the Software Requirements of Sensor Data Acquisition and Relay System (SDARS), designed and developed by Oak Systems, Bangalore for Aircraft as per RTCA DO-178B, Level-B process.

1.1 Purpose

This Software Requirements Data (SRD) describes the Software CSCI Description to be met by SDARS. The Software Requirements laid down here shall be used as a baseline in the Software Design and Development to achieve System Requirements in the form of Software product.

1.2 SCOPE

This document provides a definition of the high-level functional, performance and derived software requirements of the SDARS software (SDARS_SW_01).

1.3 REFERENCES

1.3.1 EXTERNAL DOCUMENTS

i. Software Considerations in Airborne Systems and Equipment certification: RTCA/DO-178B, Date: 01.12.1992
ii. TECHNICAL SPECIFICATIONS of SDARS.

1.3.2 INTERNAL DOCUMENTS

i. TECHNICAL SPECIFICATIONS of SDARS for Aircraft, Oak_Ts_SensorDataAcq_v1.0, Issue No: 001
ii. Software Development Plan of SDARS, Oak_SDP_SensorDataAcq_v1.0, Issue No: 001
iii. Software Requirement Standards of SDARS, Oak_SRS_SensorDataAcq_v1.0, Issue No: 001

1.4 Acronyms

Acronym	Abbreviation
CBIT	Continuous Built In Test
CSCI	Computer Software Configuration Item
ICD	Interface Control Document
NA	Not Applicable
PBIT	Power-on Built in Test
POST	Power On Self-Test
SDP	Software Development Plan
SDARS	Sensor Data Acquisition and Relay System
SRD	Software Requirements Data
SRS	Software Requirement Standards
TS	Technical Specifications

2 Software Overview

SDARS Software provides sensor information to external Systems. It is Embedded System software that Acquires data from any bus and Sends it to other systems after applying filtering. SDARS communicates through RS-422 protocol. High level block diagram is show in Figure 1.

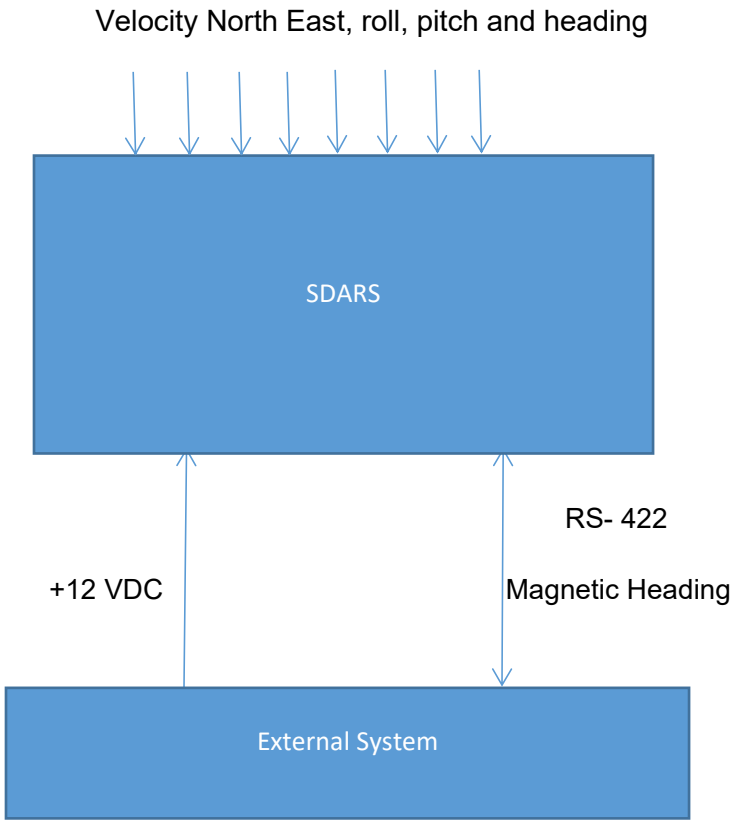

Figure 1: "System overview diagram"

2.1 User Characteristics

The Magnetic heading is a navigational data. The system sends data to Display unit which processes data and displays on display panel.

2.2 Functional allocations

1. SDARS shall receive navigation data from sensors vi pitch, yaw, roll in digital form and send magnetic heading data to external interface

2. There shall be a built-in test for runtime health monitoring.
3. Power on self-test (POST) shall indicate the status of internal voltages, internal memory
4. Continuous built in test (CBIT) shall start after POST indicates the status of external input and internal voltages. CBIT shall run continuously at intervals of 20millisecond.
5. There shall be a parameter estimation algorithm that takes the sensor data for estimation of magnetic heading.
6. The system shall provide heading information over RS422 interface in UART protocol at 9600 baud rate.

2.3 System Architecture

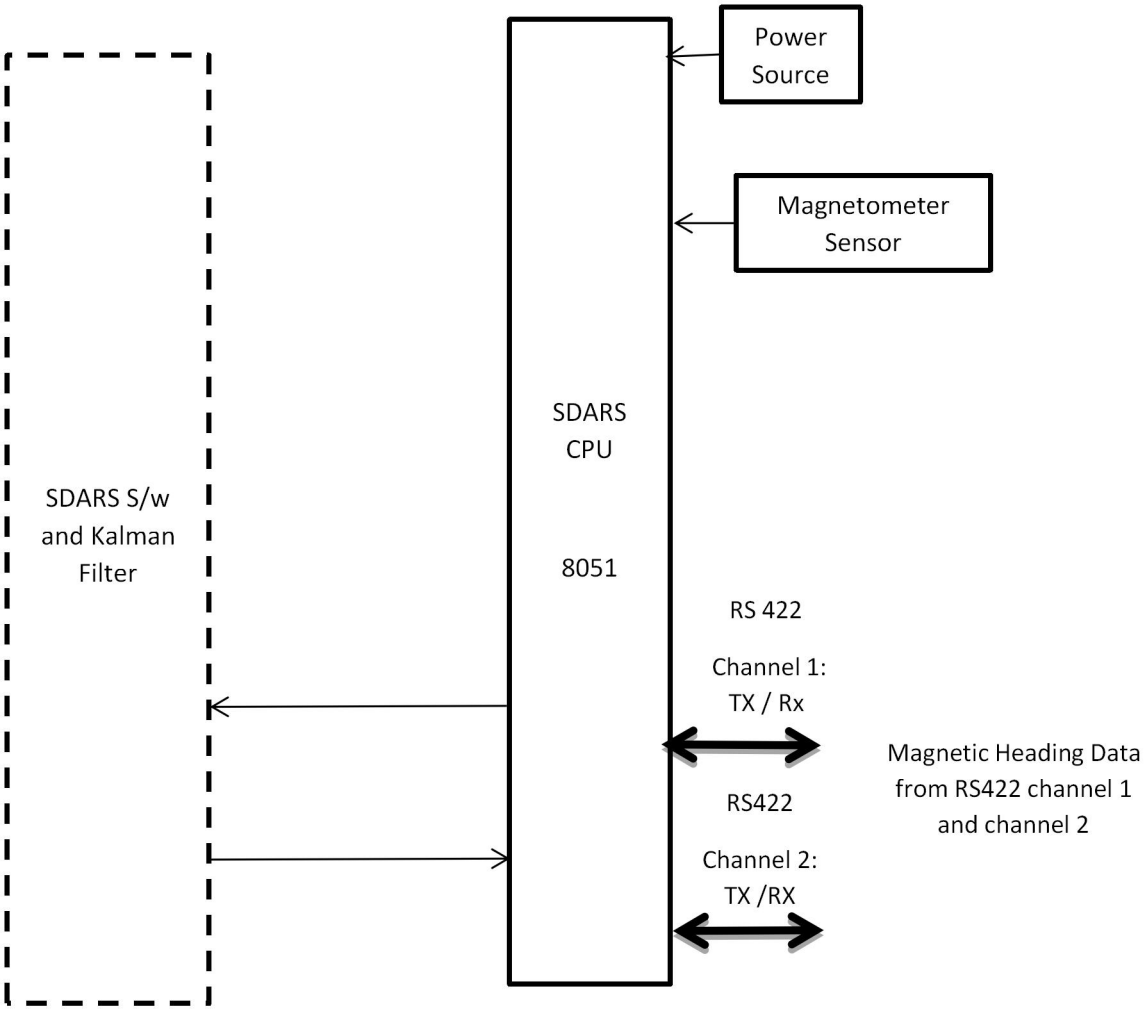

Figure 2: Overall Architecture of SDARS

SDARS software will acquire the Magnetometer sensor data, filter the data by Kalman filter software and provides magnetic heading data. It transmits the Filtered magnetic heading data via RS422 interface to other LRUs.

2.4 Software Architecture

SDARS software shall start with initialization and POST. After completion of POST it enters normal mode where s/w shall acquire the magnetic sensor data, transmits it to Kalman filter software and filtered data shall be transmitted to external interfaces via RS422. The Kalman filter algorithm takes the input of roll, pitch & yaw and its rate of change to filter and generate the magnetic heading. This heading data is then transmitted to external interfaces via two RS422 channels.

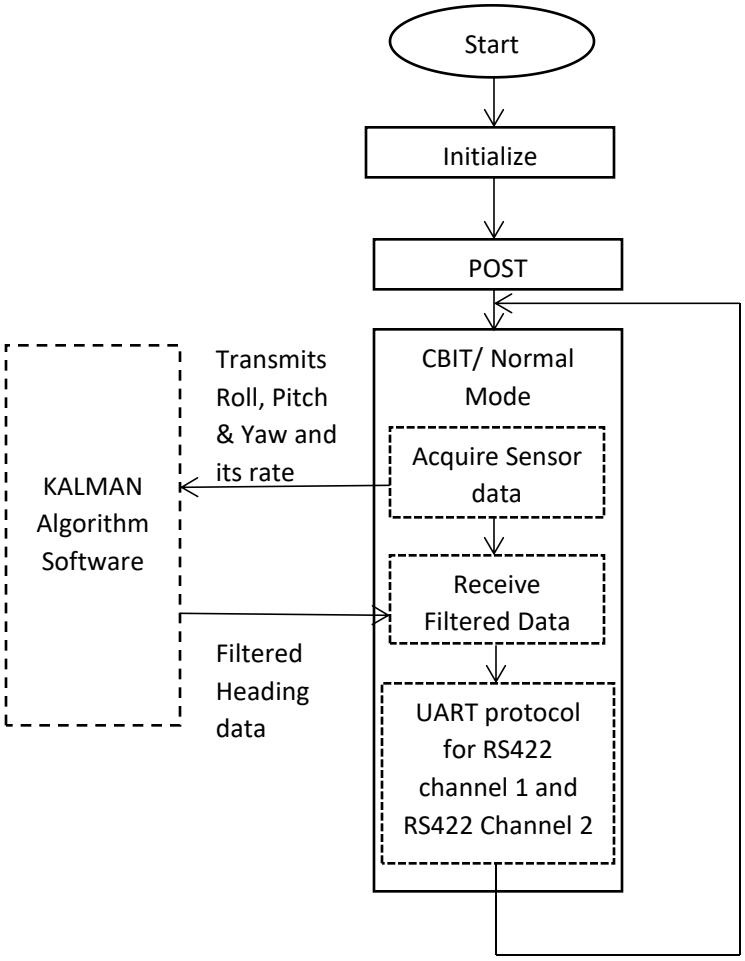

Figure 3: Software Architecture of SDARS

3 Functional Requirements

3.1 PROCESSOR REQUIREMENTS

3.1.1 SDARS_Flash_Req_001

SDARS_SW_01 shall be flashed on 8051 controller having flash memory of 16MB.

3.1.2 SDARS_Flash_Req_002

SDARS_SW_01 shall have code size less than 50% of flash memory size.

3.1.3 SDARS_Flash_Req_003

SDARS_SW_01 shall have RAM memory of 256bites and the execution consumption should not exceed 50%

3.2 POST REQUIREMENTS

3.2.1 SDARS_Init_Req_001

SDARS_SW_01 shall initialize the following:

a) RS422
b) Magnetic sensor.

3.2.2 SDARS_POST_Req_001

SDARS_SW_01 shall carry out power on self-test after initialization is completed

POST includes:

a) Processor check
b) Power supply checks
c) RS422 check
d) Magnetic sensor check.

3.3 CBIT REQUIREMENTS

3.3.1 SDARS_CBIT_Req_001

SDARS_SW_01 shall do CBIT checks at periodicity of 1 minute in Normal mode after POST check is successful. The CBIT checks are as follows:

a) RS422 check
b) Magnetic sensor check.

3.4 FUNCTIONALITY REQUIREMENTS

3.4.1 SDARS_Aquire_Req_001

SDARS_SW_01 shall acquire the magnetic sensor data containing Roll, Pitch and Yaw along with its rates at sampling rate of 1000samples/second.

3.4.2 SDARS_Aquire_Req_002

SDARS_SW_01 shall transmit the acquired data to Kalman software and receives the filtered heading data via RS232 port at baud rate of 9600.

3.4.3 SDARS_Kalman_Req_001

Kalman software shall receive the sensor data, normalize the data and shall apply the Kalman filter to the data.

3.4.4 SDARS_Kalman_Req_002

Kalman software shall transmit the filtered magnetic heading data to the SDARS_SW_01.

3.4.5 SDARS_RS422_Req_001

SDARS_SW_01 shall transmit the magnetic heading data at a baud rate of 9600.

3.4.6 SDARS_RS422_Req_002

SDARS_SW_01 shall receive the acknowledgement at a baud rate of 9600.

3.4.7 SDARS_UART_Req_001

SDARS_SW_01 shall use UART Interrupt (UART ISR) to detect the completion of data transmission or reception without continuously polling the flag bit, thus saving processor time.

4 PERFORMANCE CRITERIA

There is no functionality having any precision or accuracy

5 TIMING REQUIREMENTS

5.1 POST REQUIREMENTS

5.1.1 SDARS_time_Req_001

SDARS_SW_01 shall complete Initialization and PBIT within 90 sec

5.2 CBIT REQUIREMENTS

5.2.1 SDARS_time_Req_002

CBIT in SDARS_SW_01 shall be performed at a periodicity of 10milisecs.

5.3 RS422 REQUIREMENTS

5.3.1 SDARS_time_Req_003

SDARS_SW_01 shall transmit and receive the data at baud rate of 9600

6 MEMORY SIZE CONSTRAINTS

NA as there are no memory related requirement

7 INTERFACES

7.1 RS422 REQUIREMENTS

7.1.1 SDARS_RS422_Req_003

SDARS_SW_01 shall have Message Format is as follows: HEADER - ADDRESS - COMMAND/ERROR CODE/PARAMETERS - TRAILER – CHECKSUM.

7.1.2 SDARS_RS422_Req_004

SDARS_SW_01 shall calculate the checksum byte as follows:

The sum modulo 95 of all message characters beginning with the header byte up to and including the trailer byte. The value 32 is subtracted from each character value before taking the modulo 95 sum. The value 32 is added to the final sum to obtain the checksum value. All values are in decimal.

7.1.3 SDARS_RS422_Req_005

SDARS_SW_01 shall have the Header byte as 7BH and Trailer byte as 7DH.

7.1.4 SDARS_RS422_Req_006

SDARS_SW_01 shall have the address may take on the values from 64 to 95 decimal (40H to 5FH).

7.1.5 SDARS_RS422_Req_007

SDARS_SW_01 shall have the Command/Error Code/Parameters as all ASCII printable characters in the range of 20H to 7EH

8 FAILURE DETECTION AND SAFETY MONITORING REQUIREMENTS

POST and CBIT failures are already captured in Functional requirements

8.1 Safety REQUIREMENTS

8.1.1 SDARS_Fail_Req_001

SDARS_SW_01 shall restart the system if WDT failure occurs

8.1.2 SDARS_Fail_Req_002

SDARS_SW_01 shall not have a normal behavior if POST/CBIT check fails

8.1.3 SDARS_Fail_Req_003

SDARS_SW_01 shall handle the Power supply exceptions in the POST

9 PARTITIONING REQUIREMENTS

NA as there is no operating systems and memory.

10 COTS REQUIREMENTS

Kalman filter software a Commercial-off-the-shelf software which shall be interfaced with RS232 port. This software receives the sensor data from SDARS_SW_01, normalizes the data and shall apply the Kalman filter to the data. Kalman software shall continuously run at periodicity of 10ms and transmits the Filtered heading data to SDARS_SW_01 whenever Kalman filter *software* receives request command from SDARS_SW_01. The Kalman software shall be designed with Roll, Pitch and Yaw and its rates (Velocity north east) as inputs. These data (8bits) shall be transmitted at a rate of 100Hz and required to generate filtered Heading data at the rate of 100Hz. Roll , Pitch, Yaw and Rate combined shall be transmitted at every 40 millisecond, i.e., each data at 10 millisecond.

Kalman algorithm should use Yaw as the heading data and roll and pitch are supported for the filter as estimates. The Kalman filter should use the 10 point sampling on Yaw. if any change occurs in rates, Kalman estimates changes. Roll, Pitch and Yaw data are normalized and data will be in degrees. Rate will be in Degrees/sec. Tolerance for the data will be varied from 1 degree to 2 degree where 1 degree as maximum tolerance when no rate of change and 2 degree as maximum tolerance when rate of change is present. The data will be transmitted via RS232 internally with a baud rate of 9600.

COTS software shall convert the 8 bit octal data to decimal and normalize the decimal data to max 2pi radian. If the value exceeds 2pi radian then it should roll back from 0 to 2pi.

Quaternion formula:

Roll as X, Pitch as Y and Yaw as Z

```
cy = cos(Z* 0.5)
 sy = sin(Z* 0.5)
 cp = cos(Y* 0.5)
 sp = sin(Y* 0.5)
 cr = cos(X* 0.5)
 sr = sin(X* 0.5)
```

```
   Quaternion q;

   q.w = cr * cp * cy + sr * sp * sy
   q.x = sr * cp * cy - cr * sp * sy
   q.y = cr * sp * cy + sr * cp * sy
   q.z = cr * cp * sy - sr * sp * cy

siny_cosp = 2 * (q.w * q.z + q.x * q.y)
cosy_cosp = 1 - 2 * (q.y * q.y + q.z * q.z)
angles.Z= atan2(siny_cosp, cosy_cosp)
```

Kalman formula:

Kalman Estimation shall be done using the below generalized formula

The estimated (n) = Predicted value (n) + Factor * (Measurement – Predicted Value (n))

Where n is current state

Factor = Uncertainty in estimate / (Uncertainty in estimate + Uncertainty in measurement)

Predicted value = State Transition * current value + Control Input *Control vector + Noise.

Details of the algorithm shall be captured in design document.

Kalman software shall start giving the filtered data after 150ms. Different RS232 port shall be used for transmitting the filtered heading. It will also contain 8 bits of data with baud rate of 9600.