ACE

Advanced Customer

Evaluation Board

Description

Date: 2009 09 07

Revision: 2.0

By ARC, FSE, TCE

SensorDynamics AG

A 8403 Graz-Lebring, Schloß Eybesfeld 1e, Austria.

Tel: +43 (0) 3182 40160

Fax: +43 (0) 3128 40160 70

Email: info@sensordynamics.cc

Contents

1 Overview

The “press_demo” board is a vehicle for fast evaluation of SensorDynamics IMSS devices. It is targeted to be used for the following designs:

1. SD721: X-axis gyro, SOIC28 package
2. SD751: X-axis gyro, OC24 package
3. SD755: X-axis gyro / Y-axis accelerometer, OC24 package
4. SD777: X-axis gyro / XYZ-axes accelerometers, OC24 package
5. SD789: XY-axes gyros / XYZ-axes accelerometers, OC24 package

The concept allows the user to quickly access to the measured physical signals through digital interfaces (UART over USB and I2C) as well as quasi analog interfaces (PWM).

The board is designed for 5V power supply and -40/85℃ temperature range (SD IMSS devices can work up to -40/125℃, see specifications for detail s).

2 Board description

The following picture resumes the board configuration.

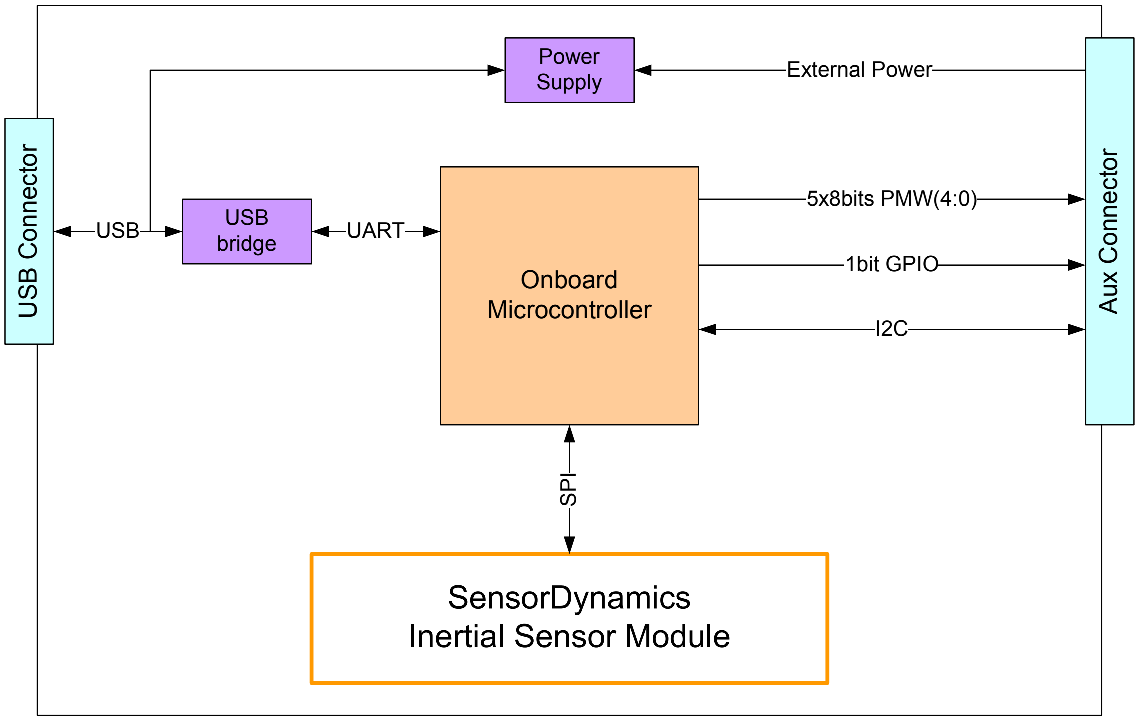

Figure 1: Board block diagram

2.1 SensorDynamics Inertial Sensor Module

This is the sensing element to detect angular rotations as well as axial accelerations (the measured physical signals depend on the used SensorDynamics device). The sensor module communicates via SPI interface.

2.2 Onboard Micro Controller

ATMEL ATMega88 micro controller is used to present the measured physical values over different outputs. Details on the running firmware are described in the following chapters. Detailed pin description is reported in the following picture.

Micro Controller Pin	Micro Controller Function			Connection at board level
PB7	XTAL2 (Chip Clock Oscillator pin 2)	TOSC2 (Timer Oscillator pin 2)	PCINT7 (Pin Change Interrupt 7)	nc
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input)	TOSC1 (Timer Oscillator pin 1)	PCINT6 (Pin Change Interrupt 6)	nc
PB5	**SCK (SPI Bus Master clock Input)**	PCINT5 (Pin Change Interrupt 5)		**SPI SCK**
PB4	**MISO (SPI Bus Master Input/Slave Output)**	PCINT4 (Pin Change Interrupt 4)		**SPI MISO**
PB3	**MOSI (SPI Bus Master Output/Slave Input)**	OC2A (Timer/Counter2 Output Compare Match A Output)	PCINT3 (Pin Change Interrupt 3)	**SPI MOSI**
PB2	SS (SPI Bus Master Slave select)	**OC1B (Timer/Counter1 Output Compare Match B Output)**	PCINT2 (Pin Change Interrupt 2)	**AUX Connector Pin9 (PWM4, 16bits)**
PB1	**OC1A (Timer/Counter1 Output Compare Match A Output)**	PCINT1 (Pin Change Interrupt 1)		**AUX Connector Pin8 (PWM3, 16bits)**
PB0	**ICP1 (Timer/Counter1 Input Capture Input)**	CLKO (Divided System Clock Output)	PCINT0 (Pin Change Interrupt 0)	**Uart RX (time measurement)**
PC6	**RESET (Reset pin)**	PCINT14 (Pin Change Interrupt 14)		reset
PC5	ADC5 (ADC Input Channel 5)	**SCL (2-wire Serial Bus Clock Line)**	PCINT13 (Pin Change Interrupt 13)	**AUX Connector Pin3 (I2C SDA)**
PC4	ADC4 (ADC Input Channel 4)	**SDA (2-wire Serial Bus Data Input/Output Line)**	PCINT12 (Pin Change Interrupt 12)	**AUX Connector Pin4 (I2C SCL)**
PC3	ADC3 (ADC Input Channel 3)	PCINT11 (Pin Change Interrupt 11)		**AUX Connector Pin2 (GPIO)**
PC2	ADC2 (ADC Input Channel 2)	PCINT10 (Pin Change Interrupt 10)		nc
PC1	ADC1 (ADC Input Channel 1)	PCINT9 (Pin Change Interrupt 9)		nc
PC0	ADC0 (ADC Input Channel 0)	PCINT8 (Pin Change Interrupt 8)		nc
PD7	AIN1 (Analog Comparator Negative Input)	PCINT23 (Pin Change Interrupt 23)		**SPI CSN**
PD6	AIN0 (Analog Comparator Positive Input)	**OC0A (Timer/Counter0 Output Compare Match A Output)**	PCINT22 (Pin Change Interrupt 22)	**AUX Connector Pin7 (PWM2, 8bits)**
PD5	T1 (Timer/Counter 1 External Counter Input)	**OC0B (Timer/Counter0 Output Compare Match B Output)**	PCINT21 (Pin Change Interrupt 21)	**AUX Connector Pin6 (PWM1, 8bits)**
PD4	XCK (USART External Clock Input/Output)	T0 (Timer/Counter 0 External Counter Input)	PCINT20 (Pin Change Interrupt 20)	nc
PD3	INT1 (External Interrupt 1 Input)	**OC2B (Timer/Counter2 Output Compare Match B Output)**	PCINT19 (Pin Change Interrupt 19)	**AUX Connector Pin5 (PWM0, 8bits)**
PD2	INT0 (External Interrupt 0 Input)	PCINT18 (Pin Change Interrupt 18)		nc
PD1	**TXD (USART Output Pin)**	PCINT17 (Pin Change Interrupt 17)		**Uart Tx (Communication)**
PD0	**RXD (USART Input Pin)**	PCINT16 (Pin Change Interrupt 16)		**Uart Rx (Communication)**

Figure 2: ATMega88 PIN associations

2.3 USB Connector

The board can be directly connected to PC via USB. The USB link is also used to supply the board.

2.4 Aux Connector

In addition to USB access, the board allows a quick and simple direct interface over the auxiliary connector. This interface includes:

1. 4V-6V external supply (needed when USB interface is not used)
2. Five 8 bits PWM outputs (5V)
3. One general purpose 5V output (GPIO)
4. I2C interface

The signals mapped on the PWM outputs as well as the GPIO depends on the board configuration; the board configuration can be freely defined by the user. Detailed pin description is reported in the following picture.

Pin	Name	Function	Micro Controller Pin
1	Ext Power	4-6V power supply	
2	D0	GPIO	PC3
3	D1	SDA	PC4
4	D2	SDL	PC5
5	D3	PWM0	PD3
6	D4	PWM1	PD5
7	D5	PWM2	PD6
8	D6	PWM3	PB1
9	D7	PWM4	PB2
10	GND	board ground	

Figure 3: Auxiliary connector pins

2.5 USB Bridge

SiliconLabs™ CP2102 Bridge is used to map the micro controller UART interface over USB. Thanks to this bridge, the PC sees a (virtual) COM port which can be accessed via standard serial terminal SW (i.e. Windows HyperTerminal).

USB drivers are included into the SD software pack but are also available on the SiliconLabs™ web page (www.silabs.com).

2.6 Power Supply

The board can be supplied with variable external sources from 4V to 6V via the USB or the auxiliary connectors. A stable 5V power signal is generated out of these variable sources thanks to a onboard 2-stages regulator: the first stage is a step up generator which brings the external power supply up to 6V; the second stage fixes this 6V source down to a clean 5V output.

3 Onboard Micro Controller Firmware

This chapter describes the firmware used for the onboard micro controller. A block diagram for the firmware architecture is drawn in the following picture.

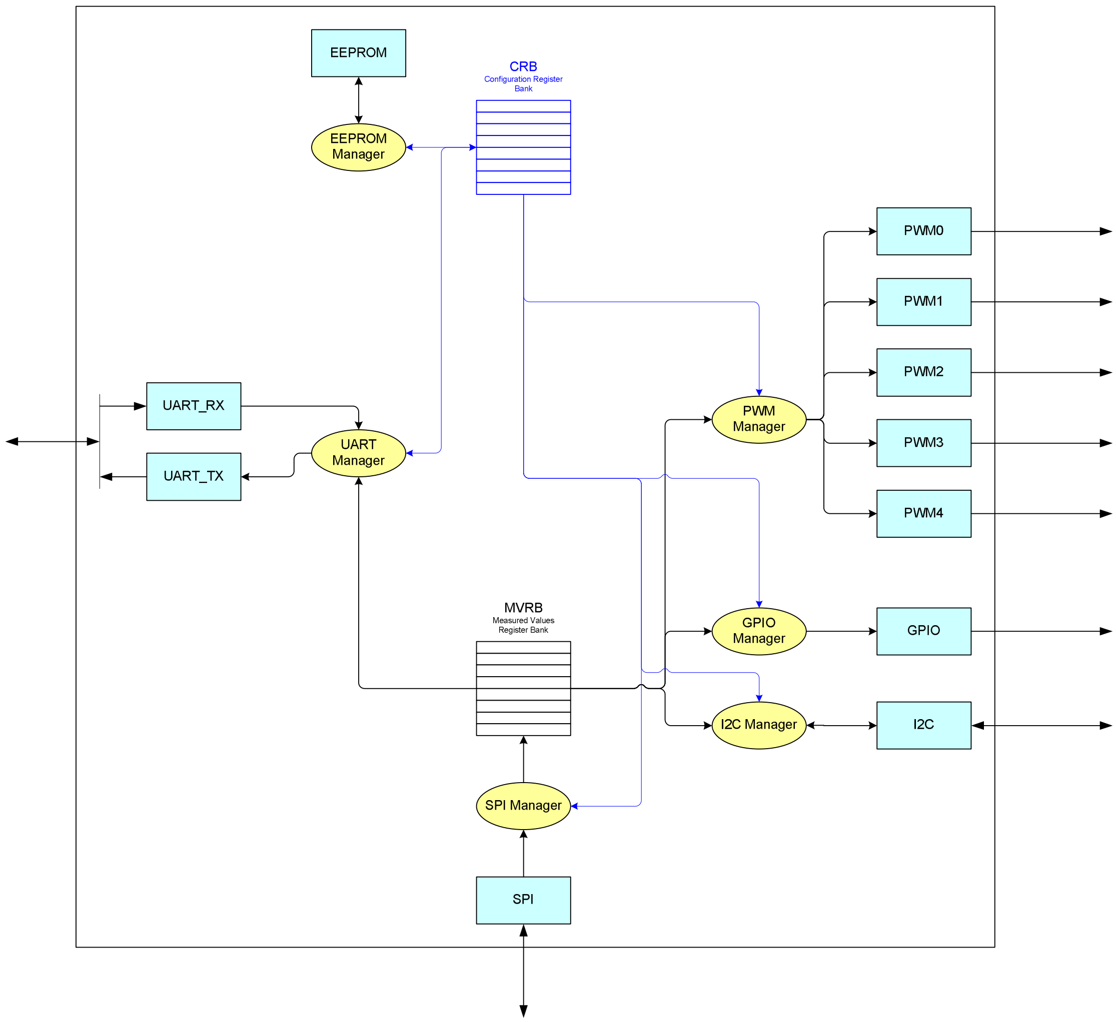

Figure 4: Firmware block diagram

As presented in the picture, the firmware architecture is based on several agents in charge of controlling micro controller hardware peripherals.

These agents get configuration parameters from 16 registers (each one 8 bit long), the “Configuration Register Bank” (CRB) which can be initialized from on chip EEPROM memory.

CBR can also be modified by user via UART (USB) interface for custom settings and stored back in the EEPROM for next power on cycles.

During normal operation, SPI manager communicates with SensorDynamics module to get measured physical signals; these values are stored into 16 registers (each one 16 bits long), the “Measured Value Register Bank” (MVRB) which can be read by other agents to drive board outputs.

Each agent provides an init routine (called once at the power up), a main routine (called periodically) and one or more interrupt routines (driven by interrupt sources of the micro controller peripherals).

3.1 MVRB

The Measured Values Register Bank is a 16 registers - 16 bits each, bank array. Its content is arranged as presented in the following table.

				SensorDynamics Modules				
Addr	**Parameter Name**	**Description**	**Register size / coding**	**SD721**	**SD751**	**SD755**	**SD777**	**SD789**
0x00	MR1 - Rate X	Angular velocity along X axis, measurement range 1	16 bits, 2'complement	±128°/s full scale 40Hz bandwidth	±128°/s full scale 40Hz bandwidth	±128°/s full scale 40Hz bandwidth	±128°/s full scale 40Hz bandwidth	±128°/s full scale 40Hz bandwidth
0x01	MR1 - Rate Y	Angular velocity along Y axis, measurement range 1	16 bits, 2'complement	NA	NA	NA	NA	±128°/s full scale 40Hz bandwidth
0x02	MR1 - Rate Z	Angular velocity along Z axis, measurement range 1	16 bits, 2'complement	NA	NA	NA	NA	NA
0x03	MR1 - Accelerometer X	Linear acceleration along X, measurement range 1	16 bits, 2'complement	NA	NA	NA	±6.2g full scale 40Hz bandwidth	±6.2g full scale 40Hz bandwidth
0x04	MR1 - Accelerometer Y	Linear acceleration along Y, measurement range 1	16 bits, 2'complement	NA	NA	±6.2g full scale 40Hz bandwidth	±6.2g full scale 40Hz bandwidth	±6.2g full scale 40Hz bandwidth
0x05	MR1 - Accelerometer Z	Linear acceleration along Z, measurement range 1	16 bits, 2'complement	NA	NA	NA	±6.2g full scale 40Hz bandwidth	±6.2g full scale 40Hz bandwidth
0x06	MR2 - Rate X	Angular velocity along X axis, measurement range 2	16 bits, 2'complement	±512°/s full scale 75Hz bandwidth	±512°/s full scale 75Hz bandwidth	±512°/s full scale 75Hz bandwidth	±512°/s full scale 75Hz bandwidth	NA
0x07	MR2 - Rate Y	Angular velocity along Y axis, measurement range 2	16 bits, 2'complement	NA	NA	NA	NA	NA
0x08	MR2 - Rate Z	Angular velocity along Z axis, measurement range 2	16 bits, 2'complement	NA	NA	NA	NA	NA
0x09	MR2 - Accelerometer X	Linear acceleration along X, measurement range 2	16 bits, 2'complement	NA	NA	NA	NA	NA
0x0A	MR2 - Accelerometer Y	Linear acceleration along Y, measurement range 2	16 bits, 2'complement	NA	NA	±13.4g full scale 100Hz bandwidth	NA	NA
0x0B	MR2 - Accelerometer Z	Linear acceleration along Z, measurement range 2	16 bits, 2'complement	NA	NA	NA	NA	NA
0x0C	Temperature	Temperature	16 bits, unsigned	1°C/LSB 80°C offset shift	1°C/LSB 80°C offset shift	1°C/LSB 80°C offset shift	1°C/LSB 80°C offset shift	1°C/LSB 80°C offset shift
0x0D	Status_L	Status info LSB	16 bits, unsigned	status	status	status	status	status
0x0E	Status_H	Status info MSB	16 bits, unsigned	status	status	status	status	status
0x0F	reserved	reserved	16 bits, unsigned	NA	NA	NA	NA	NA

Figure 5: MVRB layout

3.2 CRB

The Configuration Register Bank is a 16 registers, 8 bits each bank array. Its content is arranged as presented in the following table.

Address	Category	Register Name	Default Value	Register size / coding	Description
0x00	Reserved	Reserved	0x00	8 bits, unsigned	Reserved
0x01	Reserved	Reserved	0x00	8 bits, unsigned	Reserved
0x02	Reserved	Reserved	0x00	8 bits, unsigned	Reserved
0x03	EEPROM Manager	EEPROM_Status	0x00	8 bits, unsigned	0x00 = Not Valid / Do Nothing 0x01 = EEPROM configuration register bank successfully loaded 0x02 = Force a new EEPROM read 0x03 = Copy configuration register bank in EEPROM
0x04	I2C Manager	I2C_Address	0x00	8 bits, unsigned	Address for I2C telegrams
0x05	GPIO Manager	bit_mapping	0x1D	8 bits, unsigned	Which bit is mapped on GPIO output pin bits(3:0) = Address of MVRB. These bits select which register is used to define the value of the GPIO output pin. bits(7:4) = mapped bit. These bits select which bit in the selected register is used to define the value of the GPIO output pin.
0x06	PWM Manager	PWM0_cfg	0x0C	8 bits, unsigned	bits(3:0) = address of the selected register in MVRB which shall be used to drive the PWM signal. bits(6:4) = unused bits(7) = 0/1, use lower / higher 8 bits
0x07	PWM Manager	PWM1_cfg	0x0F	8 bits, unsigned	bits(3:0) = address of the selected register in MVRB which shall be used to drive the PWM signal. bits(6:4) = unused bits(7) = 0/1, use lower / higher 8 bits
0x08	PWM Manager	PWM2_cfg	0x0F	8 bits, unsigned	bits(3:0) = address of the selected register in MVRB which shall be used to drive the PWM signal. bits(6:4) = unused bits(7) = 0/1, use lower / higher 8 bits
0x09	PWM Manager	PWM3_cfg	0x0F	8 bits, unsigned	bits(3:0) = address of the selected register in MVRB which shall be used to drive the PWM signal. bits(6:4) = unused bits(7) = 0/1, use lower / higher 8 bits
0x0A	PWM Manager	PWM4_cfg	0x0F	8 bits, unsigned	bits(3:0) = address of the selected register in MVRB which shall be used to drive the PWM signal. bits(6:4) = unused bits(7) = 0/1, use lower / higher 8 bits
0x0B	SPI Manager	DeviceName	0x00	8 bits, unsigned	Defines which SPI protocol has to be used. 0x00 = SD721/751 0x01 = SD 755 0x02 = SD777
0x0C	UART Manager	OutputData_L	0x41	8 bits, unsigned	Each bit defines if a register in MVRB is an output when UART free running output mode is selected. Bits(i): 1/0 register with address "I" is/isn't present in UART output telegram.
0x0D	UART Manager	OutputData_H	0x70	8 bits, unsigned	Each bit defines if a register in MVRB is an output when UART free running output mode is selected. Bits(i): 1/0 register with address '8+I' is/isn't present in UART output telegram.
0x0E	FW_reg	Firmware Version	0x01	8 bits, unsigned	FW version (progressive number).
0x0F	CRC_reg	CRC	0x47	8 bits, unsigned	Sum modulus 255 of previous register then negated.

Figure 6: CRB layout

3.3 SPI manager

The SPI manager:

1. configures the SPI peripheral hardware and the PADs at power up;
2. manages the communication with the SensorDynamics module; protocol and measured physical signals depend on the used sensor device;
3. fills the MVRB according to the measured values

It also uses a real time interrupt RTI routine which defines the ticks for the telegrams transmission.

Notes: the update of MVRB registers is done with atomic operations (i.e. interrupt protected) to avoid the presence of inconsistent data which could lead to erroneous output signals.

3.3.1 Details on implementation

The firmware, on the basis of the chosen device, sends requests about its measurements and status in turn and then it updates the correspondent MVRB register.

3.4 UART manager

The UART manager configures the UART peripheral hardware and the PADs at power up with the following settings:

- Baud Rate: 38400
- Frame: 8 bits, 1 bit start, 1bit stop, no parity

During normal operation, UART manager carries out the following tasks:

1. **Write CRB**: this command allows writing custom value in a selectable CRB register. In order to execute this action, user shall send the command *WCRB*$\boldsymbol{H_1HH_2}$ (ASCII string, all capital letters, terminated with return character 0x0d), where $\boldsymbol{H_1}$ is a 1 digit Hex number representing the register address, and $\boldsymbol{HH_2}$ is a 2 digits Hex number representing the value to be written. A correct transaction terminates with the reception of the "*OK!*" string. Otherwise, the *"Error"* string is received.

2. **Read CRB**: this command allows reading the value contained in a selectable CRB register. In order to execute this action, user shall send the command *RCRB**H*** (ASCII string, all capital letters, terminated with return character 0x0d), where ***H*** is a 1 digit Hex number representing the register address to be read; the firmware replies with ASCII character representing the register content in hex format, followed by return character 0x0d. A correct transaction terminates with the reception of the requested value. Otherwise, the *"Error"* string is received.

3. **Change Baud Rate:** this command allows to change the baud rate of UART bus. In order to execute this action, user shall send the command *CHBR**n*** (ASCII string, all capital letters, terminated with return character 0x0d), where ***n*** is a number that identify the baud rate as follow:

   - ✓ ***n*** = 0 : 19200 baud
   - ✓ ***n*** = 1 : 38400 baud
   - ✓ ***n*** = 2 : 57600 baud
   - ✓ ***n*** = 3 : 115200 baud

   Hex number representing the value to be written. A correct transaction terminates with the reception of the "*OK!*" string BEFORE changing the baud rate. Otherwise, the *"Error"* string is received and the baud rate is kept unchanged.

4. **Start Free Running Mode**: in order to execute this action, user shall send the command *STAFRM* (ASCII string, all capital letters, terminated with return character 0x0d). When free running mode is enabled, the UART continuously outputs the selected measured values (OutputData_L and OuputData_H registers in CRB addresses 0x0C and 0x0D, define this selection); the output format is ***Val1,Val2, …,ValN,Chk*** terminated with return character 0x0d, where ***ValX*** is a 4 digits hex number representing the selected register value and ***Chk*** is a 2

digits hex number representing the 8bits sum of the output string (including character "**,**") negated;

5. **Start Fast Free Running Mode**: this modality (introduced in firmware version 2.0 and above) allows a fast reception of the main MVRB register values, depending on the selected device. In order to execute this action, user shall send the command *STAFFM* (ASCII string, all capital letters, terminated with return character 0x0d). When fast free running mode is enabled, the UART continuously outputs a stream of frames composed as follow:

   - ✓ **16 bit**: MR1 – Rate X (every devices)
   - ✓ **16 bit**: MR1 – Accelerometer X (only in SD777 device)
   - ✓ **16 bit**: MR1 – Accelerometer Y (only in SD755/SD777 devices)
   - ✓ **16 bit**: MR1 – Accelerometer Z (only in SD777 device)
   - ✓ **8 bit**: CRC (computed as complemented sum of full frame, byte stuff included)
   - ✓ **8 bit**: End of frame (EOF = 0x0d)

   When a ***0x0d*** byte is inside the frame, to avoid the conflict with the EOF a byte stuffing is made, inserting a ***0x0a*** byte before it. At the same way, when a ***0x0a*** byte is encountered, this byte is sent two times. Example (SD755):

   Frame: FC **0A** D3 **0D** DF 0D

   Frame with byte stuff: FC **0A 0A** D3 **0A 0D** DF 0D

6. **Start One Shot Mode**: in order to execute this action, user shall send the command *STAOSM* (ASCII string, all capital letters, terminated with return character 0x0d). The firmware replies with a frame containing the selected measured values (OutputData_L and OuputData_H registers in CRB addresses 0x0C and 0x0D, define this selection); the output format is ***Val1,Val2, …,ValN,Chk*** terminated with return character 0x0d, where ***ValX*** is a 4 digits hex number representing the selected register value and ***Chk*** is a 2 digits hex number representing the 8bits sum of the output string (including character "**,**") negated;

7. **Stop Free Running Mode**: this command allows to stop every Free Running Mode (both normal and fast). In order to execute this action, user shall send the command *STPFRM* (ASCII string, all capital letters, terminated with return character 0x0d). A correct transaction terminates with the reception of the "*OK!*" string. Otherwise, the *"Error"* string is received.

Notes:

1. free running mode is also disabled by writing 0 in OutputData_L and OuputData_H registers.
2. free running mode shall be disabled before issuing a read CRB command.
3. free running mode and one shot mode doesn't start if OutputData_L and OuputData_H registers are both equal to 0.
4. free running mode is disabled at power up (this set is different in firmware versions earlier than 2.0).

3.5 EEPROM manager

The EEPROM manager reads (stores) the CRB from (into) the on chip EPROM memory. The CRB is stores starting from EEPROM address 0x0000.

During power up, the EEPROM manager reads the stored CRB register from EEPROM and computes the checksum; if the computed checksum fits the stored checksum, the EEPROM content is copied in the CRB and EEPROM status is set to OK; if the computed checksum doesn't fit the stored checksum, CRB is initialized with default values.

During normal operation, EEPROM manager check the CRB register "EEPROM Manager Status" (Address 0x04): according to its content, EEPROM manager can perform a new EEPROM read or a copy of the current CRB content into EEPROM.

When a copy into EEPROM is performed, EEPROM status is set to "do nothing" and CRC is recalculated to be consistent with the CRB content.

3.6 PWM manager

The PWM manager is responsible to update the PWM generators with the most recent measured value. The user can decide which measured value is used to drive each PWM generator: this mapping is done thanks to PWM configuration registers in the CRB (addresses 0x07 to 0x0A).

The following table resumes the PWM resources.

Name	Pin	Micro Controller Function	Resolution	Frequency
PWM0	PD3	OC2B (Timer/Counter2 Output Compare Match B Output)	8 bits	31.3 KHz
PWM1	PD5	OC0B (Timer/Counter0 Output Compare Match B Output)	8 bits	31.3 KHz
PWM2	PD6	OC0A (Timer/Counter0 Output Compare Match A Output)	8 bits	31.3 KHz
PWM3	PB1	OC1A (Timer/Counter1 Output Compare Match A Output)	8 bits	31.3 KHz
PWM4	PB2	OC1B (Timer/Counter1 Output Compare Match B Output)	8 bits	31.3 KHz

Figure 7: PWM resources

3.6.1 Details on implementation

PWM are generated with on board micro controller waveform generators.

Main PWM manager routine just copies the selected MVRB registers to OCRnA/OCRnB registers; however MVRB are first processed as described below:

1. values are saturated to +/- 90% of the full range

2. values are offset so that 0 value give a PWM with 50% duty cycle (just inverting the MSBit)

3.7 GPIO manager

The GPIO manager is responsible to update the GPIO output bit. It simply consists on a main routine which set the PC3 pin according to the content of mapping configuration register (CRB, address 0x06).

3.8 I2C manager (implemented on demand)

The I2C manager allows the user to access the MVRB with I2C interface, available on auxiliary connector.

The board acts as an I2C slave device: external hosts can read any register in the MVRB using the following protocol.

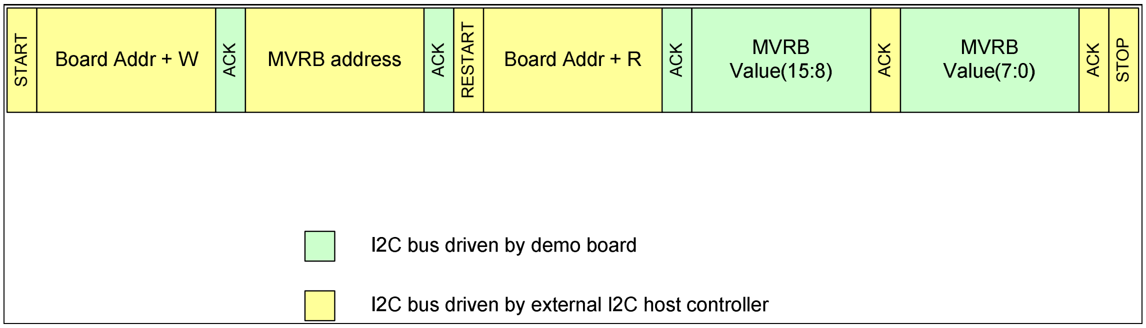

Figure 8: I2C protocol

1. External host controller starts the communication and sends address packet with write flag on
2. if the received address matches the board I2C address (CRB, address 0x05), the board replies with an acknowledge bit
3. external host sends the address of the register in the MVRB which as to be read; only bits 3:0 are considered: bits 7:4 are ignored
4. the board replies with an acknowledge bit
5. external host issues a restart command and repeats the address packet with read flag on
6. if the received address matches the board I2C address (CRB, address 0x05), the board replies with an acknowledge bit
7. the board sends the MSByte of the selected register
8. external host replies with an acknowledge
9. the board sends the LSByte of the selected register
10. external host replies with an acknowledge and closes the communication with a stop command.

3.8.1 Details on implementation

We are sorry but actually these details are not available.

4 PC demo application

The companion software on PC side communicates via USB with the micro controller. It is a GUI to allow the user to quickly set and test the sensor.

Mainly, into the software GUI we can identify three different windows:

- Configuration
- Acquisition
- Display

4.1 Configuration window

The application starts with configuration window where the user can set several fields to customize sensor evaluation. We can find:

1. *Device Name*: not editable field. It shows the on-board sensor type.
2. *FW Version*: not editable field. It shows the firmware version.
3. *I2C Address*: editable field where to write the decimal number corresponding to the address for I2C telegram reception (from 1 to 127).
4. *GPIO Bit Mapping*: two editable fields; the first corresponds to the signal to send via the on-board GPIO pins and the second is needed for choosing the bit where the GPIO output pin value is defined.
5. *PWM # Config*: two editable fields for each one of the possible 5 PWMs; the user can choose the desired signal to transmit, specifing between most or less significant byte of the relative register.
6. *Uart Frame Config*: pressing the button, a pop-up window allows the user to choose the output to transmit via UART during the free running mode, between all the possible ones. The choice is made by marking checkbox, as shown in Figure 9.

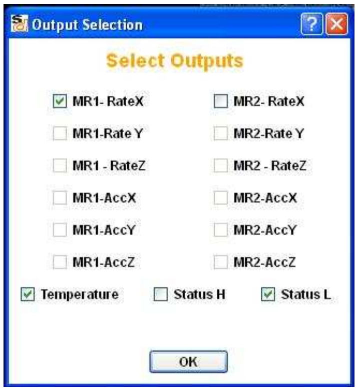

Figure 9: Output selection window

We have also four buttons:

1. *Write Cfg*: button to be pushed to write the chosen configuration into the board.

2. *Read Cfg*: button to be pushed to read the actual configuration present into the board.

3. *Return to Default Values*: if pressed, the configuration menu backs to default sensor values.

4. *Connect/Disconnect*: button to be pressed to establish or interrupt board connection. Every time the user tryes an operation with the board not connected, an error pop-up window appears.

In this window there are present the file menu, the view menu and the info one. The first contains the load and save configuration function. Each configuration can be stored and recalled as a *.arc* file. The default folder containing user configuration is called "configuration_files". The second is to view designers' reserved advanced features (a password is required, ask Sensor Dynamics to have more informations). The third contains helpful infos as an online guide, the software version and Sensor Dynamics contacts.

Figure 10 shows the configuration window.

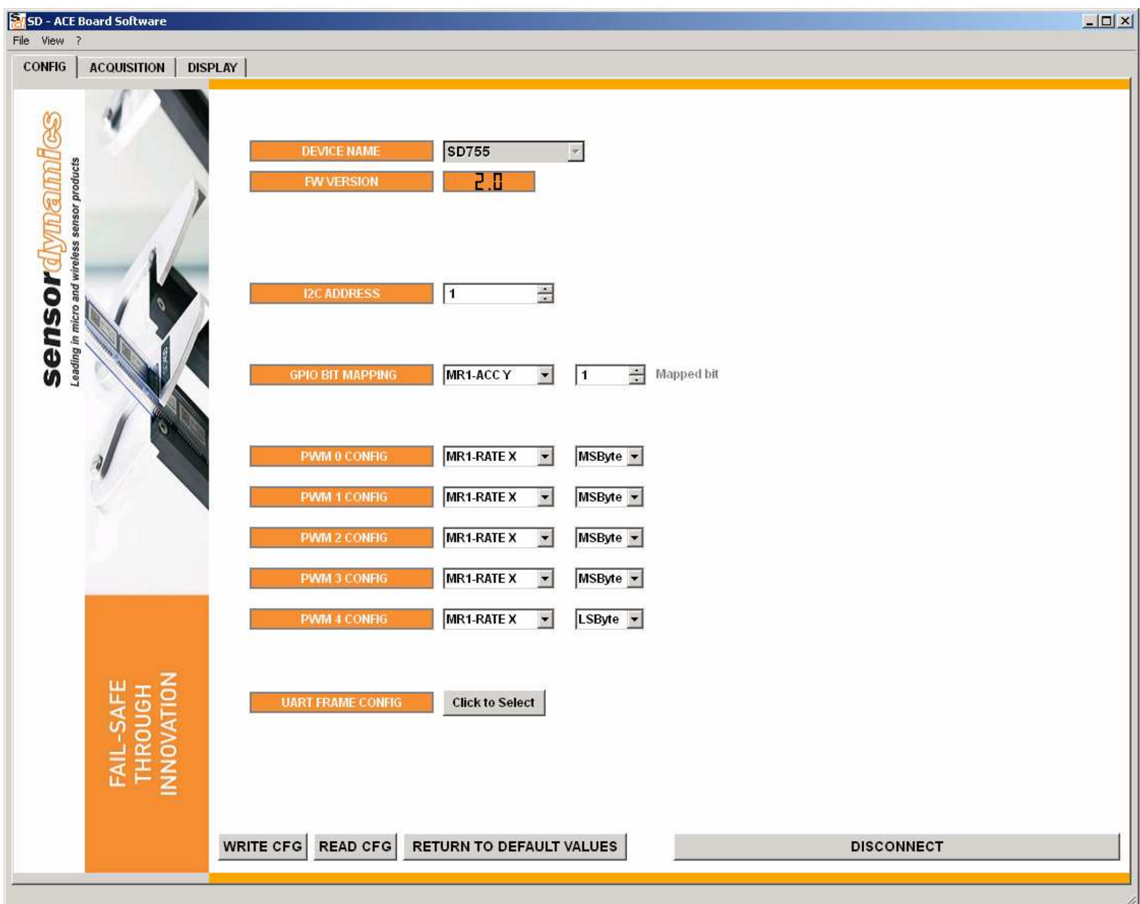

Figure 10: configuration window

4.2 Acquisition window

The acquisition window permits the user to:

- save sampled data in a log file;
- analyze some critical parameters associated with sampled data;
- visualize alarms status in real time

This window is divided in three blocks, as showed in Figure 11.

The group box (1) offers the possibility to save the sampled data in a chosen file. The saved file could be an excel or a text file type, and stores the information in *n* columns, where *n* is the number of fields checked in the *Uart Frame Config* pop-up window.

Note that, if the configuration has not been written in the device, the window will store the data on the basis of the values contained in the device.

The field *Filename* contains the chosen file to save, which could be chosen more easily pushing the *Browse* button.

The group box, also, permits also to choice the decimal separator used to save the decimal data. This permits not to change the international configuration of the Excel reader application.

To start the logging activity you must push the "*Start*" button in the group box.

The group box (2) permits to analyze some critical parameters extracted from the sampled data. It evaluates the mean, the noise (evaluated as Standard Deviation) and the peak to peak amplitude. These parameters are obtained considering a set of samples included in a mobile window of 100 samples. This box is not enabled by default: to show it, it is necessary to check the "*Show Rate/Acc Parameters*" in the group box (3).

The group box (3) permits to visualize the status of each alarm. It is possible to select which alarms to show pushing the "*Select alarms…*" button.

To start the latter two activities you need to push the "*Start Acquisition*" button in the group box (3). Both activities run simultaneously, differently to the first one that runs independently, because it necessitates to write in a file, so it is conceptually different.

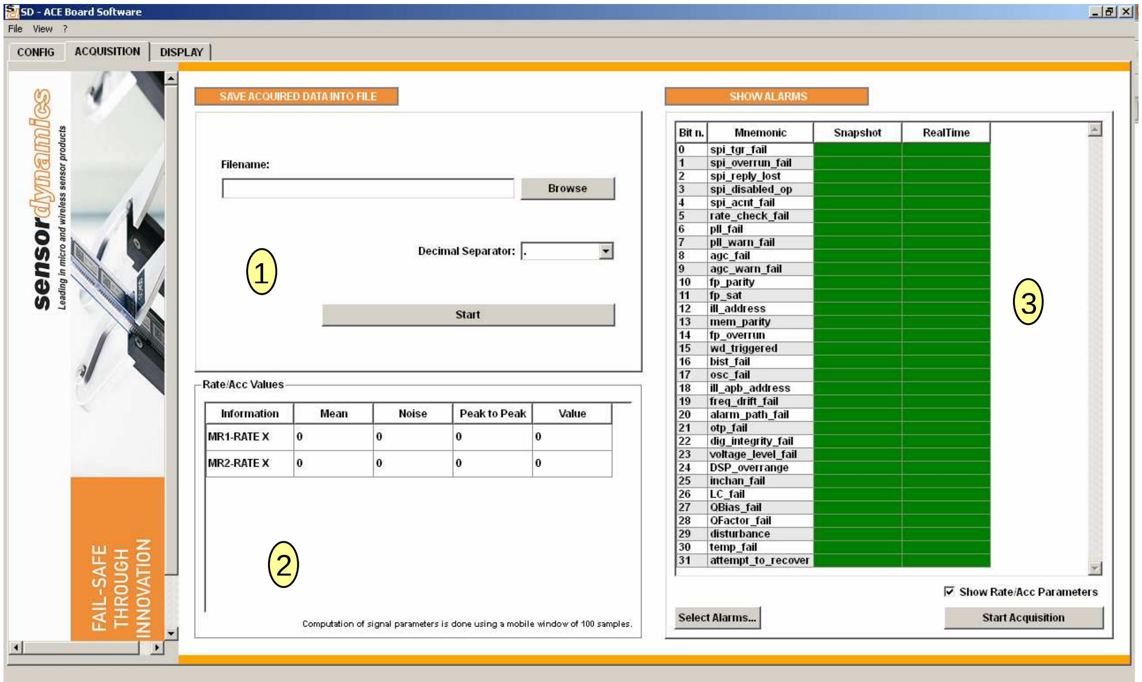

Figure 11: Acquisition window

Following figures show each groupbox in detail.

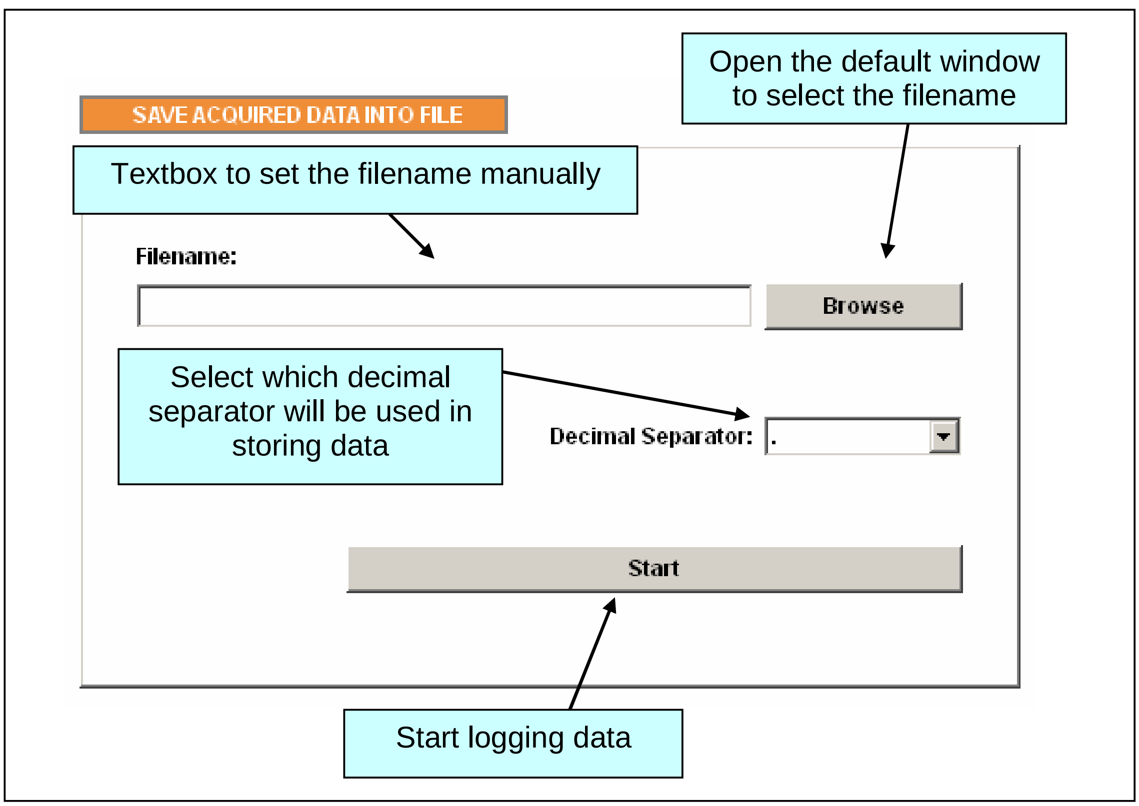

Figure 12: Groupbox 1

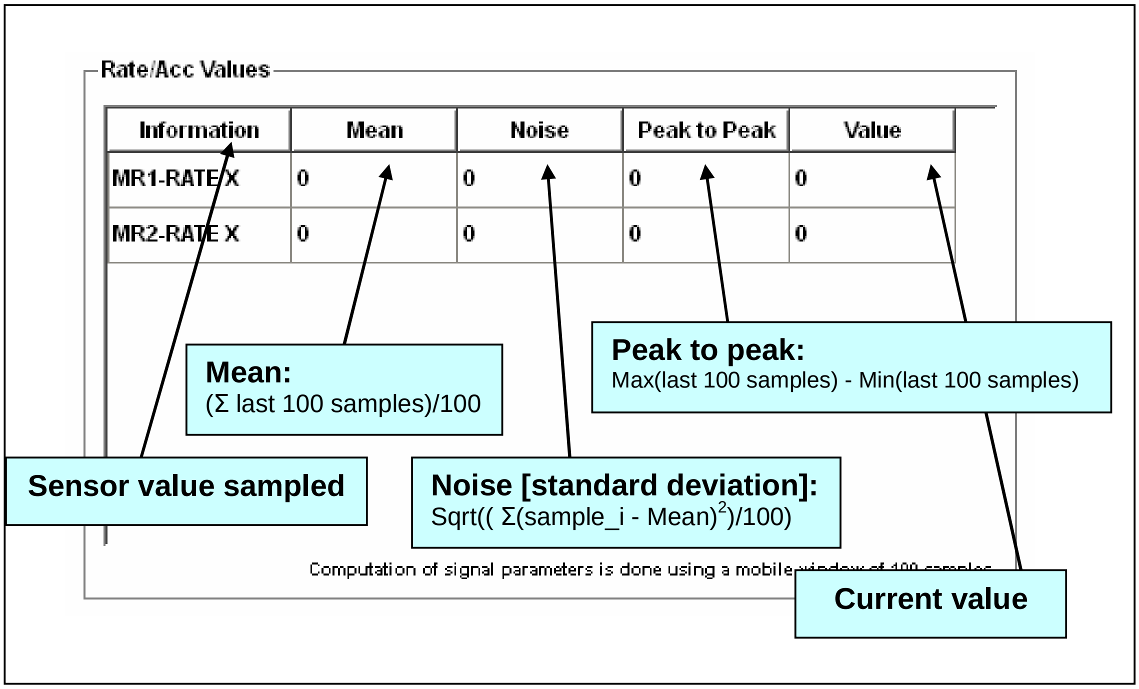

Figure 13: Groupbox 2

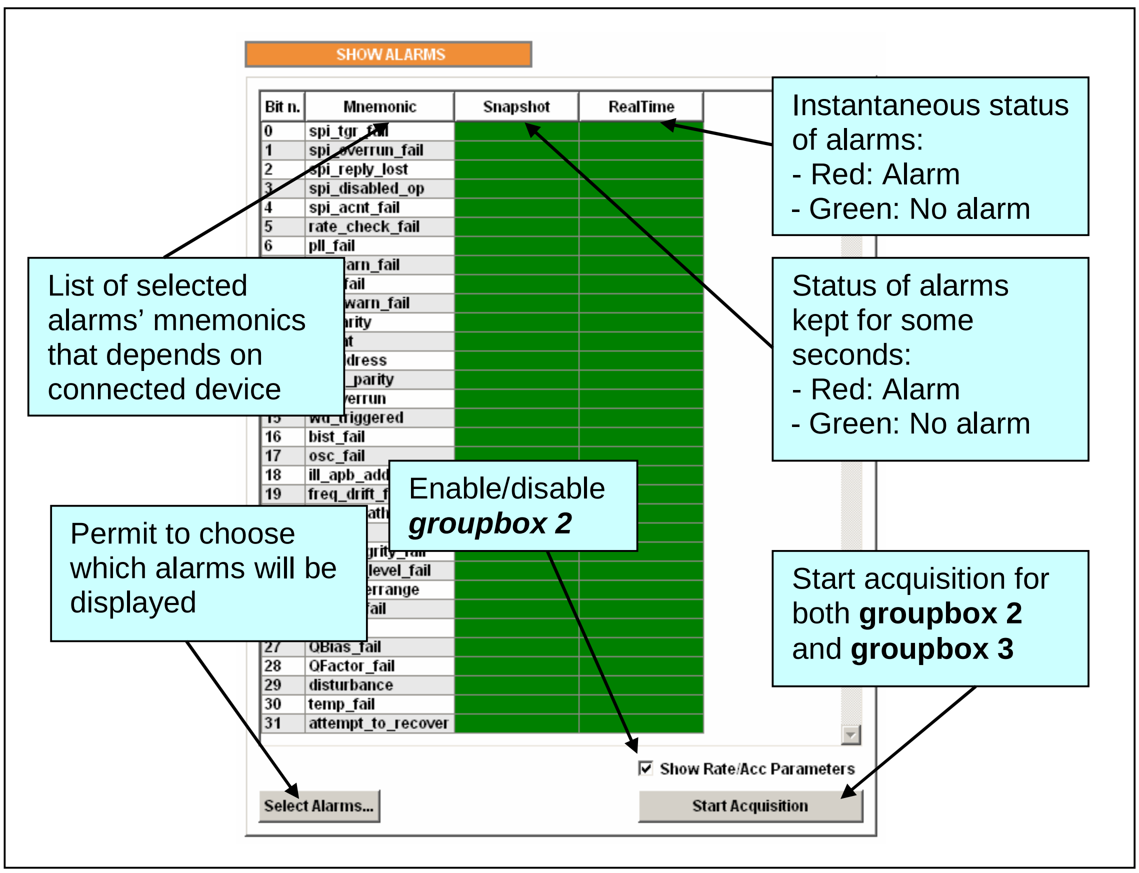

Figure 14: Groupbox 3

4.3 Display window

The display window has the main purpose to show the measured values in graphical way. The first plot contains the rates and the second the accelerations. Temperature measured and sensor status are indicate at the bottom of the window.

The look depends on the used sensor device (how many plots, scale…). When entering in this application status, the application programs the free running mode to output all interesting physical signals (rate/acc) plus temperature and status. Each plot can viewed or not by marking the relative check-box. The “Zoom In” boxes set the auto-scale on the actual displayed values. With the run/pause button the user can choose to interrupt the data flow.

Figure 15 shows the display window.

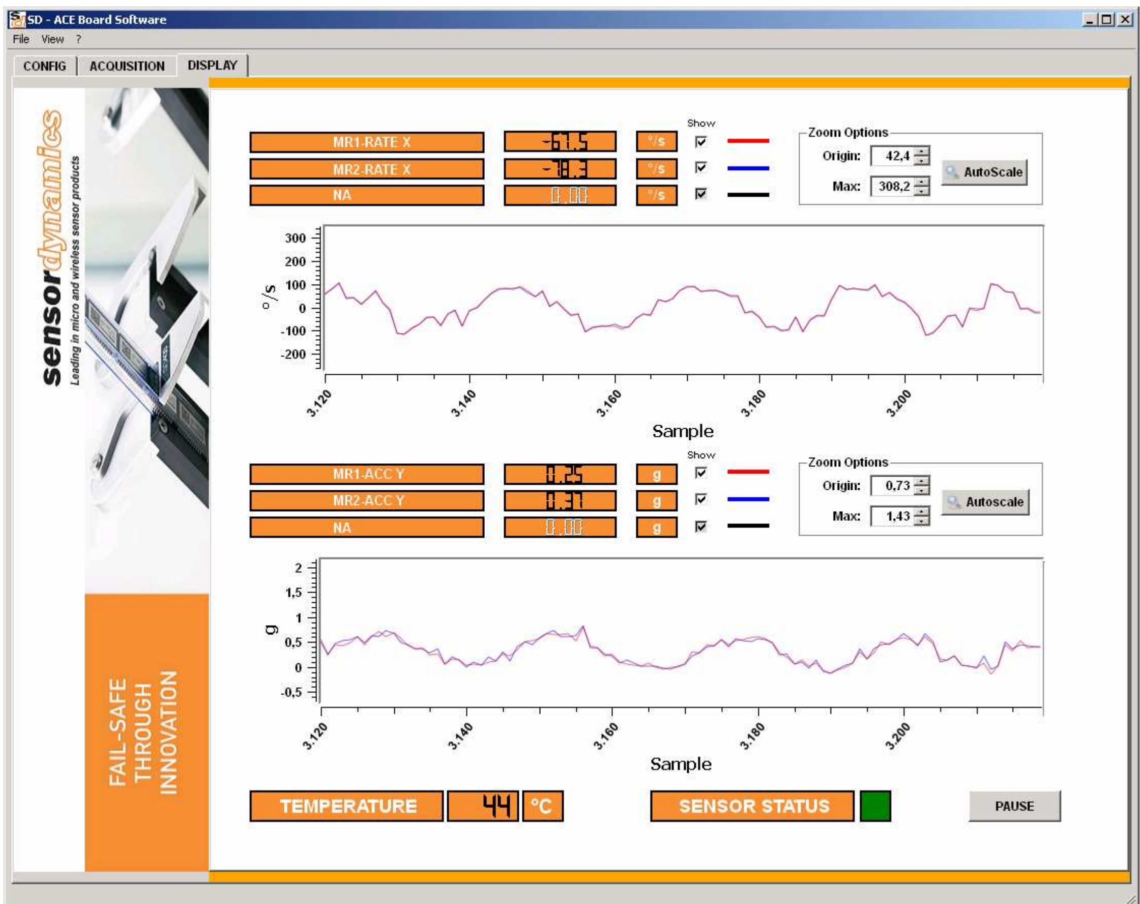

Figure 15: Display window

Finally, the following tables resume the measured values and the maximum ratings for each device.

	SD721		SD751		SD755		SD777		SD789	
	Signal Name	conversion factor	Signal Name	conversion factor	Signal Name	conversion factor	Signal Name	conversion factor	Signal Name	conversion factor
Rate1	MR1 - Rate X	1/256	MR1 - Rate X	1/256	MR1 - Rate X	1/256	MR1 - Rate X	1/256	MR1 - Rate X	1/256
Rate2	MR2 - Rate X	1/64	MR2 - Rate X	1/64	MR2 - Rate X	1/64	MR2 - Rate X	1/64	MR1 - Rate Y	1/256
Rate3	not visible	-	not visible	-	not visible	-	not visible	-	not visible	-
Acc1	not visible	-	not visible	-	MR1 - Accelerometer Y	1/4828	MR1 - Accelerometer X	1/4828	MR1 - Accelerometer X	1/4828
Acc2	not visible	-	not visible	-	MR2 - Accelerometer Y	1/2414	MR1 - Accelerometer Y	1/4828	MR1 - Accelerometer Y	1/4828
Acc3	not visible	-	not visible	-	not visible	-	MR1 - Accelerometer Z	1/4828	MR1 - Accelerometer Z	1/4828

	SD721	SD751	SD755	SD777	SD789
Rate Plot Scale	±512°/s	±512°/s	±512°/s	±512°/s	±128°/s
Acc Plot Scale	-	-	±15g	±7g	±7g

Figure 16: Measured values and the maximum ratings for each device