

Project	VIBRO CONTROL BOARD
Title	<b>FUNCTIONAL SOFTWARE SPECIFICATION</b>
Doc. Id.	ASMR&D038
Revision	01
Issue date	31 January 2020
Author	Maurizio Bonesi, Marco Toschi
Approver	Marco Toschi

The documentation is © Sesotec ASM S.r.l., which holds all the rights of reproduction, disclosure and any other rights identified by current legislation on copyright. Assigned, copied, distributed or reproduced, nor quoted, summarized, or modified, even separated, without the explicit consent of Sesotec ASM S.r.l.

## REVISION HISTORY

Rev.	Date	Paragraph	Description
01	31 January 2020		First issue

## INDEX

1	INTRODUCTION.....	4
1.1	Reference Documents .....	4
1.2	Reference Standards .....	4
2	OVERVIEW.....	5
2.1	Scope and Key Objectives.....	5
2.2	Functional Block Diagram .....	6
3	FUNCTIONS .....	7
3.1	Working Modes .....	7
3.2	Finite State Machine for Automated Flow Control .....	8
3.3	Alarms.....	9
4	INTERFACES.....	10
4.1	Interface with Other Systems .....	10
4.2	Interface Protocol.....	10
4.2.1	Holding Registers .....	11
4.2.2	Input Registers.....	11
5	GLOSSARY .....	12
6	ANNEXES .....	13
6.1	Annex 1.....	13

## 1 INTRODUCTION

This document was generated under the authority of the Sesotec ASM S.r.l. company, for the purpose of developing the software that get control of Vibro Control Board.

### 1.1 Reference Documents

Document Code	Document Title

### 1.2 Reference Standards

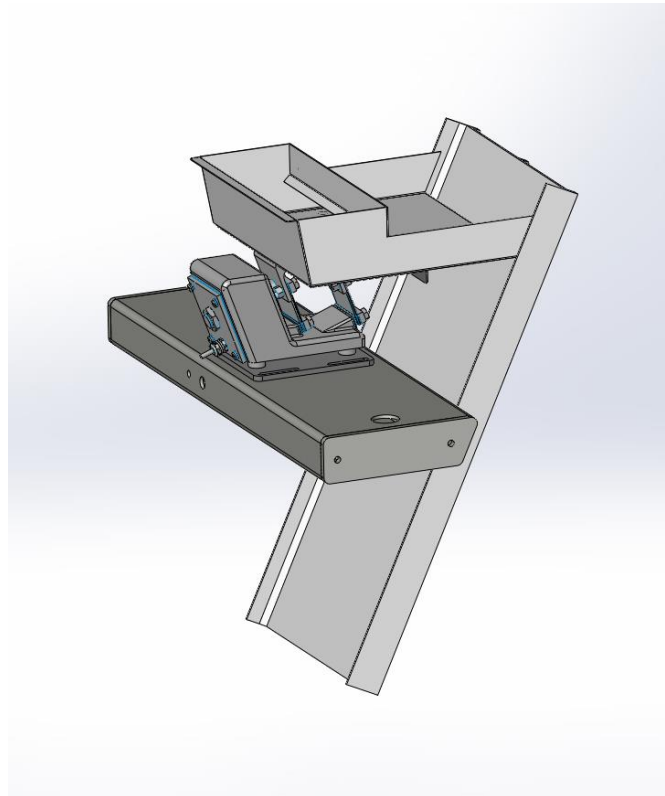
Document Code	Document Title

## 2 OVERVIEW

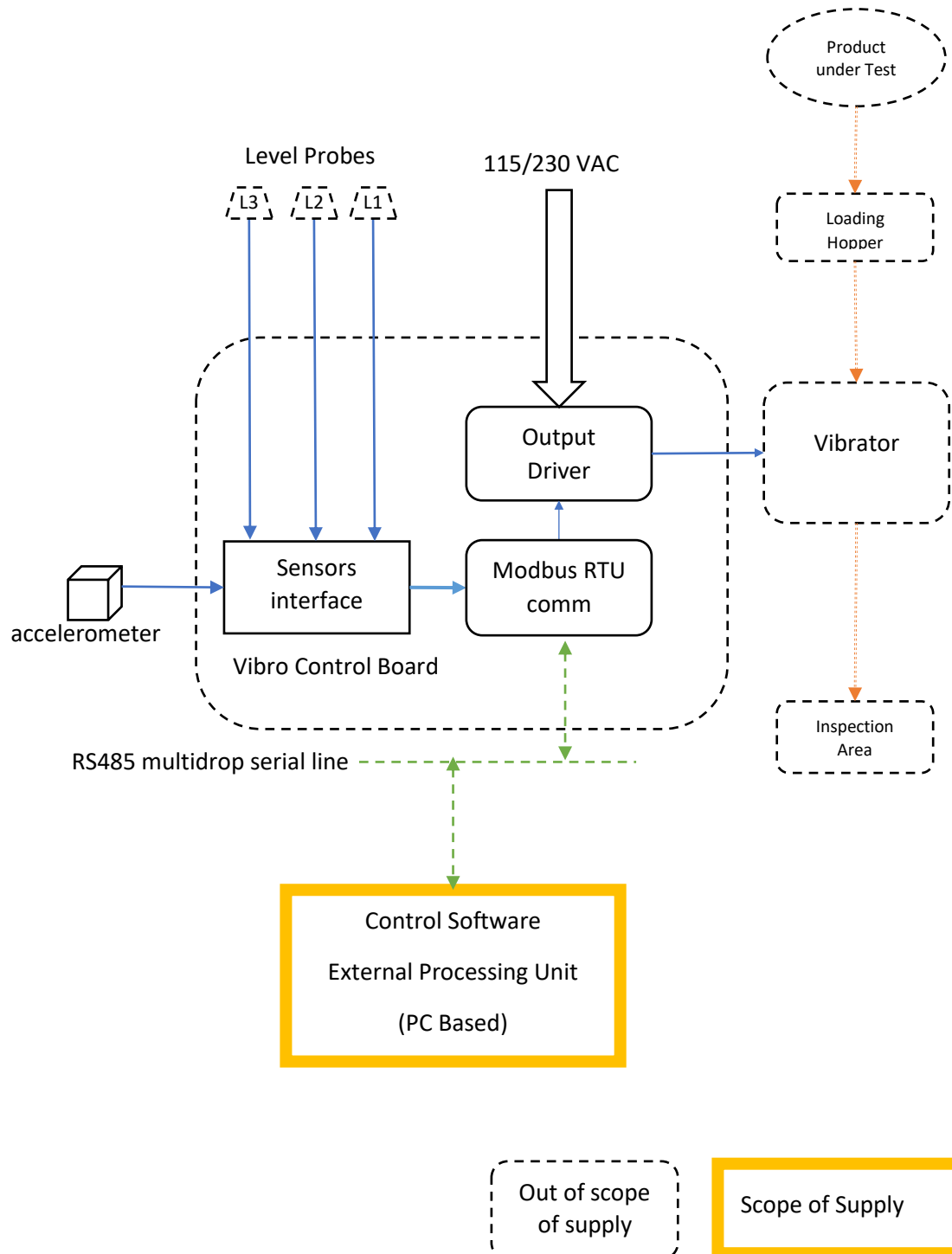
The system consists on an electronic board that will be used to drive an external vibrating device.

### 2.1 Scope and Key Objectives

The system will provide proper current amplitude values to achieve a desired product flow rate on the tray connected to the vibrating device.



## 2.2 Functional Block Diagram



### 3 FUNCTIONS

Code	Function
F01	<b>Constant Flow Delivery</b> The system will deliver a constant product flow regardless of the weight load that insists on the vibrating tray.
F02	<b>Similar Flow on Different Chutes</b> The system will deliver similar product flow on chutes belonging to the same section that have been configured with the same setpoint.
F03	<b>Proportional Flow as Chutes Width Vary</b> The system will deliver proportional product flow on chutes when they are installed in different sizes: $\frac{1}{4}$ , $\frac{1}{2}$ , $\frac{3}{4}$ , 1 on the same section. This will be accomplished via correction factor that will scale the setpoint as needed.

#### 3.1 Working Modes

Code	Function
F04	<b>Manual Flow Control</b> The system will allow the user to set the flow between: <ul style="list-style-type: none"> <li>a minimum value of 0% (no product flows out);</li> <li>a maximum value of 100% (product flows out at full speed).</li> </ul> The feature will be enabled by default. The user can select the operating mode.
F05	<b>Automated Flow Control</b> The system will be equipped with n. 3 level probe sensors (L1, L2, L3) which will be feed to the algorithm that will control the vibration setpoint %. The user will be able to enable/disable it at any time.

### 3.2 Finite State Machine for Automated Flow Control

Code	Function
F06	<p><b>Finite State Machine</b></p> <p>When set to operate in Automatic flow control mode, the system will implement FSM to compute the proper value for the output signal to be fed to the vibrating unit.</p> <p>The FSM will read the level probes sensors (L1, L2, L3) at regular intervals and compute the necessary output value to guarantee product flow control, also taking into consideration algorithm configuration parameters as defined in the following section.</p> <p>The full states transition map and pseudo-C control code for the automatic flow control FSM is reported in <i>Table 1 - FSM states transitions map</i>.</p>
F07	<p><b>Configuration Parameter</b></p> <ul style="list-style-type: none"> <li>interval: the time interval at which the algorithm is fed (input sampled, FSM processed, outputs driven);</li> <li>dly: number elapsed intervals;</li> <li>setpoint: the user specified setpoint for vibration control (0÷100%);</li> <li>step: the increment/decrement step when increasing/decreasing output value;</li> <li>variation: a fixed amount that is added/subtracted to the output value in case coarse variation in the output value is needed.</li> </ul>
F08	<p><b>Internal Parameters</b></p> <ul style="list-style-type: none"> <li>currState: the current FSM state {A, B, C, D, E, F, G};</li> <li>nextState: the state that the FSM is set to and will be used on next iteration {A, B, C, D, E, F, G};</li> <li>counter: used internally to measure the number of time intervals that one state persists.</li> </ul>



Code	Function																		
F09	<p><b>Input Parameters</b></p> <ul style="list-style-type: none"> <li>level: a numerical mapping of the probes readout levels according to the following table</li> </ul> <table> <tr> <th>Probes Readout (L3 L2 L1)</th><th>Level</th></tr> <tr><td>000</td><td>0</td></tr> <tr><td>001</td><td>1</td></tr> <tr><td>010</td><td>2</td></tr> <tr><td>011</td><td>2</td></tr> <tr><td>100</td><td>3</td></tr> <tr><td>101</td><td>3</td></tr> <tr><td>110</td><td>3</td></tr> <tr><td>111</td><td>3</td></tr> </table>	Probes Readout (L3 L2 L1)	Level	000	0	001	1	010	2	011	2	100	3	101	3	110	3	111	3
Probes Readout (L3 L2 L1)	Level																		
000	0																		
001	1																		
010	2																		
011	2																		
100	3																		
101	3																		
110	3																		
111	3																		
F10	<p><b>Output Parameters</b></p> <ul style="list-style-type: none"> <li>output: the variation value that is applied to the vibrator control system (0÷100%)</li> </ul>																		

### 3.3 Alarms

Code	Function
F11	<p><b>Alarms</b></p> <p>The system will manage the following alarms via an internal management system and reporting strategy:</p> <ul style="list-style-type: none"> <li>Power fault;</li> <li>Feeder accelerometer error;</li> <li>Faulty feeder;</li> <li>Level probes fault: odd configurations of three-probes setups will be detected (e.g.: if the rightmost signal is the machine infeed one, configurations like 101, 100, 110 will be treated as probes system failures).</li> </ul>

## 4 INTERFACES

### 4.1 Interface with Other Systems

Code	Function
F12	<b>Serial Port</b> The system will be equipped with n. 2 RS485 terminal blocks for daisy-chain connections and configuration for termination resistor on one port.
F13	<b>Level Probes</b> The system will be equipped with a terminal block to connect with n. 3 sensors (L1, L2, L3) via dry contact interface.
F14	<b>Synchronization</b> If multiple systems are installed on the same machine and the vibrator frequency has been setup to grid frequency (50 or 60 Hz), they will synchronize their output signal with the rising edge of the input AC voltage.

### 4.2 Interface Protocol

Code	Function
F15	<b>Modbus RTU</b> Communication protocol will be based on Modbus RTU, configured as slave.
F16	<b>Address</b> The system will be equipped with hardware rotary switches to set the node address.

#### 4.2.1 Holding Registers

Register	Address (hex)	Register Type	Description	
OpMode	0x1000	1 word	0	Manual
			1	Automatic
Control	0x1001	1 word	Vibration setpoint from 0 to 100.	
TrayWidth	0x1002	1 word	Tray dimension	

#### 4.2.2 Input Registers

Register	Address (hex)	Register Type	Description	
Status	0x4000	1 word	Current vibration setpoint	
Probes Status	0x4001	1 word	Probes bit mask	
Alarms <sup>1</sup>	0x4002	1 word	0	No Alarm
			1	Power section fault
			2	Accelerometer fault
			4	Feeder fault
			8	Level Probes Fault

<sup>1</sup> The alarm signal can be OR'ed to indicate the presence of multiple failure.

## 5 GLOSSARY

Term	Description
FSM	Finite State Machine

## 6 ANNEXES

### 6.1 Annex 1

level	currState						
	A	B	C	D	E	F	G
0	counter=0; nextState=A; output=0;	counter=0; nextState=A; output=0;	counter=0; nextState=A; output=0;	counter=0; nextState=A; output=0;	counter=0; nextState=A; output=0;	counter=0; nextState=0; output=0;	counter=0; nextState=A; output=0;
1	counter=0; nextState=B; output=0;	counter++; if (counter>dly) { nextState=C; }	output=setpoint- variation;	output=output-step; if (output<(setpoint- variation)) { output=(setpoint- variation); nextState=C; }	output=setpoint; counter=0; nextState=D;	nextState=C;	output=setpoint+variation; nextState=D;
2	counter=0; nextState=B; output=0;	counter++; if (counter>dly) { nextState=C; }	counter=0; nextState=D; output=setpoint- variation;	output=output+step; if (output>setpoint) { output=setpoint; nextState=E; }	output=setpoint;	output=output-step; if (output<setpoint) { output=Setpoint; nextState=E; }	output=setpoint+variation; nextState=F;
3	counter=0; nextState=B; output=0;	counter++; if (counter>dly) { nextState=C; }	counter=0; nextState=E; output=setpoint- variation;	counter=0; nextState=F;	output=setpoint; counter=0; nextState=F;	output=output+step; if (output>(setpoint+variation)) { output=(setpoint+variation); nextState=G; }	output=setpoint+variation;

Table 1 - FSM states transitions map