Project	VIBRO CONTROL BOARD
Title	**FUNCTIONAL SOFTWARE SPECIFICATION**
Doc. Id.	ASMR&D038
Revision	02
Issue date	18 March 2020
Author	Maurizio Bonesi, Marco Toschi
Approver	Marco Toschi

REVISION HISTORY

Rev.	Date	Paragraph	Description
01	31 January 2020		First issue
02	18 March 2020	§3	F1 updated
		§3.1	F17 added
		§4.1	F12 updated, F13 removed
		§4.2	F15, F16 updated
		§4.2.1	Updated
		§4.2.2	Updated

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 Slave Holding Registers 11
4.2.2 Slave 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

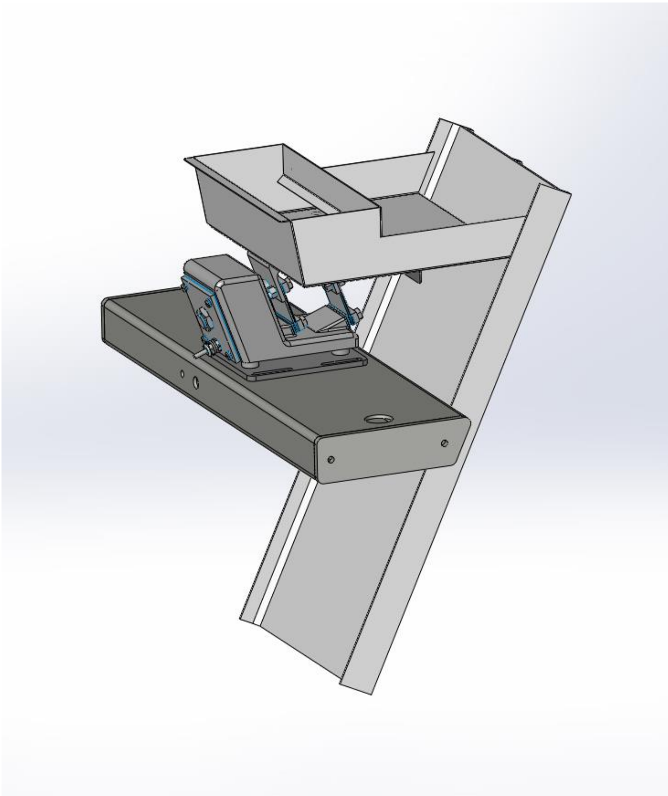
Document Code	**Document Title**

2 OVERVIEW

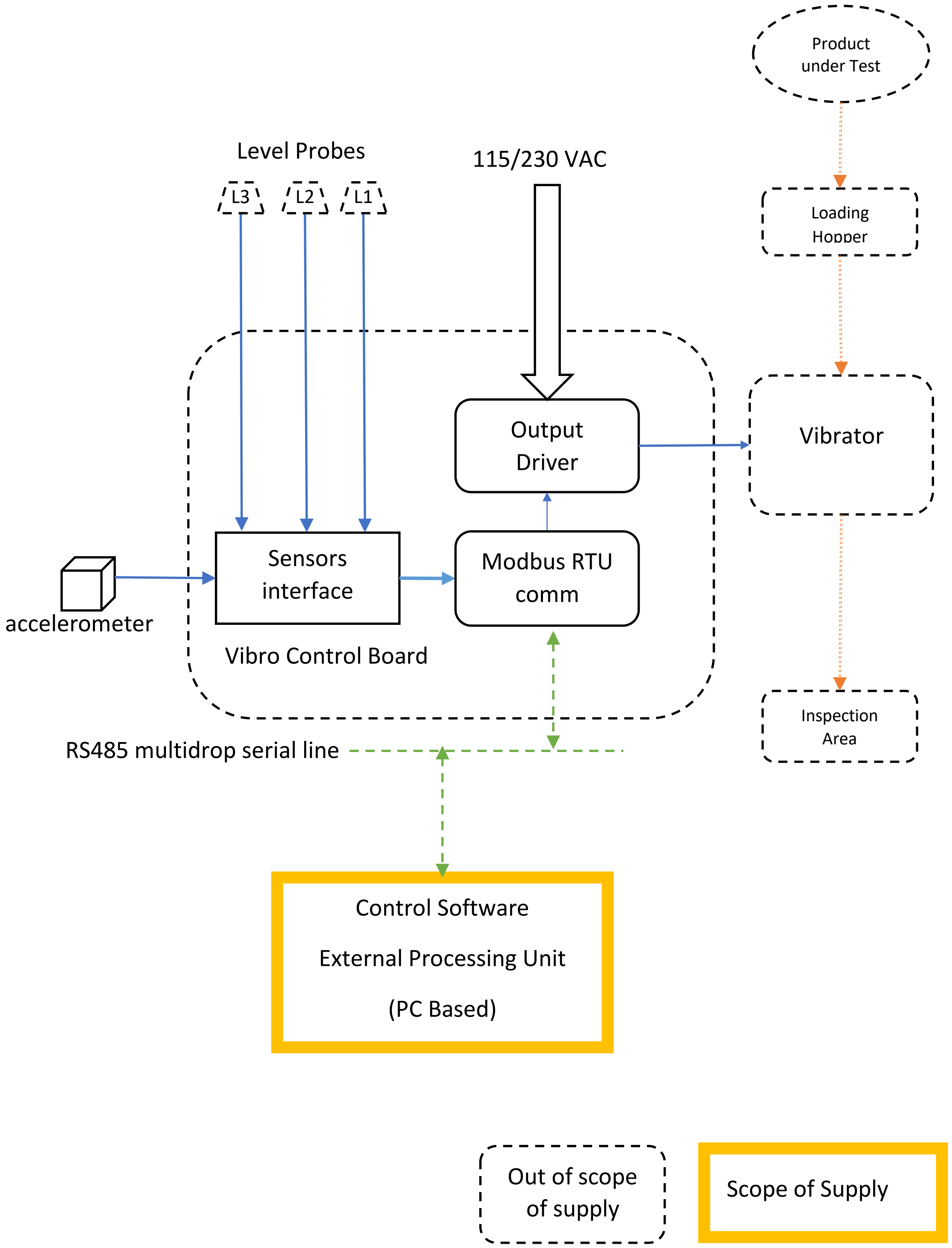
The system consists on an electronic board that will be used to drive an external vibrating device.

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	**Constant Flow Delivery** The system will deliver a constant product flow regardless of the weight load that insists on the vibrating tray, by tracking the RMS acceleration value.
F02	**Similar Flow on Different Chutes** The system will deliver similar product flow on chutes belonging to the same section that have been configured with the same setpoint.
F03	**Proportional Flow as Chutes Width Vary** The system will deliver proportional product flow on chutes when they are installed in different sizes: ¼, ½, ¾, 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
F17	**Resonance Frequency Tuning** The system will allow the user to set the optimal vibrating frequency by: 1. executing a sequence of vibration trials from 40 to 240 Hz; 2. calculating the maximum acceleration RMS value within that range. The results will be plot on a graph.
F04	**Manual Flow Control** The system will allow the user to set the flow between: • a minimum value of 0% (no product flows out); • a maximum value of 100% (product flows out at full speed). The feature will be enabled by default. The user can select the operating mode.

Code	Function
F05	**Automated Flow Control** 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	**Finite State Machine** 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. 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. The full states transition map and pseudo-C control code for the automatic flow control FSM is reported in *Table 1 - FSM states transitions map*.
F07	**Configuration Parameters** • interval: the time interval at which the algorithm is fed (input sampled, FSM processed, outputs driven); • dly: number elapsed intervals; • setpoint: the user specified setpoint for vibration control (0÷100%); • step: the increment/decrement step when increasing/decreasing output value; • variation: a fixed amount that is added/subtracted to the output value in case coarse variation in the output value is needed.
F08	**Internal Parameters** • currState: the current FSM state {A, B, C, D, E, F, G}; • nextState: the state that the FSM is set to and will be used on next iteration {A, B, C, D, E, F, G}; • counter: used internally to measure the number of time intervals that one state persists.

<table>
<tr><th>Code</th><th>Function</th></tr>
<tr><td>F09</td><td>Input Parameters
level: a numerical mapping of the probes readout levels according to the following table
<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>
</td></tr>
<tr><td>F10</td><td>Output Parameters
output: the variation value that is applied to the vibrator control system (0÷100%).
</td></tr>
</table>

3.3 Alarms

<table>
<tr><th>Code</th><th>Function</th></tr>
<tr><td>F11</td><td>Alarms
The system will manage the following alarms via an internal management system and reporting strategy:

Power fault;
Feeder accelerometer error;
Faulty feeder;
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).

</td></tr>
</table>

4 INTERFACES

4.1 Interface with Other Systems

Code	Function
F12	**Serial Port** The system will be equipped with a RS485 port with DB9 connector.
F14	**Synchronization** 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	**Modbus RTU** Communication protocol will be based on Modbus RTU, configured as Master.
F16	**Address** The system will allow to set the slave address/es as a software parameter.

4.2.1 Slave Holding Registers

Register	Address (hex)	Register Type	Description
Frequency	0x1000	1 word	Vibration setpoint from 40 to 240.
Amplitude	0x1001	1 word	Vibration setpoint from 0 to 255.

4.2.2 Slave Input Registers

<table>
<tr><th>Register</th><th>Address (hex)</th><th>Register Type</th><th colspan="2">Description</th></tr>
<tr><td>RMS Accelerometer Status</td><td>0x4000</td><td>1 word</td><td colspan="2">RMS value of current acceleration</td></tr>
<tr><td>Vibration Amplitude Status</td><td>0x4001</td><td>1 word</td><td colspan="2">Current amplitude setpoint of vibration</td></tr>
<tr><td>Probes Status</td><td>0x4002</td><td>1 word</td><td colspan="2">Probes bit mask</td></tr>
<tr><td rowspan="5">Alarms[1]</td><td rowspan="5">0x4003</td><td rowspan="5">1 word</td><td>0</td><td>No Alarm</td></tr>
<tr><td>1</td><td>Power section fault</td></tr>
<tr><td>2</td><td>Accelerometer fault</td></tr>
<tr><td>4</td><td>Feeder fault</td></tr>
<tr><td>8</td><td>Level Probes Fault</td></tr>
</table>

[1] 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

	currState						
level	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