Robot cell MARS

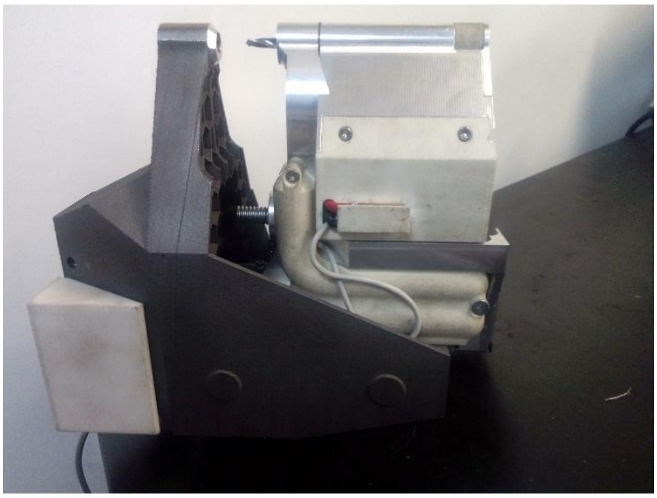
End Effectors C-Clamping drilling

Product datasheet

Version

Version	Date	Objet
0.1	24/06/2022	Version préliminaire
0.2	14/11/2022	English release
0.3	05/12/2022	MRL 6
0.4	31/05/2023	API logging capabilities added

TABLE OF CONTENT



flange end effector

1. MAIN FEATURES

- Dimensions for flange : 190mm x 190mm x 110 mm
- Dimensions for web : 325 x 222 x 110 mm
- Weight for flange EE : 4,6 kg
- Weight for web EE : 5,2 kg
- 3 motorized axes with on-board digital drives
- Electrical supply : 24V
- Pneumatic supply : 6bars
- Chip suction through the spindle nose
- Storage of chips in on-board tray
- Vibration drilling with configurable frequency and amplitude
- Embedded probing camera (accuracy of +/- 0.2 mm)
- Pre-programmed automatic cycles with configuration of the feed rate and speed of the cutting tool, the clamping force, the frequency and the amplitude of the vibratory drilling
- Recording for each cycle executed of the position and force on the feed axis, position and force on the clamping axis, speed and torque on the spindle axis, detection of the start of drilling, detection of the knockout.

2. SPINDLE AXIS

- Motor 2 kw
- Offset tool axis with spur gear drive
- Tool speed : 200 à 4000RPM
- Tool attachment : Diam 6,2 threaded M6

3. FEED AXIS

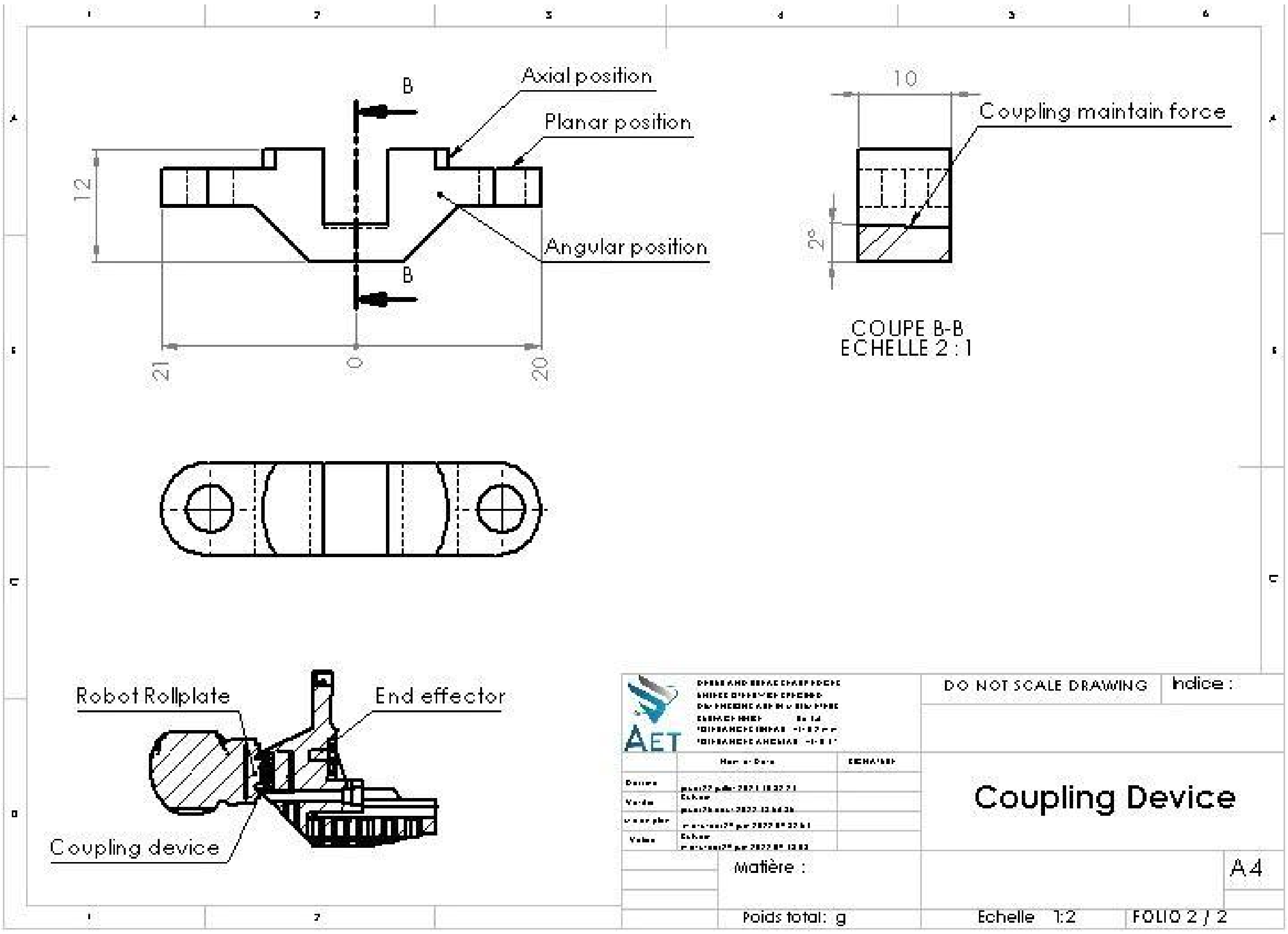
- Motor 1 kw
- Ball screw drive with 2mm pitch
- Feed rate : 0 à 100 mm/s
- Stroke : 17mm

4. CLAMPING AXIS

- Motor 1 kw
- Ball screw drive with 2mm pitch
- Clamping speed : 0 à 50 mm/s
- Max clamping force : 1000N
- Stroke for flange EE : 30mm
- Stroke for web EE : 80mm

5. MECHANICAL ATTACHMENT

- Locking to AET standard



6. PNEUMATIC CONNECTION

PU6 pneumatic tubing

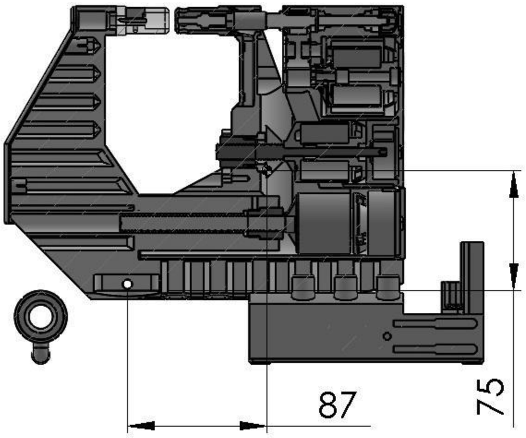
7. POWER SUPPLY CONNECTION

Anderson Power Products PP75 Powerpole®

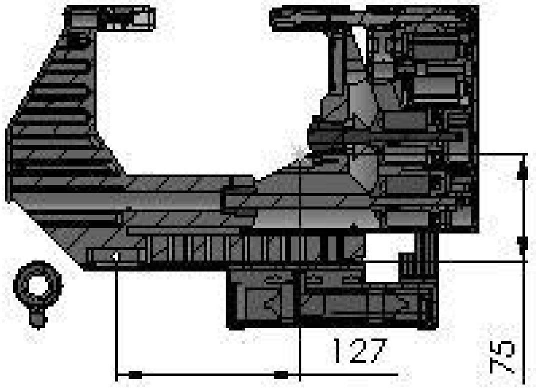
8. DATA CONNECTION

Medium: Fast Ethernet (IEEE 802.3u)
Connector: RJ45

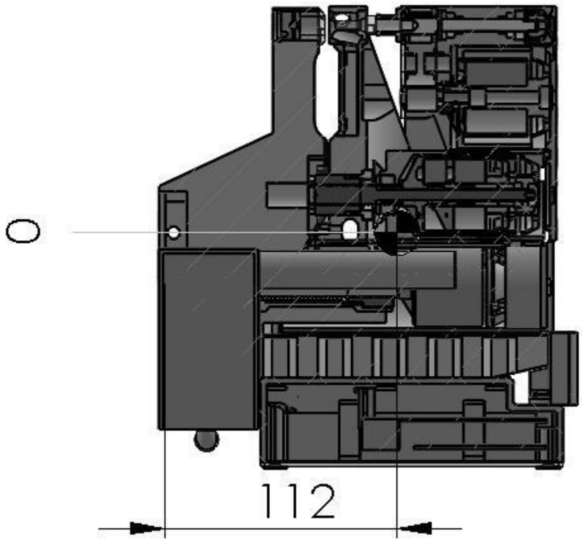
9. WEB EE PAYLOAD DIAGRAM

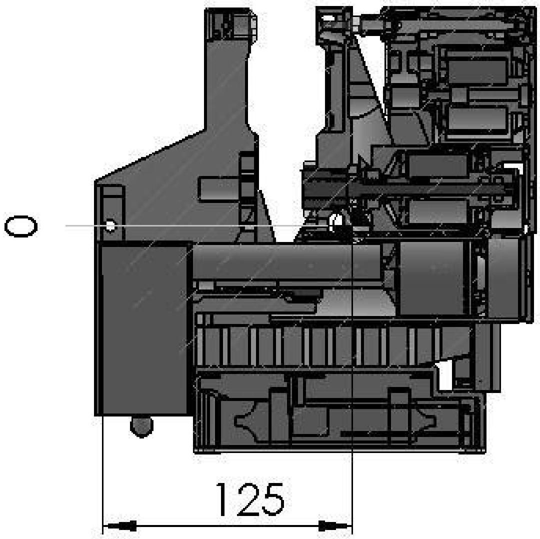


COUPE A-A



10. FLANGE EE PAYLOAD DIAGRAM





11. COMMUNICATION PROTOCOL

Communication with End-effector is implemented using a client/server architecture upon raw TCP/IP socket. (End-effector's server is listening on port 10000)

11.1. « External manager » request fields

Request Id	ASCII string to uniquely identify client's request (for example, 8 byte ASCII HEX formatted integer with auto-increment)
Command Predicate	Command specific ASCII string (predicate is case sensitive)
Command parameters	Command specific Parameters are space delimited (ASCII char code 32)
Request delimiter	Line feed character (ASCII code 10)

N.B : Fields are space delimited (ASCII char code 32)

11.2. End-effector response fields

<table>
<tr><td><b>Request Id</b></td><td>Retrieve End-effector identification information</td></tr>
<tr><td><b>Response Code</b></td><td>8 byte ASCII HEX-formatted integer
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>00000000</td><td>Request completes successfully</td></tr>
<tr><td>FFFFFFFF</td><td>Unknown command</td></tr>
<tr><td>FFFFFFFE</td><td>Content length error</td></tr>
<tr><td>FFFFFFFD</td><td>Parameters count error</td></tr>
<tr><td>FFFFFFFC</td><td>Parameters format error</td></tr>
<tr><td>FFFFFFFB</td><td>Parameters range error</td></tr>
<tr><td>FFFFFFFA</td><td>Process limit reached</td></tr>
<tr><td></td><td></td></tr>
<tr><td>FFFFFFF0</td><td>Operation not permitted (Invalid state)</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Other</td><td>Command specific</td></tr>
</table>
</td></tr>
<tr><td><b>Response Data</b></td><td>Command specific
Data fields are space delimited (ASCII char code 32)</td></tr>
<tr><td><b>Response delimiter</b></td><td>Line feed character (ASCII code 10)</td></tr>
</table>

N.B : Fields are space delimited (ASCII char code 32)

12. API

12.1. End-effector information

End-effector identification

Functionality	Retrieve End-effector identification information	
Predicate	getId	
Parameters	**Name**	**Description (format)**
Response data	**Name**	**Description (format)**
	ee_id	(human readable ASCII integer)
	ee_desc	Manufacturer defined ASCII string (32 bytes max)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set cutting tool identification information

Functionality	Set cutting tool identification information	
Predicate	setCutterData	
Parameters	**Name**	**Description (format)**
	cutter_id	(human readable ASCII integer)
	cutter_ref	Customer defined ASCII string (32 bytes max)
Response data	**Name**	**Description (format)**
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get cutting tool identification information

Functionality	Set cutting tool identification information	
Predicate	getCutterData	
Parameters	**Name**	**Description (format)**
Response data	**Name**	**Description (format)**
	cutter_id	(human readable ASCII integer)
	cutter_ref	Customer defined ASCII string (32 bytes max)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get cutting tool identification information

Functionality	Get cutting identifier	
Predicate	getCutterId	
Parameters	**Name**	**Description (format)**
Response data	**Name**	**Description (format)**
	cutter_id	(human readable ASCII integer)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get cutting tool identification information

Functionality	Get cutting identifier	
Predicate	getCutterReference	
Parameters	**Name**	**Description (format)**
Response data	**Name**	**Description (format)**
	cutter_ref	Customer defined ASCII string (32 bytes max)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set End-Effector Tool Center Point (TCP)

Functionality	Set end-effector tool center point (TCP)	
Predicate	setTCP	
Parameters	**Name**	**Description (format)**
	X	X axis coordinate in mm (human readable ASCII float)
	Y	Y axis coordinate in mm (human readable ASCII float)
	Z	Z axis coordinate in mm (human readable ASCII float)
	W	Yaw angle in degrees (human readable ASCII float)
	P	Pitch angle in degrees (human readable ASCII float)
	R	Roll angle in degrees (human readable ASCII float)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get End-Effector Tool Center Point (TCP)

Functionality	Get end-effector tool center point (TCP)	
Predicate	getTCP	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	X	X axis coordinate in mm (human readable ASCII float)
	Y	Y axis coordinate in mm (human readable ASCII float)
	Z	Z axis coordinate in mm (human readable ASCII float)
	W	Yaw angle in degrees (human readable ASCII float)
	P	Pitch angle in degrees (human readable ASCII float)
	R	Roll angle in degrees (human readable ASCII float)
Specific Error codes	**Value**	**Description (format)**
	NONE	

12.2. Probing device

Probe

<table>
<tr><th>Functionality</th><td colspan="2">Perform probing</td></tr>
<tr><th>Predicate</th><td colspan="2">probe</td></tr>
<tr><th rowspan="2">Parameters</th><th>Name</th><th>Description (format)</th></tr>
<tr><td>seq_id</td><td>Optional parameter to identify probing sequence.
Passing this parameter means a new probing sequence at a different location is started
(human readable ASCII integer)</td></tr>
<tr><th rowspan="7">Response data</th><th>Name</th><th>Description (format)</th></tr>
<tr><td>Probing result</td><td><table><tr><th>Code</th><th>Description</th></tr><tr><td>00000000</td><td>No valid information found</td></tr><tr><td>00000001</td><td>Probing successful</td></tr></table>(8 bytes ASCII HEX-formatted integer)</td></tr>
<tr><td>X</td><td>Measured X offset
(8 bytes ASCII HEX-formatted IEEE754 float)</td></tr>
<tr><td>Y</td><td>Measured Y offset
(8 bytes ASCII HEX-formatted IEEE754 float)</td></tr>
<tr><td>Z</td><td>Suggested Z offset for next probing.
(8 bytes ASCII HEX-formatted IEEE754 float)</td></tr>
<tr><td>Contact flag</td><td><table><tr><th>Code</th><th>Description</th></tr><tr><td>00000000</td><td>Contact not reached</td></tr><tr><td>00000001</td><td>Contact reached</td></tr></table>(8 bytes ASCII HEX-formatted integer)</td></tr>
<tr><td colspan="2">N.B : Extra reported fields are reserved for manufacturer use and can be safely ignored</td></tr>
<tr><th rowspan="2">Specific
Error codes</th><th>Value</th><th>Description (format)</th></tr>
<tr><td colspan="2">NONE</td></tr>
</table>

12.3. Clamping parameters

Set clamping cycles limit

Functionality	Set the maximum number of clamping cycles that the end-effector is allowed to perform before preventing further clamping requests.	
Predicate	setClampingCycleLimit	
Parameters	**Name**	**Description (format)**
	limit	Number of clamping cycles at which end-effector will not be able to perform further clamping cycle without the manager explicitly resets the clamping counter. (human readable ASCII integer)
Response data	**Name**	**Description (format)**
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get clamping cycles limit

Functionality	Get the maximum number of clamping cycles that the end-effector is allowed to perform before preventing further clamping requests.	
Predicate	getClampingCycleLimit	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	limit	Number of clamping cycles at which end-effector will not be able to perform further clamping cycle without the manager explicitly resets the clamping counter. (human readable ASCII integer)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set clamping cycles counter

Functionality	Set the clamping cycles counter at a given value.	
Predicate	setClampingCycleCounter	
Parameters	**Name**	**Description (format)**
	counter	Number of clamping cycles to be stored into end-effector memory (human readable ASCII integer)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get clamping cycles counter

Functionality	Get current value of the clamping cycles counter.	
Predicate	getClampingCycleCounter	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	counter	Number of clamping cycles done by end-effector (human readable ASCII integer)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set clamping force

Functionality	Set clamping force value to be used during further clamping cycles.	
Predicate	setClampingForce	
Parameters	**Name**	**Description (format)**
	force	Force to be applied during clamping cycle, in kg (human readable ASCII float)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get clamping force

Functionality	Get clamping force value to be used during further clamping cycles.	
Predicate	getClampingForce	
Parameters	**Name**	**Description (format)**
Response data	**Name**	**Description (format)**
	force	Force parameter which will be applied during clamping cycle, in kg (human readable ASCII float)
Specific Error codes	**Value**	**Description (format)**
	NONE	

12.4. Drilling parameters management

Set drilling cycles limit

Functionality	Set the maximum number of drilling cycles that the end-effector is allowed to perform before preventing further drilling requests.	
Predicate	setDrillingCycleLimit	
Parameters	**Name**	**Description (format)**
	limit	Number of drilling cycles at which end-effector will not be able to perform further drilling cycle without the manager explicitly resets the drilling counter. (human readable ASCII integer)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get drilling cycles limit

Functionality	Get the maximum number of drilling cycles that the end-effector is allowed to perform before preventing further drilling requests.	
Predicate	getDrillingCycleLimit	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	limit	Number of drilling cycles at which end-effector will not be able to perform further drilling cycle without the manager explicitly resets the drilling counter. (human readable ASCII integer)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set clamping cycles counter

Functionality	Set the drilling cycles counter at a given value.	
Predicate	setDrillingCycleCounter	
Parameters	**Name**	**Description (format)**
	counter	Number of drilling cycles to be stored into end-effector memory (human readable ASCII integer)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get drilling cycles counter

Functionality	Get current value of the drilling cycles counter.	
Predicate	getDrillingCycleCounter	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	counter	Number of drilling cycles done by end-effector (human readable ASCII integer)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set drilling RPM

Functionality	Set spindle rpm speed to be used during further drilling cycles.	
Predicate	setDrillingRPM	
Parameters	**Name**	**Description (format)**
	rpm	Spindle speed to be applied during drilling cycle Range : [500 ; 4000] rpm (human readable ASCII integer)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get drilling RPM

Functionality	Get spindle rpm speed to be used during further drilling cycles.	
Predicate	getDrillingRPM	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	rpm	Spindle speed applied during drilling cycle (human readable ASCII integer)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set drilling feedrate

Functionality	Set feed rate value to be used during further drilling cycles.	
Predicate	setDrillingFeedrate	
Parameters	**Name**	**Description (format)**
	feedrate	Feedrate to be applied during drilling cycle, in mm/min Range : [1 ; 3000] mm/min (human readable ASCII integer)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get drilling feedrate

Functionality	Get feed rate value to be used during further drilling cycles.	
Predicate	getDrillingFeedrate	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	feedrate	Feedrate applied during drilling cycle, in mm/min (human readable ASCII integer)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set drilling parameters

Functionality	Set both spindle rpm speed and feedrate to be used during further drilling cycles.	
Predicate	setDrillingParameters	
Parameters	**Name**	**Description (format)**
	rpm	Same as set drilling RPM
	feedrate	Same as set drilling feedrate
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set peck drilling frequency

Functionality	Set feed rate value to be used during further drilling cycles.	
Predicate	setPeckDrillingFrequency	
Parameters	**Name**	**Description (format)**
	frequency	Frequency parameter to be applied for chips fragmentation Range : [0 ; 40] Hz (human readable ASCII float)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get peck drilling frequency

Functionality	Get feed rate value to be used during further drilling cycles.	
Predicate	getPeckDrillingFrequency	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	frequency	Frequency parameter applied for chips fragmentation (human readable ASCII float)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set peck drilling amplitude

Functionality	Set feed rate value to be used during further drilling cycles.	
Predicate	setPeckDrillingAmplitude	
Parameters	**Name**	**Description (format)**
	amplitude	Amplitude parameter applied for chips fragmentation Range : [0 ; 0.3mm] (human readable ASCII float)
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Get peck drilling amplitude

Functionality	Get feed rate value to be used during further drilling cycles.	
Predicate	getPeckDrillingAmplitude	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	amplitude	(human readable ASCII float)
Specific Error codes	**Value**	**Description (format)**
	NONE	

Set peck drilling parameters

Functionality	Set both peck drilling frequency and amplitude to be used during further drilling cycles	
Predicate	setPeckDrillingParameters	
Parameters	**Name**	**Description (format)**
	frequency	Same as set peck drilling frequency request
	amplitude	Same as set peck drilling amplitude request
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

12.5. End-Effector cycles management

Get End-effector current status

<table>
<tr><th>Functionality</th><td colspan="2">Get End-effector status</td></tr>
<tr><th>Predicate</th><td colspan="2">getStatus</td></tr>
<tr><th rowspan="2">Parameters</th><th>Name</th><th>Description (format)</th></tr>
<tr><td colspan="2">NONE</td></tr>
<tr><th rowspan="3">Response data</th><th>Name</th><th>Description (format)</th></tr>
<tr><td>status</td><td>End-effector’s current status
(human readable ASCII integer)
<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>0</td><td>Boot-up sequence in progress</td></tr>
<tr><td>16</td><td>Boot-up sequence completed
Waiting for initialization request</td></tr>
<tr><td>17</td><td>Feed initialization in progress</td></tr>
<tr><td>18</td><td>Feed initialization completed</td></tr>
<tr><td>19</td><td>Clamp initialization in progress</td></tr>
<tr><td>20</td><td>Clamp initialization completed</td></tr>
<tr><td>32</td><td>Ready to operate</td></tr>
<tr><td>49</td><td>Clamping in prgress</td></tr>
<tr><td>63</td><td>Error occured while clamping</td></tr>
<tr><td>64</td><td>End-effector clamped</td></tr>
<tr><td>65</td><td>Unclamping in progress</td></tr>
<tr><td>79</td><td>Error occured while unclamping</td></tr>
<tr><td>81</td><td>Drilling in progress</td></tr>
<tr><td>95</td><td>Error occured while drilling</td></tr>
<tr><td>96</td><td>Drilling finished</td></tr>
<tr><td></td><td></td></tr>
<tr><td>65535</td><td>Error occured</td></tr>
</table>
</td></tr>
<tr><td>message</td><td>Optional description message
(human readable ASCII string)</td></tr>
<tr><th rowspan="2">Specific
Error codes</th><th>Value</th><th>Description (format)</th></tr>
<tr><td colspan="2">NONE</td></tr>
</table>

Initialize End-effector

Functionality	Perform end-effector initialization after power-up or error	
Predicate	init	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**

	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Launch clamping process

Functionality	Perform end-effector clamping sequence	
Predicate	clamp	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Launch drilling process

Functionality	Perform end-effector drilling sequence	
Predicate	drill	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

Launch unclamping process

Functionality	Perform end-effector unclamping sequence	
Predicate	unclamp	
Parameters	**Name**	**Description (format)**
	NONE	
Response data	**Name**	**Description (format)**
	NONE	
Specific Error codes	**Value**	**Description (format)**
	NONE	

13. LOGGING CAPABILITIES

13.1. LOG FILE FETCHING

Logged data are available through **HTTP GET requests** on port **8080**.

User can access logged data with the following URL paths :

URL Path	Type
/logs/process/*[id]*/probe	Probing capture JPEG image (with blue highlighted hole location if any found)
/logs/process/*[id]*/clamp	CSV log file
/logs/process/*[id]*/drill	CSV log file
/logs/process/*[id]*/unclamp	CSV log file

where *[id]* has to be replaced with :

- current to access logged data associated with the current process
- last to access logged data associated the previous process

N.B. : In case of successive probing, logged data will always return the result of the last probing.

13.2. CSV LOG FILE STRUCTURE

Header	Unit	Descrition
Time	µs	Time elapsed since process' start
T°	°C	End-effector's internal temperature
RPM	rpm	Measured Spindle rotation speed
Spindle Amps	%	Spindle motor's measured current relative to motor's rated current (75A)
Spindle Cmd	%	Voltage applied to spindle motor relative to power supply
Feed Pos.	µs	Feed axis position feedback
Feed Amps	%	Feed motor measured current relative to motor's rated current (55A)
Feed Cmd	%	Voltage applied to feed motor relative to power supply
Clamp Pos.	µs	Clamp axis position feedback
Clamp Amps	%	Clamp motor's measured current relative to motor's rated current (55A)
Clamp Cmd	%	Voltage applied to clamp motor relative to power supply