FGD - Fragranite Machining Cell

Items count interface

The Fragranite Machining Cell subsystems are:

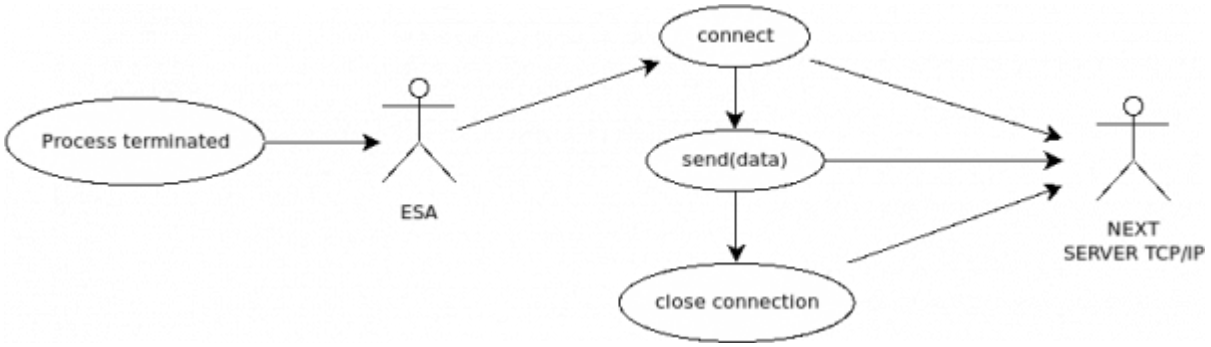
- Robot 1
- Robot 2

Each item that enter into the production process will be taken by the Robot 1 and then by the Robot 2.

So for each item the process is:

1. Robot 1 starts the task
2. Robot 1 terminates the task
3. Robot 2 starts the task
4. Robot 2 terminates the task

For each terminated process ESA will send data to Next via TCP/IP channel.



The data sent by ESA will consist in the following structure:

Field	Description	Example
Command	A specific string of 4 characters	ITEM
Product	The code of the product	114.0055.882
Data of starting task in Robot 1	Data in the format YYYYMMDD	20230430
Time of starting task in Robot 1	Time in the format HH:MM:SS	00:03:17
Data of terminating task in Robot 1	Data in the format YYYYMMDD	20230430
Time of terminating task in Robot 1	Time in the format HH:MM:SS	00:06:12
Data of starting task in Robot 2	Data in the format YYYYMMDD	20230430
Time of starting task in Robot 2	Time in the format HH:MM:SS	00:06:24
Data of terminating task in Robot 2	Data in the format YYYYMMDD	20230430
Time of terminating task in Robot 2	Time in the format HH:MM:SS	00:08:40

The string must be prepared considering the following rules:

- Each field is separated by the next one by the character ';'
- At the end of the string it will be placed the exadecimal code 0x04 (4 as a number)

Follows an example of a string that ESA can send at the end of the process:

```
ITEM;114.0055.882;20230430;00:03:17;20230430;00:06:12;20230430;00:06:24;20230430;00
```

```
:08:40\04
```

Machine status interface

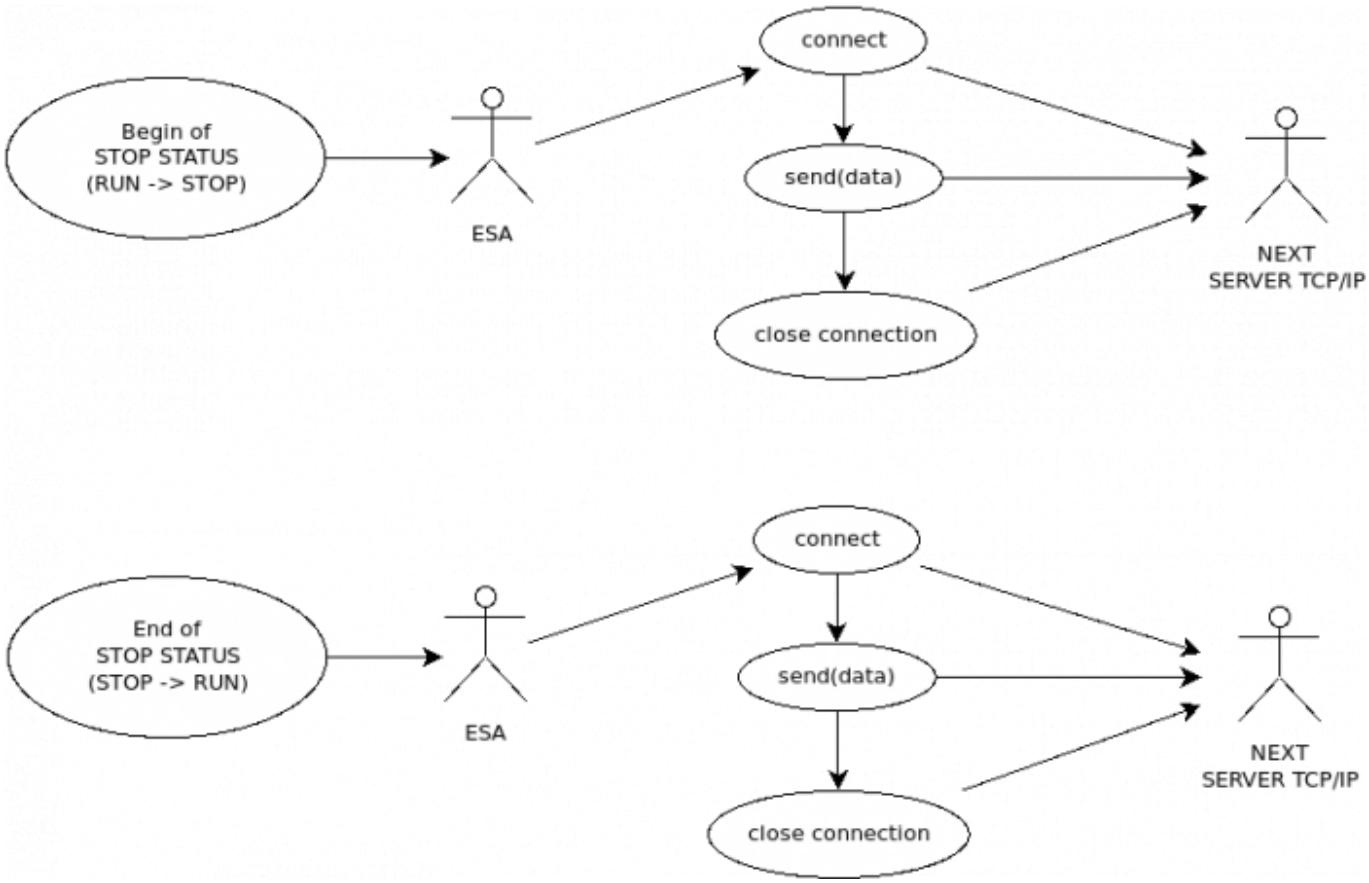
The two robots can be in one of the following status:

- RUN: the robot is available or is producing
- STOP: the robot is not available so it cannot produce

We are interested in the following state transition:

- RUN → STOP: from RUN to STOP
- STOP → RUN: from STOP to RUN

ESA will send to Next via TCP/IP a specific message for each of the two state transition.



Considering a STOP event for one of the two robots, it is necessary to send the 2 messages in a specific order, in particular:

1. RUN → STOP transition
2. STOP → RUN transition

Next will not consider any different sequence, for example:

1. RUN → STOP transition
2. RUN → STOP transition

or

1. STOP → RUN transition
2. STOP → RUN transition

In the follow the data structure for the 2 transitions.

RUN→STOP data structure		
Field	**Description**	**Example**
Command	A specific string of 4 characters	STOP
Robot	The robot interested by the event: ROBOT1 or ROBOT2	ROBOT1
Data of starting the event	Data in the format YYYYMMDD	20230430
Time of starting the event	Time in the format HH:MM:SS	00:03:17
Reason of the stop	A numeric code that identify the reason of the stop, from a specific list of the available stops	20

* Each field is separated by the next one by the character ';'

- At the end of the string it will be placed the exadecimal code 0x04 (4 as a number)

In the follow an example of a string that ESA can send when a STOP began:

```
STOP;ROBOT2;20230430;00:03:17;20\04
```

STOP→RUN data structure		
Field	**Description**	**Example**
Command	A specific string of 3 characters	RUN
Robot	The robot interested by the event: ROBOT1 or ROBOT2	ROBOT1
Data of starting the event	Data in the format YYYYMMDD	20230430
Time of starting the event	Time in the format HH:MM:SS	00:03:17
Reason of the stop	A numeric code that identify the reason of the stop, from a specific list of the available stops	20

* Each field is separated by the next one by the character ';'

- At the end of the string it will be placed the exadecimal code 0x04 (4 as a number)

In the follow an example of a string that ESA can send when a STOP began:

```
RUN;ROBOT2;20230430;00:08:17;20\04
```