Using the Teabag Boxing System Simulator

The Teabag Boxing System Mechanical-process simulator is composed of two modules:

1. The Teabag Boxing System Mechanical-Process graphical simulator (file: TBSMPSimulator.exe)

2. An HMI module that is used for controlling the "Teabag Boxing System MP graphical simulator" (file: TBSMPSimHMI.exe).

Both applications have been developed using C# and **require .NET framework v1.1 to be executed.**

.**NET Framework** Version 1.1
Download details: .**NET Framework** Version 1.1 Redistributable Package
http://www.microsoft.com/downloads/details.aspx?FamilyId=262D25E3-F589-4842-8157-034D1E7CF3A3&displaylang=en

A. The TBS Mechanical-Process graphical simulator.

This application simulates the Mechanical process of the Teabag Boxing System. It provides visualization of the TBS mechanical process and a socket interface for interfacing with sensors and actuators (see ArchimedesTBSExample1_4.pdf.)

To execute the application use the following command from windows command prompt:

```
tbs_simulator <SPIFB IP>
```

where
`<SPIFB IP>` defines the IP address of the device where the Simulated Process Interface Function Blocks (SPIFBs) that are incorporated in the TBS Controller are located. SPIFBs are used in the development process of the TBS controller to allow its test with the TBSMPSimulator.

B. The TBSMPSimHMI module.

This application provides a command prompt interface for the user to manually issue commands to the TBSMPSimulator.

The TBSMPSimHMI application should be executed on the same machine with the TBSMPSimulator.

TBSMPSimHMI allows the user to:

1. Generate one teabag that enters the teabag boxing process.

Command: **`factory <type>`**

Where
`<type>` can be either “pass” or “fail” and determines whether the generated teabag will pass weight control or not.

2. Periodically generate teabags that enter the teabag boxing process.

   Command: **`gen <type> <period>`**

   Where
   `<type>` can be either “pass”, “fail” or “mix” and determines whether the generated teabags are all valid, all invalid or randomly mixed accordingly.
   `<period>` determines the time interval between two successive teabag generations. Period is measured in tenths of a second. A value of -1 is used to stop the generation of teabags.

3. Issue commands to actuators (motors).

   Command: **`<motor_name> <TBSMPcommand>`**

   Where
   `<motor_name>` can be one of “arm”, “sbelt”, “belt1”, “belt2” and “belt3” and

   `<TBSMPcommand>` can be “forwards”, “backwards” and “stop”.

Examples:

```
> gen mix 40
```

For the teabag producer to provide teabags every 4 seconds. Generated Teabags are randomly valid and invalid (regarding weight).

```
> gen mix -1
```

To stop the generation of teabags.

```
> arm forwards
```

For the arm (feeder) to move forwards.

```
> arm backwards
```

For the arm (feeder) to move backwards.

```
> arm stop
```

For the arm (feeder) to stop moving.