# Project 7

# Automatic powder materials packing line

Created by:
Sylwester Dawida and Michał Machura

# Table of contents

# 1. Introduction

Project Github link:

https://github.com/MichalMachura/PowderMaterialPackingLine

Powder materials like flour, grain or cement after production process are very often transported by conveyor belts. When end user is a big company, it is packaged into large containers and transported by various types of vehicles eg cement for large construction site is transported by vehicles without any packages. Some times, when end user is nearly production site, material is still transported by conveyor belts eg. coal from mine goes to power station by conveyor belts.

Another situation is when end user is small company or a person. Then large containers are too big, conveyor belts transport is also excluded. Materials must be packaged into small packages or boxes. Next each package must be weighted. Necessary is also checking for some damages. Damaged package could not go for sell, it must be rejected. When packages are weighted and checked for some damages, they should be moved onto pallet. Discussed tasks of course could be executed be people, but sometimes it could be dangerous for them. People also working slower than machines, so it could slow down works of whole system. But also these things could be done, be large group of people, but then risk of accidents is grown. Another option is use of automated packaging system, which could do all necessary things, be safe and allow for control from higher level. Of course automated systems not always could detect damages correctly, but then can be called an alarm and it could be validated by person.

The main topic of this is creating SCADA application of automated powder material packaging line using Wonderware InTouch. The packaging line should contain:

- filling station
- weight validation station
- damage validation station
- palletization station.

In case of package tearing should be called alarm and problem

should be solved manually, by person. Application is divided into two parts. first is graphical - frontend. Second is functional part - backend. All is controlled by suitable scripts.

# 2.Description

Whole backend of application is implemented as a state machine. Each box has assigned value of percentage filling. While box is empty is invisible. Boxes positions are incremented, while they are not achieve start position of next box. Then next necessary procedures are executed.

Some of process are need random values, to work and better described real processed. Unfortunately Wonderware InTouch is not equipped into random number generator. For this task was created another script, with counter variable increased every script period, and divided modulo 100. It let to obtain pseudo-random generator.

Whole functional of system was divided into specialized stations. Next subsections are dedicated for their short descriptions.

### 2.1. Filling station

Station(Fig. 1) is filling packages to at least 70% of max box weight. Real current flow is constant value 3.5 % of max box weight. Controller takes current flow value from sensor, with random disturbance in range -70% to +84%.

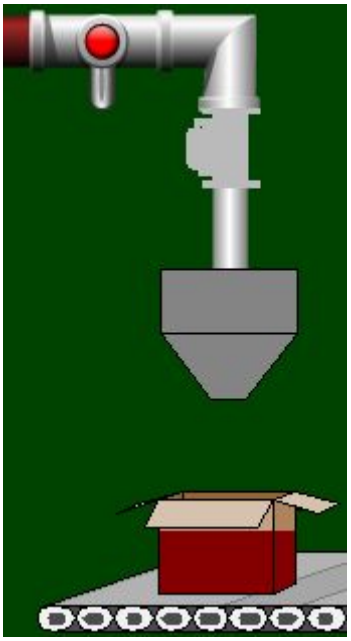

*Fig. 1 Filling station*

### 2.2. Weight validation

Station(Fig. 2) is checking packages weight which should be in range (65 -75)%. When weight is below 65% package is rejected. Boxes which are rejected are sliding on side line to special container. When weight is above upper limit human needs to check package manually.

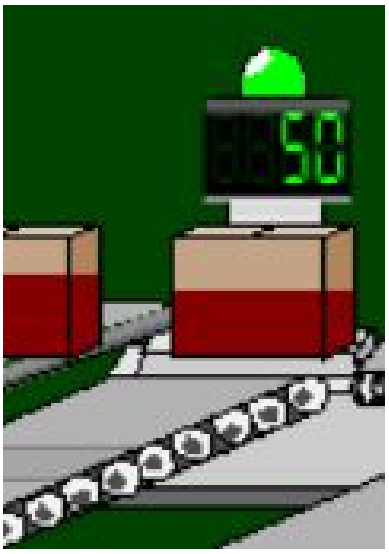


*Fig. 2 Weight validation station*

### 2.3. Damage validation

This station(Fig. 3) is responsible for damage recognition. In this case if package is rejected by station, human needs to check package manually. Boxes which are rejected are sliding on side line to special container. Box is considered as damaged, if random value is bigger than 90.

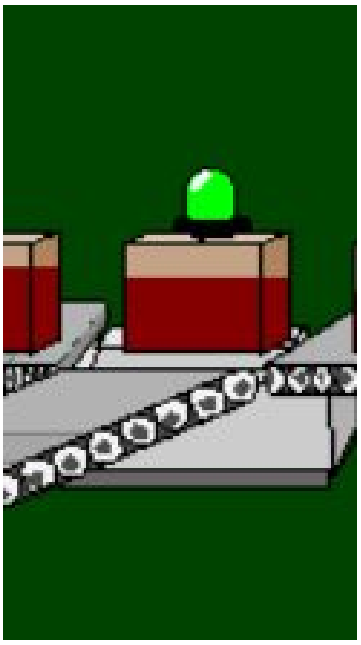

*Fig. 3 Damage validation station*

### 2.4. Palletization

Finally validated packages are stored on pallet(Fig. 4) and if it's full (8 boxes) line stops and pallet is replaced with new empty one.

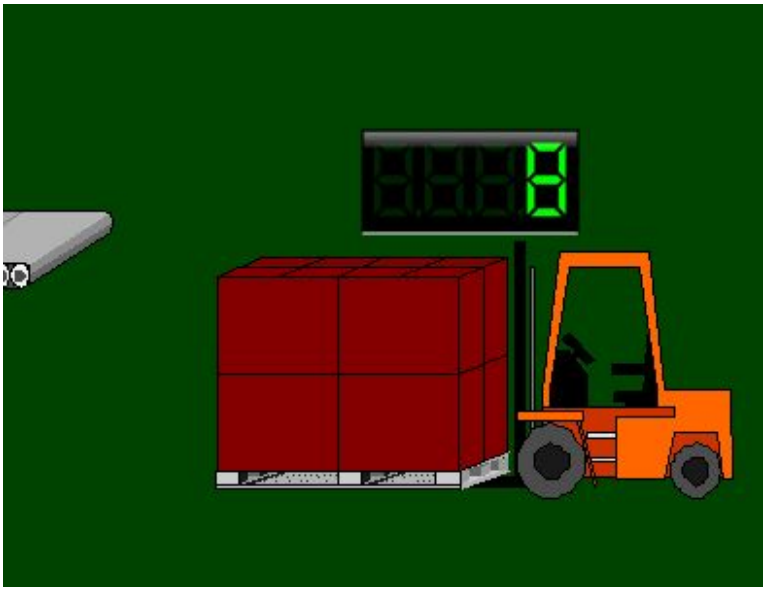

*Fig. 4 Palletization station*

### 2.5. Alarms

InTouch mechanism of alarms is used to control weight of fourth box on

line. While fourth box's weight is out of limits alarm is called. Current alarm is available to see in frame(Fig. 5) located in right-bottom part of window.

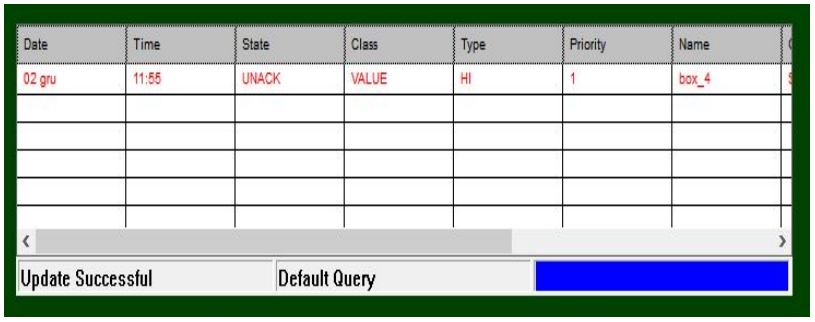


*Fig. 5 Alarm list*

### 2.6. Security

In case of any accident, emergency button(Fig. 6) is available. While button is pushed, line is stopped, otherwise line is on.

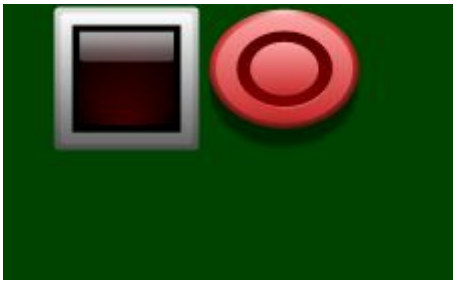

*Fig. 6 Security button*

# 3.Visualisation

Whole package line is show on Fig. 7. Whole visual part is implemented as a token propagation. As was mentioned in previous chapter, value of one box is assigned to next one, when it achieved it's max position, equaled to start of next box trajectory.

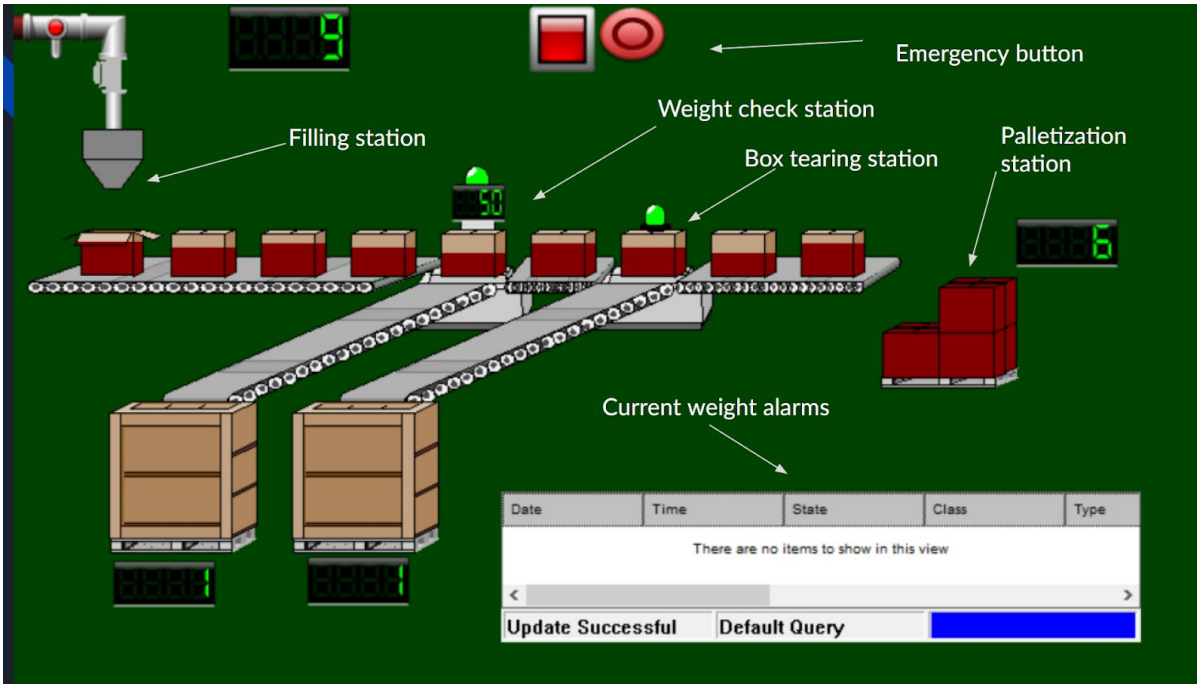


*Fig. 7 Packaging line*

# 4.Conclusions

To sum up, packaging line is well constructed, it has main stop button for security reasons, filling station simulated with random generator. Weight validation station with specific ranges, secured with “manual check” alarm. Finally palletization station with animated forklift.

Problems of realization were non included random number generator and limited number of variables used in script. First problem was solved. Second could not be solved.

In conclusion could be said that the simulation is close to realistic packaging line and it’s control of processes could be implemented in real production line, after some corrections and while variables number is not limited.