Projekt zaliczeniowy

Z przedmiotu

Implementacja komputerowych architektur sprzętowych

***Symulacja sortowni w programie Factory IO***



Prace wykonali:

Łukasz Kozak

Arkadiusz Kruszyński

AiR WARiE A3 2020

Spis treści

[Założenia projektowe. 3](#_Toc41151657)

[Budowa projektu linii sortowni. 3](#_Toc41151658)

[Linia początkowa składa się z: 3](#_Toc41151659)

[Linia paletyzacji: 3](#_Toc41151660)

[Linia sortowania kolorowych przedmiotów składa się z: 3](#_Toc41151661)

[Program drabinkowy w TIA Portal. 4](#_Toc41151662)

[Weryfikacja działania programu. 4](#_Toc41151663)

[Literatura. 4](#_Toc41151664)

# Założenia projektowe.

Projekt bazuję na ścisłej współpracy programu Factory IO oraz TIA Portal v13 z Symulatorem. W aplikacji TIA Portal stworzyliśmy program drabinkowy, natomiast Factory IO ułożyliśmy imitację sortowni, która ma za zadanie:

* Sortować paczki oraz je paletyzować, a następnie umieścić je w magazynie
* Sortować kolorowe surowe materiały.
* Zostało również zasymulowana usterka w materiale za pomocą niebieskiej pokrywki.

# Budowa projektu linii sortowni.

W symulacji można wyróżnić trzy gałęzie linii:

* Linia początkowa, startowa z której pojawiają się losowo elementy do posortowania wstępnego.
* Linia paletyzacji.
* Linia sortowania kolorowych produktów.

Taki podział usprawnił współpracę w projekcie, ponieważ każdy miał swoje niezależne zadanie do zrealizowania. Chcieliśmy również stworzyć jak najbardziej realną symulację linii wstępnego sortowania. Dlatego skrzynia elektryczna znajduje się bezpośrednio przy emiterze przedmiotów oraz posiada niezbędne elementy do funkcjonowania sortowni rzeczy.

## Linia początkowa składa się z:

* Emitera przedmiotów, który generuję: kolorowe surowe materiały, niebieskie pokrywki oraz paczki do sortowania.
* Czujnika wizji, którego wyjściem są liczby odpowiadające kolorowi materiału znajdującego się bezpośrednio pod nim oraz czujnika optycznego.
* Sortera dla lekkich przedmiotów……..

## Linia paletyzacji:

## Linia sortowania kolorowych przedmiotów składa się z:

* Dwóch taśm transportowych dla lekkich przedmiotów.
* Czujnika wizji, którego wyjściem są liczby odpowiadające kolorowi materiału znajdującego się bezpośrednio pod nim.
* Trzech czujników optycznych i jednego odbiciowego.
* Czterech ramion z taśmą.
* Trzech wyjść które zbierają surowe elementy każdego koloru po posortowaniu.

# Program drabinkowy w TIA Portal.

Do zrealizowania programów użyliśmy oddzielnych bloków funkcyjnych, które są wywoływane w głównym wątku sterownika. Drabinka również znajduję się w załączniku nr 1.

Pomijając elementy konieczne do prawidłowego działania TIA portal z Factory IO, które znajdują się w bloku FC9000 to nasz program składa się:

* Ustawień wstępnych: Start, Stop, Emergency. Każdy z tych statusów są konieczne do działania linii produkcyjnej. W Emergency znajdują się wszystkie błędy jakie mogą się pojawić, między innymi nasz zasymulowany błąd produktu jako pokrywka niebieska.
* Ustawień wstępnych, które odpowiadają za załączenie się taśm i innych niezbędnych rzeczy do działania sortowni.
* Z dużej części kodu sortowania elementów po kolorze surowego materiału.
* Z obsługi symulowanego błędu.

(Tutaj dodaj swoją część)

Podczas wykonywania sortowni kolorowych przedmiotów za pomocą tablicy karnaugho został zdefiniowany warunek działania taśm transportowych.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| lp. | Czujka na początku tasmy | Czujka na końcu taśmy | Bufor sortowni | Wyjście |
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| 00 | 1 | 1 |
| 01 | 1 | 1 |
| 11 | 1 | 0 |
| 10 | 1 | 0 |

Z tabeli jasno wynika, że warunek wynosi:

Dlatego zrezygnowaliśmy z czujki na początku taśmy, bo była nadmiarowa.

# Weryfikacja działania programu.

# Literatura.

1. <https://docs.factoryio.com/> - dokumentacja Factory IO
2. <https://www.youtube.com/user/RealGamesLDA> - kanał YouTube twórców Factory IO