|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт искусственного интеллекта

Кафедра проблем управления

**КУРСОВАЯ РАБОТА**

по дисциплине **Основы программирования систем управления**

**Тема курсовой работы: «**Разработка и симуляция автоматического склада с помощью Automation Studio и Factory IO»

|  |  |
| --- | --- |
| **Студент группы:** КРБО-01-22 | Гизетдинов М.Р. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **Руководитель курсовой работы:** | Морозов А. А.**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

|  |  |
| --- | --- |
| Работа представлена к защите: | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
| Допущен к защите | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

Москва 2023

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ…………………………………………………………………….4
2. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ……………………………………...5
3. РАЗРАБОТКА АЛГОРИТМИЧЕСКОГО ОБЕСПЕЧЕНИЯ…………….6-18
4. РАЗРАБОТКА ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ…………………….19-24
5. ТЕСТИРОВАНИЕ И ОТЛАДКА…………………………………………….25
6. ЗАКЛЮЧЕНИЕ………………………………………………………………26

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ……………………………..27

1. ВВЕДЕНИЕ

В современной логистике наиболее важна скорость, пунктуальность и точность. Для этого предпринимателям нужны отлаженные и быстро работающие склады, однако, поскольку на складах в основном работают люди, это может приносить некоторые проблемы: людям необходим отдых и хорошие условия труда, человек может ошибиться и принести убытки. На помощь приходит робототехника – робот не может устать, абсолютно точен и менее затратен в долгосрочной перпективе. Склад – идеальное место для автоматизации: сортировка, упаковка, загрузка и разгрузка могут быть реализованы без участия в этих процессах человека.

1. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ

Цель данной курсовой работы – создание в среде симуляции Factory IO модели автоматического склада и её настройка для дальнейшей работы. Автоматический склад должен:

1. Сортировать и упаковывать коробки на палеты;
2. Загружать их на стойки и при необходимости разгружать.

Для этого нужно разработать алгоритм его работы, создать из данных в среде Factory IO элементов физическую модель склада, а с помощью Automation Studio – разработать и отладить ПО. Взаимодействовать Automation Studio с Factory IO будет путем встроенной утилиты – Simulation. Будет запускаться виртуальный сервер, имитирующий реальный ПЛК и подключающийся переменными напрямую к портам Factory IO.

Версия Automation Studio – v 4.5.2, Factory IO – v 2.5.4.

1. РАЗРАБОТКА АЛГОРИТМИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Работа склада начинается с сортировки коробок двух размеров – S и M, которые будут отправляться в сектор упаковки и хранения. Для этого на входном конвейере расположена вертикальная стойка из горизонтальных сенсоров для измерения высоты коробок. На одной стороне источники лазерного луча, на другой – приемники. При перекрытии лучей подается информация о том, сколько из них перекрыто. На основании этих данных становится понятно, какого размера коробки:

1. Перекрыт только один нижний лазер – коробка размера S;
2. Перекрыто два нижних лазера – коробка размера M.

С помощью этой информации составляется решение о дальнейшем направлении коробки – малые коробки на конвейер для упаковки малых коробок, средние – на конвейер для упаковки средних.

Отправка происходит с помощью двух роботов-манипуляторов с одной поворотной осью. Каждый манипулятор необходим для отправки коробок исключительно одного типа: первый по движению конвейера – для малых коробок, второй – для средних. Как только датчики фиксируют приближение коробки – нужный манипулятор отклоняет механическую «руку» с намотанной на нее конвейерной лентой, которая запускается в сторону нужного для дальнейшей транспортировки конвейера. При приближении коробки другого размера, неиспользуемый манипулятор отключается, включается другой.

В случае, если склад не может принять коробки (происходит упаковка и коробки временно не могут пройти дальше, либо склад заполнен) они проезжают дальше по входному конвейеру (подразумевается, что дальше по конвейеру расположены такие же секторы упаковки и хранения, что и в данной курсовой работе). Для появления коробок используется Emitter – условный инструмент среды Factory IO, который помещает коробки на конвейер, а для

|  |
| --- |
|  |
| Рисунок 1 – входная группа и сортировка. |
|  |
| Рисунок 2 – пример работы манипулятора. Снизу – конвейеры для отправки коробок на упаковку: справа – для малых коробок, слева – для средних. |

их исчезновения – Remover – противоположность Emitter.

Далее коробки отправляются на упаковку. Для этого используется устройство Palletizer, его суть заключается в правильном расположении коробок и помещении их на палету. Состоит Palletizer из двух уровней: на верхнем коробки расставляются и загружаются на палеты, с нижнего поднимается платформа с палетой, после загрузки которой опускается обратно вниз и отправляет палету к загрузочному крану.

|  |
| --- |
|  |
| Рисунок 3 – общий вид Palletizer. |

Верхний уровень состоит из нескольких механизмов:

1. Конвейерная лента.
2. Механизм Plate - шторки, которые раздвигаются для помещения на палету коробок.
3. Механизм Push, который толкает коробки на шторки.
4. Механизм Clamp, который тромбует и фиксирует коробки для того, чтобы при раздвижении шторок коробки не разъехались в разные стороны.
5. Механизм Turn, который заключается в выдвижении прорезиненного препятствия, при соприкосновении с которым коробки разворачиваются на 90 градусов из-за продолжающегося движения конвейера.

К верхнему уровню подведен конвейер с ранее отсортированными коробками. На конвейере расположен направленный датчик (Diffuse Sensor), который регистрирует появление коробки. С помощью него можно понять, когда коробка проехала на верхний уровень Palletizer. Также на этом конвейере установлены направляющие, чтобы коробки располагались длинной стороной параллельно конвейеру.

|  |
| --- |
|  |
| Рисунок 4 – верхний уровень Palletizer. |

|  |
| --- |
|  |
| Рисунок 5 – верхний уровень Palletizer с активными механизмами. |

Схема упаковки малых коробок на верхнем уровне заключается в следующем:

1. На конвейер проезжают три коробки, предварительно развернутых на 90 градусов (перпендикулярно конвейеру) механизмом Turn. Это будет первый ряд коробок;
2. Первый ряд коробок помещается механизмом Push на шторки механизма Plate. Во время данного процесса конвейер Palletizer и подводящие коробки конвейера стоят на месте;
3. Второй ряд коробок формируется аналогично п. 1;
4. Второй ряд коробок помещается на шторки аналогично п. 2, таким образом формируется группа из 6-и коробок, что составят первый слой на палете. Конвейер Palletizer и подводящие коробки конвейера останавливаются вплоть до п. 8;
5. Группа коробок тромбуется и фиксируется механизмом Clamp, который остается активен до п. 8;
6. Механизм Plate раздвигает шторки и коробки проваливаются на заранее подведенную палету на платформе;
7. Платформа с палетой опускается на расстояние, достаточное для того, чтобы шторки механизма Plate смогли закрыться;
8. Все механизмы, кроме Turn, возвращаются в изначальное состояние. Конвейер запускается обратно.
9. Пункты 1-8 повторяются, формируется второй слой коробок.

Для средних коробок процедура отличается тем, что механизм Turn необходимо отключить (перпендикулярно конвейеру коробки не поместятся), из-за чего на конвейер Palletizer помещается только одна коробка. Механизм Push срабатывает три раза, формируется группа из трех коробок в один слой высоты, после этого палета считается укомплектованной.

|  |  |
| --- | --- |
| Рисунок 6 - упаковка коробок по пунктам вышеописанной схемы. | |
|  |  |
| Начало. | Поворот механизмом Turn. |
|  |  |
| Пункт 1. | Пункт 2. |
|  |  |
| Пункт 3. | Пункт 4. |
|  |  |
| Пункт 5. | Пункт 6. |
| Рисунок 6 (продолжение). | |
|  |  |
| Пункт 7. | Пункт 8. |
|  |  |
| Расположение средних коробок на конвейере Palletizer. | Группа средних коробок. |
|  |  |
| Укомплектованная малыми коробками палета. | Укомплектованная средними коробками палета. |
| Рисунок 6 (продолжение). | |

После упаковки палета с коробками опускается на нижний уровень, что состоит из:

1. Платформы, что служит лифтом и конвейером для палет. В неё встроены ролики, на которые встает палета, при необходимости ролики запускаются и палета движется в нужном направлении. Так же на платформе расположены концевые датчики, которые помогают выставить палету ровно посередине.
2. Подведенного к платформе загрузочного конвейера. На нем располагается Emitter для палет. Когда платформа находится в нижнем положении, она равняется с ним, что позволяет загрузить палету.
3. Разгрузочного конвейера, по которому укомплектованные палеты движутся к загрузочному крану. Находится на одном уровне с конвейером из п. 2.

|  |
| --- |
|  |
| Рисунок 7 – нижний уровень Palletizer. |

На нижнем уровне палета отправляется на разгрузочный конвейер с помощью роликов на платформе, после чего попадает на погрузочный конвейер крана.

Погрузочный кран состоит из:

1. Несущей рамы.
2. Транспортировочной платформы.
3. Рельс, по которым перемещается платформа.

|  |
| --- |
|  |
| Рисунок 8 – Общий вид зоны хранения и погрузочного крана. |

Зона погрузки на кран состоит из:

1. Погрузочного конвейера с прорезью под вилы платформы и уголком для ровного расположения палеты.
2. Датчика приближения палеты (Diffuse Sensor).

|  |
| --- |
|  |
| Рисунок 9 – погрузочная зона крана. |

Погрузка происходит следующим образом:

1. Датчик палеты сигнализирует об её приближении;
2. Платформа выдвигает вилы в сторону палеты и приподнимается, чтобы палета повисла на вилах;
3. Вилы задвигаются обратно – палета погружена на платформу;
4. Платформа по рельсам перемещается к свободной ячейке на стеллаже;
5. Платформа выдвигает вилы в сторону ячейки и опускается, чтобы палета погрузилась на выступы ячейки;
6. Кран возвращается в изначальное положение.

|  |  |
| --- | --- |
| Рисунок 10 - погружение палеты на стеллажи по вышеописанным пунктам | |
|  |  |
| Пункт 1. | Пункты 2-3. |
|  |  |
| Пункт 4. | Пункт 5. |
|  | |
| Пункт 6. | |

Следующий, и последний этап – это разгрузка. Принцип действия такой же, как и у разгрузочного крана, только наоборот – у крана нет погрузочной зоны, но есть разгрузочная. Состоит из разгрузочного конвейера и Emitter. Активируется по нажатии кнопок.

|  |
| --- |
|  |
| Рисунок 11 – зона разгрузочного крана. Слева – короб с кнопками для запуска разгрузки. |

|  |
| --- |
|  |
| Рисунок 12 – общий вид склада. |

1. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В основе ПО данной работы лежит следующий принцип:

1. Существует переменная – счетчик для обозначения различных состояний оборудования;
2. Счетчик работает исключительно в связке с датчиками, необходимыми для отслеживания выполнения действий;
3. После выполнения действия к счетчику прибавляется единица, что сигнализирует о смене состояния оборудования, начинается выполнение следующего действия.

Таким образом реализуется выполнение последовательных действий в системах реального времени. Подавляющее большинство действий осуществляется с помощью переменной bool. С учетом особенностей работы устройств в Factory IO были добавлены промежуточные состояния, работа которых объяснена на примере механизма Push: после того, как образовался ряд из коробок, необходимо толкнуть их на шторки механизма Plate, после чего перевести механизм Push в изначальное положение. Решение проблемы кажется очевидным, пример алгоритма с переменной-счетчиком – i:

ЕСЛИ (i == 1 И датчик\_коробки == 1) ТО

PUSH = 1

i++

ЕСЛИ (i == 2 И конечное\_положение\_push == 1) ТО

PUSH = 0

i++

ЕСЛИ (i == 3 && конечное\_положение\_push == 0) ТО

ПРОДОЛЖИТЬ РАБОТУ КОНВЕЙЕРА

i++

Однако это решение не сработает: во-первых, почти во всех механизмах Factory IO конечным положением считается начальное и конечное, то есть существует только крайнее состояние и промежуточное, что доставляет большие неудобства при работе с ними. Во-вторых, механизмы работают с задержкой, то есть если мы в первом состоянии заставляем механизм двигаться, а в следующем, при активации датчика крайнего положения заставляем механизм вернуться в изначальное состояние, то при выполнении алгоритма механизм не сдвинется с места. Это произойдет из-за того, что после выполнения первого состояния сразу же сработает условие для выполнения второго состояния, т.к. в этот момент датчик крайнего положения будет выдавать положительное значение, ведь циклы в системах реального времени длятся миллисекунды, а за это время механизм не успевает прийти в движение.

Таким образом, было решено ввести промежуточные состояния:

ЕСЛИ (i == 1 && датчик\_коробки == 1) ТО

PUSH = 1

i++

ЕСЛИ (i == 2 && крайнее\_положение\_push == 0) ТО (промежуточное сост.)

i++

ЕСЛИ (i == 3 && крайнее\_положение\_push == 1) ТО

PUSH = 0

i++

ЕСЛИ (i == 4 && крайнее\_положение\_push == 0) ТО (промежуточное сост.)

i++

ЕСЛИ (i == 5 && крайнее\_положение\_push == 1) ТО

ПРОДОЛЖИТЬ РАБОТУ КОНВЕЙЕРА

i++

На примере данного алгоритма написано 90% кода в данной работе, за исключением того, что вместо чередующихся условных операторов была введена машина состояний switch case, внутри которой уже и расположены условные операторы, аргументом для switch case является переменная-счетчик.

Иногда if используется в чистом виде, к примеру, когда нужно запустить цепочку действий switch case с помощью показаний множества датчиков:

if ((BR\_PLT\_FRONT\_1 == 1) && (BR\_PLT\_BACK\_1 == 1) && (k==0))   
 **{** BR\_PLT\_CNV\_1=0;  
 BR\_PLT\_GO\_1=0;  
 BR\_PLT\_EL\_FULL\_1=1;  
 BR\_PLT\_UP\_1=1;  
 k++;  
 **}**

Для различных механизмов код написан в отдельных программах:

1. Код Palletizer основан на машине состояний, разделенной на несколько switch case для удобства его чтения. Присутствует несколько условных операторов для запуска цепочки switch case. Основным условием является появление коробок перед Diffuse Sensor.
2. Код кранов основан на машине состояний, состоящей из одного switch case. Есть несколько условных операторов. Информация о заполнении стеллажей хранится в глобальных массивах типа bool, при заполнении ячейки, присущее ей значение в массиве становится равным 1.
3. Код конвейеров представляет дерево if, зависящее от глобальных переменных, предназначенных для этого, и активирующихся либо внутри кода кранов, либо Palletizer.
4. Код сортировки представляет дерево if, зависящее от показаний датчиков высоты коробок. Так же есть исключения для случаев невозможности отправления коробок на упаковку.
5. Код разгрузочного крана представляет собой копию кода погрузочных кранов, только действия в нём представлены наоборот. Взаимодействует с глобальными массивами с информацией о заполнении стеллажей. Путем цикла находит первую заполненную ячейку и разгружает её. Запускается с кнопок. Во-время работы по разгрузке невозможно нарушить работу крана путем нажатия соседней кнопки, если стеллажи пустуют – кран в действие при нажатии кнопки не придет. Сделано это с использованием условных операторов, в которые обрамлены все switch case в этом коде.

Далее предложен набор переменных для разных программ, переменные названы по такой схеме: **BR\_\*название устройства, к примеру PLT – Palletizer, CRANE – Кран, HEIGHT – сенсор высоты\*\_\*название механизма на устройстве или действия\*\_\*номер устройства\***.

Приставка LIM означает, что переменная отвечает за концевой датчик, MOVING – датчик движения механизмов, SENS – выносной датчик, FULL – при включенной переменной действия выполняются по полной амплитуде. Переменные i, k, n, go, go1 – счетчики.

|  |
| --- |
| Рисунок 13 – переменные программ проекта. |
|  |
| Переменные файла SORTING. |
|  |
| Переменные файла PLT\_1. |
| Переменные файла PLT\_2. |
|  |
| Переменные файла CONVEYORS. |
|  |
| Переменные файла CRANE\_1. |
| Рисунок 13 (продолжение). |
|  |
| Переменные файла CRANE\_2. |
|  |
| Переменные файла CRANE\_EXIT. |
| Рисунок 13 (продолжение). |

Информация по механизмам Factory IO бралась с официального сайта Factory IO (пункт 1 списка литературы), информация по работе с Automation Studio – с образовательного портала РТУ МИРЭА (пункт 2 списка литературы). Листинг исходного кода и проект представлен в пункте 3 списка литературы.

1. ТЕСТИРОВАНИЕ И ОТЛАДКА

Тестирование проводилось путем запуска симуляции и проверки работы визуально и через встроенную утилиту Automation Studio – Watch, которая позволяет отслеживать состояние механизмов, датчиков и счетчиков.

При нахождении сбоя в работе программа останавливалась на тех значениях переменной-счетчика, на которых сбой и происходил. Далее находился тот участок кода, на котором счетчик принимал такие значения, и анализировался с последующим исправлением ошибок.

|  |
| --- |
|  |
| Рисунок 14 – вид Watch. В списке представлены отслеживаемые переменные. |

1. ЗАКЛЮЧЕНИЕ

Была проведена масштабная работа по созданию модели склада, по итогам которой полученная модель работает быстро и без ошибок. Для её работы, а точнее – разгрузки, нужен только один человек-оператор, а в случае необходимости – можно автоматизировать и разгрузку, подключив к единой системе контроля производств, что отсылает нас к технологии Web 4.0. Склад может масштабироваться в зависимости от требований предпринимателей, ведь показанная в данной работе модель является лишь одной из множества подобных ячеек и может работать как в группе, так и обособленно. В дальнейшем подобные склады придут на смену нынешним за счет своей надежности и выгоды в долгосрочном использовании.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Информация по устройствам в Factory IO. — Текст : электронный // Официальный сайт Factory IO : [сайт]. — URL: https://docs.factoryio.com/manual/ (дата обращения: 01.05.2023).
2. Информация по работе с B&R Automation Studio. — Текст : электронный // Учебный портал РТУ МИРЭА : [сайт]. — URL: https://online-edu.mirea.ru/ (дата обращения: 01.05.2023).
3. Гизетдинов, М. Р. Репозиторий проекта курсовой работы / М. Р. Гизетдинов. — Текст : электронный // GitHub : [сайт]. — URL: https://github.com/markus0440/Automatic-Warehouse (дата обращения: 16.05.2023).