RoboNox-2020

Team:

Terminators

28th June 2020

Table Of Contents

Introduction

Couriers and online shopping have proved to be an important part of today's fast-growing business firms as well as regularly this is mainly done by humans especially in remote areas. We have for many decades anticipated the era of robotics. We are concerned that robots may steal our jobs and spy on us. We imagine that they will arrive not in ones and twos but in vast armies ready to alter forever life as we know it. But the reality is far more different. Robotic technologies are arriving slowly among us to easy human efforts.

In this report, we will throw light on how Robots can ease the emerging technology trend and can make the process fast and more efficient. The basic idea is to unload parcels by scanning barcodes of the top row and sort them according to their pin codes. And send them via conveyor belt to center of the arc so that they are ready to dispatch for delivery. Activating one conveyor at a time will help to save electricity.

Pre-Existing Designs

1. Siemens AG and Honeywell International Inc. have built machines that pull packages from the back of a tractor-trailer and place them on conveyor belts, whizzing the parcels off for sorting.
2. Few Japanese companies use a Robo arm to pick the parcel from the back of a tractor-trailer.

Assumptions

1. All the boxes are similar in size. Approx 1x1x1 m
2. Bar codes are on the visible front side
3. Motors provide adequate torque
4. Crates are of unlimited size
5. Suction enough to pull boxes

Warning : Don’t try to scan this barcode XD

Our Design

We have picked the best possible design to achieve maximum efficiency and sturdiness, in all sorts of weather condition. This design has been known to followed by **Amazon Warehouses**.
We have changed the unloading design. Following the cutting edge research by **Bestian Solutions**.
Combining the best of both we have reached a unique design, which is fast, effective and has been tested by experts.

Working Principle

1. Camera, on the arm scans the whole environment.
2. Then the ,manipulator is placed at the position of pickup.
3. The suction member is activated and pulls the package on ramp
4. Rotary Joint 1 rearranges its position to go to the respective conveyor
5. Then the Package slides down through the conveyor to the crate
6. The cycle repeats till all packages are recieved
7. In case there is any error, the machine will stop automatically, as human safety is off utmost importance :)

Circuit

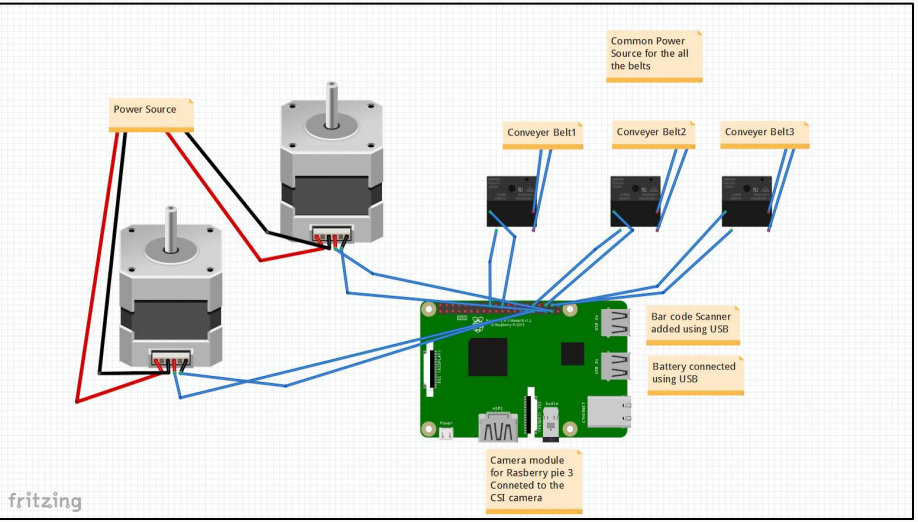


Fig. Board View

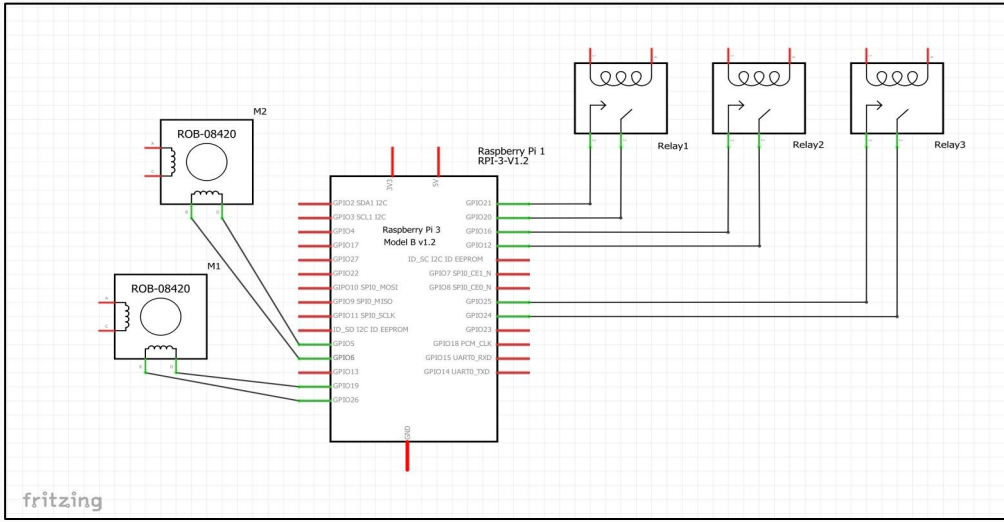


Fig. Schematic View

Code :

C: > Users > P > Downloads > barcode (1).py

```python
'''
Author: Manan Shah
Team: Terminators - Robonox
Date: June 28, 2020
'''

# ***** Import Block *****

from __future__ import print_function
from pyzbar import pyzbar
import numpy as np
import cv2

# ***** End of Import Block *****

'''
Function to decode image to get information from barcode
Find barcodes and QR codes
'''
def decode(im):
  '''
  params: im - The image file

  return: decodedObjects - The result of the decoded image
  '''
  decodedObjects = pyzbar.decode(im) # decode image

  # Print results
  for obj in decodedObjects:
    print('Type : ', obj.type)
    print('Data : ', obj.data, '\n')  # return type is bytes: b''

  return decodedObjects

# ***** End of decode function *****


'''
Function to sort the parcels according to the pincodes
Put your ranges to sort the parcel based on its pincode
'''
def pincode(pincode_int):
  '''
  params: pincode_int - Integer - The pincode id of the places

  It runs the conveyer belt according to the pincode

  return : none
  '''

  # used for this demo: https://finkode.com/hp/mandi.html  Mandi pincodes

  if pincode_int <= 175040:                  # For the first place
    conveyor_1()
  elif 175040 < pincode_int <= 175080:       # For the second place
    conveyor_2()
  elif 175080<pincode_int <=175126:          # For the third place
    conveyor_3()
  else:
    print("Faulty code, unable to determine!")

# ***** End of pincode determiner *****


# ***** start conveyor belts functions *****

# ''' Function for running belt 1 '''
def conveyor_1():
  print("belt 1 executing..")

# ''' Function for running belt 2 '''
def conveyor_2():
  print("belt 2 executing..")

# ''' Function for running belt 3 '''
```

barcode (1).py

C: > Users > P > Downloads > barcode (1).py

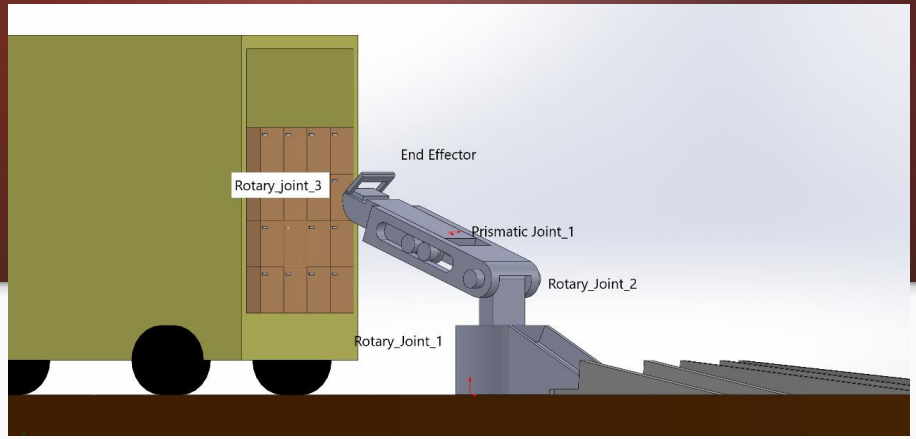
```python
# ***** start conveyor belts functions *****

# ''' Function for running belt 1 '''
def conveyor_1():
  print("belt 1 executing..")

# ''' Function for running belt 2 '''
def conveyor_2():
  print("belt 2 executing..")

# ''' Function for running belt 3 '''
def conveyor_3():
  print("belt 3 executing..")

# ***** End of conveyor belt functions ******


'''
start Display barcode and QR code location(optional)
'''
def display(im, decodedObjects):
  '''
  params:
    im - The image file
    decodeObjects -  The decoded objects

  returns: none
  '''

  # Loop over all decoded objects
  for decodedObject in decodedObjects:
    points = decodedObject.polygon

    # If the points do not form a quad, find convex hull
    if len(points) > 4:
      hull = cv2.convexHull(
          np.array([point for point in points], dtype=np.float32))
      hull = list(map(tuple, np.squeeze(hull)))
    else:
```

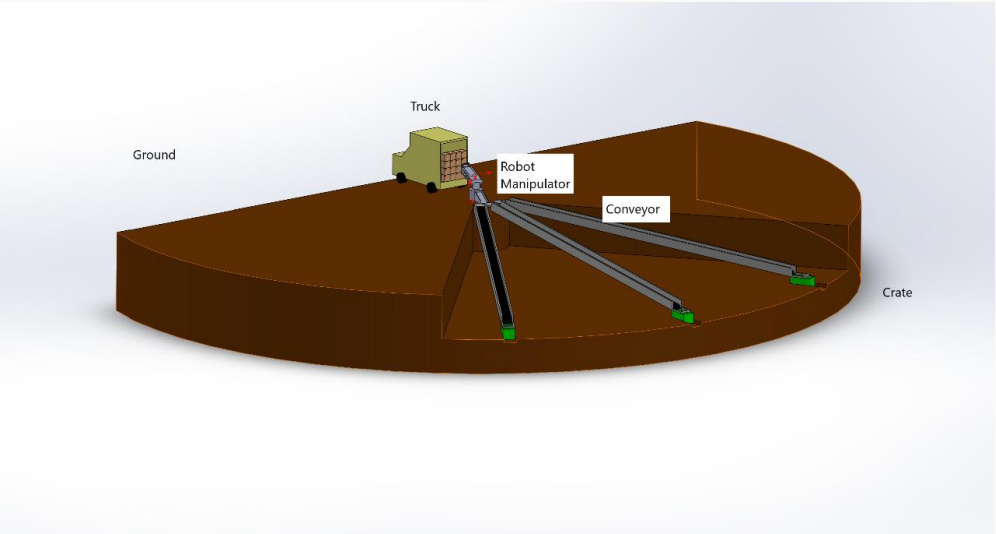
barcode (1).py

C: > Users > P > Downloads > barcode (1).py

```python

  # Loop over all decoded objects
  for decodedObject in decodedObjects:
    points = decodedObject.polygon

    # If the points do not form a quad, find convex hull
    if len(points) > 4:
      hull = cv2.convexHull(
          np.array([point for point in points], dtype=np.float32))
      hull = list(map(tuple, np.squeeze(hull)))
    else:
      hull = points

    # Number of points in the convex hull
    n = len(hull)

    # Draw the convext hull
    for j in range(0, n):
      cv2.line(im, hull[j], hull[(j + 1) % n], (255, 0, 0), 3)

  # Display results
  cv2.imshow("Results", im)
  cv2.waitKey(0)

# ***** End  of Display barcode and QR code location function *****


'''
Function to scan box start
If box found found, make suction, scan the barcode and execute the appropriate conveyer.
else return;
'''
def scan_boxes(x_cord,y_cord,z_cord):
  '''
  params:
    x_cord - The x coordinate
    y_cord - The y coordinate
    z_cord - The Z coordinate

```

Design


End Effector
Rotary_joint_3
Prismatic Joint_1
Rotary_Joint_2
Rotary_Joint_1


Truck
Ground
Robot
Manipulator
Conveyor
Crate

Video of simulation

Video explaining idea

THANK YOU !!!

Jitaoge toh party milegi :)