Springfield Technical Community College

ELEC-485-E51 ADVANCED AUTOMATION

Holiday Ornament Packaging System

December 22, 2014

Course Instructor

Richard Jagodowski

Team Members

Eric

Gayle Smith

Jay

Joe

Kyle

Text

Table of Contents

Introduction

The purpose of this project is to integrate several systems learned throughout the curriculum of the Electrical Engineering Technology program to achieve one common goal. Our group decided to utilize several classes related to automation and controls to create an automated production system. The production system chosen was to box and palletize Christmas ornaments using an automated system. By using Microprocessors, PLCs, Robotics, Pneumatics, Control Circuits, and various sensors, we were able to achieve a process that with minimal operator intervention would efficiently box and palletize Christmas ornaments.

Holiday Ornament Packaging System

System Overview

Our process begins by using a microprocessor to control the feed of open cardboard boxes onto a motorized conveyor. Once a box was set on the conveyor, the box would move a short distance and another box would be feed behind it. This process would repeat until no boxes were left on the slide. After the first box reached “Station 1”, an automated filling process began. With the box set in the proper place the conveyor would stop, and four ornaments would simultaneously drop into the box. If all 4 balls were sensed being dropped into the box, then the conveyor would start again and continue until the next box was sensed at “Station 1”. If all 4 balls were not sensed, the process would not continue and an operator would have to inspect the hoppers for a jam and reset the station. Next, when a full box made it to the sensor at the end of the conveyor, “Station 2” would begin.

From here a robotic arm would grab a lid and place it on the box. Following this, the robot would move the box and place it onto a shipping pallet nearby.

This robot is programmed to stack 4 boxes before the pallet would be considered full. From there a new pallet would be put in place and the process would be restarted.

3D-Printing

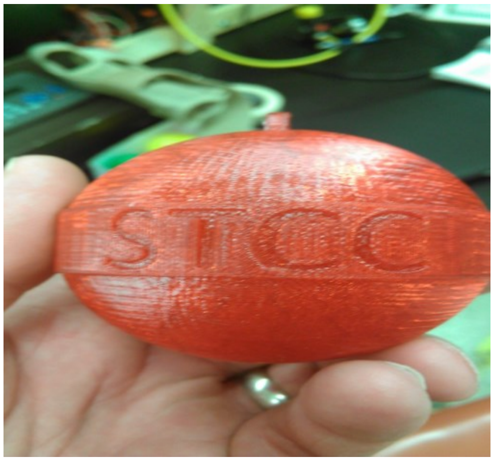
The Design Element

Our team wanted to design and create custom made Christmas ornaments utilizing the MakerBot 3D Printer. Before we made the ornaments, we took into account the materials needed to construct our packaging assembly. On that same note, we also had to account for the number of ornaments built and the time to construct them. Our packaging system used PVC pipes to drop ornaments into boxes, which held 4 ornaments each. These ornaments were staged in the piping by two pneumatic actuators that divided the stack of ornaments. The maximum outside diameter of the ornaments was calculated at 2.85". Our ornaments utilized a solid body design; therefore, the actuator didn't get caught in the ornaments and caused no damage.

There were plenty of designs available for us to try on the MakerBot 2. The designs were found on a website called Thingiverse. Thingiverse is dedicated to those who like to create and share their designs with the online community. After trying several available designs, it became apparent that we needed a hollow spherical design.

Design Software

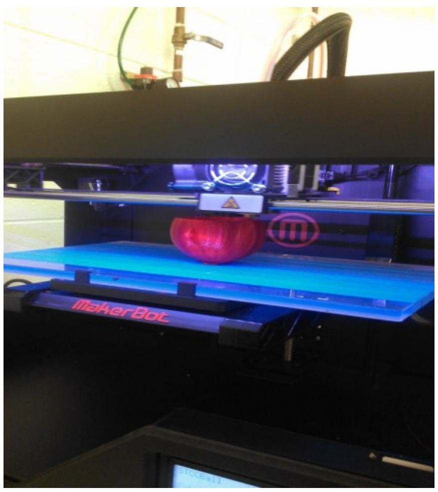
Since the STCC ELEC program did not have any design software for us to use, we attempted to utilize free software called SketchUp. After several days of attempting to create a design on our own, time was becoming an issue. We needed between 24 to 32 ornaments to satisfy the projects requirements. So we had to call in the cavalry! We contacted Tom Minor from the Mechanical Engineering Technology Department, and he was kind enough to help. Tom used the design software SolidWorks to tweak a hollow sphere design that we had found on Thingiverse. Following this, he came up with the design you see below.



The MakerBot Replicator 2

The MakerBot 3D printer was used to build our custom ornaments. It uses a PLA filament to create the designs. The filament is drawn through an extruder that is heated up to 230 degrees Centigrade. The extruder travels along X,Y and Z axises and creates the 3D design layer by layer. We uploaded the STL file to the memory card of the printer and were able to run the print jobs straight from the MakerBot user interface.

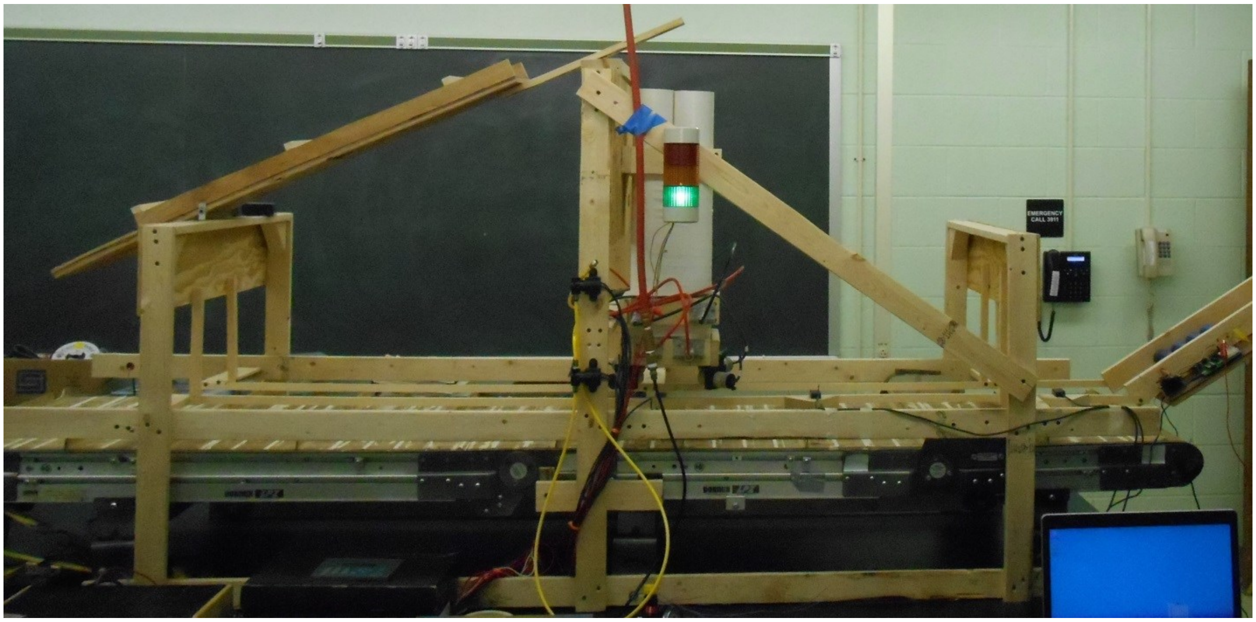
Therefore, we eliminated the need to connect a laptop to upload the file and allowed each team member to print off the required number of ornaments. We made the ornaments in a variety of colors including magenta and yellow, which are S.T.C.C. school colors. Once we had the ornaments we required, it was time to send them off to packaging. See the MakerBot in action below.



The Structure

By Jay Peters

The structure of the project was a frame made of 1” x 3” pine strapping bolted onto the metal frame of a 102” x 30” motorized conveyor. The wooden frame spanned the conveyor by use of three towers made of 1” x 3” pine strapping and ¼” plywood. The towers at the head and foot of the conveyor were approximately 30” tall, while the center tower was 43” tall. The towers were used to strengthen the frame and to attach the ornament hopper, box chute and lid chute to the conveyor. The center tower also allowed for the mounting of the wiring and tubing, along with the hardware for the ornament hopper’s actuators.



Materials list

1" x 3" pine strapping

¼" plywood

¾" x ¾" pine strips

1" fine thread drywall screws

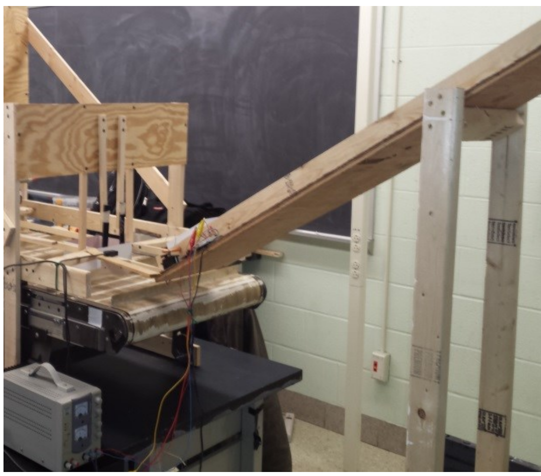
1 ¼" coarse thread drywall screws

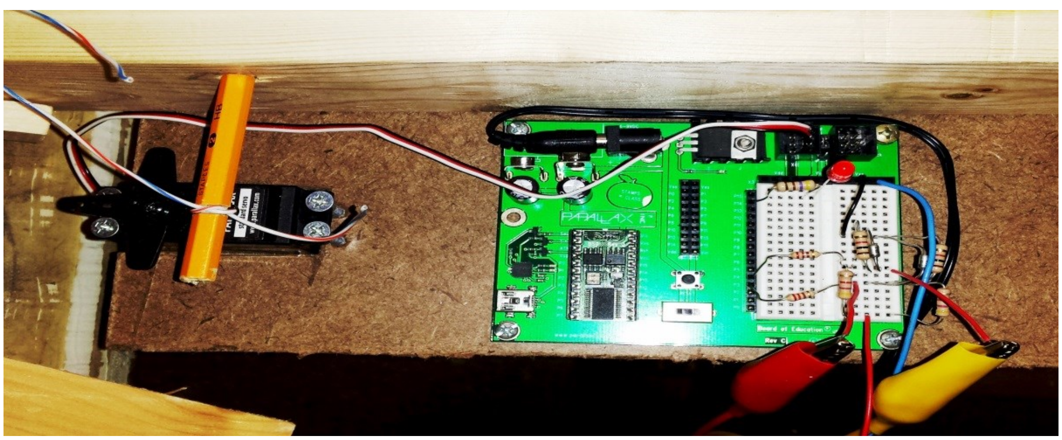
3/8" diameter machine bolts with washers and nuts

Shim material

The box chute was 48” long and angled plywood structure with 2” x 3” lumber attached as legs. The box chute was a gravity fed device with a tethered weight to ensure the timely feeding of the boxes onto the conveyor.   The feed rails of the chute tied directly into the guide rails for the conveyor, which were mounted to the conveyor frame. Both the box chute and guide rails were adjusted to allow 6 ½” to 7” of space between them. The guide rails also allowed for the placement of sensors, wiring and pneumatic tubing along the length of the conveyor. The box chute had a simple gate operated by a servo motor controlled by a Basic stamp. Once a box had activated a diffuse beam sensor, the servo was programmed to close the gate so that only one box at a time would travel on the conveyor.   Once the box had travelled a safe distance, a second diffuse beam sensor would be activated to open the gate and allow another box to enter onto the conveyor. The box chute gate was an independent system and not controlled by the PLC.

A wiring diagram of the circuit needed to convert the 24v diffuse beam sensor signal to the >5v signal used by the Basic stamp, along with the program used to control the gate are in the Appendix.





Materials list

½" plywood

¼" hardboard

1" fine thread drywall screws

1" x 3" pine strapping

2" x 3" lumber

1 Parallax Board of Education

1 Parallax Basic Stamp

1 Variable DC power supply

2 Infrared diffuse beam sensors Te xt

1 Rubber tipped plunger

6' 18ga wire

2 3.3v zener diodes

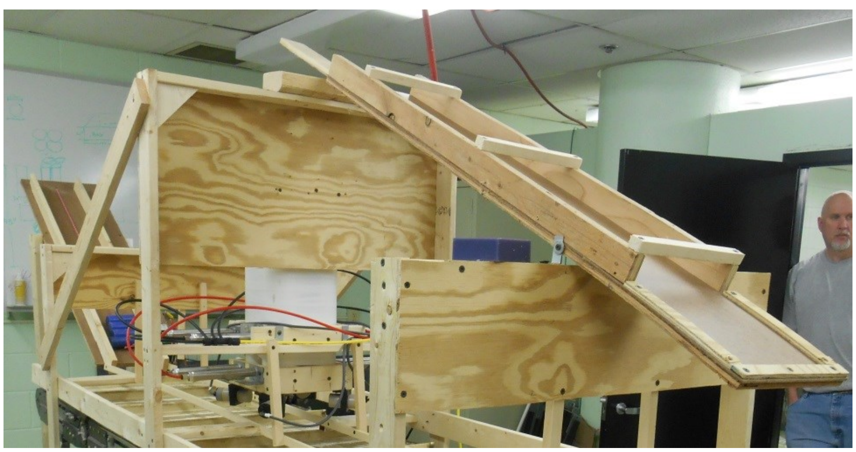
2 1K ohm resistors

2 4.7K ohm resistors

3 4oz wax rollers

5’ nylon rope

The lid chute was a 42” long angled plywood structure attached to the central tower and the tower at the foot of the conveyor. It was positioned so that the each lid was within easy reach of the end effector of the robotic arm. The lid chute was a gravity feed device with a weight to ensure the box lids were positioned properly.



Materials list for Chute

½" plywood

¼" hardboard

½" x ½" pine strips

1" fine thread drywall screws

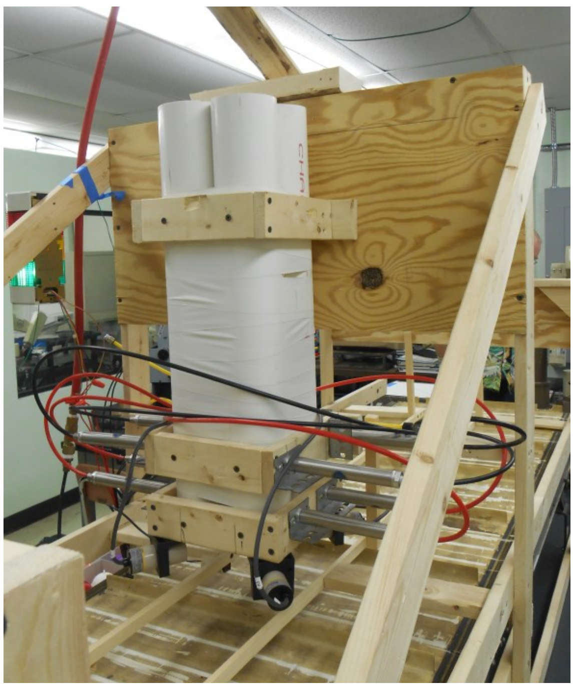
2 2"angle brackets

The Hopper

By Joe Sayles

In order to make our hopper, I purchased 4 PVC pipes that were 3 inches in diameter from the Hardware store. Following this, they cut the piping to 24 inches in length. After that I took to class were I taped the pipes together, and we bolted the hopper to the structure that Jay built.

The purpose of the hopper was to allow the ornaments to fall down into box. Our hopper was very sturdy and strong because it held the sensors and actuators firmly. In summary, the hopper was an important tool for the success of our project.



Programmable Controller

The Ball Drop:

The entire system required a central processor to control all stations and keep track of the overall process. This controller also needed to provide an operator interface. We ended up using a Unitronics Programmable Logic Controller to receive station sensor inputs and send signals to control all stations accordingly. The PLC contained 10 digital inputs as well as 6 relay outputs. Using a DC power supply I was able to power the PLC with 24vdc. The following is a list of Inputs and Outputs.

I1 = Hopper 1 Cap Prox
I2= Hopper 2 Cap Prox
I3= Hopper 3 Cap Prox
I4= Hopper 4 Cap Prox
I5= Station 1 Sensor
I6= Station 2 Sensor
O0= Motor Run
O1= Top Cylinders
02= Bottom Cylinders
O3= Robot Start
O5= Green Light

Kyle was able to program the PLC using Unitronics' u90 ladder software and a Unitronics specific programming cable to interface between the computer and PLC. All inputs were digital, and 24VDC required for each sensor. Outputs used relays to control the 24vdc signals. As a safety device, button #0 on the HMI will not allow anything to run. Button #1 must be pressed in order to tell the PLC to let the motor run. The program begins by allowing the 1/3 HP DC motor to run and move the conveyor forward. A green light turns on to indicate the conveyor is running. If the conveyor ever stops, a red light will go on. As the conveyor moves, the program looks for any changes in the Station 1 or 2 input sensors. When Sensor 1 is high, it beings the hopper fill subroutine. This subroutine will stop the conveyor, and pause 2 seconds before dropping the ornaments in the box. This is done by discharging a pneumatic solenoid on the bottom section of all 4 hoppers. When this happens, the bottom pneumatic cylinders open, and allow 4 balls to drop in the box. While the balls are dropping, the PLC monitors 4 capacitive proximity sensors. As a ball passes by a sensor, it causes the PLC input to go LOW.

When the signal goes low per sensor, it adds together to see a total count of balls that fell. If the sensors add up to 4 total, it will reset the counter, and continue with the subroutine. The sensors conditions can be viewed on the HMI, 2 line display of the PLC. It will display 0 if not sensed and 1 if sensed. This way if the operator sees that not all four have changed to 1, they know to inspect hoppers. Ex. When station starts display reads 0000. If it displayed 0111 after balls have dropped, the operator knows that hopper #1 is caught up. The operator can fix the respective hopper and press button #5 on the HMI to reset the process. If all 4 are sensed, then the bottom cylinders will extend, and top cylinders will retraced for 5 seconds to allow the hoppers to refill and get ready for the next box.

Once this subroutine is complete the output for the motor will close again and the conveyor will begin moving and the full box will move out of the way of Sensor 1. This will continue until either Sensor 1 or Sensor 2 goes low again.

If Sensor one goes low, it will repeat the hopper station subroutine, if Sensor 2 goes low, it will begin the robot subroutine. When subroutine 2 begins, the motor stops and the plc simply pulses and output to the robot interface through a 24vdc signal. At this point the plc will then wait until Sensor 2 goes high again, then engage a buffer timer for 2 seconds to make sure the box has been completely removed from the conveyor. After these 2 seconds are up, the motor beings moving the conveyor forward again and it continues until Sensor 1 or 2 go low again. If for some reason Subroutine 2 cannot be completed, pressing button #6 on the HMI display will reset station 2 subroutine. This will repeat indefinitely unless a box is NOT removed from station 2. This indicates the pallet is full and operator will have to change out pallet. Each time subroutine 2 is completed, the PLC will add 1 count to a memory integer. This integer will be displayed on the HMI as total number of boxes completed. This can be reset by pressing #4. If at any point the process needs to be stopped, the operator can simply press button #0 on the HMI to stop the PLC from allowing the motor to run and a red light will stay on.

Power Source

Full Wave Rectifier

We used a full wave rectifier (ECG-5306 P312) to transform the 120VAC (from the wall outlet) to approximately 133 VDC to power our DC Motor. Three 2P5-P50 capacitors were utilized to smooth out the DC rippling, which caused the DC motor to run smoother. Overall, our full wave rectifier circuit was very successful because it made the conveyor run with out failure.

Packaging with Kawasaki Robot

By Gayle Smith

The Process

The robotic arm is hovering over the cover until the box triggers the photo sensor at the end of the conveyor. The programmable controller then sends a signal change to the I/O board. The Kawasaki program is in a continuous loop waiting for this signal change. When the signal change the 'IF THEN" statement executes the program. The robotic arm approaches the cover, closes the end effectors and lifts the cover from the slide. There are "SET" locations for the robotic arm to locate to in order to avoid collision with the framework. The cover will move away from the slide and approach the box. Here it places the cover on the box, opens the end effectors and then relocates on the box, closing the end effectors and picking it up off the conveyor. The robotic arm then transports the box/cover to the pallet and places it down.

The counter on the computer monitor reads 1. The end effector opens, then rises up, and returns to a set position over the cover to hover until the next box triggers the photo sensor on the conveyor. This process will repeat four times. The computer monitor will read "pallet is full, empty & type proceed" when the 4th box is placed on the pallet. The box/cover are stacked as follows: Box 1 (boxA) is placed in the far back position; Box 2 (boxB) is placed in front of box 1; Box 3 (boxC) is placed on top of Box 1; and Box 4 (boxD) is placed on top of box 2.

Impediment

There were a few impediments during programming. Selecting positioning points of the cover and box happened at the end of the project and program adjustments had to be corrected. The sensor signal from the programmable controller sent a signal but the program would not read it correctly. A loop was created to overcome this. Reaching the robotic arm positioning limits would create an error in the controller. This led to a robotic losing the RY-axis.

The LIMP command needed to be used to set everything back to a calibrated placement. This alone took several hours of troubleshooting.

Appendix

Gate Program

```
' Box Chute program
' {$STAMP BS2}
' {$PBASIC 2.5}
' Purpose is to control the gate of the ELEC-485 project using a servo and two diffuse light sensors.
' Written 12/3/2014

COUNTER VAR Word
DEBUG "Begin program."
FOR COUNTER = 1 TO 5
PULSOUT 14, 600
     PAUSE 20
     NEXT
DO

  IF IN3 = 0 THEN
   FOR COUNTER = 1 TO 5
  PULSOUT 14, 800
   PAUSE 20
   NEXT

ELSEIF IN4 = 0  THEN
 DO
 LOOP UNTIL IN4 = 1
 IF IN4 = 1 THEN
 FOR COUNTER = 1 TO 5
  PULSOUT 14, 600
    PAUSE 20
    NEXT

ENDIF
ENDIF

LOOP
```

Appendix Packaging Program used by Kawasaki Robot

PROGRAM, Packaging, Kawasaki Unimate Controller, model 280

```
1.  SETI COUNT=0
2.  5APPRO SETA, 10
3.  SPEED 20
4.  OPENI 0
5.                      10 IFSIG -3,,,, THEN 20
6.                      20 APPRO COVERA, 10
7.                      COSEI 0
8.                      APPRO COVERAA, 10
9.                      MOVE COVERAA
10.                     DEPART 10
11.                     APPRO SETB, 10
12.                     APPRO SETBB, 10
13.                     DEPART 10
14.                     APPRO COVERB, 10
15.                     OPENI 0
16.                     APPRO BOXA, 10
17.                     MOVE BOXA
18.                          CLOSEI 0
19.                          DEPART 10
20.                          APPRO SETC, 10
21.                          SPEED 40
22.                          SETI COUNT = COUNT + 1
23.                          TYPEI COUNT
24.                          IF COUNT EQ 1 THEN 100
25.                          IF COUNT EQ 2 THEN 200
26.                          IF COUNT EQ 3 THEN 300
27.                          IF COUNT EQ 4 THEN 400
28.               100 APPR0 STACKA, 10
29.                          MOVE STACKA
30.                          OPENI 0
31.                          APPRO STACK AA, 10
32.                          GOTO 20
33.               20 DEPART 10
34.               SPEED 50
35.                          GOTO 5
36.               200 APPR0 STACKB, 10
37.                          MOVE STACKB
38.                          OPENI 0
39.                          APPRO STACKBB, 10
40.                          GOTO 20
41.               300 APPR0 STACKC, 10
42.                          MOVE STACKC
43.                          OPENI   APPRO STACKCC, 10
44.                          GOTO 20
45.               400 APPR0 STACKD, 10
46.                          MOVE STACKD
47.                          OPENI 0
48.                          APPRO STACKDD, 10
```

49. PAUSE PALLET IS FULL, EMPTY & TYPE PROCEED

Appendix Materials List

Unimate Puma 200 Robot

Kawasaki Unimate Controller model 280

I/O Module model 320

Teach Pendant

Computer Monitor

Keyboard

5 ¼" Floppy Disk Drive

5 ¼" Floppy Disk

Wood Platform (robot base)

Cardboard Box (pallet)

Custom End Effectors

(4) Puma mtg hardware

(4) #8-32 Pan Head Machine Screws, Washers & Nuts

(4) #8-32 Pan Head Machine Screws

1" Painters Tape

Power Strip

Pneumatic Tubing

Pneumatic Regulator

Screw Driver

1- Unitronics u90 R16 PLC

1- Variable DC Voltage power supply

3 – 24vdc ice cube relays/bases

1- 12 pole terminal strip

30' – 18ga blue wire

4 – Capacitive Proximity Sensors

1 – Infrared Sensor

1 – Thu-Beam Sensor

1 – 1/3HP DC Motor

1 – 6' Belt Conveyor

1 – AC/DC Rectifier Circuit

1- Heavy Duty Shoebox Enclosure