AUE 6620 Digital Automotive Manufacturing

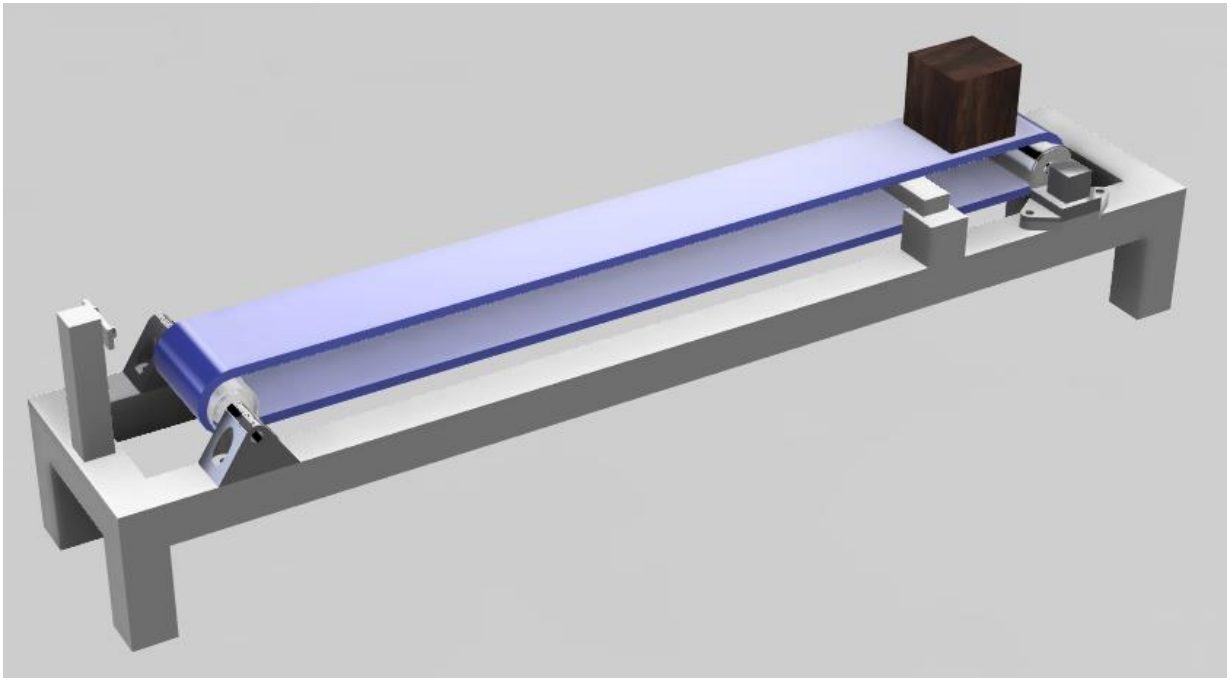
Semester Project Report (Milestone 5)

Automated Conveyor System

Team **'Automates'**

Team members:

1. Chirag Mutha
2. Shubham Garde
3. Mihir Bangale



Problem Statement -

A single standard conveyor belt speed cannot be used for parts of varying shape, size, and weight. There are high chances of parts getting damaged when they are transported at higher speed as they can topple. This can also hurt line operators when heavier parts are transported at high speeds on the conveyor.

Moreover, when a standard conveyor belt runs at a constant specified speed for parts of varying weights, this speed is more than necessary. This would take more operation time and thus more operational cost. A solution for this can be setting up individual conveyor belts for parts varying in shape, size, and weight. However, a setup like this will require far more resources such as floor space, operators, and capital.

Why should someone read your report or what should they learn from your work?

This report is useful for someone who is interested in solving a real-life problem in the manufacturing industry but has little experience in working with electronic components or programming. Also, one can learn sensor fusion after reading this report and can get motivated to work on cool projects based on sensor fusion. This report will also give an insight into why proper planning is essential before executing any task. One will learn from our mistakes and can get a step ahead of others as the person would be aware of the challenges and issues, we faced that were common in real life and be prepared for it.

Goal of the Project -

The goal of this project is to build an automated conveyor system that would carry parts from the point they are loaded onto the conveyor to the respective stations. This conveyor system should reduce run at a speed decided based on the weight of the part. For a heavier part, the belt should run at a slower speed than a threshold value and vice-versa. As a result, the overall operational time should decrease.

Metrics used -

1. The operation time of the new conveyor belt should take 25% less time than the manually operating conveyor (i.e., pick up the part, place it on a weighing scale, set the corresponding speed, place it onto the conveyor)
2. The conveyor belt should carry a 1 kg part at the corresponding running speed.
3. It should be precise in stopping at accurate stations within a tolerance of ± 20 mm
4. The actual belt speed achieved should be within a tolerance limit of ± 2 rpm (depends on the motor selection)

Project Scope -

The scope of the project can be understood using the following points:

Inputs-

The inputs fed to the system help in gathering the real-time data from the system and making necessary decisions based on this data.

1. Position of the part using an ultrasonic sensor:
   An ultrasonic sensor is placed at one of the ends of the conveyor belt. It measures the distance of the part along the length of the conveyor belt. This information is fed continuously to Raspberry Pi.

2. Position of the part using an infrared sensor
   An infrared sensor is used to detect if the part placed on the conveyor has reached the location where a load cell is mounted under the conveyor belt. Once the part reaches the location, the motion-detecting sensor sends a signal to the Raspberry Pi.

3. Weight of the part using a load cell
   This is the primary data for determining the speed of the conveyor system. The load cell used is a strain gauge load cell which converts the magnitude of the load into an electrical signal. This value is fed to the control logic to determine stepper motor speed.

Outputs-

1. Motor speed control
   Control logic is built into the code which has a decision criterion. This decision is based on the signal input from the load cell which essentially tells the weight of the part loaded onto the belt. Using this input, the stepper motor speed is decided in terms of the number of pulses the motor should run.

2. Weight of the part
   This is a byproduct of the decision made using control logic. This output can be used in a manufacturing setting for other applications such as packaging of the part, material handling processes down the line, etc.

Limitations –

Taking into consideration the constraints that we had while working on this project such as the time frame for completion, previous experience of working with electronics and programming, etc., we decided to limit the scope of this project to the following points:

1. Transporting only one part at a time using the conveyor:
   This can be deemed as one of the major drawbacks if we think of scaling this concept into a manufacturing facility. A manufacturing facility has a complex operation. This consists of child parts, unfinished parts, and finished products being carried on the conveyor system. A conveyor belt that runs for long lengths on a shopfloor carries more than one part simultaneously.

2. Part weight limited to 1 kg due to belt tension constraints
   Considering the size of the conveyor and the material used for the conveyor belt, we had set this restriction to the part weight. The belt sags if a part heavier than this is placed on the belt. As a result, the motor has to generate more torque to drive the belt. However, the stepper motor used cannot generate enough torque.

3. Part size limited to 10cm x 10cm x 10cm
   Taking into account the available belt width of 6 inches and conveyor length of 1 meter, we decided to not use a part more than the specified dimensions. Also, if the height of the part is too much, the part would fall off the belt when traveling at high speed due to its higher center of gravity.

Did you meet your entire scope?

Yes, we met our entire scope of the project. The only part we would have improved upon was using a high-powered DC motor instead of a stepper motor. Also, we need to redesign the wiring of the model. Rather than making the whole connection look messy, we would make it more clean and hidden which would add aesthetics to the model.

Work Distribution -

The entire project work was divided into several tasks listed below. The work distribution for those tasks is as follows:

Sr. No.	Tasks	Chirag	Shubham	Mihir
1	Finalize Bill-of-Material			
2	Components Delivery			
3	CAD modeling and 3D printing			
4	Component assembly			
5	Integration of sensors with conveyor			
6	Python programming			
7	Sensor calibration			
8	Testing the model			
9	Report writing			

Approach taken -

The project work was divided into various tasks which could help us plan the timeline and work distribution better. Some of the tasks had to be carried on simultaneously because of the time constraint and the need to learn the basics. For example, during the step of sensor integration, we had to study each of the sensors individually before making further connections. With this going on, we also decided to start building the actual physical model of the prototype. Even though the physical structure seems simple, it takes more time than anticipated to collect the necessary parts and work on the assembly.

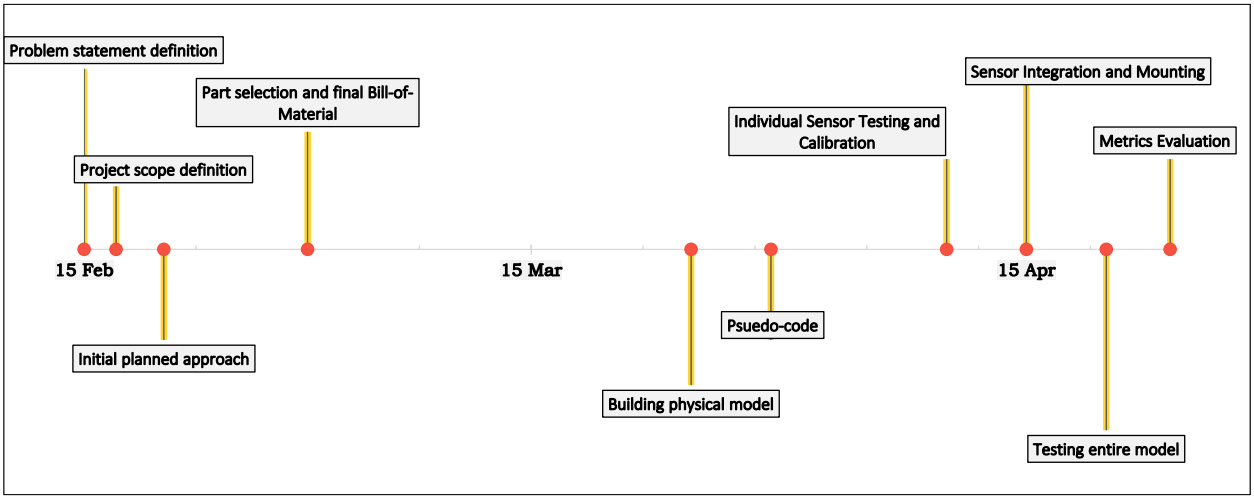
Following are the tasks that were decided for effective work planning:

1. Problem statement and Project scope definition: This is the very basic step when taking upon any project. These two definitions act as foundations for subsequent tasks. A problem statement gives the right direction whereas the project scope helps us stay on track. We defined these two in the initial phase of the project.
2. Sensors, components, and their specific models were decided as per design requirements and compliance with each other. This is one of the most crucial steps. The type of sensors and motor to be used depends on the desired function of the prototype. As per the project goal, we decided on the components and finalized the Bill-of-Material.
3. Understanding the operation of Raspberry Pi, motor hat, and sensors is an essential step. An ultrasonic sensor, infrared sensor, and load cell are studied for their operational range and limitations. For example, the load cell has to be calibrated as per our loading conditions.
4. Prepare pseudo-code and finalize the flow of operation. Once the individual sensors are studied and we understand the basic working, we can start working on finalizing the flow of operation. Then, we can write a pseudo-code which essentially is the logic behind the actual code to be implemented.
5. Build a sturdy physical model: While working on preparing a pseudo-code and studying the electronics part, we simultaneously started building a physical model. It has a basic aluminum frame structure with the necessary sensor mountings as shown in the picture below.
6. We tested the sensors individually and then integrated them by referring to the specific connections for each component. This is an imperative step as it helped us troubleshoot easily during the testing phase of the prototype. For example, the load cell has to be calibrated as per our requirements.
7. Test the entire setup and tune it according to the predetermined metrics. Once the entire assembly was done, we tested the setup to check if the conveyor is running as per expectations. We also checked if the metrics that were decided in the earlier phase of this project are being met.

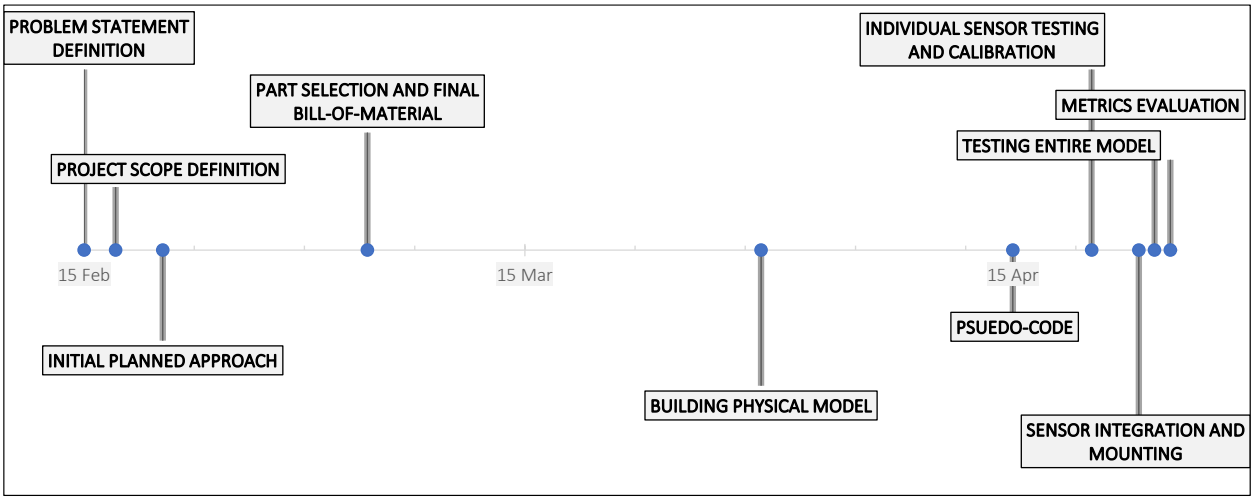
Planned tasks vs Actual tasks -

The initial approach includes listing out all the necessary electronic components (sensors, motor, and other hardware components) concerning their specifications, cost, and availability which we planned to use. Also, writing a pseudo-code to design the control logic for the speed control way earlier in the project development phase. At a later stage, appropriately planning the building of a physical conveyor model is an important task considering 3D printing constraints and sensor mounting fixtures. We also need to check the compliance of each team member with each other to successfully execute the project.

Planned timeline -



Actual timeline -



Required Resources -

Item/Description	Quantity
Stepper driver hat (12V PSU required)	1
Ultrasonic Sensor (requires grove cable)	1
Raspberry Pi 3 B	1
12V power supply (5 Amp)	1
MicroSD Card (32 GB, 2 pack)	1
3-pin JST SM Connectors (pre-wired, 20 pairs)	2
Bearing	4
Barrel plug M/F pair	1
Jumper wires	1
Fuses (3A)	1
RPi case (plastic, GPIO, and camera compatible)	1
Terminal bridges	1
Crimp-on spade terminals (for 2mm terminals)	1
Fuse Holder	1
Standoff 12mm with M2 BC	2
Spade terminals	12
Load Cell/HX711 - 1kg	1
DC Motor	1
5v power supply	1
Conveyor Roller	1
Infrared Distance Sensor	1
Conveyor Belt	1
6mm flange coupler	1
45x45 frame	
Fasteners (Pack of 4)	16
Brackets	32

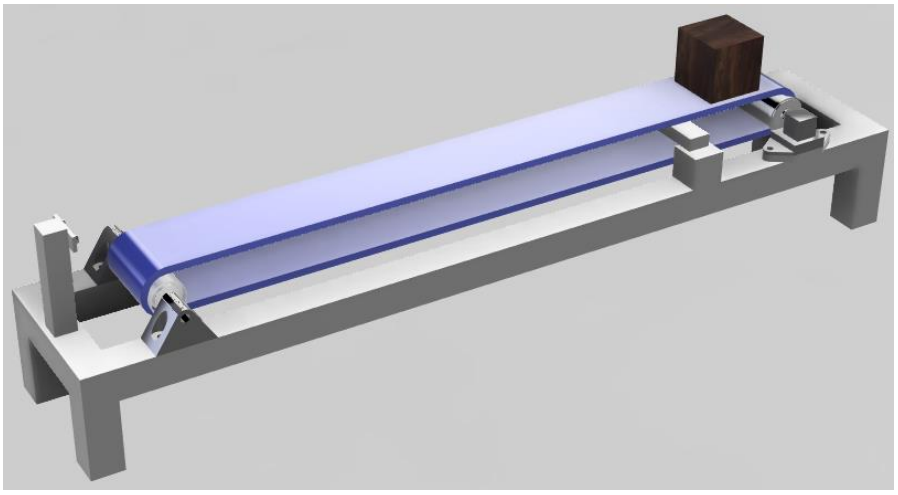
Project Results -

Yes, we achieved the solution to our problem and the model performed as it was meant to. The only downside was the unavailability of a high-powered DC motor, so we ended up using a stepper motor which didn't give the required output of smooth belt motion.

Initial Phase of Project –

CAD Model of Conveyor Belt -

Designing the CAD model for the conveyor belt system to estimate the exact dimensions for structural components as well as planning the sensor mounting location on the actual model. The cad model helped us to plan and visualize all our mounting locations on the model with correct accuracy.



CAD model

Component selection and part ordering –

Made the list of all the required components for designing the conveyor belt physical model, sensors, drivers (Motor driver, loadcell driver, etc.), and other electronic components. The bill of materials below contains all the components (with their price and links to buy) used by us for this project.

Final Phase of the project –

Sensor Testing with python code and full integration of sensors with code -

Wrote the code for individual sensors and tested them individually first. After confirming the proper functioning of the sensors, we did the sensor fusion and code integration for all the components according to their use and tested them. Made the required changes to the code.

Fabrication and Sensor placement –

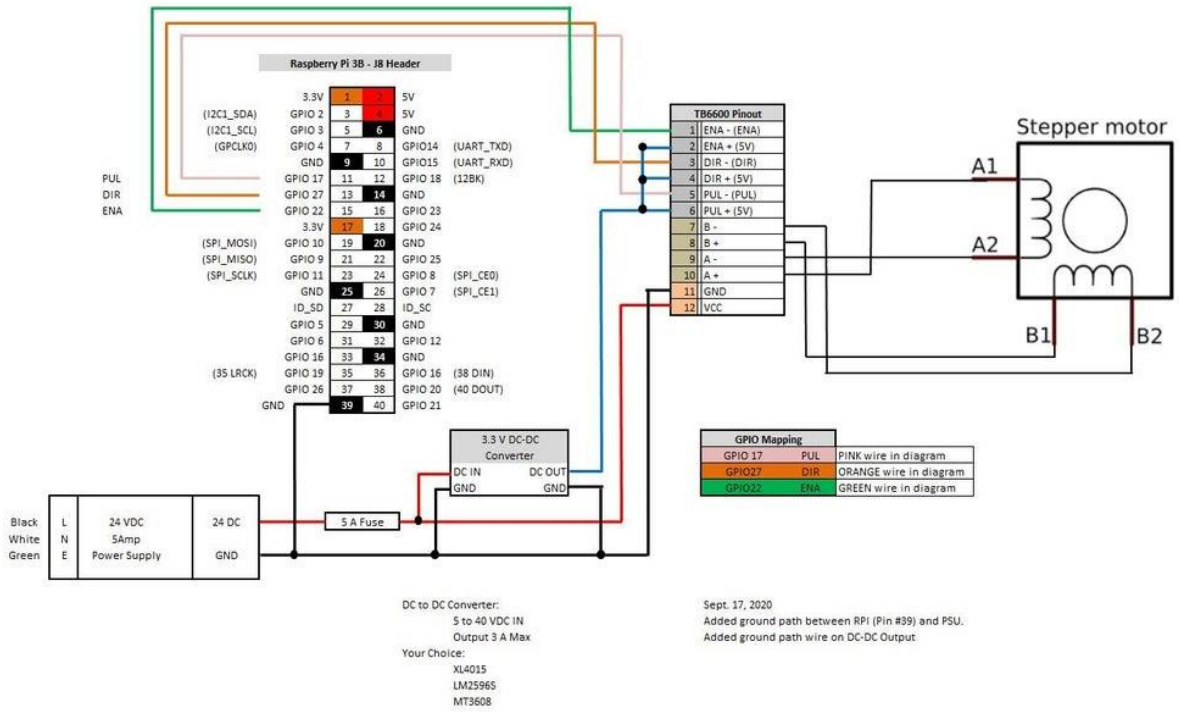
Fabricated the whole model and mounted all the sensors in their respective locations. Fabrication was carried out at CMI.

Wiring Connections of all the electronic components with Raspberry pi –

(Note - All the connections were given GPIO pin no. rather than board pin no. in the python code. Almost all the connections for the components are the same, the changed GPIO pin connections are mentioned with the respective component wiring diagram.)

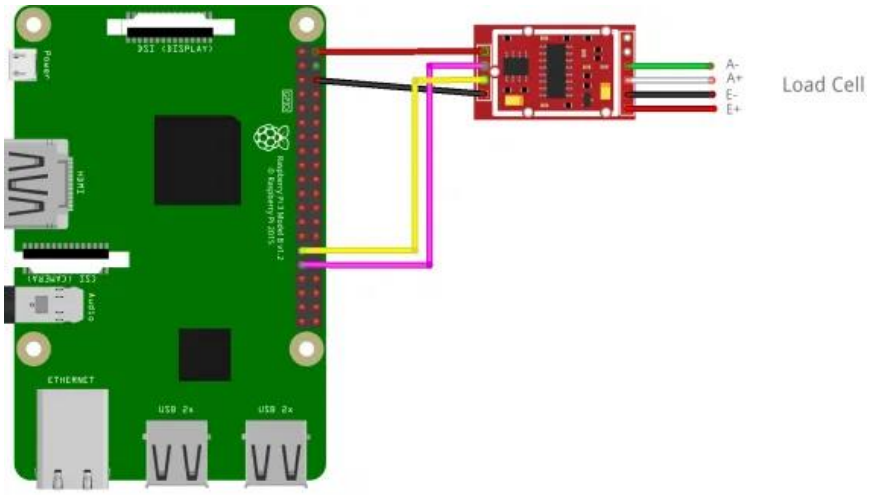
Stepper Motor + TB6600 Motor Driver + Raspberry Pi Wiring diagram –

Reference - https://www.instructables.com/Raspberry-Pi-Python-and-a-TB6600-Stepper-Motor-Dri/



Raspberry pi + Load cell + HX711 (Load cell driver) -

Reference - https://tutorials-raspberrypi.com/digital-raspberry-pi-scale-weight-sensor-hx711/

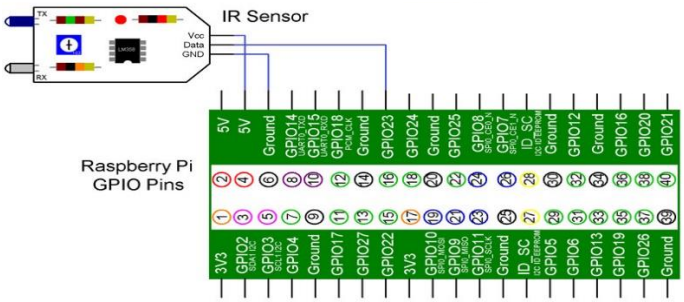


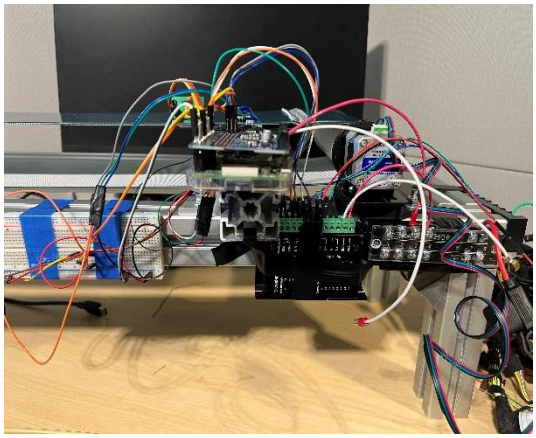
Ultrasonic sensor + Raspberry pi –

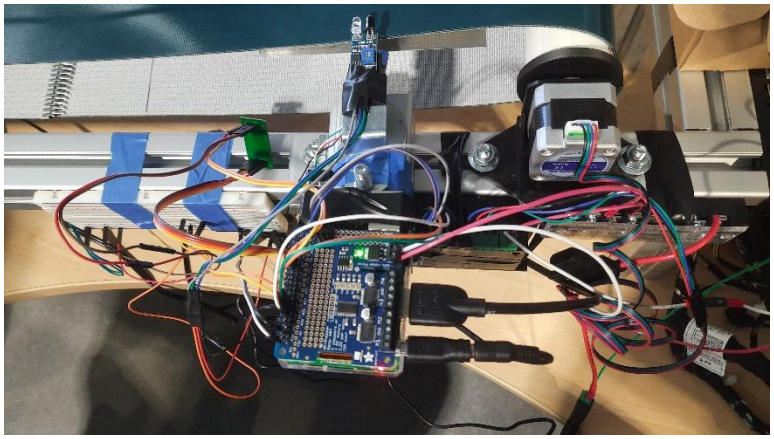
Note – SIG pin is connected to GPIO pin 23, VCC to 5 V, and GND to the ground of Raspberry pi.
Reference - https://wiki.seeedstudio.com/Grove-Ultrasonic_Ranger/

Infrared sensor + Raspberry pi –

Note – Data pin was connected to GPIO pin no. 24







Actual Wiring of model

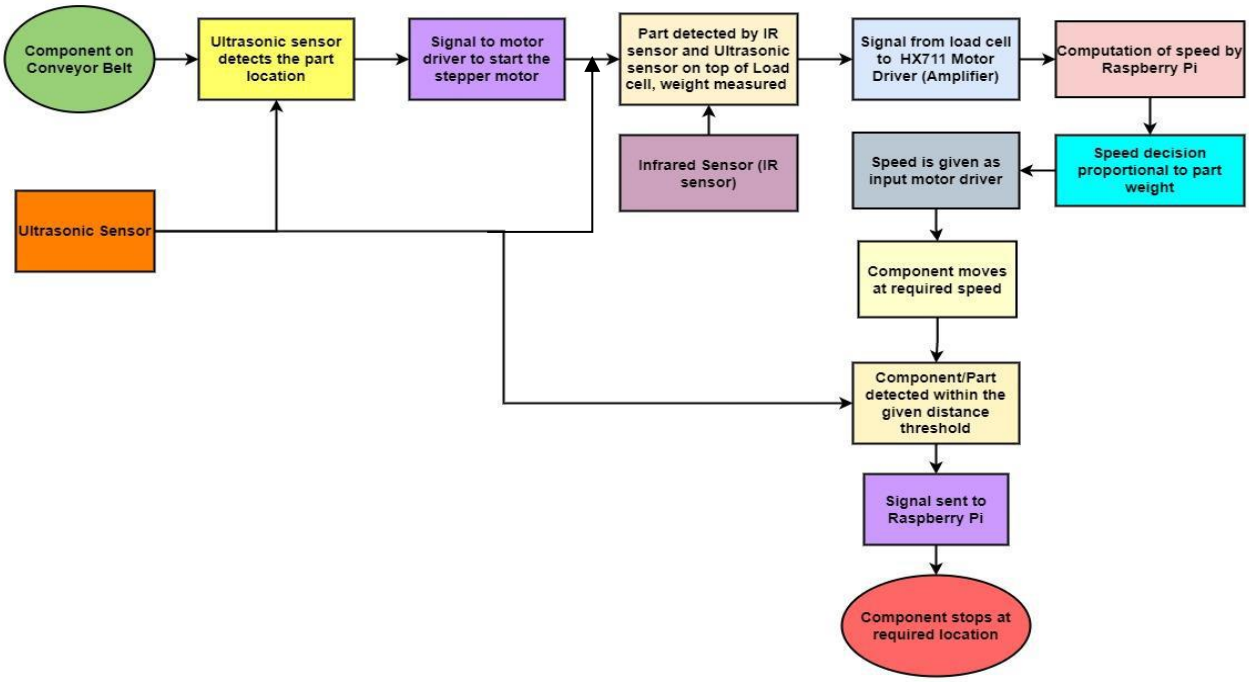
How does the model work? –

Pseud code and Flow chart of Working model of conveyor system

 # Calculate the distance of component from ultrasonic sensor
Place the Component
Measure the distance (Time of flight to distance conversion)

If distance > threshold:
    Motor speed = low, till the component reaches threshold distance (within tolerance)
    Infrared detects parts
    Measure weight
    Compare weight value with the class of speed
    Assign class of speed as input to motor

Stopping criteria
If distance <= threshold:
    Stepper motor speed = 0



Explanation -

When the conveyor system is switched on the ultrasonic sensor starts to measure the distance (in cm) continuous of an object on the conveyor belt. The load cell also calibrates to zero value when the system is switched on. Initially, the motor runs at a constant low speed for all types of part weights to bring it safely on the load cell to measure its weight.

As soon as the infrared sensor detects the part and the distance of the object/part from the ultrasonic sensor is within the threshold value set for the exact stopping of the part on the load cell the power from the motor is withdrawn by the raspberry pi which stops the part on the load cell which can be seen in the below replica of the actual working model.

Then the load cell measures the weight of the component in grams whose value is sent to the raspberry pi. The raspberry pi then computes and allocates the motor speed to the stepper motor depending on the already set weight and speed ratio.

After that, the conveyor belt starts running at the designated speed and the ultrasonic sensor again starts measuring reducing the distance between the part and the ultrasonic sensor.

As soon as the part reaches the destination (stop location) ultrasonic sensor sends a signal to raspberry pi to stop the motor as the threshold distance is reached. So, the raspberry pi stops the stepper motor and brings the conveyor to a static state. This is how the entire automated conveyor system was working.

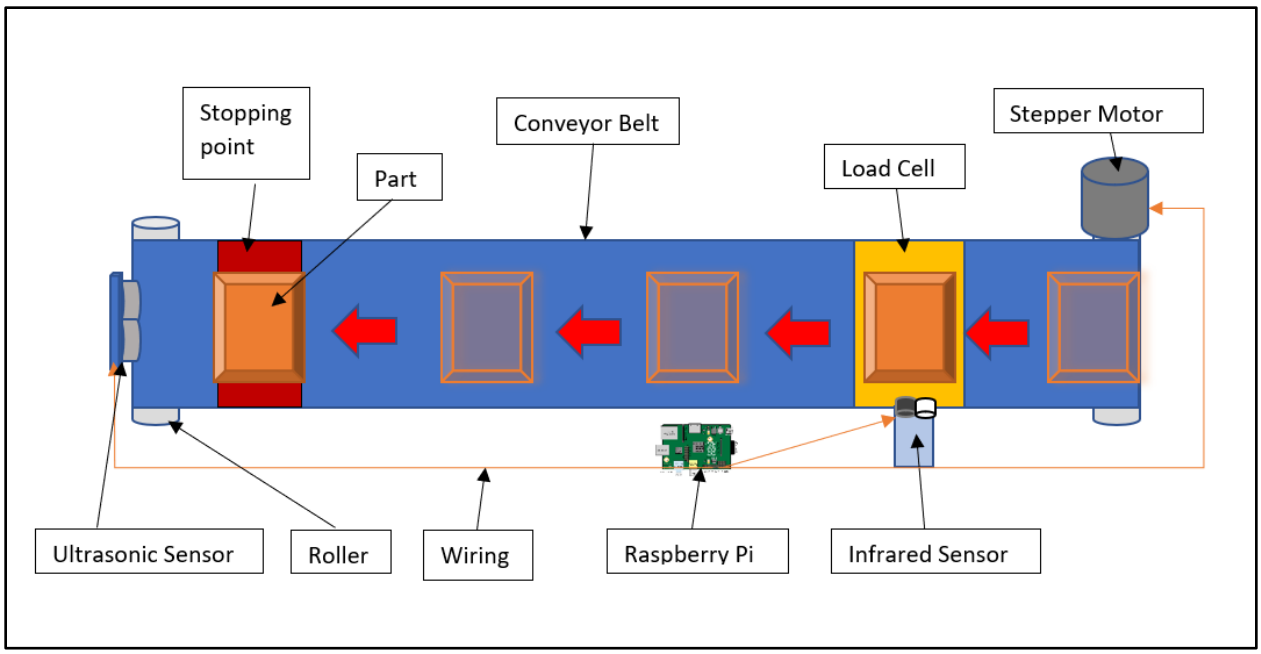
Changes made due to stepper motor –

Initially, we used a DC motor, but it didn't have enough torque to run the conveyor, so we switched to a stepper motor as we didn't have enough time to get a new high-powered dc motor.

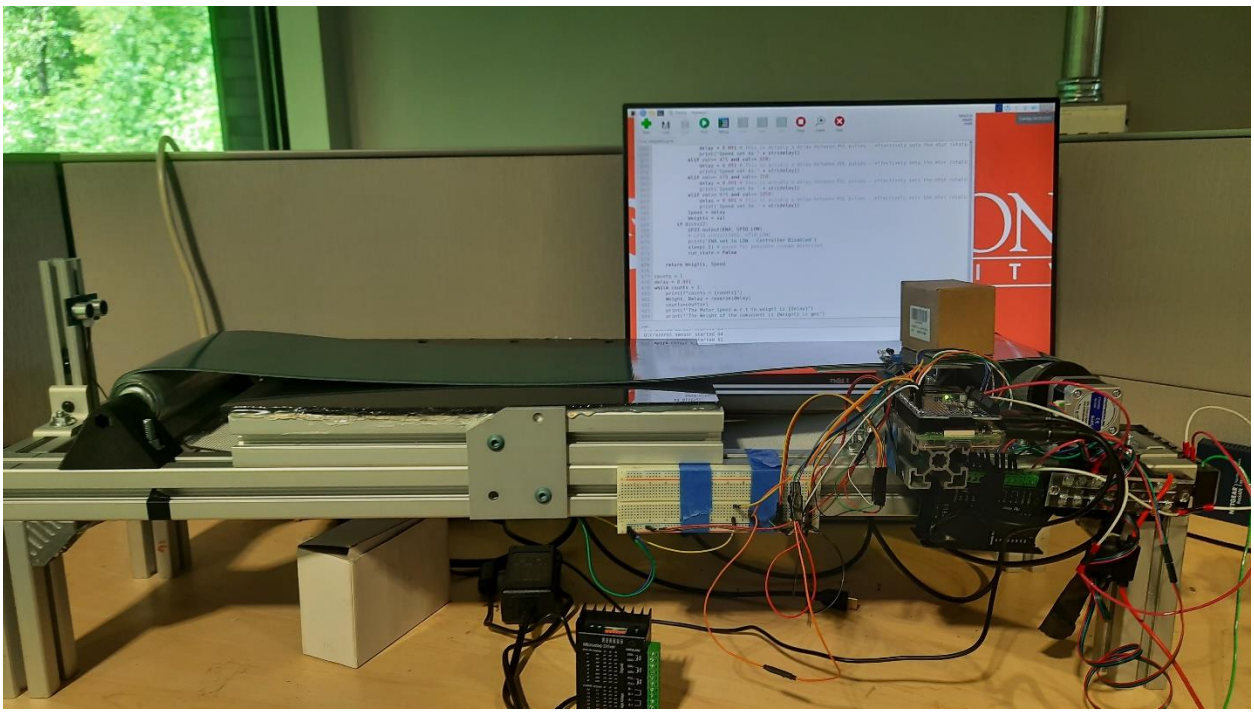
We had to do the necessary changes to the code as we switched from a DC motor to a stepper motor whose speed is controlled with pulse modulation. The switching from dc motor to stepper motor in the code part as well as the physical model was easy with some few modifications in the design for a motor mount.

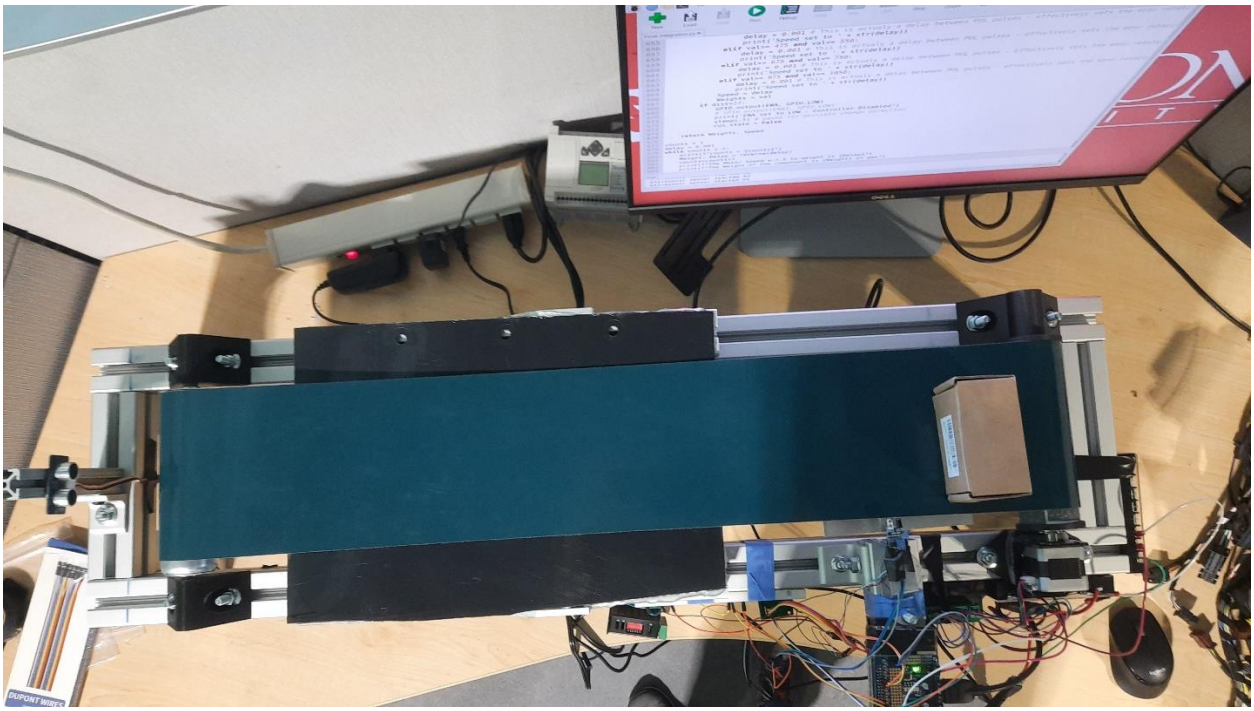
TB6600-4A40V stepper motor driver was added to drive the motor. Also, we added the high current board connectors to make the required connections.

The biggest hurdle with a stepper motor is you must control the speed of the stepper motor in pulses which causes some jerky movements as well as it was not giving us the required top speed. This issue was solved by increasing the pulse duration. Also, the stepper motor didn't have enough power to move the conveyor belt to the speed we wanted even after running it at 3 Amps though it was designed for 1.7 Amps. (Important Note - We supplied the stepper motor high current in presence of the Professor just for a couple of minutes and the current was again lowered to 1.7 Amps after the testing).



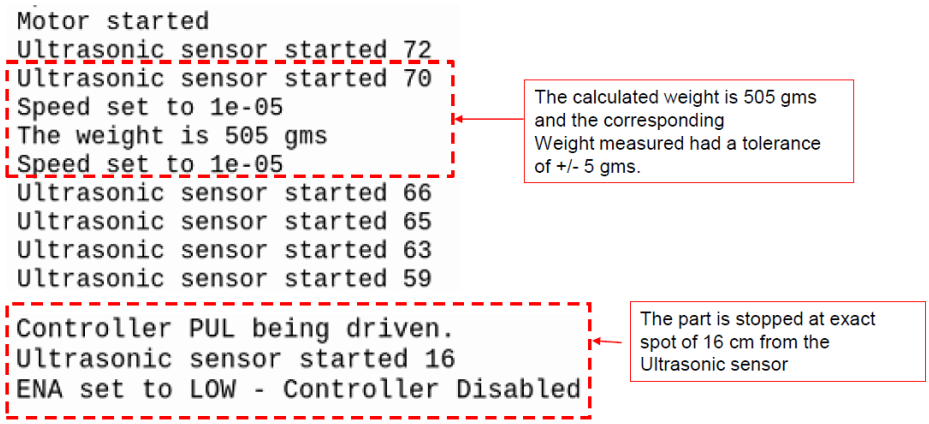
Schematic Diagram





Actual Model

Results -



Obtained Results

The above screenshot is showcasing the results obtained from the actual run of the model. The model is designed to work with a maximum weight of 1000 grams because of component specification constraints. The output of the system is weight in grams and the stopping distance (in cm) when the part reaches its destination. The load cell had an accuracy of +/- 5 grams which was better than initial expectations. This was achieved only because of the stopping of the part at the exact point on the load cell. The system is highly robust in functioning.

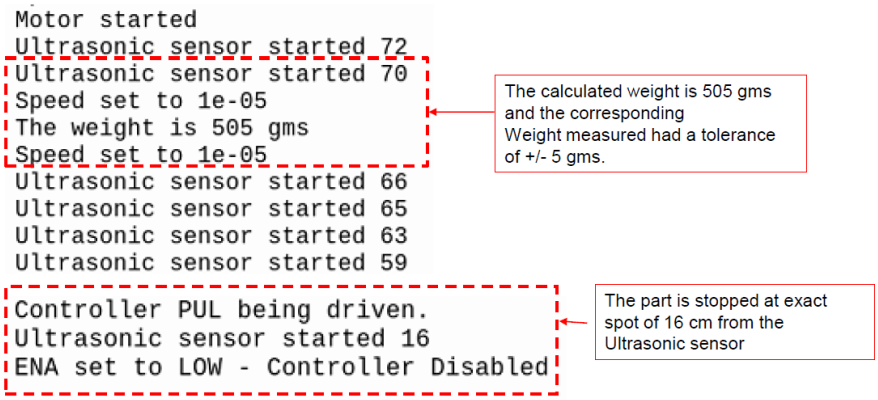
Bill of Material

Item/Description	Quantity	Cost Per	Subtotal	Link
Stepper driver hat (12V PSU required)	1	-	-	https://www.amazon.com/Adafruit-Stepper-Motor-HAT-Raspberry/dp/B00TIY5JM8
Ultrasonic Sensor (requires grove cable)	1	$ 10.99	$ 10.99	Digikey
Raspberry Pi 3 B	1	-	-	CVAC
12V power supply (5 Amp)	1	$ 7.50	$ 7.50	Digikey
MicroSD Card (32 GB, 2 pack)	1	$ 9.81	$ 9.81	https://www.amazon.com/gp/product/B087JCL881
3-pin JST SM Connectors (pre-wired, 20 pairs)	2	-	-	https://www.amazon.com/BTF-LIGHTING-Connectors-WS2812B-WS2811-WS2812/dp/B01DC0KIT2
Bearing	4	-	-	CVAC
Barrel plug M/F pair	1	-	-	CVAC
Jumper wires	1	$ 6.79	$ 6.79	https://www.amazon.com/EDGELEC-Breadboard-Optional-Assorted-Multicolored/dp/B07GD2BWPY/ref=sr_1_3?crid=J06WJ6PFXG5V&keywords=jumper%2Bwire%2Bfor%2Braspberry%2Bpi&qid=1647301722
Fuses (3A)	1	-	-	CVAC
RPi case (plastic, GPIO, and camera compatible)	1	-	-	CVAC
Terminal bridges	1	-	-	CVAC
Crimp-on spade terminals (for 2mm terminals)	1	-	-	CVAC
Fuse Holder	1	-	-	CVAC
Standoff 12mm with M2 BC	2	-	-	CVAC
Spade terminals	12	-	-	CVAC
Load Cell Driver / HX711 - 1kg	1	$ 8.00	$ 8.00	https://www.amazon.com/CHENBO-Weight-Weighing-Pressure-Arduion/dp/B078KS1NBB
Stepper Motor with driver	1	$ 24.98	$ 24.98	https://www.amazon.com/gp/product/B075WTSGK6/ref=ox_sc_act_title_1?smid=A10WM2F57SB7R4&psc=1
5v power supply	1	-	-	CVAC
Conveyor Roller	1	-	-	https://www.mcmaster.com/2277T427/
Infrared Distance Sensor	1	$ 9	$ 9	https://www.amazon.com/HiLetgo-Infrared-Avoidance-Reflective-

				Photoelectric/dp/B07W97H2WS/ref=sr_1_4?keywords=ir+sensor&qid=1647304006&sr=8-4
Conveyor Belt	1	$18.34	$18.34	https://www.amazon.com/gp/product/B08JHB1QHQ
6mm flange coupler	1	-	-	CVAC
45x45 frame		-	-	https://www.mcmaster.com/t-slots/t-slotted-framing-rails/rail-profile~single/
Fasteners (Pack of 4)	16	-	-	https://www.mcmaster.com/t-slot-fasteners/t-slotted-framing-fasteners/
Brackets	32	-	-	https://www.mcmaster.com/brackets/framing-type~t-slot-1/t-slotted-framing-structural-brackets/
Total			$ 95.41	

Project Goals achieved –

We were able to achieve the majority of our project goals. Our model was designed to run with a part of a maximum of 1000 grams of weight with a certain speed depending on the weight of the part. But our model could run with 1500 grams of part weight. The weighing of the part metric was achieved successfully which can be verified from the result table screenshot. Also, the metric of stopping the part at a certain distance before the conveyor belt ends was achieved which can also be verified from the same screenshot. The output of the system is weight in grams and the stopping distance (in cm) when the part reaches its destination. The load cell had an accuracy of +/- 5 grams which was better than initial expectations. This was achieved only because of the stopping of the part at the exact point on the load cell. Hence, confirming that the sensor fusion, as well as the code integration, was robust. The measured distance is in cm and the weight is in grams. We even printed the speed of the conveyor belt.



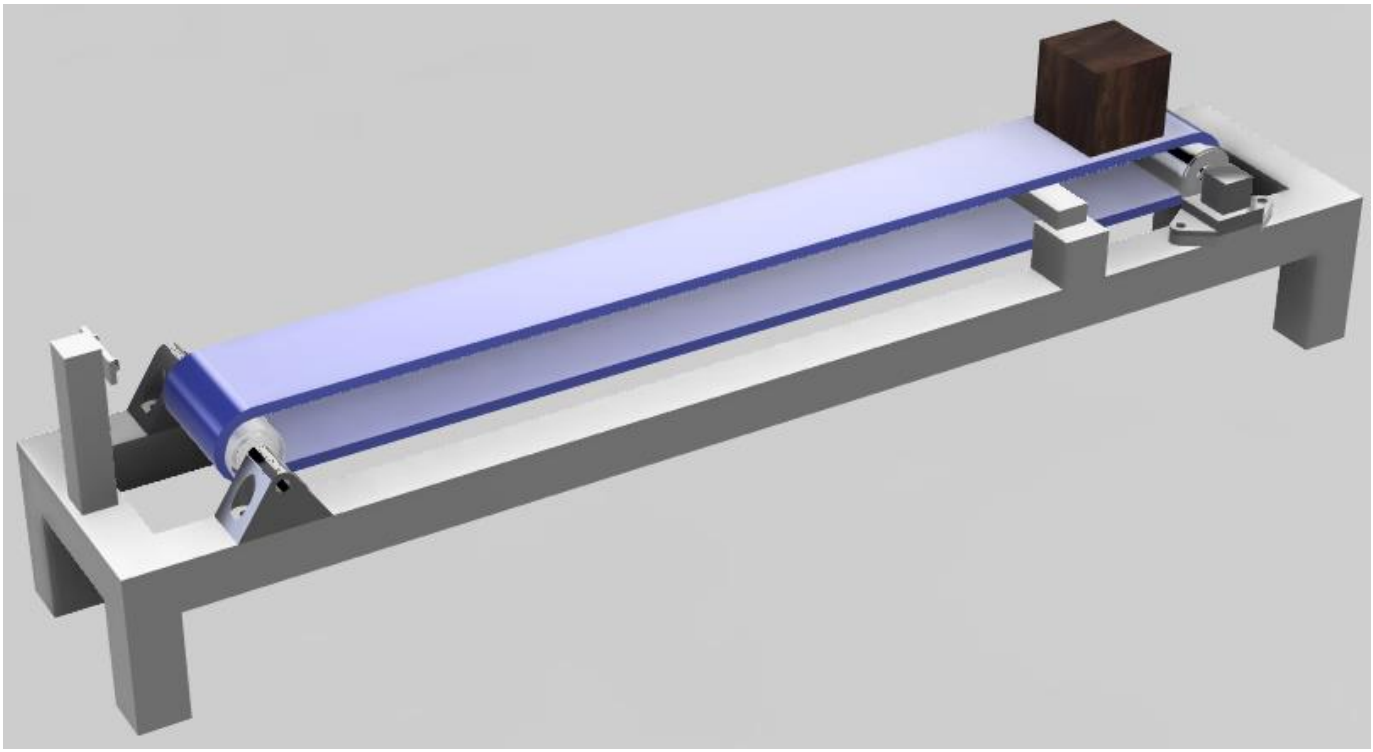
The only issue we faced was the dc motor didn't have enough torque to rotate the roller. So, we had to switch to a stepper motor which helped to achieve our goals but due to speed control in pulses and low speed, we didn't have enough speed distribution for different kinds of weight ranges. The speed of the motor was low compared to a high-speed dc motor.

If we would redo the project, then we would have first calculated the exact amount of torque required for the conveyor belt to rotate and would have ordered the dc motor with specifications based on the calculated torque values. This would allow us to get multiple speeds for many weights to range from 1 gram to 1000 grams. Also, high powered dc motor would have run the belt continuously and smoothly compared to the stepper motor. Also, we would redesign the frame for the proper placing of all the sensors so that it looked aesthetically good as well as easy to make changes to connections. If you see our current model's wiring, it's not that good though it is easy to make changes if we end up with wiring issues. We would focus on the overall packaging of the model.

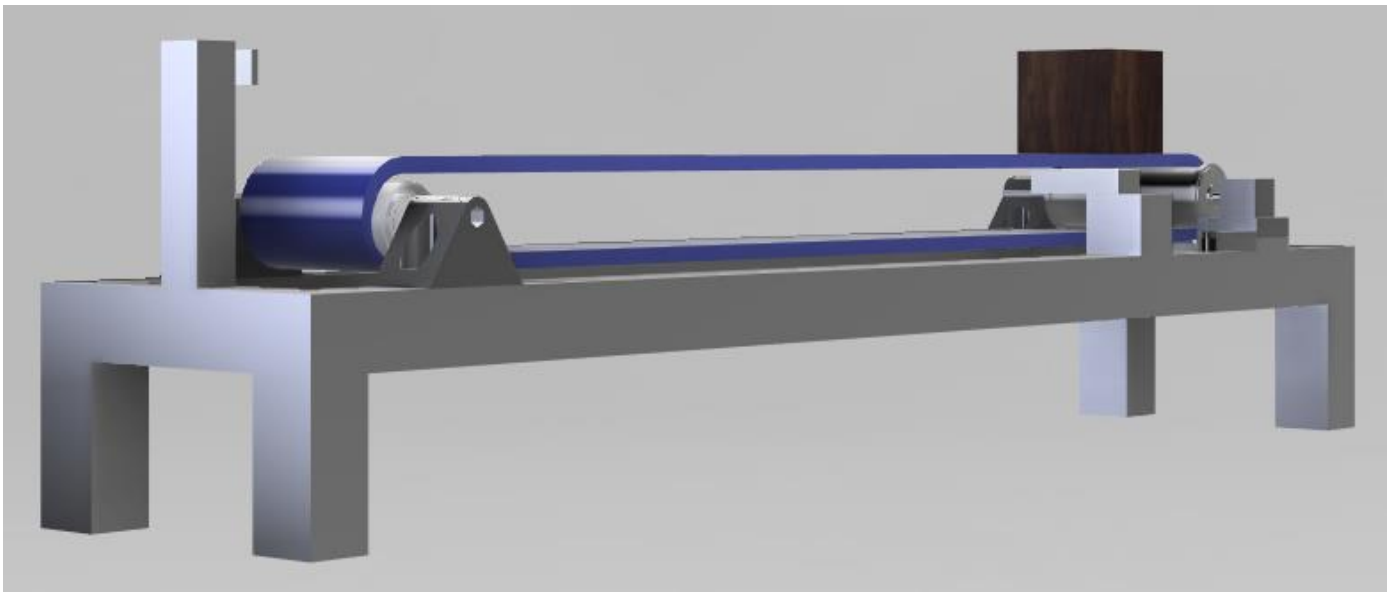
To the next class, we would suggest that they plan and try to check and get updated on all the aspects of design, code integration, component selection, sensor calibration sensor fusion, packaging, etc. wisely. Last moment changes may end up screwing up a good design which unfortunately happened to us (the wiring part). Also, try to complete the project as soon as possible so that you have enough time to make

changes and have enough time for multiple iterations. Completing the project ahead of time would give you a heads up on additional improvements and focus on other work such as reports and presentations which are also as important as the project. Just enjoy what you do and learn from what you achieve even failure is a great thing to achieve something in the future as you would learn an important lesson from it.

CAD Model –



Isometric View



Isometric View

Top View

Scope for further improvements and Future scope -

- High-power DC motor for smooth and fast belt movement. The motor used for the project in the initial stages was DC – motor. The DC motor didn't have enough torque to rotate the conveyor rollers. Therefore, we had to switch to the stepper motor, the conveyer belt was not moving continuously, and it was hard to tune the stepper motor. In the future, using a high-power DC motor would be easy to adjust the motor speed and will help to rotate the conveyer belt continuously and smoothly.
- Part sorting system or pick-&-place robot for part loading and unloading. Adding these robots would be a replica of the industrial system. The robots will help set the part in the same space accolated to weigh the product.
- Add multiple stations for different part categories. So, the sorting system can drop the parts at multiple stations when the part moves forward after the weight measurement.
- For better components placement and wiring packaging considering aesthetics the wiring of the components needs to be routed cleanly.

Comments for next year’s class

1. Start designing and building the physical model at the earliest.
2. Do hand calculations for component selection and check market availability (for example, motor)
3. Work on pseudo-code and component calibration simultaneously
4. Select the parts considering pros and cons and ease of operation, also check what were the previous applications where the same component was used.
5. Spare more time for testing and iterations.