Lebanese American University

Material Sorting Machine

Hazem Daher

Hoda Ghosn

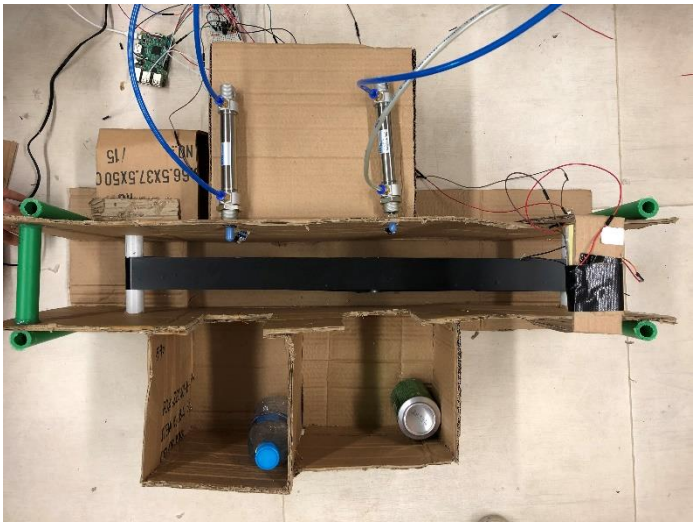
Sara Oud

Table of Contents

Introduction

The “Material Sorting Machine” is an autonomous system that serves the objective of sorting items based on whether they are metals or nonmetals. This system’s main aim is to smoothen the recycling process.

The machine consists of a conveyor belt that will move the items to be sorted to their designated position, pneumatic pistons that will act as the actuators of the system and will push the items into their respective storage units, a conductive metal sensor to decide if the item placed is a metal or nonmetal, and different circuit components and sensors which will be discussed further throughout this report.



Procedure

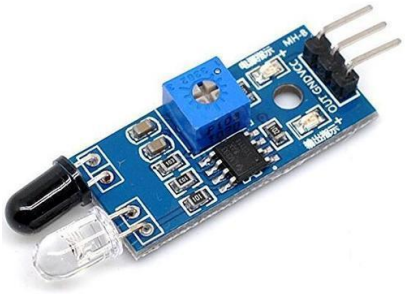
In order to implement this project, we used the following:

Controllers:

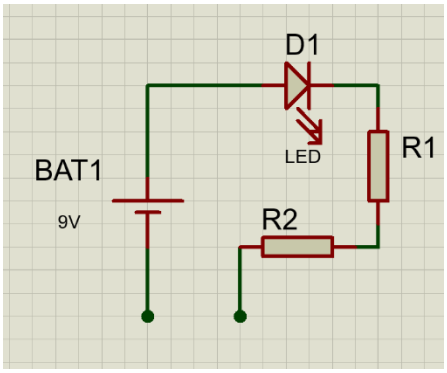
- **Raspberry Pi:** In order to connect the system and run it based on a Python code we wrote to suit the project requirements.

Sensors:

- **IR Sensors:** To detect when an item reaches the point where the piston is supposed to push the item into its storage unit. The digital outputs of the sensors are taken as inputs to GPIOs on the Raspberry Pi.



- **Conductive Metal Sensor:** A basic circuit we assembled consisting of an LED and a voltage divider of resistances with two floating wires, one from the circuit's ground and the other from the resistor, which, when connected to a metallic object, will conduct electricity and turn on the LED. The voltage of one of the resistances is taken as an input to one of the GPIO pins of the Raspberry Pi.



Actuators:

- **DC Motor:** In order to ensure the proper rotation of the conveyor belt (will be elaborated more on later).

- **Pneumatic Pistons:** In order to push the items into their storage units when they reach the correct position.

Conveyor Belt

The conveyor belt system works as follows:

A DC motor is driven through an H-Bridge (L293d IC) that is connected to a 12V power supply and whose enables are connected to GPIO pins on the Raspberry Pi. The enable that allows the motor to turn clockwise was connected to GPIO21 and the enable that allows the motor to turn counterclockwise was connected to GPIO20. The shaft of the motor was placed in a gearbox, and the shaft of the gearbox was super-glued to a plastic cap that was in turn stuck to a pipe that rotates the belt of the conveyor belt and drives another pipe at the end of the belt with it.

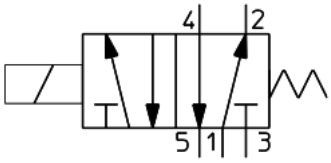
First, the system must check the input of the metal detector circuit which is connected to another GPIO of the Raspberry Pi (GPIO18). Then, regardless of what the value we take from it is (0 or 1 indicating LOW or HIGH), the motor would move until it reaches the right position of the item (in the code, GPIO20 would be set true and GPIO21 would be set to false, this is the case for the entire part of the project where the sorting happens). If the metal detector gave a value of 1, that means that the material sensed was metallic, and it should be dispensed in the first storage unit. Thus, the code enters an “if metal detector = 1 statement” that states that if the first IR sensor reads a HIGH value (indicating that an item has reached the position it is at), the motor must stop (GPIO20 will become False), and the piston at that position would advance and push the item into the storage unit. In the case where the metal detector reads 0 (a nonmetallic item), then the same would happen at the second storage unit, and when the item passes by the first IR sensor, nothing would happen as that IR sensor’s input is insignificant in the part of the code (if metal detector = 0) that runs when the metal detector output is LOW.

In the case where GPIO21 is set to true and GPIO20 is set to false, the conveyor belt would move in the opposite direction.

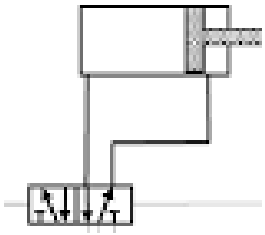
Pneumatics

The pneumatics part of the project works as follows:

Each piston was connected to a 5/2-way valve having a spring return and reading the following symbol:

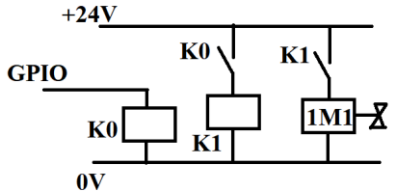


Since this symbol indicates that the initial position is the one at the side of the spring and we wanted the double acting cylinders we used to be initially retracted, we connected port 2 of the valve to the right of the piston, port 4 to the left side of the piston, port 5 to the air compressor output, and left ports 1 and 3 to be discharges into the atmosphere, as shown below:



This connection ensures that the pistons are initially retracted.

The solenoids of the valves are connected to normally open (NO) relays which pass 24V to the valves when switched closed. These relays are in turn connected to smaller NO relays that take 5V (amplified from 3.3V) from the GPIO pins of the Raspberry Pi in order to switch closed and pass voltage to the larger relays that energize the valves (the ones talked about in the previous sentence). When the relays are switched, the solenoids of the valves are energized and the valves switch to the second position that allow the pistons to advance.



Since the main switch that actuates the valves are the small relays that are activated through the GPIO pins of the Raspberry Pi, these pins are activated in the code as follows:

If the first IR sensor that is activated when the metal detector detects a metallic item is activated, the GPIO that switches the relay is set to true (3.3V amplified to 5V) and the valve solenoid would be energized causing the piston at that position to advance and push the item into its respective storage unit. At this point, the GPIO that switches the relay that energizes the other valve solenoid (the one at the nonmetal storage unit) would be set to false (or LOW) and thus that piston would remain still and won't advance.

In the case where the metal detector had detected a nonmetallic material, the same would happen, only this time the energized solenoid would be that of the second valve (at the nonmetal storage unit) and the one set to LOW would be the one at the metal storage unit.

Results

As we ran our code on the Raspberry Pi, we were able to achieve the desired results we described in the procedure.

When a metallic item is placed on the conveyor belt, the metal detector circuit would sense it, the conveyor belt would then start moving after a small time delay (induced by the time.sleep(number of seconds) function in Python), and then it would move and keep moving until the item reaches the position where it is placed in front of the metal storage unit, where the IR sensor would sense it, cause the conveyor belt to stop and the piston to advance, pushing the item into the unit.

When a nonmetallic item is placed on the conveyor belt, the metal detector would not sense it, the conveyor belt would then also start moving after a small time delay. This time, nothing happens when the item passes near the metal storage unit and the first IR sensor as this sensor's input is not commanded to do anything in the part of the code dealing with the nonmetallic item case (**note: in the case of a metallic object, the IR sensor set to detect nonmetallic objects does not induce any change in the system either, however, the item would not reach it anyways as it would have been placed in its storage unit before reaching it**). When the item reaches the position where the nonmetallic storage unit is placed, the IR sensor at that point would sense it and would cause the conveyor belt to stop and the piston at that position to advance, pushing the item into the unit.

Conclusion

In conclusion, the Material Sorting Machine is a successful system that takes materials, senses their conductivity, classifies them as metals or nonmetals, and turns on a system that would sort them into different storage unit based on their category. The aspects of the system are all effective as the code runs perfectly and within more than acceptable response time, and the sensors and actuators provide their inputs and outputs in a very timely manner as well, respectively.