ROBOT MANIPULATOR DESIGN ASSIGNMENT

Souto T.L.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA

In this project a mechatronic system is developed, this system is composed of a mechanical, a pneumatic and an electrical sub-systems. Ultrasonic sensors and microcontrollers are used to achieve the goal of distributing tennis balls on a basic packing system. The design was developed using tools like Proteus for the control and sensors, solidworks for the system mechanics and the Arduino IDE to program the microcontroller.

***Keywords*—Mechatronic systems, pneumatics, ultrasonic sensors**

This paper is a Individual design Project and It is part of the second assignment of the course Master of engineering - Mechatronics at Massey university, Auckland.

An electronic version of this report can be found by following this link, and the programming documentation can be found by following this link, also a GitHub repository with all the files used in this project is available here.

Thiago Lima Souto is a student register under the number 19044686 at Massey university. Questions, comments or communications can be addressed via email thiago.souto@yahoo.com.br

Please Check the links.

CONTENTS

LISTINGS

I. Introduction

The main objective of this report is to design a mechatronic system capable of uniformly distribute tennis balls from a hopper onto a production line using pneumatics. This system shall be composed of a electrical, mechanical and software sub-systems in order to practice such skills.

To accomplish this objective a system composed basically by one pneumatic actuator, a motor, two ultrasonic sensor and the mechanical system that integrates the hopper to the production line was designed. A pneumatic cylinder is used to push the tennis ball, that comes from the hopper and is unable to enter the production line because of a gate, into a rail that leads to a conveyor with a line of tennis ball's packages. The first ultrasonic sensor is set to detect any object that is below 6cm of distance and sends a signal to the solenoid valve that activate the cylinder, this step is required to verify if there is any balls coming from the surveyor. After the ball goes through the rail it ends up on the package and a second ultrasonic sensor, set to detect object at a distance below 20 cm, senses if the package is full and send a signal to the motor that pushes another package one step forward completing the cycle.

This report begins explaining why the use of pneumatics in conjunction with other systems, such is electrical actuation, is widely used. Then the actuation used on the model is discussed as well as the electronics and sensors. The simulation tools used are then explained and also how they interact with each other. Finally the outcome of this project is discussed and the conclusions exposed.

II. Methodology

At this section of the report, for each aspect of the project, a small discussion and the necessary explanations that will support the decisions made around pneumatics, actuation, electronic control and simulation, will be rendered. The intention is to, first, give a general view of the parts of the subject that will have an impact in what was used on the project. Second to explain why this was done. For example, on pneumatics, compressors and reservoirs are discussed and why they are not on simulations.

Also the tools and software used to achieve the results and how these results were achieved. Finally, the relation of each part with the objectives of the report is exposed. Figures and listings will illustrate the explanation wherever necessary.

A. Pneumatics

In a pneumatic system normally air is drawn from the atmosphere via an air filter and raised to the system required pressure by an air compressor, which is normally driven by a AC motor. [1] Because the air contains a significant amount of water vapour the air must be cooled and treated before being used.

Pressurized air is then stored in a reservoir and can be released to the cylinder, without the reservoir the valve action could be slow, if the pressure was to be raised every time the valve was to be used.

The pneumatic systems may look complex and for small applications such as this demonstration project it is no feasible, although many factories produce compressed gas at a central and distribute it on a air ring main to all places on the site, in a similar fashion as electricity or water.

The development of a pneumatic system will make possible, by studying small systems like this project to have a better idea on large scale plants and how they integrate their systems.

For a matter of convenience the pressure reservoir and compressor will be excluded of the simulations included in this report.

B. Actuation

The actuation systems are responsible for transforming the control system output into an action that will control the system [2], on this project solenoid valves are used to control the flow of air which will start the pneumatic cylinder. The pneumatic cylinder is an example of a linear actuator, it consists of a cylindrical tube along which a piston can slide.

To control the motor which will control the conveyor belt a relay is used, it is represented by the LED at the electronic simulation, it works as a mechanical switch acting as the motor actuator.

C. Electronic system

The electronic system is composed by two ultrasonic sensors, a LCD display, and a microcontroller. The LCD monitor displays the actual distance between the ultrasonic sensors and the subjects of measurement, there will be displayed $UTcm:$ for the first measurement and $UTcm2:$ for the second. This is necessary so the correct functioning of the system can be judged.

1) LCD displays

The LCD module used was a 16x2 very commonly used in embedded projects, because its cheap, available and easy to program. They are very common on calculators. It has 16 Columns and 2 Rows. So, it will have 32 characters in total and each character will be made of 5×8 Pixel Dots. [5]

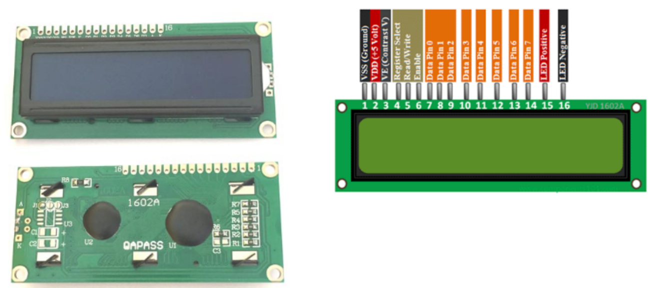

Fig. 1. 16x2 Display

2) Ultrasonic sensors

D. Microcontrollers

1) Control Methodology

III. Design Process

The Design Process section is excellent. It discusses the employed design process, describing key stages and highlighting its advantages and disadvantages. It also discusses the

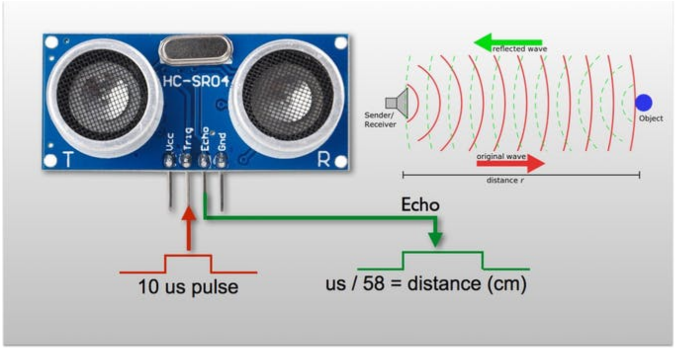

Fig. 2. Ultrasonic Sensor

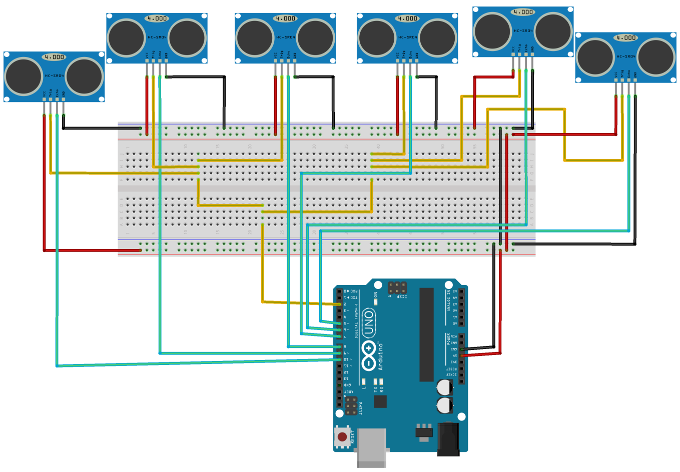

Fig. 3. Multiple sensors

process in relation to the assessment's aims and objectives, and the defined specifications. Appropriate figures are used to effectively illustrate the design process.

A. Electronic simulation

Virtual Prototyping enables system testing before fabrication and electronic devices such as sensors and microcontrollers ca be simulated in specialized software such as Proteus, which has over 15 million parts designed and can accelerate the design process.

1) Proteus

The Proteus Design Suite is used across various industry sectors as a solution for professional PCB design and as a rapid prototyping tool for R&D [4]. This tool was chosen due to the easy of use and simulating capacity, as well as per the amount of parts available, including the ones o=used on the project. With Proteus there is the possibility of importing the programming to the part and simulate accurately the results.

In order to simulate the ultrasonic sensors input, a potentiometer is connected to the ultrasonic sensor test and a DC generator is connected in one side and ground on the other side.

The first ultrasonic sensor, connected on pins 8 and 9, receives a DC generator of $50mV$ in order to simulate a distance below 6 cm stipulated. If a distance below 6 cm is detected then it means that there is a tennis ball waiting to be pushed thorough the gate to the conveyor and the cylinder is activated, as shown on Figure 4.

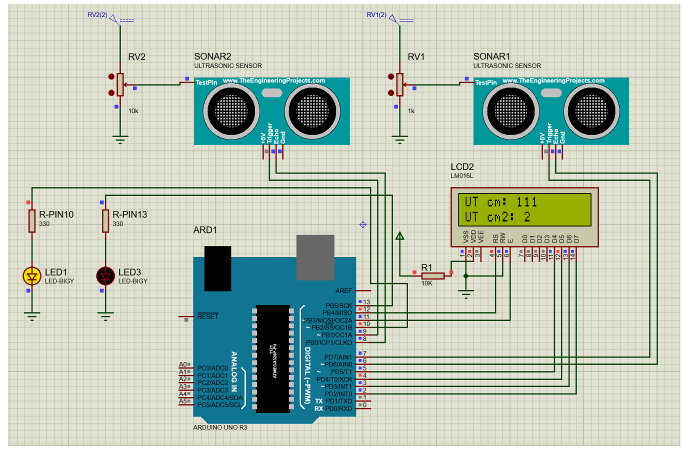

Fig. 4. Simulation in Proteus - Sensor 2 below 6cm

The second sensor receives a DC generator of $500mV$ and simulates a 20 cm distance. It is located on top of the packing station and when there are three balls inside the pack it activates a motor that makes the conveyor belt roll a distance of one pack, and the full pack is projected forward while another one empty takes its place. Figure 5 shows the system simulating a full pack being detected.

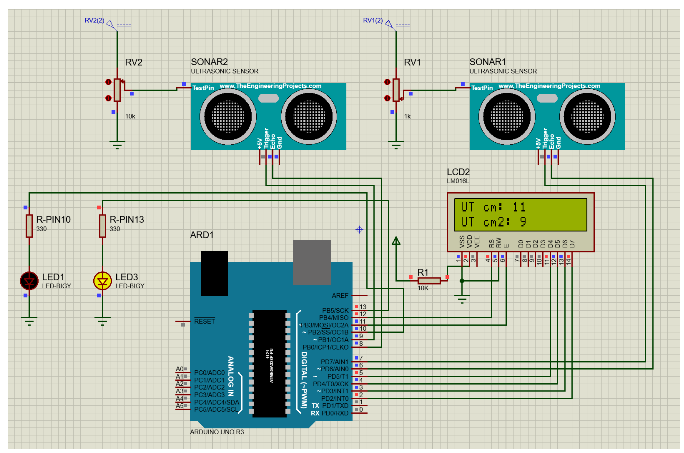

Fig. 5. Simulation in Proteus - Sensor 1 below 20cm

B. Mechanical Simulation

1) Motion and Physics on CAD simulation

2) Routing

IV. RESULTS

The results of the simulation are not ideal due to the many factors that have to be considered on a real project. The physical interaction of all the components have to be taken into consideration. All these factor can be seen on a real model and that's the idea behind a design process like this, where we round the intrinsic factors that are of minor importance in favour of having the big picture revealed and then measure and adjust the real product at the end.

Although, since the aim of the project was to practice the design process, it was fully achieved since an electrical, mechanical and software, therefore, mechatronic system was fully developed and integrated.

V. Discussion and Conclusions

The Conclusion section is excellent. It describes the assessment's aims and objectives, and the defined specifications. It discusses the developed system's performance, linking very clearly to the aims, objectives, and specifications. A lot of recommendations to improve the system are discussed.

The system can be improved by changing the LED's for a real motor and the solenoid valve on the electronic simulation. If the project were to be physical, the first idea was to make a continuous belt around the and to place the packing into a box at the end of the process.

References

[1] A. Parr, *Hydraulics and Pneumatics - A technician's and engineer's guide*, 2nd ed., Ed. Butterworth-Heinemann, Oxford - UK 1998.

[2] P. Fritzson, et al. *The OpenModelica Integrated Modeling, Simulation and Optimization Environment* (Conference paper), PROCEEDINGS OF THE 1ST AMERICAN MODELICA CONFERENCE.Cambridge, USA: Massechusetts, 2018.

[3] W. Bolton, *Mechatronics - Electronic control systems in mechanical and electrical engineering*. United Kingdom: Pearson Education Limited, 2019.

[4] (Proteus wesite Online Help) Labcenter. (2020, May). Available: https://www.labcenter.com/.

[5] (16x2 LCD Module) Datasheet. (2020, May). Available: https://components101.com/16x2-lcd-pinout-datasheet. Thanks gs.solidworks.com/solidworksblog/2016/10/growing-solidworks-nation.html.

[6] (SolidWoks - Wikipedia)(2020, May). Available: https://en.wikipedia.org/wiki/SolidWorks.

[7] K.N. Frazer, School of Food and Advanced Technology, Massey University, Slides from Lecture 12, March 2019.

[8] *RDrive Datasheet* ROZUM Robotics LLC., Mountain House, CA - USA, 2020.

[9] (Atan2 - Wikipedia)(2020, May). Available: https://en.wikipedia.org/wiki/Atan2

Appendix

```
#include "Ultrasonic.h"
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int echoPin = 6;
const int pingPin = 7;
const int echoPin2 = 8;
const int pingPin2 = 9;
int led13 = 13;
int led10 = 10;

void setup() {
  lcd.begin(16, 2);

  pinMode(led13, OUTPUT);
  pinMode(led10, OUTPUT);

  pinMode(pingPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(pingPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
}

void loop() {
  long duration, duration2, cm, cm2;
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(pingPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  cm = microsecondsToCentimeters(duration);

  digitalWrite(pingPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(pingPin2, LOW);
  duration2 = pulseIn(echoPin2, HIGH);
  cm2 = microsecondsToCentimeters(duration2);

  lcd.setCursor(0, 0);
  lcd.print("UT cm: ");
  lcd.print(cm);
  lcd.setCursor(0, 1);
  lcd.print("UT cm2: ");
  lcd.print(cm2);

  if (cm < 20) {
    digitalWrite(led13, HIGH);
    delay(1000);
    digitalWrite(led13, LOW);
  } else {
     digitalWrite(led13, LOW);
  }

  if (cm2 < 6) {
    digitalWrite(led10, HIGH);
    delay(1000);
    digitalWrite(led10, LOW);
    delay(1000);
  } else {
     digitalWrite(led10, LOW);
  }
  delay(100);
}

long microsecondsToCentimeters(long microseconds){
return microseconds / 29 / 2;
}
```

Listing 1. Arduino Programming