UNIVERSITI PUTRA MALAYSIA

TABLETOP UNIVERSAL TESTING MACHINE FOR SOFT SOLIDS

SHUKRY JAMERI

TABLETOP UNIVERSAL TESTING MACHINE FOR SOFT SOLIDS

SHUKRY JAMERI

183578

BACHELOR OF ENGINEERING (PROCESS AND FOOD)

FACULTY OF ENGINEERING

UNIVERSITY PUTRA MALAYSIA

2018/2019

TABLETOP UNIVERSAL TESTING MACHINE FOR SOFT SOLIDS

SHUKRY JAMERI

183578

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF ENGINEERING (PROCESS AND FOOD)

DEPARTMENT OF PROCESS AND FOOD ENGINEERING

FACULTY OF ENGINEERING

UNIVERSITY PUTRA MALAYSIA

JUNE 2019

ACKNOWLEDGEMENT

I am very grateful to be able to complete my final year project. I would like to express my heartiest gratitude to Dr. Mohd Afandi P Mohammed, my supervisor for giving me the chance to do this project under his supervision and for all the support and encouragement he has given from beginning till the end. Thank you so much for believing in me even though there were times where I do not have faith in myself. I would also like to extend a special thanks to Ahmad Tarmezee Talib for his constructive comments and feedback. To my parents Mdm. Siti Norita Binti Dani and Mr. Jameri Bin Tahir, thank you so much for your kindness and support. You are my source of strength to carry on whenever I feel low. The journey towards completing this project has taught me a lot. Juggling through classes, assignments, part-time jobs, student activity meetings, and allocating time for final year project was no easy feat. Lastly I would also like to express my gratitude to those who are directly and indirectly involved on the completion of this project. You guys are awesome.

ABSTRACT

This report shows the design, construction, and calibration of a Tabletop Universal Testing Machine which are made up of Plastic Lego Bricks. A commercial 5kg load cell was used to measure the load applied on a sample. MH infrared sensor with a sensitivity of 0.1cm was used to measure displacement between the platform and compression plate which was used to determine the material strain. Load cell and MH infrared sensor was connected to a personal computer through Arduino Uno microcontroller. Load was supplied from EV3 large servo motor which was controlled by an EV3 Mindstorms Intelligent Brick. The user interface was designed using NI LabView 2019. Hapi Gummy candy was used to perform test run to determine the capability of the machine to perform compression test.

ABSTRAK

Laporan ini menunjukkan kerja-kerja mereka-bentuk, pembinaan, dan penentukuran sebuah mesin penguji mampatan universal yang diperbuat daripada bata Lego plastik. Cell daya yang berkeupayaan 5kg telah digunakan bagi mengukur daya yang dikenakan keatas sampel. Sebuah sensor inframerah MH digunakan bagi menentukan sesaran di antara tapak mampatan dan plat mampatan yang akan digunakan bagi menentukan nilai *strain* yang dialami oleh bahan sampel. Cell daya dan sensor inframerah MH disambungkan ke komputer peribadi menggunakan kontroler-mikro Arduino Uno. Daya yang dikenakan kepada sampel dibekalkan oleh motor servo besar EV3 dan dikawal oleh *EV3 Mindstorms Intelligent Brick.* Rekaan antaramuka pengguna telah dlaksanakan menggunakan *software* NI LabView 2019. Gula-gula Hapi Gummy telah digunakan bagi menjalankan ujian bagi menguji kebolehupayaan mesin untuk menjalankan ujian mampatan.

Contents

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

1.1 UNIVERSAL TESTING MACHINE

One of the known effective methodology to study the physical properties of soft solids are via the usage of the universal testing machine (UTM). the UTM is mainly used to determine force and deformation of the material tested, which can then be used to obtain important parameters such as Young's modulus, Poisson's ratio, tensile strength, and compressive strength of a material. Mechanical properties of soft solids materials are a topic of interest, especially in the food industry. Commercial UTM uses load cells that can measure and withstand a large amount of force which is convenient for very hard materials such as metal and composites but are less sensitive to be used for soft solids. In addition, the cost of a commercial UTM is relatively expensive. According to the previous study, the factors that contribute to the high production cost of UTM are due to its complicated design, expensive choice of material, and also abundance in a variety of hardware installed which in return increased the inventory (Annappa & Panditrao, 2012). The same author also suggested to replace the materials with something lighter and has a lower cost. Therefore the use of user-developed UTM is a good alternative, especially for scientific and educational purpose.

1.2 LEGO AND LEGO MINDSTORMS EV3

Lego is a construction toy known worldwide that is made up of plastic blocks which can be interlocked together to form a solid structure according to the creativity of those who assemble it. The origin of the Lego group actually began as a carpenter's workshop from Billund, Denmark which was owned by Ole Kirk Christiansen who at first producing wooden toys in 1932 (Sandgaard & Jensen, 2015). The Lego Company started developing the infamous interlocking bricks which became the company's featured product up to this day. The Lego Mindstorms was developed by the Lego Company and Media Laboratory in Massachusetts Institute of Technology (MIT), which acts as a hardware-software platform. The Lego Mindstorms consists of sensors, motor parts, and Lego parts which can be assembled to create a mechanical system, and also an intelligent brick computer which functions as the control system. The four generations of the Mindstorms platform the Robotics Invention System, NXT, NXT 2.0, and EV3. With the addition of the intelligent brick computer, it is possible to create a programmable device by using Lego parts.

1.3 PROBLEM STATEMENT

Universal Testing Machine UTM is a very useful instrument that serves a lot of purposes. The Universal Testing Machine has a lot of purposes on studying the mechanical properties of the material are however fetches a very high price in the market. The previous study reported that an unnecessary increase in the cost of the Universal Testing Machine is due to the expensive choice of material, complicated design, and also having too many varieties of hardware which increases the number of inventory of the machine which in turn causes an insignificant increase in price.

Furthermore, the load cells used for commercial Universal Testing Machine have a load cell that can withstand a high amount of load are less sensitive when it comes to small changes in value. While it is most suitable to test materials with high mechanical strength, soft solids, on the other hand, require sensors with higher sensitivity in order to be able the precise amount of load needed to cause deformation on soft samples in order to obtain a more accurate stress-strain curve.

Besides that, it is important for a device to have a proper general user interface which is user-friendly. Having a complex user interface will require an operator to be trained before being able to operate the machine. The previous version of the Tabletop Universal Testing Machine requires some basic knowledge in programming in order to be able to execute the program. It will require time and money and a lot of commitment in order to train operators to have the necessary skills and knowledge just to be able a single machine.

1.4 RESEARCH OBJECTIVES

The main objective of this research is to :

1) Develop a functioning Tabletop Universal Testing Machine based on interlocking Lego Bricks. and microcontrollers such as the Arduino UNO which can be interfaced with a larger variety of hardware,

2) Design an interface which is user-friendly and promotes ease of handling of the device in order to improvise on the previous design of the Tabletop Universal Testing Machine for soft solids

2.0 LITERATURE REVIEW

2.1 UNIVERSAL TESTING MACHINE AND LEGO

The previous project on lab scale design and assembly of UTM has showed that it is appropriate to obtain reliable mechanical properties of complaint materials using the self-made UTM which provides the advantage of lower production cost and smaller frame size in comparison with commercial UTM (Huerta, Corona, Oliva, Avilés, & González-Hernández, 2010). LEGO bricks are commercially available interlocking bricks pieces that offer the benefits of the creation of structures based on the creativity of the designer which might be less expensive, reconfigurable and rather scalable. It provides a system layout that permits the satisfaction of positive design constraints with a particularly simple setup for the reduction of risk in systematic errors. The advancement of robotics technology has been a vital part in education due to the robot system including computer, control systems theory, mechanism, information sensing and processing technology, artificial intelligence, etc. (Del Vecchio, Reis, & da Costa Mattos, 2014). With further advancement in programming of the Lego system, many instruments are now being invented and innovated based on LEGO bricks structuring and are being complemented by a various control system to perform varieties of experiments. Lego-based measurement device interfaced with MATLAB function has previously been tested for the investigation of in-situ (low temperature) ice cream compression behavior and has proven to provide reliable results. Lego bricks was also used for this project while using a different kind of scripting tools to maneuver the UTM. Instead of using MATLAB, Arduino microcontrollers will be used with the support of EV3 intelligent brick to control the UTM. Bread dough was used as a sample to determine the performance of the tabletop UTM in comparison with commercial UTM.

2.2 ARDUINO UNO

Arduino is a company which produces microcontrollers and microcontroller kits for building digital devices and also enabling sense and control of a system. The microcontrollers are usually programmed using the C and C++ programming language. The Arduino also serves as a platform for an open-source community where people around the world are sharing their projects for educational purposes. Arduino Uno is a microcontroller board based on the ATmega328P . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. Arduino UNO can be used to tinker without worrying too much about doing something wrong, worst case scenario, the chip can be replaced for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform.

2.3 INSTRUMENT – LABVIEW

National Instruments Corporation, more commonly known as NI is a multinational company which is based in the United States of America. The NI provides hardware for automated test equipment and visual instrument software which can be used to be interfaced with many types of hardware. The Arduino UNO microcontroller used for this project LabVIEW is basically a system engineering software that has rapid access with many types of hardware and data insights which can be used for applications that requires testing, taking measurements, and process control. The LabVIEW promotes graphical programming approach that helps visualizes the coding process of a program. It helps in simplifying the process of integrating measurement hardware from different companies via the representation of complex logic on the diagram. This software can be used for data developing, algorithms analysis, and for designing a custom engineering user interfaces.

2.4 UNIAXIAL COMPRESSION TEST

In a society of elderly people, the demand for proper textured food for safe consumption is increasing by those who have difficulty masticating or chewing. The elderly society often prefer food with soft texture for ease of tongue-palate compression and does not need require teeth mastication. (Ishihara et al., 2013). Compression tests can be used to determine the mechanical and rheological characteristics of different foods with regard to consumer perception. Bread dough rheology allows substantial stretching of samples before fracture (Tanner, Qi, & Dai, 2018). The same author also concluded that dough rheology is sensitive to water content changes, changes in starch content, mixing and wheat genetics, and temperature. Researches have been conducted on the compression of bread dough. Tanner et. al (2018) conducted a study on the comparison between biaxial tests and compression test for bread dough and found that the mode of deformation does not change the damage function, but only depends on the variety of dough. Another study on bread dough compression was performed on determining the effect of friction on the uniaxial compression of bread dough where the researchers found that, the presence of friction can affect the result of the uniaxial compression test on bread dough (Charalambides, Goh, Wanigasooriya, & Williams, 2005). Compression test was also used to determine the effect of how different level of dough softness on the stickiness of the bread dough which pose a problem in the industry where sticky dough may decrease the life expectancy of industrial machines (Tock et al., 2013).

3.0 METHODOLOGY

3.1 HARDWARE, DESIGN, AND CONSTRUCTION OF UTM

The UTM was designed to determine the stress-strain curves of soft materials such as bread dough or other food samples. The dimension of the overall design was relatively small compared to commercial UTM which makes it easy to transported. The device is capable in analyzing samples of up to 5cm of height. The UTM is composed of several parts which are the mainframe, the drive system, compression plate, compression base and load cell, and control system.

3.1.1 Main Frame

The main frame of the UTM was made of plastic interlocking Lego bricks which are hard enough to withstand the testing of soft solid samples. The compact design of the main frame and the lightweight of the chosen material contributes to the portability of the device. Using Lego parts as the mainframe offers the benefits of improvising the structure of mainframe which makes the design process of the mainframe more versatile. Figure 1 shows the set of LEGO bricks used.

Figure 1: Lego Mindstorms EV3 expansion set

3.1.2 Drive System

The drive system of the UTM is composed of an EV3 Large Servo Motor from Lego Mindstorms combined with Lego Technics Linear Actuator, as shown in Figure 2. The Lego Technics Linear Actuator from the motor converts the rotary motion into a uniaxial linear motion needed for the uniaxial compression testing.

Figure 2:Linear actuator (left) and EV3 large servo motor (right)

3.1.3 Compression Plate and Compression Platform

The compression plate is basically a flat plate of joint Lego bricks attached at the end of the actuator. The actuator is complemented with support structure at its sides which are important to ensure that the plate does not rotate while moving up and down. Such action during sample testing may produce the undesired experimental result.

The base plate, which is also a flat plate of joint Lego bricks are attached at the load cell. The compression plate and the base plate are arranged in a way that it is perpendicular with the motion of the actuator to ensure the uniform load is applied perpendicularly onto the sample and the load cell during testing uniformly.

3.1.4 Load Cell and Sensors

Figure 3 shows the load cell and electronics used to measure force. A 5kg load cell was installed on the device. The load cell was placed at the bottom of the base plate receiving loads during the experiment from a perpendicular direction. The load cell was connected with Arduino UNO boards for data processing. The signals from the load cell were amplified by HX711 driver before being sent to Arduino UNO boards for data processing.

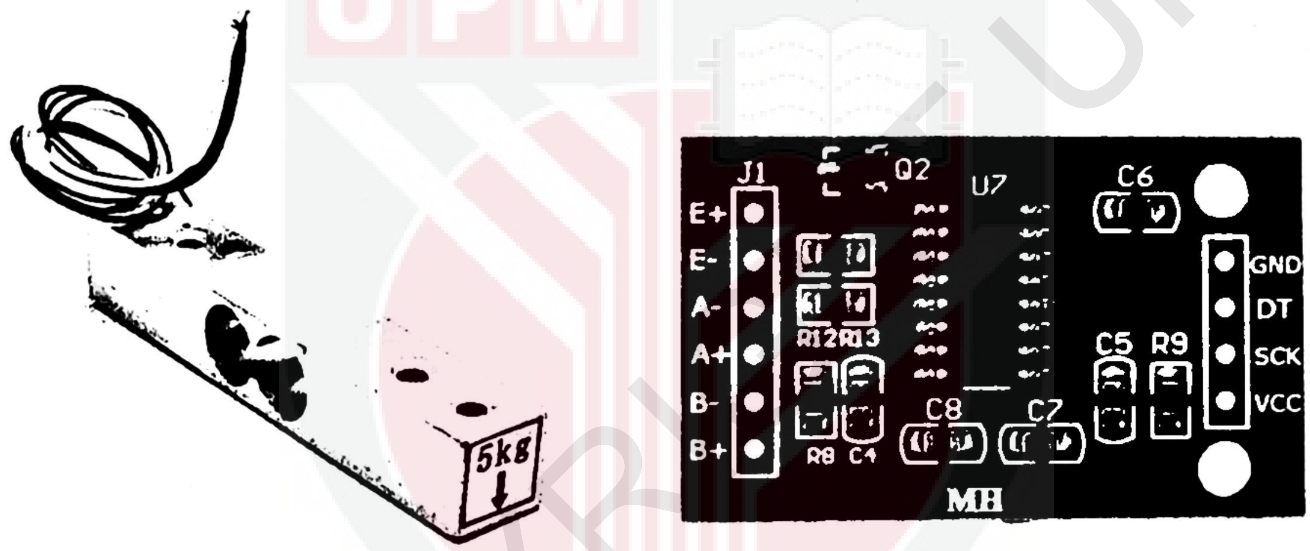

Figure 3:From left, 5kg load cell and HX711 driver.

In order to produce a stress-strain curve to study the mechanical properties of soft solids, it is important to have a deformation detection system. An infrared sensor was used to measure distance which can be useful to gather the displacement data of a sample during an experiment. The range that an infrared sensor detects range based on the calculation of the speed of an infrared wave. The infrared sensor consists of a transmitter and receiver. Figure 4 shows the infrared sensors used in this work.

Figure 4:MH Infrared sensor

The transmitter on the sensor will transmit infrared wave which will travel until it is in contact with an object positioned vertically with the transmitter. Upon contact, the infrared wave will be reflected back onto reverse direction from where it is being transmitted where the receiver will receive back the signal. The difference in time in which the infrared wave being transmitted and returned back to the receiver will determine the displacement between the sensor and an object subjected to detection.

3.1.5 Control System

The drive system was controlled by EV3 Mindstorms Intelligent Brick which will be connected to the personal computer via wireless Bluetooth connection. Sensors were connected to the Arduino UNO microcontroller. Arduino UNO was connected to a personal computer via USB ports. Programs will be initiated from the computer. Outputs from the sensors were sent and processed through specially programmed codes. The drive system and the sensors were integrated by LabVIEW software program from a personal computer, as shown in Figure 5.

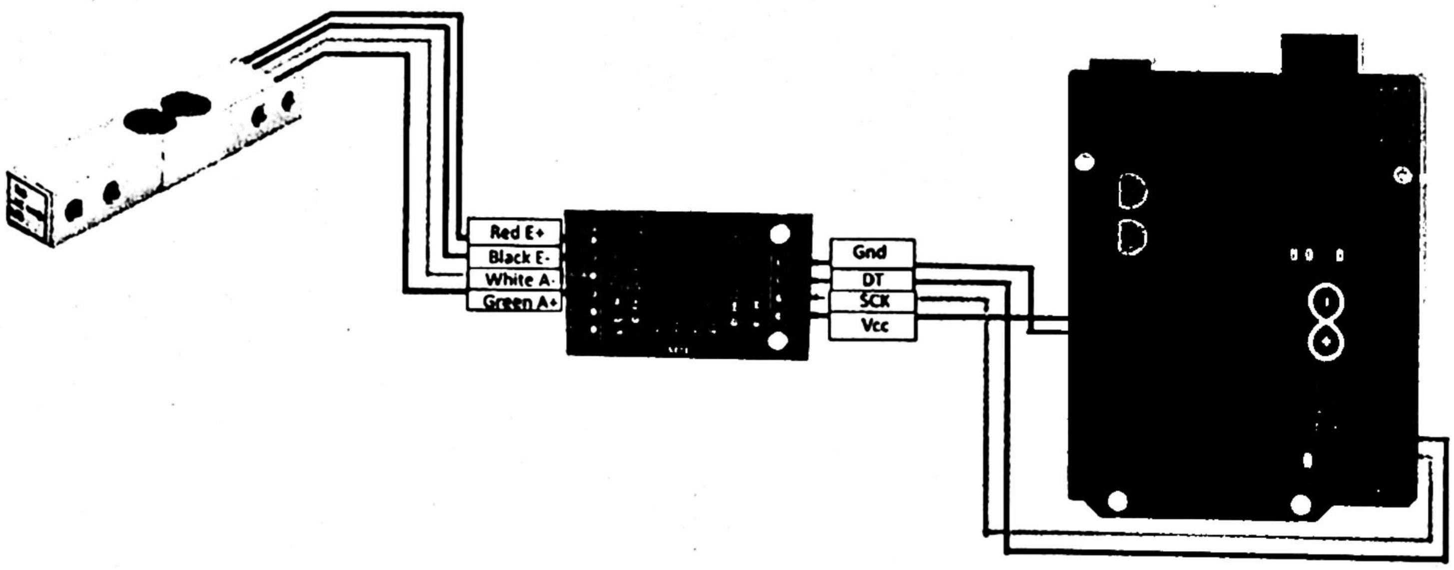

Figure 5:Load Cell, HX711 Driver, and Arduino UNO connection schematics.

3.2 SOFTWARE AND USER INTERFACE DEVELOPMENT

Relying on Arduino UNO microcontroller alone will require an operator to at least have some basic knowledge in the C++ programming language in order to execute different functions of the Tabletop Universal Testing Machine. Therefore, in order to generate a general user interface which is user-friendly, the Arduino UNO boards can be interfaced with LabVIEW which uses a graphical programming language.

3.2.1 Motor Control

Drive system will be controlled by the Mindstorms software which can be connected with the PC by a wireless network via Bluetooth. Figure 6 shows the graphical programming platform for Lego Mindstorms which can be transmitted directly into the EV3 Mindstorms Intelligent Brick. It will then process the program and execute it accordingly. Figure 6 shows an example of a series of command for motor control.

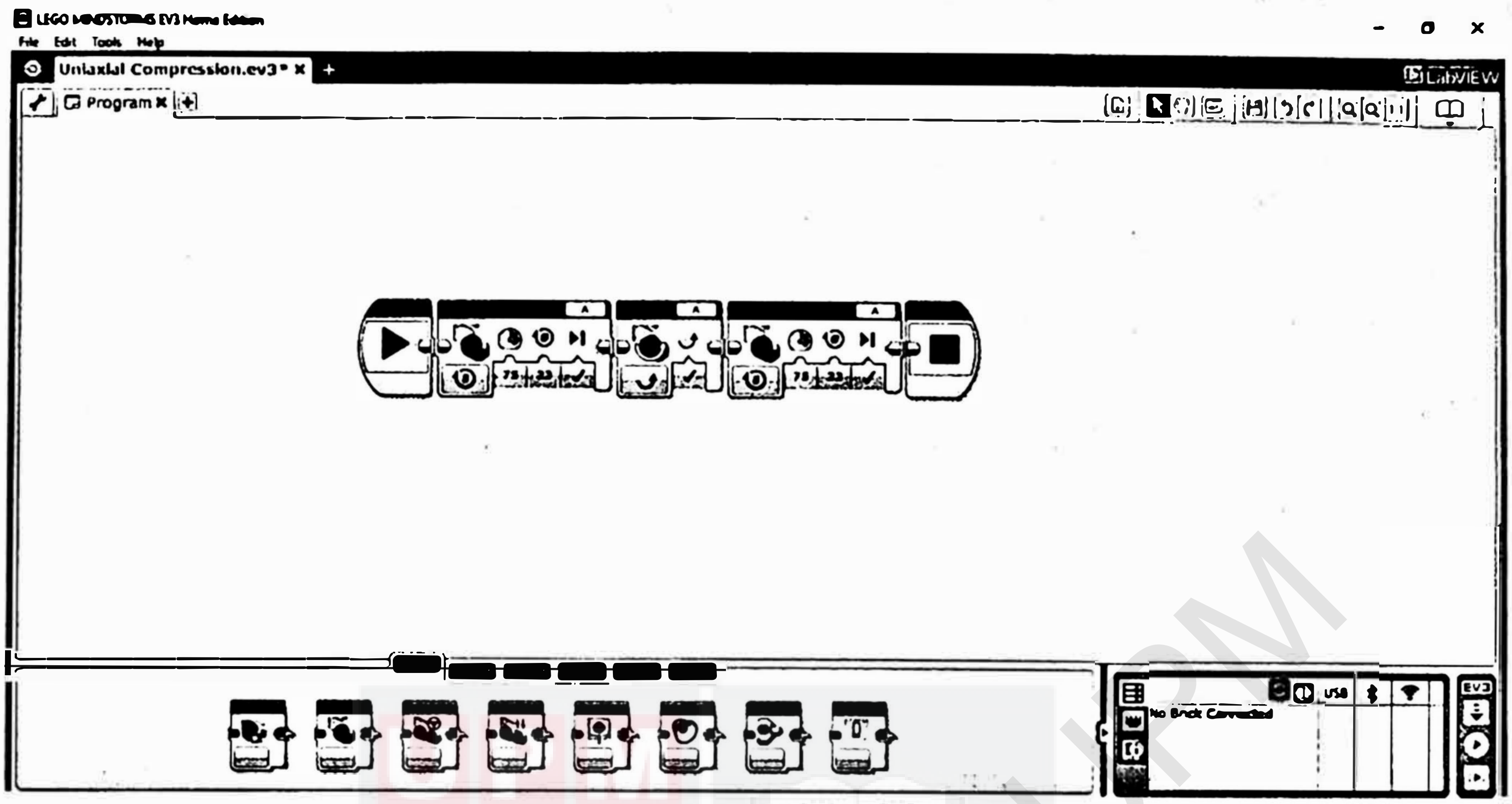

Figure 6:Drive program for Uniaxial Compression test.

The yellow panel is a “begin” command which initiates a program. The green panels are commands for motor controls. The exact number of rotations and motor speed can be adjusted from these panels. A set amount of rotation was pre-determined from the commands to accurately specify the amount of displacement needed for the experiment. The displacement of the compression plates is linearly proportional with the number of rotation of the motor. Hence the amount of displacement needed can easily be calculated. The graphical programming for EV3 Mindstorms Intelligent Brick is readily available to be accessed from LabVIEW.

3.2.2 Interfacing Arduino and LabVIEW

For the data acquisition part, load cell and infrared sensors are connected with Arduino which receives and transmit the signals from the sensor into a personal computer. The LabVIEW software from the personal computer will then execute the data acquisition function to interpret the signal converting it into valuable experimental data. To make this happen, Arduino needs to be interfaced with LabVIEW. The Arduino UNO Serial Command to interface Arduino UNO and LabVIEW are as shown in the Appendix 2. The Arduino UNO Serial Command does not code the test program but only serve to connect the Arduino UNO and LabVIEW.

3.2.3 Sensor Calibration

For the calibration process, the relationship between the analog output signals from both sensors and some pre-determined values were studied. For load cell, the average of a voltage reading was recorded and tabulated for when no load was applied onto the load cell. The same method was applied with a load weighing 5.66 g, 11.32 g, 16.98 g, 22.64 g, 28.30 g, 33.96 g, 39.62 g, 45.28 g, 50.94 g, and 56.60 g. The result was tabulated and plotted on a graph.

Calibrating the infrared sensor to measure distance uses the same appraoch. Infrared sensors work by measuring the time taken for infrared wave transmitted from a transmitter to be received back by a receiver of the same sensor. The closer an object is to the sensor, the faster the reflected infrared wave will be received back by the sensor's receiver. The sensor will generate an analog output based on the time taken for the infrared wave to be returned back to the receiver upon being transmitted. The voltage output from the infrared sensor will be read when an object is placed right in front of the sensor. The same steps were repeated by placing an object further from the sensor with a starting distance of 2 mm until 34 mm with an increment of 2 mm for each sampling. The data obtained were then tabulated and plotted on a graph.

Figure 7 shows the code for the program that is compiled by LabVIEW which uses a graphical language program. The program includes reading analog output signals from the infrared sensor and also output signals from the load cell which has been amplified through the HX711 driver. Since there is no built-in code to call the HX711 driver for LabVIEW, it is necessary to create a custom command for it. Analog output is simply in the form of voltages hence, it has to be translated into a value with units in accordance with the functionality of the sensor.

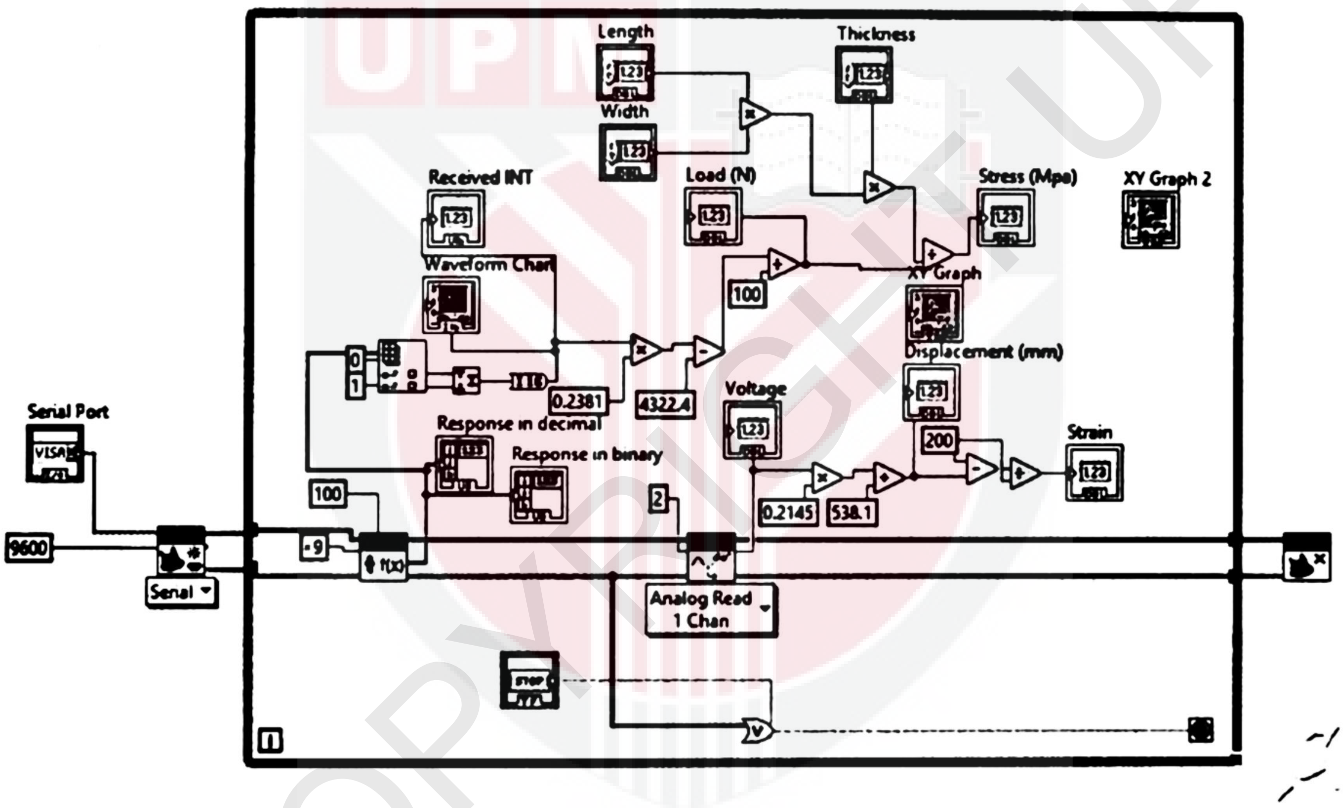

Figure 7:Block flow diagram of a VI code for developing a user interface for the Tabletop Universal Testing Machine.

3.3 PERFORMANCE TEST RUN FOR UNIAXIAL COMPRESSION TEST

To conduct compression testing using the LEGO UTM developed, food samples (soft solids) were used. The samples used are Hapi Gummy candy, which gummy candy samples manufactured by Twinfish Malaysia were bought from local stores. The dimension of Hapi Gummy is shown in Figure 8. Uniaxial compression test was performed on Hapi Gummy candy at 1 mm/s. Figure 9 shows the sample being loaded to the LEGO tester.

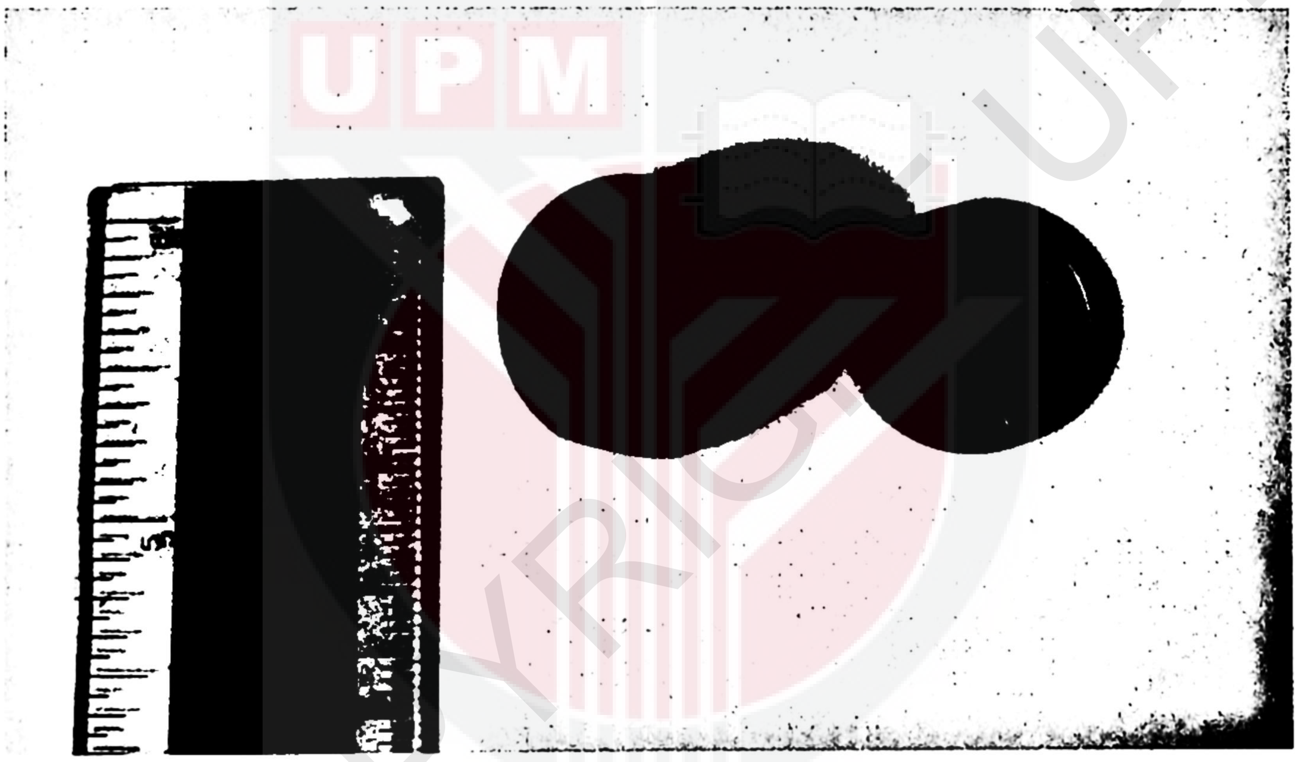

Figure 8:Hapi Gummy sample

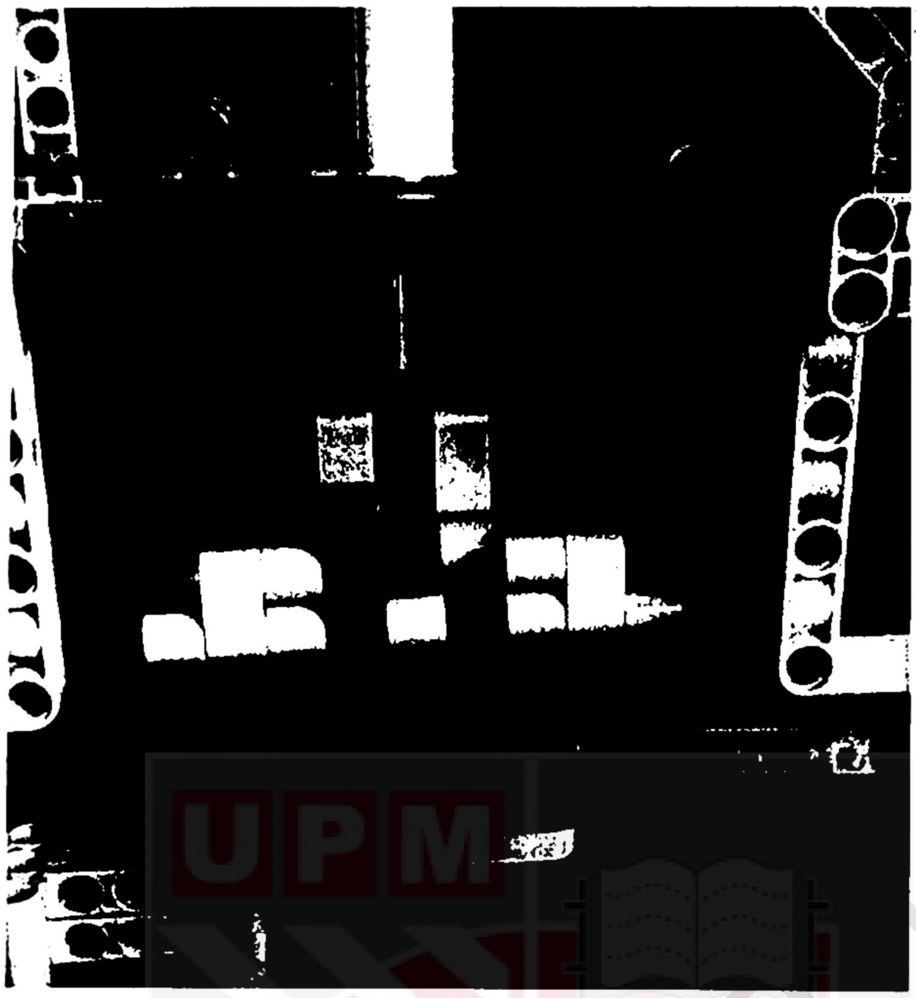

Figure 9:Hapi Gummy placed on Lego Tabletop Universal Testing Machine

4.0 RESULTS AND DISCUSSION

4.1 OVERVIEW

The discussion in this topic is about the construction of the machine and problems faced during the construction of the device. The soft solid sample was tested to observe the operation and performance of the machine. Further discussion about the machine will be discussed based on the results obtained.

4.2 CONSTRUCTION OF THE MAIN FRAME

Mainframe of the machine developed is shown in Figure 10. The whole structure was made up of Lego bricks Mindstorms kit version. The only unique part is the load cell, infrared sensor, an Arduino UNO microcontroller which is not an original item provided by the Lego Mindstorms kit. The design features a load cell being positioned below a platform. The drive system is mostly positioned on the upper part of the machine. The way that load was supplied onto the sample from top to bottom vertically. The design was very light hence providing ease of mobility for the machine. Due to the light structure, the machine tends to vibrate lightly when the drive system is active. It is observed that during initial testing, the vibration from the motor movement is significant, which might affect sensor readings. Apart from the vibration, the structure seems to be stable and does not topple easily.

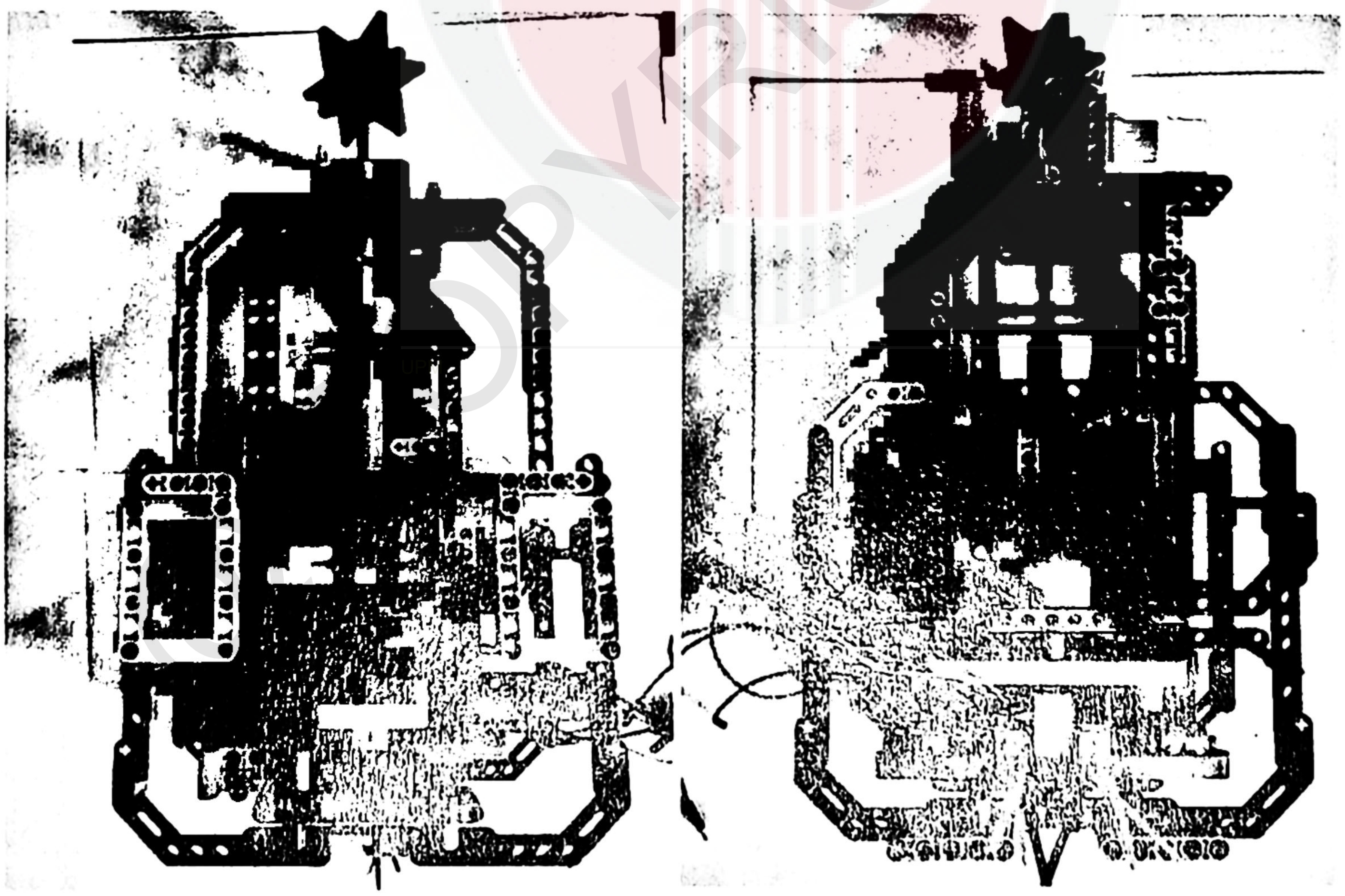

Figure 10:Front view (left) and back view (right) of Tabletop Universal Testing Machine

4.3 DRIVE SYSTEM AND COMPRESSION PLATE

The drive system used for the LEGO tester is shown in Figure 11. Motor movement was linearized by the shaft which is connected with the motor through a series of gears. No apparent significant power loss can be observed for power transmission from the motor into the actuator. The system worked as predicted and the sidebars installed beside the compression plate prevents the compression plate from spinning.

Figure 11: Compression plate attached with actuator and motor drive

4.4 COMPRESSION PLATFORM AND LOAD CELL

Platform and load cell was set up is shown in Figure12. The platform and load cell were positioned on the machine as shown in the red circle in Figure 13. The size of the platform was quite small hence limiting the dimension size of the sample to be used on this machine. During pre-testing of the machine, load supplied from the motor causes major bending on the platform. It was further found out that the plastic gaskets used to interlock the bricks were not able to withstand the force from the motor which causes the structure of the gasket to fail, as shown in Figure 14.

Figure 12: Side view (left) and top view (right) of load cell and compression platform setup

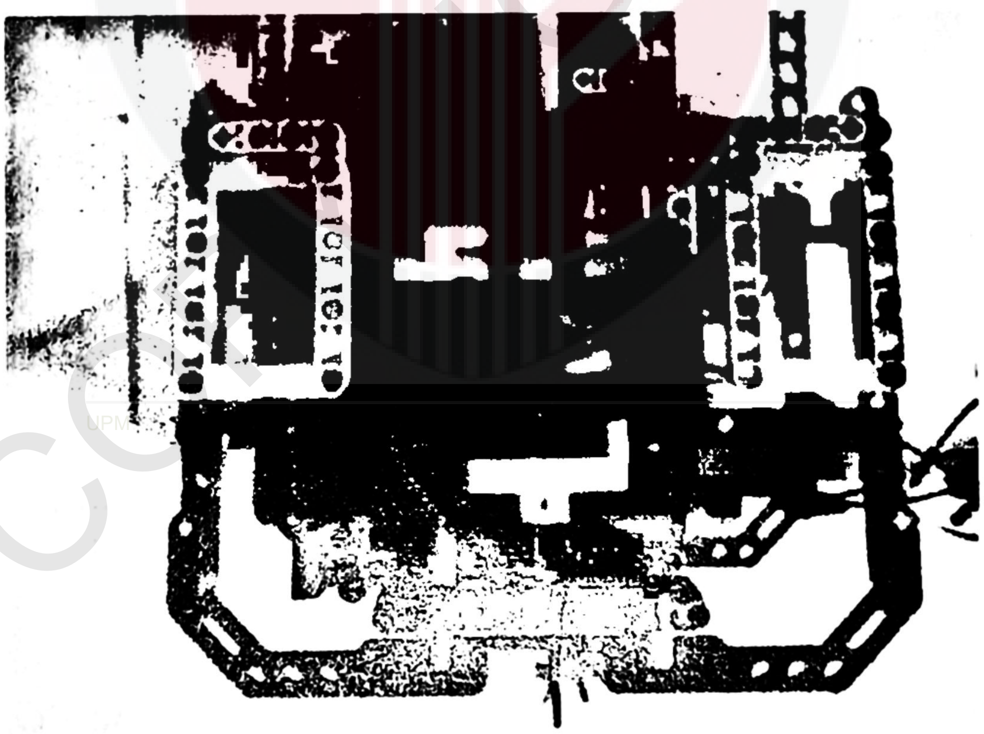

Figure 13: Placement of platform and load cell on the machine

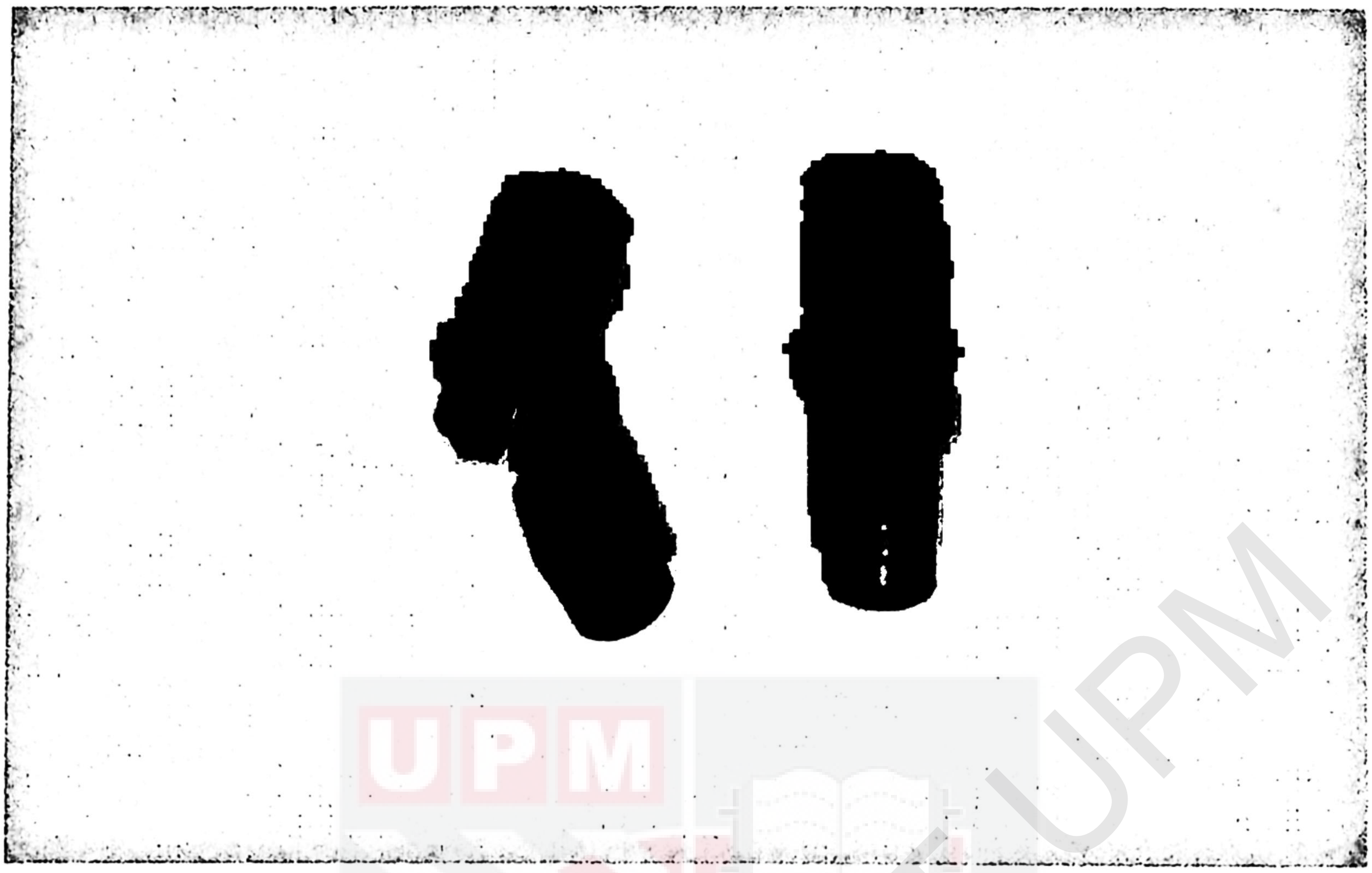

Figure 14: Fractured gasket during pre-testing of machine

A stronger alternative was required to interlock the Lego bricks while being able to withstand loads during the machine is operating. Screws and nuts were used to replace the Lego plastic gaskets. However, it limits the flexibility of the Lego brick to interconnect with several bricks at once. Cable ties were also used to further support the structure of the platform, as shown in Figure 15.

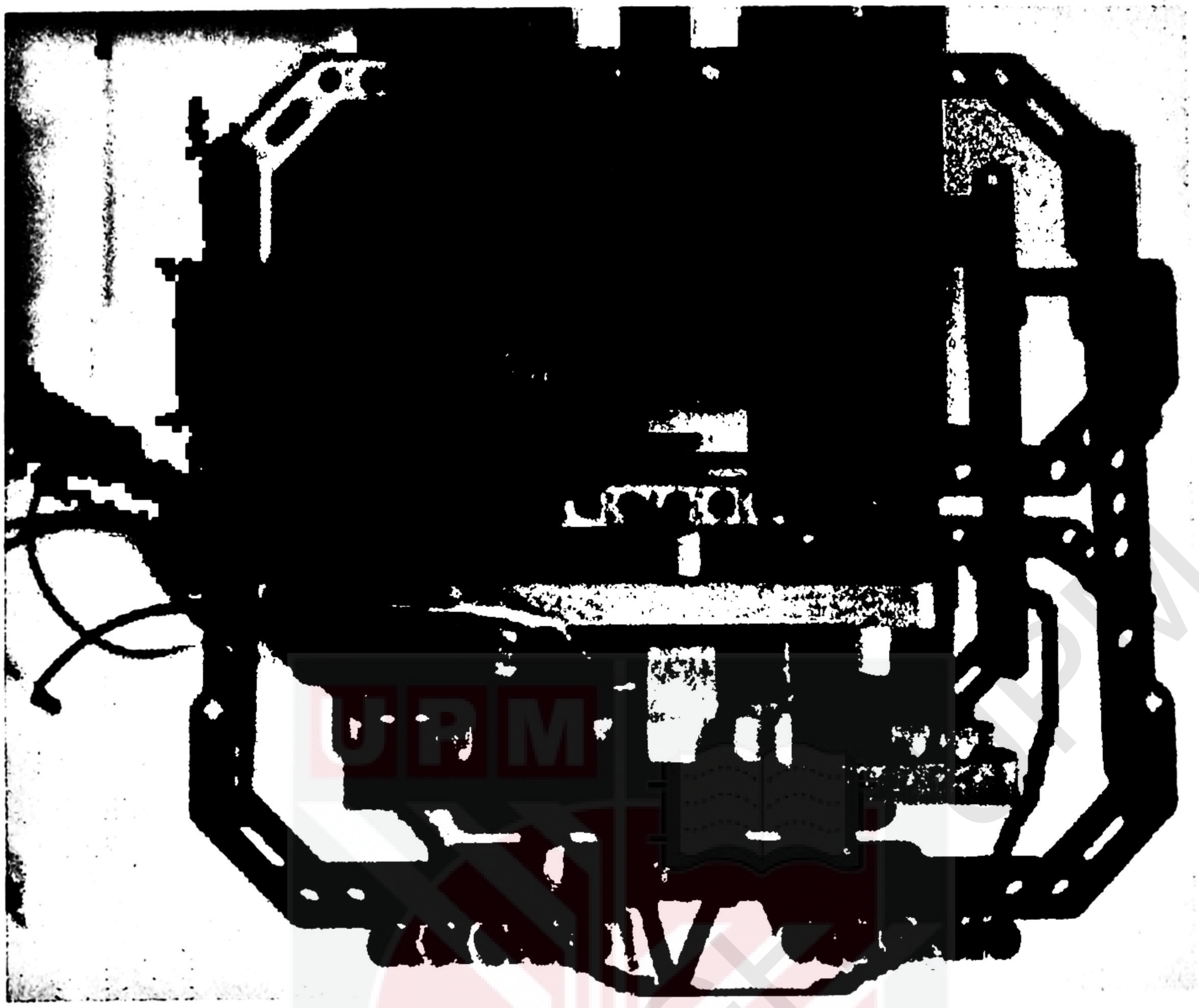

Figure 15: Load cell and platform structure supported with a cable tie

The improvised version of the platform was able to withstand constant supply of load and was capable of performing a repeated number of tests performed on the machine without failing.

4.5 SENSOR CALIBRATION

Based on Figure 16, a linear relation can be observed between the relationship of the weight of a load sample and the output amplitude of the load cell. Based on the relation, an equation relating the values of the two variables was obtained. The slope obtained from Figure 16 was 0.2367 g^{-1} and an intercept value of -4298 g. Based on the plot the following equation was obtained.

$$Weight\,, g = 0.2367\ (Amplitude) - 4298$$

The value was included in the programming code which enables the program to process real-time weight data supplied by the motor.

Table 1: Table of amplitude reading in correspondence with the amount of weight

Weight (g)	Sensor Amplitude Output Reading						Average Amplitude Reading
	V1	V2	V3	V4	V5	V6	
5.66	18176	18177	18181	18180	18183	18182	18179.83
11.32	18200	18208	18204	18207	18201	18206	18204.33
16.98	18225	18230	18226	18229	18228	18224	18227.00
22.64	18248	18256	18249	18255	18225	18254	18247.83
28.30	18272	18281	18273	18278	18274	18277	18275.83
33.96	18294	18302	18297	18300	18298	18299	18298.33
39.62	18320	18327	18321	18325	18323	18321	18322.83
45.28	18345	18346	18349	18348	18347	18346	18346.83
50.94	18368	18374	18370	18369	18372	18373	18371.00
56.60	18390	18398	18396	18391	18394	18397	18394.33

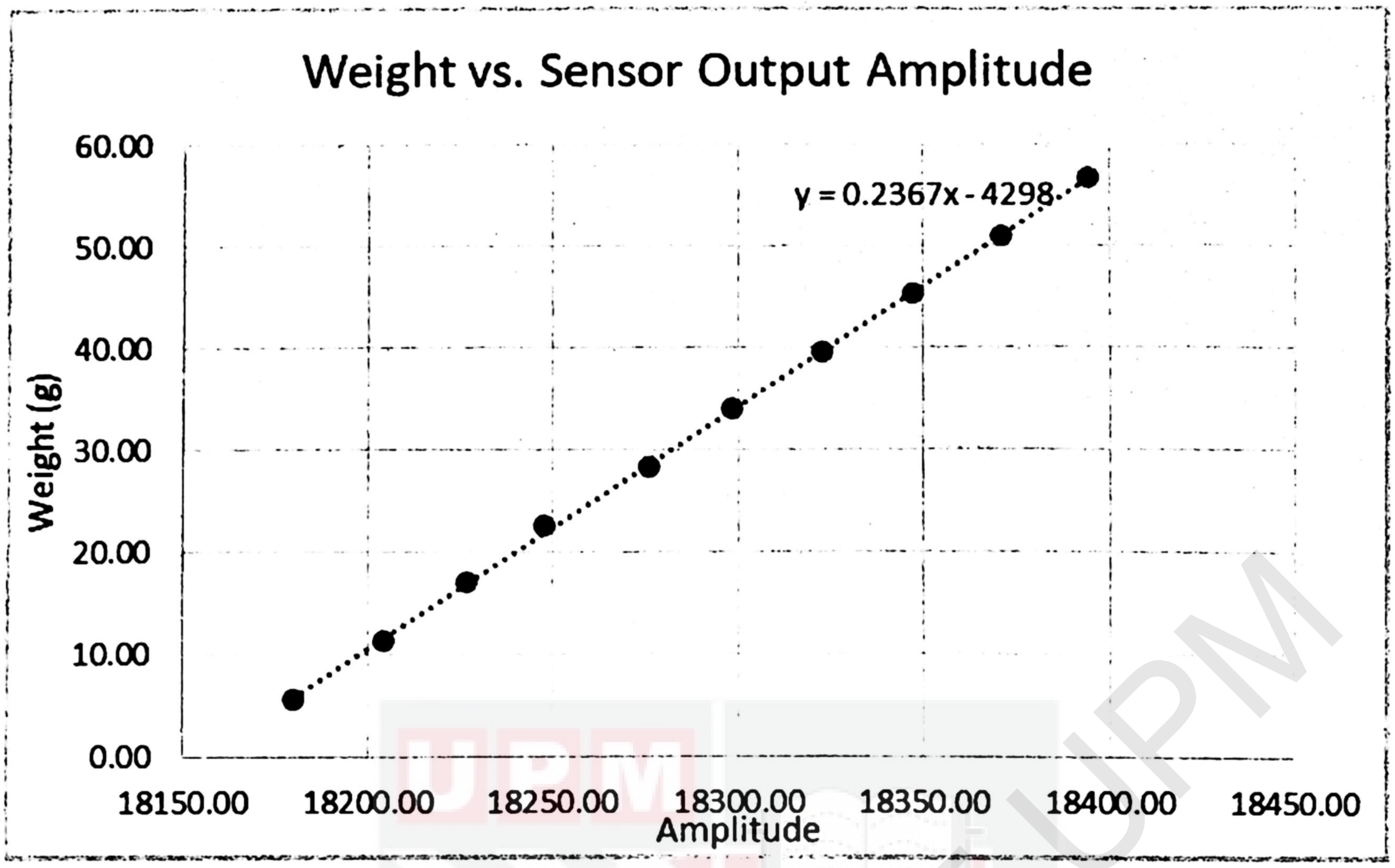

Figure 16: Graph of the weight of load against load cell output amplitude

Based on Figure 17, it was observed that the relationship between the distance of the object from the infrared sensor in mm and the output voltage given out by the infrared sensor is that of a cubic equation. Based on the graph, the following relation was obtained.

$$Distance, mm = 0.8195\,V^3 - 4.0503\,V^2 + 8.7636\,V + 10.215$$

By applying the constants obtained from the equation and implementing it on the software program, a real-time sample displacement data can be collected.

Load cell and infrared sensor were both activated simultaneously allowing the machine to gather accurate data of weight and displacement data on a test sample in real time.

Further mathematical programs and constants were included in the program to obtain stress and strain value needed to properly study the mechanical strength of a sample.

Table 2: Table of voltage reading from infrared sensor and distance of the object from the infrared sensor

Distance	Sensor Output Voltage (V)			Average Voltage
	V1	V2	V3	
32	3.93	3.89	3.95	3.92
30	3.81	3.84	3.76	3.80
28	3.71	3.68	3.73	3.71
26	3.42	3.37	3.45	3.41
22	2.84	2.78	2.83	2.82
20	2.58	2.60	2.60	2.59
18	2.04	1.95	2.98	2.32
16	1.13	1.11	1.12	1.12
14	0.33	0.28	0.27	0.29
12	0.21	0.19	0.20	0.20
10	0.16	0.15	0.15	0.15
8	0.15	0.15	0.15	0.15

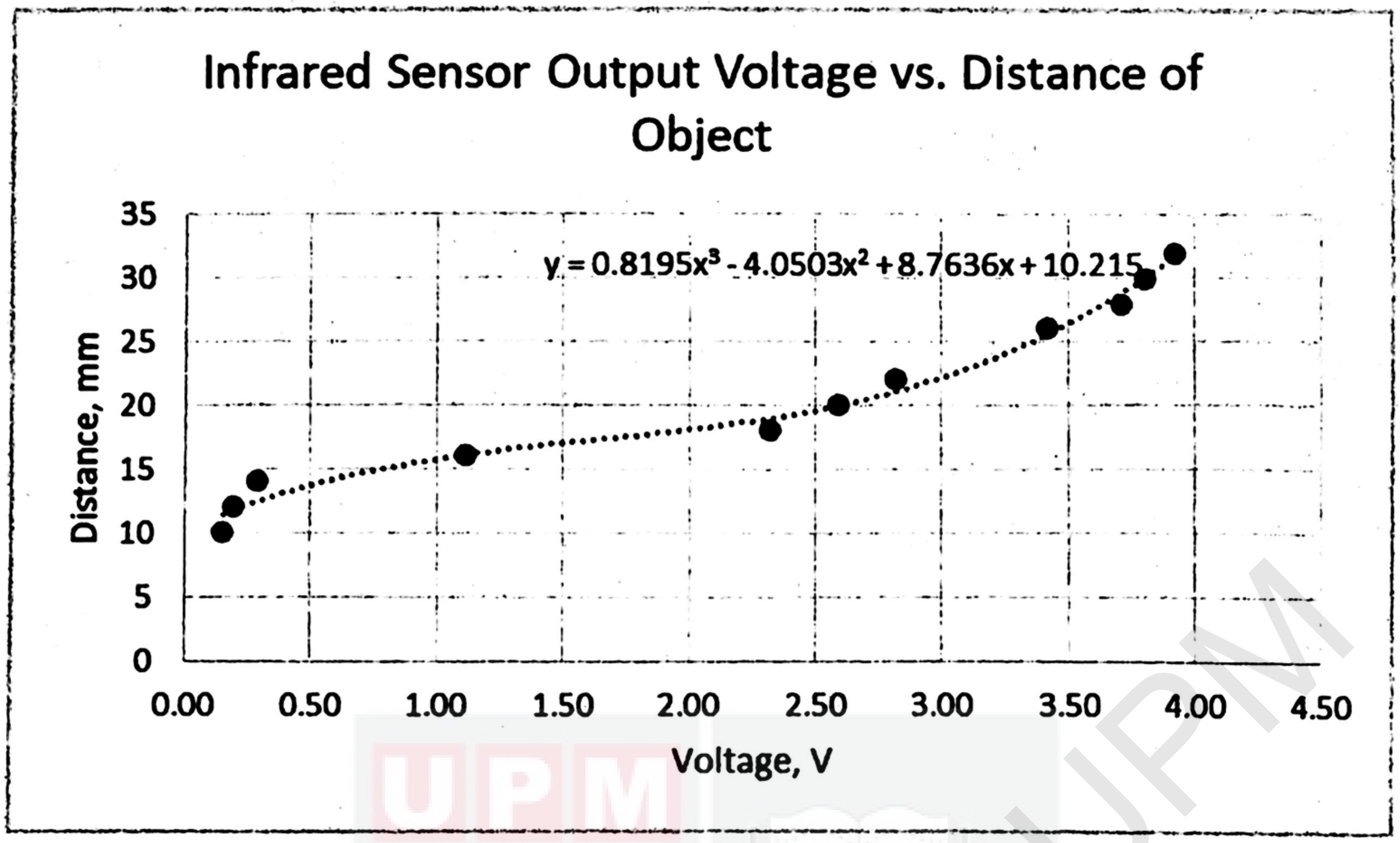

Figure 17: Graph of infrared sensor output voltage against the distance of the object from the sensor

4.6 USER INTERFACE

Figure 18 shows the user interface for the Lego Tabletop Universal Testing Machine. There are several easy to understand buttons, real-time graph, and also tables which will record raw data to be kept by the user. Users only need to choose the mode of test that needs to be performed on the sample. On the left panel, the interface simply asks the user to fill in the required information regarding the sample that will be tested and confirmed. Once confirmed, the user needs to move on to the right panel to initiate the drive system and wait for the test to finish.

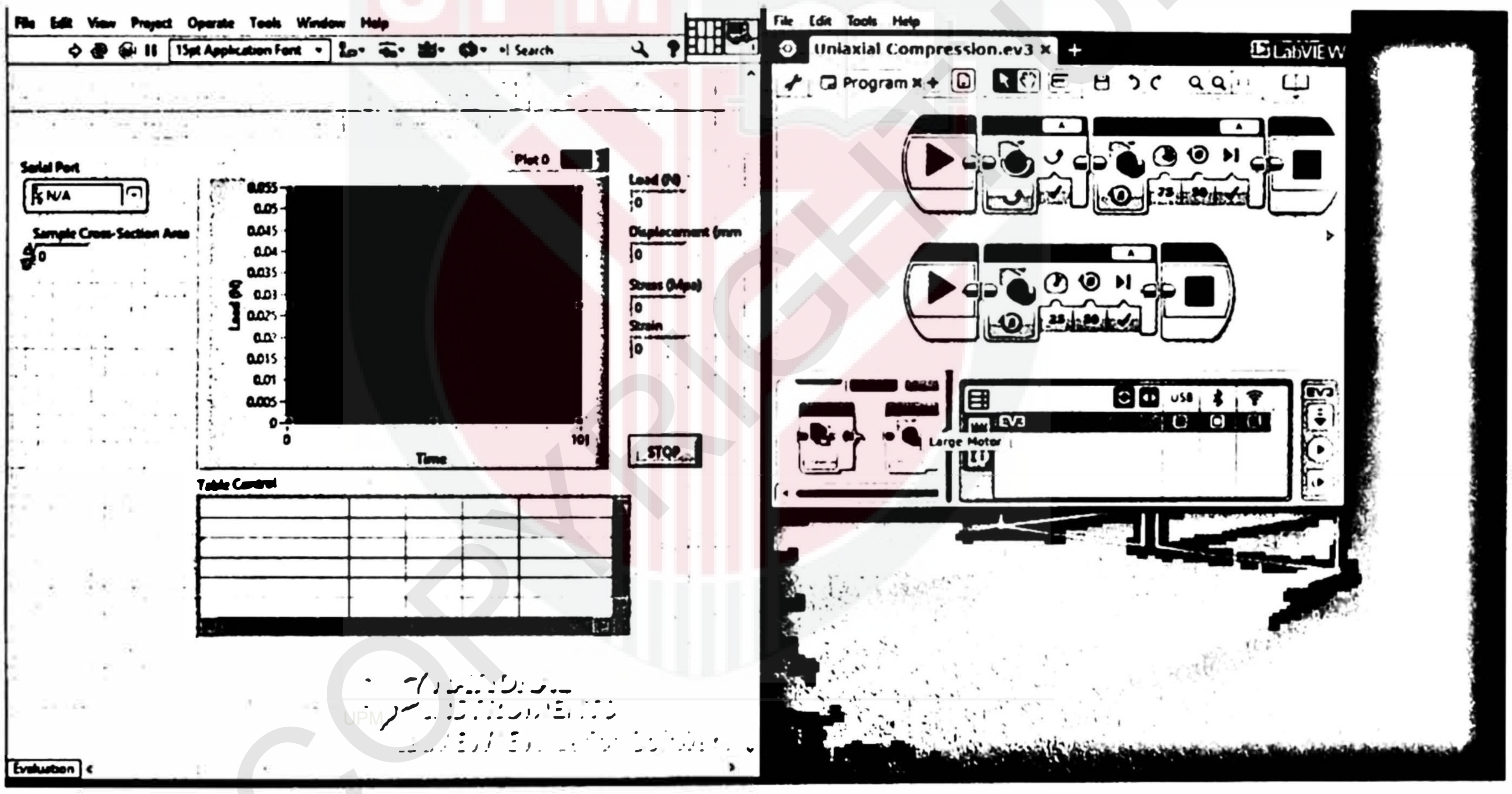

Figure 18: User interface for Lego Tabletop Universal Testing Machine

Once finished the raw data from the table can directly be exported into Microsoft Excel and a stress-strain graph can be plotted. The data will be automatically deleted once the program is reset. User can choose to save it as an excel file for keepsake. This interface is very simple and easy to understand thus enabling new users to operate the machine with a simple guide provided. This program, however, operates by two different separate programs. While, combining both programs may reduce code efficiency and reduce execution speed, combining both programs could decrease the amount of hardware required by the machine, thus reducing its total production cost.

4.7 PERFORMANCE TEST

Figure 19 shows a stress-strain graph produced by the Lego Tabletop Universal Testing Machine. The graph shows a lag reading of the strain value. The lag reading was probably due to the sensor's limitation. The Infrared sensor can only measure a minimum distance of 10 mm from its transmitter and a maximum of 30 mm object distance from the sensor with a sensitivity of 1.0 mm. Correct positioning of the sensor and appropriate dimensioning of the sample may help overcome this issue. Despite that, the device has proved its capability to generate a stress-strain curve based on the actual test

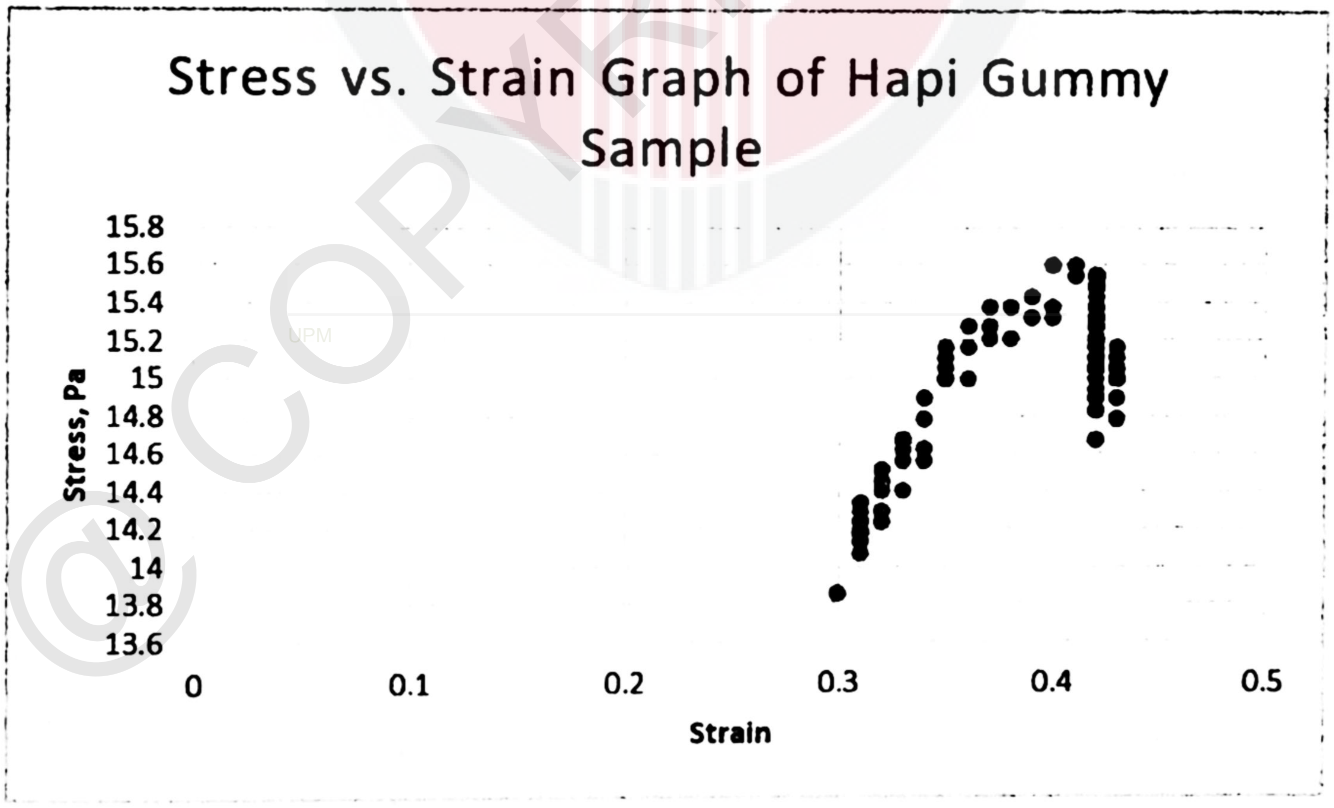

Figure 19: Stress-strain graph of Hapi Gummy sample produced by Lego Tabletop Universal Testing Machine

5.0 CONCLUSIONS AND RECOMMENDATION

5.1 CONCLUSION

The design, construction, calibration, performance of the Tabletop Universal Testing Machine were discussed. Different types of hardware with low cost were tested in this design such as 5kg bar load cell, MH infrared sensor, and Arduino UNO microcontroller. The design was able to detect displacement as small as 1.00 mm and a maximum load of 50 N. The sensitivity of the load cell can be calibrated up to 0.01 g. A user-friendly interface was also successfully developed by interfacing Arduino UNO, EV3 Mindstorms Intelligent Brick, and LabVIEW software.

5.2 RECOMMENDATION

Recommendations for future development of the Tabletop Universal Testing Machine are:

1. Designing custom Lego part for the compression platform and interchangeable parts for different test modes using computer aided drawing CAD and 3D printer.
2. Use Arduino to control the drive system of the device and eliminating the dependency on EV3 Mindstorms Intelligent Brick for motor control.

6.0 REFERENCES

Annappa, C. M., & Panditrao, K. S. (2012). Application of value engineering for cost reduction- a case study of universal testing machine. *International Journal of Advances in Engineering & Technology, July 2012.*

Charalambides, M. N., Goh, S. M., Wanigasooriya, L., & Williams, J. G. (2005). Effect of friction on uniaxial compression of bread dough, *0*, 3375–3381.

Del Vecchio, F. J. C., Reis, J. M. L., & da Costa Mattos, H. S. (2014). Elasto-viscoplastic behaviour of polyester polymer mortars under monotonic and cyclic compression. *Polymer Testing*, *35*, 62–72. https://doi.org/https://doi.org/10.1016/j.polymertesting.2014.02.007

Huerta, E., Corona, J. E., Oliva, A. I., Avilés, F., & González-Hernández, J. (2010). Universal testing machine for mechanical properties of thin materials. *Revista Mexicana de Fisica*, *56*(4), 317–322.

Ishihara, S., Nakao, S., Nakauma, M., Funami, T., Hori, K., Ono, T., … Nishinari, K. (2013). Compression Test of Food Gels on Artificial Tongue and Its Comparison with Human Test. *Journal of Texture Studies*, *44*. https://doi.org/10.1111/jtxs.12002

Sandgaard, M. V., & Jensen. (2015). THE LEGO GROUP. Retrieved from https://www.lego.com/en-us/aboutus/lego-group

Tanner, R. I., Qi, F., & Dai, S. (2018). Bread dough rheology and recoil, (January 2008). https://doi.org/10.1016/j.jnnfm.2007.04.006

Tock, C., Gates, F., Speirs, C., Tucker, G., Robbins, P., & Cox, P. (2013). A Method for Measuring Stickiness of Bread Dough, *21*, 195–198.

APPENDICES

APPENDIX 1: LabVIEW LINX firmware for use with Arduino Mega 2560

Arduino_Mega2560_HX11 | Arduino 1.8.9

File Edit Sketch Tools Help

Arduino_Mega2560_HX11 §

```
/****************************************************************************************
**  This is example LINX firmware for use with the Arduino Mega 2560 with the serial
**  interface enabled
** Custom command for reading out HX711 was added by Bytoe, based on
** arduino Queuetue_HX711_library, available in the Arduino library repositories
**  (Under Arduino IDE, include library > Manage Libraries)
** Only one subfunction was implemented (read)  many more are available. See
** https://github.com/queuetue/Q2-HX711-Arduino-Library for more info
** For more functions [illegible] could be used as well, with
**
**  For more information see           www.labviewmakerhub.com/linx
**  For support visit the forums at    www.labviewmakerhub.com/forums/linx
**
**  Written By Sam Kristoff
**
**  BSD2 License
****************************************************************************************/

//Include All Peripheral Libraries Used By LINX
#include <SPI.h>
#include <Wire.h>
#include <EEPROM.h>
#include <Servo.h>

//Include Device Sepcific Header From Sketch>>Import Library (In This Case LinxArduinoMega2560.h)
//Also Include Desired LINX Listener From Sketch>>Import Library (In This Case LinxSerialListener.h)
#include <LinxArduinoMega2560.h>
#include <LinxSerialListener.h>


//Create A Pointer To The LINX Device Object We Instantiate In Setup()
LinxArduinoMega2560* LinxDevice;

// Library inclusion, pin declaration and initialisation:
#include <Q2HX711.h>
const byte hx711_data_pin = D3; // HX711 data line connected to arduino D3
const byte hx711_clock_pin = D3; // HX711 clock line connected to arduino D3

Q2HX711 hx711(hx711_data_pin, hx711_clock_pin);

//Initialize LINX Device And Listener
void setup()
{
  //Instantiate The LINX Device
  LinxDevice = new LinxArduinoOne();

  //The LINX Listener Is Pre Instantiated, Call Start And Pass A Pointer To The LINX Device And The UART Channel To Listen On
  LinxSerialConnection.Start(LinxDevice, 0);

}

void loop()
{
  //Listen For New Packets From LabVIEW
  LinxSerialConnection.CheckForCommands();

  //Your Code Here, But It will Slow Down The Connection With LabVIEW

LinxSerialConnection.AttachCustomCommand(0x9, readout_HX11);
// Custom commands should have number 0-15

// here, custom command nr 9 was chosen => = hex '9'

}


// %%%%%%   Custom command definition    %%%%%%%%%
int readout_HX11(unsigned char numInputBytes, unsigned char* input, unsigned char* numResponseBytes, unsigned char* response){
// Author: Bytoe
// date: 12-05-2016
//
// Functionality:
// Converts a float into an int, then into 2 bytes and sends this to Labview,
// You can apply a scaling factor before, to choose the required resolution.
//
// Inputs (from Labview)
// * Res_inverse (default 100); // inverse of resolution  Default: 100  Range  1-254
//       e.g. 8 means a resolution of 0.125, res_inverse = 100 a resolution up to to 0.01
// Outputs
//       2 bytes containing data of a float
//       1 byte containing the (inverse of) the requested resolution
// Parameters.
// * float input_value. This is an example code with FIXED value that will be send to Labview
//          You can easily extend this code by calling another (sub)function
        // Range: 16 bits => number between -32767/res_inverse to +32767/res_inverse.



//float input_value = -55.625; // °C Fictive example  Multiply by 8 to get an int with a resolution of 0.125
//unsigned int res_inverse = input[0]; // inverse of resolution. Default  8  Range  1-254
//signed int int_to_send = input_value* res_inverse ; // with input_value  =  -55.625 and res_inverse = 8  this is -445, or in bits  '1111 1110'  '0100 0011' => High byte: '254', low byte '67'
signed int rawsignal = int (float(hx711.read()/100.0));

unsigned int value =rawsignal;    // Make the value an unsigned integer, to shift 0 in from the left, instead of ones.

// send two bytes and res_inverse
*numResponseBytes = 2;
//response[0] = (value & 0x0000) >> 16;   //mask all but MSB, shift 16 bits to the right; // to test  try B10101010 ; // default  MSB
response[0] = (value & 0xFF00)  8; //mask top bits and bottom bits, shift 8
response[1] = (value & 0x00FF) ; //mask top bits, no need to shift
// response[2] = byte (res_inverse & 0x00FF);
return 0;
};
```

APPENDIX 2: Modified LabVIEW LINX firmware for use with Arduino Uno

```
//Include All Peripheral Libraries Used By LINX
#include <SPI.h>
#include <Wire.h>
#include <EEPROM.h>
#include <Servo.h>

//Include Device Sepcific Header From Sketch>>Import Library (In This Case
LinxArduinoUno.h)
//Also Include Desired LINX Listener From Sketch>>Import Library (In This Case
LinxSerialListener.h)
#include <LinxArduinoUno.h>
#include <LinxSerialListener.h>

//Create A Pointer To The LINX Device Object We Instantiate In Setup()
LinxArduinoUno* LinxDevice;

// Library inclusion, pin declaration and initialisation:
#include <Q2HX711.h>
int hx711_data_pin = A1; // HX711 data line connected to arduino D3
int hx711_clock_pin = A2; // HX711 clock line connected to arduino D2

Q2HX711 hx711(hx711_data_pin, hx711_clock_pin);

//Initialize LINX Device And Listener
void setup()
{
  //Instantiate The LINX Device
  LinxDevice = new LinxArduinoUno();

  //The LINX Listener Is Pre Instantiated, Call Start And Pass A Pointer To The LINX
Device And The UART Channel To Listen On
  LinxSerialConnection.Start(LinxDevice, 0);

}

void loop()
{
  //Listen For New Packets From LabVIEW
  LinxSerialConnection.CheckForCommands();

  //Your Code Here, But It will Slow Down The Connection With LabVIEW
LinxSerialConnection.AttachCustomCommand(0x9, readout_HX11);
```

```
// Custom commands should have number 0-15
// here, custom command nr 9 was chosen ==> = hex '9'
}
// %%%%%%   Custom command definition    %%%%%%%%%
int readout_HX11(unsigned char numInputBytes, unsigned char* input, unsigned char*
numResponseBytes, unsigned char* response){
//
// Functionality:
// Converts a float into an int, then into 2 bytes and sends this to Labview,
// You can apply a scaling factor before, to choose the required resolution.
//
// Inputs (from Labview):
// * Res_inverse (default 100); // inverse of resolution. Default: 100. Range: 1-254
//     e.g. 8 means a resolution of 0.125, res_inverse = 100 a resolution up to to 0.01.
// Outputs:
//     2 bytes containing data of a float
//     1 byte containing the (inverse of) the requested resolution.
// Parameters:
// * float input_value. This is an example code with FIXED value that will be send to
Labview.
//      You can easily extend this code by calling another (sub)function.
    // Range: 16 bits => number between -32767/res_inverse to +32767/res_inverse.
//float input_value = -55.625; // °C Fictive example. Multiply by 8 to get an int with a
resolution of 0.125
//unsigned int res_inverse = input[0]; // inverse of resolution. Default: 8. Range: 1-254
//signed int int_to_send = input_value* res_inverse ; // with input_value  =  -55.625 and
res_inverse = 8, this is -445, or in bits: '1111 1110'  '0100 0011' => High byte: '254', low
byte '67'
signed int rawsignal = int (float(hx711.read()/100.0));

unsigned int value =rawsignal;    // Make the value an unsigned integer, to shift 0 in from
the left, instead of ones.

// send two bytes and res_inverse:
*numResponseBytes = 2;
//response[0] = (value & 0x0000) >> 16;   //mask all but MSB, shift 16 bits to the right; //
to test: try B10101010 ; // default: MSB
response[0] = (value & 0xFF00) >> 8; //mask top bits and bottom bits, shift 8
response[1] = (value & 0x00FF) ; //mask top bits, no need to shift.
// response[2] = byte (res_inverse & 0x00FF);
return 0;
};
```