

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



## **College of Engineering and Technology**

### **Mechanical Engineering department**

#### **Graduation Project**

## **Rehabilitation and Development of an Automatic Compaction Machine**

#### **Project Team**

Rami Hasib Naser Al-din

Luai Husam Zatari

#### **Supervisor:**

Dr. Mohamed Al-Qawasmi

**Hebron-Palestine**

2013

# Abstract

Compaction machine is used to compact a specimen of soil for proctor (100mm diameter) and CBR or California Bearing Ratio (150mm) tests to eliminate the laborious hand compaction method, in which a rammer is left to fall freely through a guide from a specific height to the specimen, and lifted then to the same height until all the specimen is compacted. The university's previous machine was completely damaged, in the project the machine is developed to keep up with new technologies, in which the machine is controlled by a logical controller, the safety in the machine is increased by closing the working area and prevent the machine from working whenever the door is opened, and it was also improved to be more user friendly using a touch screen that shows the options, errors, number of turns and anything blocking the machine from functioning, and successfully the machine was built to help in preparing a more accurate specimen.

# Acknowledgment

We would like to express our gratitude to our parents, our families and our friends, whom we admire and have brought love, warmth, patience, and understanding to our lives, and without whom we would be unable to complete our project. We want to thank Dr. Mohamed Al-Qawasmi, our supervisor, for his patience, help and guidance through the entire project who appreciated us for our work and motivated us and before all of them to God who gave us strength, intelligence and patience to succeed, and special thanks to Eng.hussien Amro, for his support and help.

# List of Contents

<b>Abstract</b> .....	<b>I</b>
<b>Acknowledgment</b> .....	<b>II</b>
<b>List of Contents</b> .....	<b>III</b>
<b>List of Figures</b> .....	<b>VI</b>
<b>List of Tables</b> .....	<b>VIII</b>

## **CHAPTER 1: Introduction.**

1.1 Introduction .....	1
1.2 Recognition of the need .....	2
1.3 literature Review .....	2
1.4 Project Objectives .....	3
1.5 Time plan .....	3
1.6 Cost Estimation and Budget Breakdown .....	5
1.7 Risk Management .....	6
1.7.1 Technology Risk .....	6
1.7.2 Risks Avoidance .....	7

**CHAPTER 2: California Bearing Ratio (CBR) and Proctor tests.**

2.1 Introduction .....8  
2.2 Proctor test .....8  
2.3 CBR test .....11

**CHAPTER 3: Analyses of the Current Compaction Machine.**

3.1 Introduction .....12  
3.2 The Machine’s Mechanical Structure. ....12

**CHAPTER 4: System control Architecture for the Developed Compaction Machine.**

4.1 Introduction .....21  
4.2 Flow Chart and Block Diagram of the Machine .....21  
4.3 System control Architecture .....23  
    4.3.1 User Interface .....23  
    4.3.2 Controller.....27  
    4.3.3 Sensing Element .....30

**CHAPTER5: Software Design.**

5.1 Introduction .....31  
5.2 Pin Mapping .....33

5.3 Counter .....	36
5.4 CBR and Proctor .....	37
5.5 TFT Touch shield .....	40

**CHAPTER6: Electrical Design.**

6.1 Introduction .....	43
6.2 Power Supply .....	44
6.3 Inputs .....	45
6.4 Outputs .....	48

**CHAPTER7: Building and Testing.**

7.1 Introduction .....	50
7.2 Limit switches .....	51
7.3 Counter .....	51
7.4 Outputs .....	52
7.5 Assembly .....	53

<b>References .....</b>	<b>54</b>
-------------------------	-----------

# List of Figures

<b>Figure 1.1</b>	Myers M-100 compaction machine .....	2
<b>Figure 2.1</b>	Hammer for proctor .....	9
<b>Figure 2.2</b>	Standard Proctor compaction test results for a silty clay .....	10
<b>Figure 2.3</b>	Hammer for CBR.....	11
<b>Figure 3.1</b>	The ELE compaction machine (EL28-625) diagram .....	13
<b>Figure 3.2</b>	Photo for the rammer of the current machine .....	14
<b>Figure 3.3</b>	Photos for the yoke of the compaction machine .....	15
<b>Figure 3.4</b>	Photo of the machine’s motor and ratchet .....	16
<b>Figure 3.5</b>	load characteristics curve of the motor .....	17
<b>Figure 3.6</b>	Photo of the machine’s cage.....	18
<b>Figure 3.7</b>	Electrical linear actuator .....	18
<b>Figure 3.8</b>	Sketch for movement of the cage .....	19
<b>Figure 3.9</b>	Control panel .....	20
<b>Figure 4.1</b>	General flow chart for operation of the developed compaction machine.....	22
<b>Figure 4.2</b>	Block diagram of the developed compaction machine .....	23
<b>Figure 4.3</b>	Schneider Magelis touch screen .....	24
<b>Figure 4.4</b>	Viewsonic tablet .....	25
<b>Figure 4.5</b>	2.8” TFT Touch Shield .....	26
<b>Figure 4.6</b>	PIC4550 microcontroller .....	27
<b>Figure 4.7</b>	Mplab IDE .....	27
<b>Figure 4.8</b>	IOIO Board .....	28
<b>Figure 4.9</b>	IOIO manager .....	28

<b>Figure 4.10</b> Arduino Mega.....	29
<b>Figure 4.11</b> Limit switches with roller.....	30
<b>Figure 5.1</b> Input and outputs of the system .....	32
<b>Figure 5.2</b> Pin mapping of peripherals to the Arduino .....	33
<b>Figure 5.3</b> Definition of inputs and outputs .....	34
<b>Figure 5.4</b> Touch shield pins usage on Arduino .....	35
<b>Figure 6.1</b> 220v AC powering circuit .....	44
<b>Figure 6.2</b> Limit Switch Connection .....	45
<b>Figure 6.3</b> Snapshot of switch bounce on an oscilloscope.....	46
<b>Figure 6.4</b> The counter circuit with low-pass filter .....	47
<b>Figure 6.5</b> Control circuit of the outputs .....	48
<b>Figure 7.1</b> sequential design vs. concurrent design .....	51



# List of Tables

<b>Table 1.1</b>	Time table for the first semester time (week) .....	4
<b>Table 1.2</b>	Time table for the second semester time (week).....	4
<b>Table 1.3</b>	Hardware cost .....	5
<b>Table 3.1</b>	The motor plate .....	15
<b>Table 3.2</b>	Torque and speed values of the motor .....	16
<b>Table 4.1</b>	Specification of 2.8" TFT Touch Shield .....	26
<b>Table 4.2</b>	Specification of PIC4550 .....	27
<b>Table 4.3</b>	Specification of the Arduino Mega .....	30
<b>Table 4.4</b>	Specification of the Limit Switch .....	30
<b>Table 5.1</b>	Clock selection bits description in register (TCCR5B).....	36

# CHAPTER 1

## Introduction

### 1.1 Introduction

As the basic structural foundation for highway embankments, earth dams, and almost all constructions, soil materials play an important role in the ultimate success of a project. Whether the working is in clay, silt, sand, gravel, peat, or loam, understanding the soil properties of the working site helps in making good construction decisions. In another hand the loose soils must be compacted to increase their unit weights.

Compaction, in general, is the densification of soil by removal of air, which requires mechanical energy. The degree of compaction of a soil is measured in terms of its dry unit weight. When water is added to the soil during compaction, it acts as a softening agent on the soil particles. Compaction increases the strength characteristics of soil which increases the bearing capacity of foundation constructed over them; it also decreases the amount of undesirable settlement of structure and increase the stability of slopes of embankments. Smooth-wheel roller, sheepsfoot roller, rubber tired roller, and vibratory rollers are generally used in the field for soil compaction.

In laboratory a mechanical compaction machine is used to eliminate the laborious hand compaction method, where the soil is compacted in a mold that has a specific volume and diameter depending on the test. The mold is attached to a base plate at the bottom and to an extension at the top. The soil is mixed with varying amounts of water and then compacted in three layers by a rammer with a specific mass and drop height to suit test requirements, then it delivers a number of blows to each layer. The rammer itself travels across the mold and the table rotates the mold in equal steps on a base that is extremely stable.

## 1.2 Recognition of the Need

The compaction machine is needed in Soil Mechanics Lab at Palestine Polytechnic University Civil Engineering Department, the current one is a damaged old fashion machine. The project is to redesign, develop the existing compaction machine as a Mechatronics system, and make the use of the machine much easier, also safety measures will be taken into consideration such as closing the working area during operation and switching off the machine automatically if any danger to the user or the machine occurs. The Purpose of the project is to give the students in Civil Engineering Department the ability to do the CBR (California Bearing Ratio) test and proctor test with a well prepared specimen, so it can help in giving more accurate results. A better user interface will also be helpful to the students, where one can tell the machine exactly what to do, and what test will the specimen be prepared for. The safety of the user is an important issue therefore every effort will be made to insure the safety of the user, machine, and surroundings.

## 1.3 Literature Review

For the proposed problem there can be more than one solution, following are some of the solutions:-

- 1- Purchasing a brand new compaction machine, according to Alibaba website ([www.alibaba.com](http://www.alibaba.com)) it can cost 2000\$-3000\$ without shipping.
- 2- Design and build a new machine, where new motor and mechanical parts are need in addition to the electrical parts for control; so more costs are added to the project therefore this choice can be eliminated.
- 3- Redesign the current machine in which fewer parts must be purchased so less cost the project is.

There are several companies that manufacture compaction machines for laboratories according to the specifications, such as Myers associates and Troxler companies for laboratories equipment, but the cost is considerably high. Here is a hint about the Myers M-100 compaction machine shown in (figure 1.1):-



**Fig (1.1):** Compaction Machine

*“Unit is designed to automatically compact soil samples rapidly and to produce the accuracy of hand compaction. For standard or modified compaction tests with 5.5 lb. hammer with 12" height of drop or 10 lb. hammer with 18" drop. Compactor accommodates 4" or 6" I.D. molds. 4" round hammer, or 6" pie-shaped hammer compensate for sample drop during compaction. The weight is concentrated at the foot, allowing free fall of the rammer. Hammer changes are made from in front of the compactor. Automatic counter registers number of blows per compaction and can be preset to stop the machine. A start-stop switch controls the operation independently from the counter control. Compaction mold is mounted on the round base; the circular base indexes automatically with each blow to ensure uniform sample compaction. A pie-shaped compactor foot is used with 6" dia. molds to completely cover the soil surface. 115V, 60Hz, 1ph AC”*

#### **1.4 Project Objectives**

- 1- Redesign and develop the existing machine.
- 2- To use a microcontroller to logically control the machine.
- 3- To increase the safety by closing the working area.
- 4- Use sensors and limit switches to feedback the controller.
- 5- Switching off the machine when detecting any danger.
- 6- Building a user interface.

#### **1.5 Time Plan**

The time plan explains the stages in designing and building the system components. The section includes the first table that shows the activities and task scheduling for the first semester, while the second table shows the tasks for the second semester.

**Table 1-1:** Time table for the first semester (week)

Activity	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Selection of the project	█	█													
Literature review			█	█	█	█	█	█	█	█	█				
Analysis of the machine									█	█	█	█			
Evaluating & Reviewing													█	█	
Documentation					█	█	█	█	█	█	█	█	█	█	█
Final edition															█

**Table 1-2:** Time table for the second semester (week)

Activity	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
software design	█	█	█	█	█										
software testing						█	█	█	█	█	█				
hardware testing				█	█	█	█	█	█	█	█	█	█		
interface system												█	█	█	
system testing														█	█
Documentation	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█

## 1.6 Costing and Budget Breakdown

This section lists the overall cost of the project; the cost includes hardware costs, and software costs.

- Ñ **Hardware cost:** includes the cost of the components that are used to implement the project. These costs are shown in Table1-3, which describes the cost of each unit and number of units required.
- Ñ **Software cost:** most of the needed software is available from the university such as Catia and Matlab, the other software that is needed for the PLC and touch screen are included with the hardware price.

**Table 1-3:** Hardware cost

Components	Unit Price	Quantity	Price
Touch screen	300 NIS	1	300 NIS
Cabinet	200 NIS	1	300 NIS
Arduino mega	340NIS	1	300 NIS
Soldering wire	50NIS	1	50 NIS
Soldering board	20NIS	1	20 NIS
Limit switch	30 NIS	4	120 NIS
		<b>Total cost</b>	<b>1020 NIS</b>

## 1.7 Risk Management

The implementation of any project may face many risks during each stage of the project, such as: determining and analyzing the system requirements, designing, implementing and testing the whole system. This section illustrates the problems which might occur during the implementation.

### 1.7.1 Technology Risk

Technology risks can be classified as hardware and software risks:

#### Ñ **Hardware Risks Include:**

- 1- Malfunctions of expensive components such as Arduino mega, or any of the other electrical components that are sensitive devices.
- 2- Some of the components might be hard to find and purchase, so there might be a wasted time to wait for these components.

#### Ñ **Software Risks Include:**

The program may follow undesirable algorithm, but this project algorithm was clear and understandable.

## 1.7.2 Risks Avoidance

- Ñ Be careful when using hardware components and using them according to their specifications.
- Ñ Use the right steps in working safely during the project development.
- Ñ Purchasing an extra spare parts of the inexpensive components, so when any problem occurs the alternative components could be found and replaced easily.
- Ñ The availability of each item must be checked before choosing any of the project's items.



# CHAPTER 2

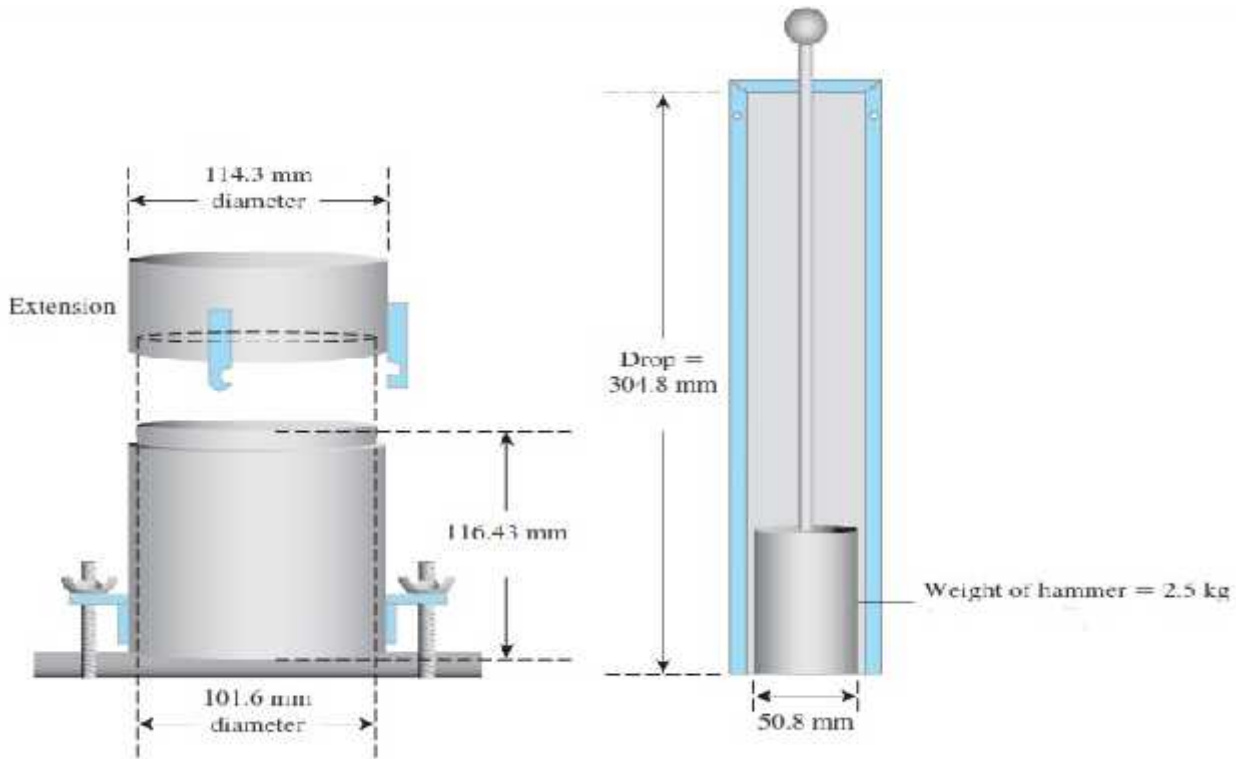
## CBR and Proctor Tests

### 2.1 Introduction

Almost all available design methods for low volume roads are empirical in nature and rely on simple input parameters like CBR for subgrade, subbase and base materials. There is however a strong need to use mechanistic based design analyses to check the empirical based designs on the consequences of overloading, or unexpected poor material quality etc. As discussed in the previous chapter, compaction is the densification of soil by the removal of air particles, compaction is very important in CBR and Proctor tests, where both tests are related in a way such that applying CBR test requires the completion of Proctor test, where it will be discussed briefly in the next sections. In general the tests are used to identify the California bearing ratio (CBR) of the soil, which is the ratio of pressure required to penetrate the compacted soil to the pressure required to penetrate a standard soil.

### 2.2 Proctor Test

The Proctor compaction test is a laboratory method of experimentally determining the optimal moisture content at which a given soil type will become most dense and achieve its maximum dry density. During test, the soil is compacted to determine the maximum dry unit weight of the soil ( $\gamma_d$ ), in a mold that has a volume of  $944\text{cm}^3$ ; the diameter of the mold is 101.6 mm. The soil is mixed with varying amount of water and then compacted in three layers by a hammer (on the right side of figure 2.1) that is placed on top of the mould(left side of figure 2.1) , the hammer delivers 25 blows to each layer, and it has a mass of 2.5 kg and has a drop height of 30.5cm.



**Fig (2.1):** Hammer for Proctor

For each test, the moist unit weight of compaction ( $\gamma$ ), can be calculated as

$$\gamma = \frac{W_2 - W_1}{V} \quad (2.1)$$

Where  $W_2$  = weight of the mold and the compacted soil

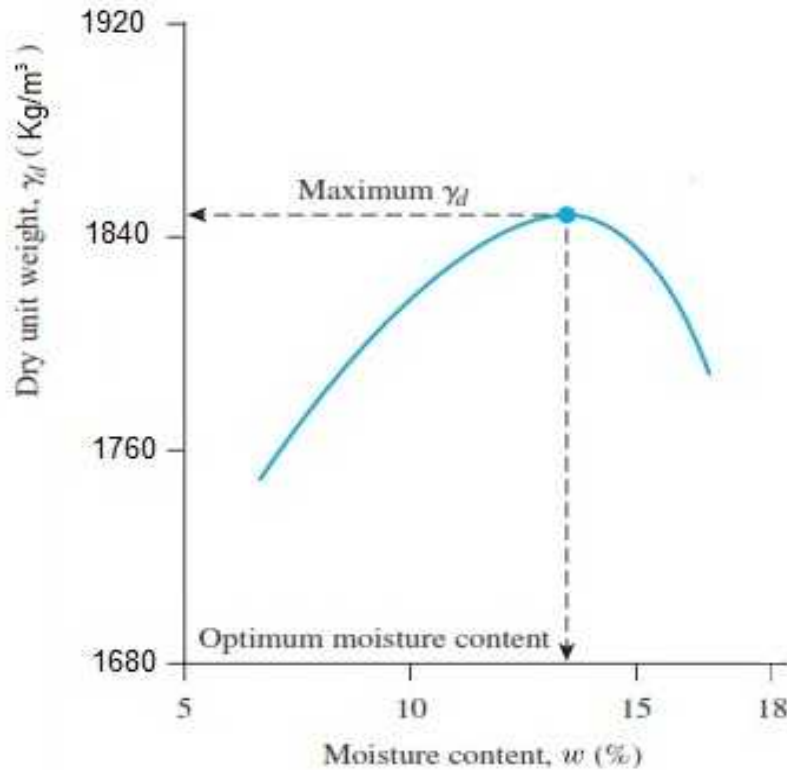
$W_1$  = weight of the mold

$V_m$  = volume of the mold ( $944\text{cm}^3$ )

The moisture content of the compacted soil is determined in laboratory. With the known moisture content, the dry unit weight ( $\gamma_d$ ) can be calculated as

$$\gamma_d = \frac{\gamma}{1 + \frac{\omega(\%)}{100}} \quad (2.2)$$

Where  $\omega(\%)$  = percentage of moisture content.



**Fig (2.2):** Standard Proctor compaction test results for a silty clay

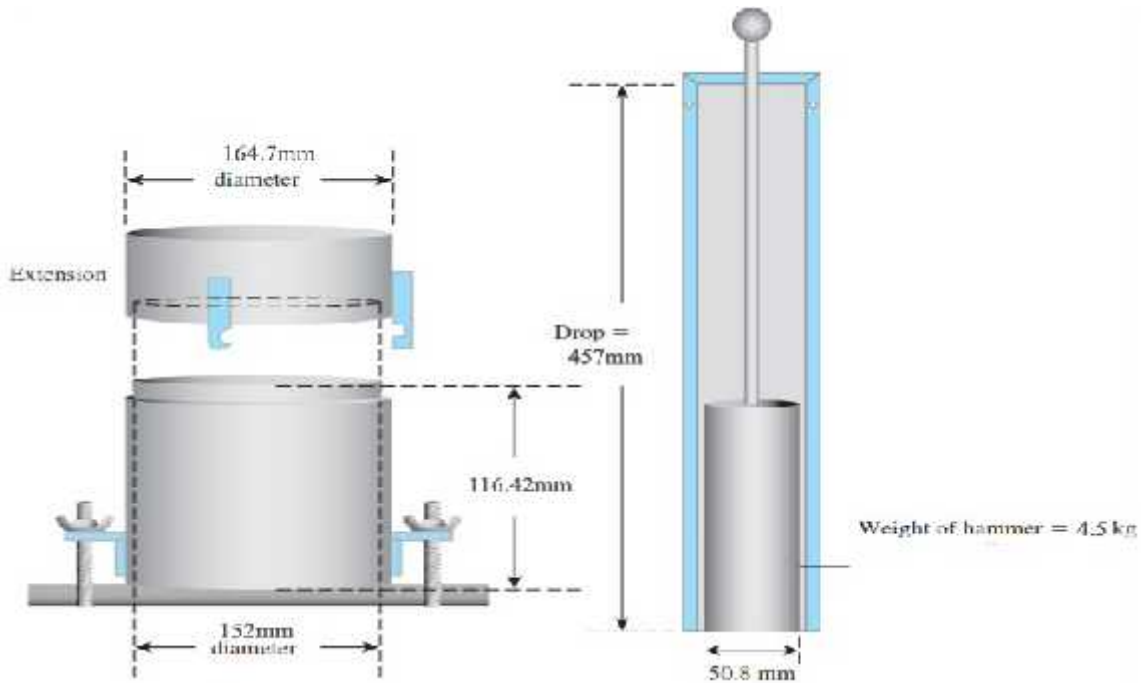
The values of  $\gamma_d$  determined from Eq. (2.2) can be plotted against the corresponding moisture content to obtain the maximum dry unit weight and optimum moisture content for the soil. (Figure 2.2) shows such a. After determining the moisture content that gives the maximum unit dry weight of the soil sample, The CBR test can be performed using the determined moisture content.

### 2.3 California Bearing Ratio (CBR) Test

The California bearing ratio (CBR) is a penetration test for evaluation of the mechanical strength of road subgrades, subbase, and basecourses. The test is performed by measuring the pressure required to penetrate a soil sample with a plunger of standard area. The measured pressure is then divided by the pressure required to achieve an equal penetration on a standard crushed rock material, the standard material for this test is crushed California limestone which has a CBR value of 100%. The test specimen is prepared in a mold that has a volume of 2125 cm<sup>3</sup>; the diameter of the mold is 152 mm see (figure 2.3), the soil is mixed with a percentage of moisture content that gives the maximum dry unit (determined earlier in

proctor test section 2.2) and then compacted in five layers by a hammer with 4.54Kg that gives 56 blows to each layer, and has a drop of 45.5 cm.

$$CBR = \frac{\text{penetration pressure of sample}}{\text{penetration pressure of standard soil}}$$



**Fig (2.3):** Hammer for CBR

# CHAPTER 3

## Analyses of the Current Compaction Machine

### 3.1 Introduction

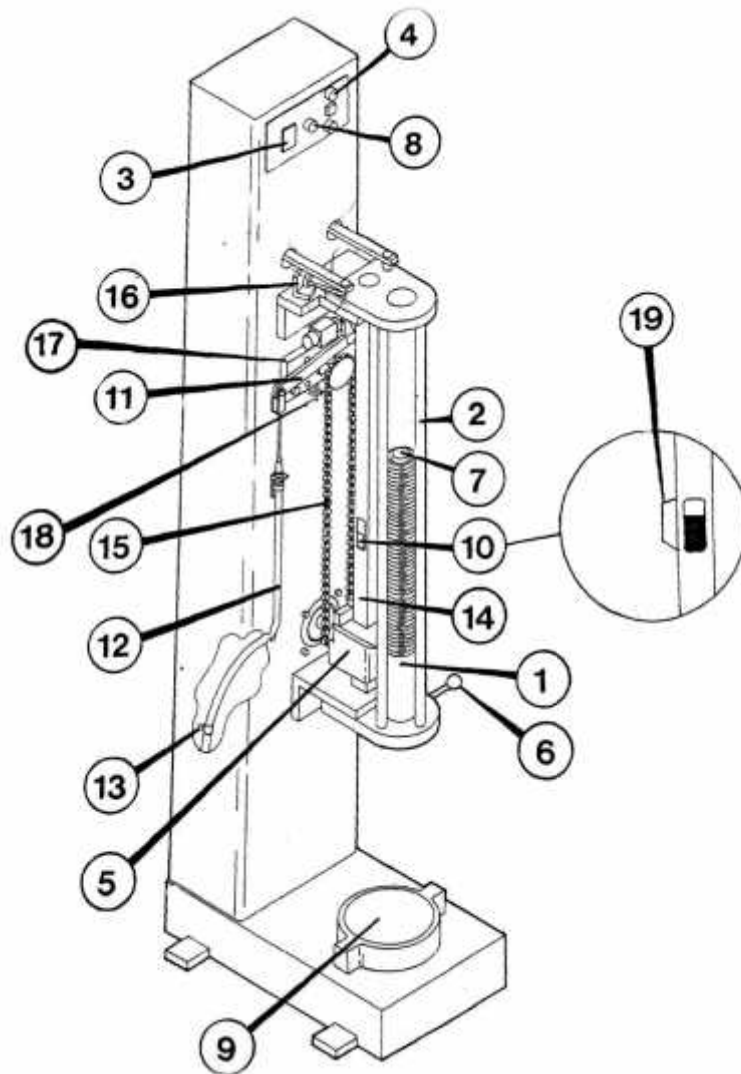
The (Engineering Laboratory Equipment) ELE's machine EL28-625 which is the University's existing compaction machine is designed to automatically compact specimens for proctor and CBR tests; the machine's height of the rammer drop is adjustable from 300mm to 450mm by means of a simple switch. The weight of the rammer can be altered from 2.5Kg to 4.5Kg to suit test requirements. The rammer itself travels across the mould and the table rotates the mould in equal steps on a base that is extremely stable.

The blow counter and rammer movements are controlled by a printed circuit board with traditional ways of control using relays. The number of blows for each layer can be set at the beginning of the test by means of a simple thumbwheel control. At the completion of each layer the machine resets itself for the next layer.

### 3.2 The Machine's Mechanical Structure

This section will discuss the function of each part in the existing compaction machine to help understanding its operational techniques; the following figure (3.1) shows a diagram for the EL28-625

1. The outer rammer
2. The cage
3. Key pad
4. Start button
5. The yoke
6. Guard
7. The inner rammer
8. CBR or Proctor switch
9. The base
10. Thumb button
11. Lever
12. Cable
13. Grease nipple
14. The yoke's guide
15. Chain
16. Self lock
17. The lever's base
18. Limit switch
19. Knockout block



**Fig(3.1):** The ELE compaction machine (EL28-625) diagram

- **The Rammer**

The rammer shown in figure (3.2A) is the main part in the compaction of a specimen where it's the part that falls from the specified height to the mould. The rammer consists of two parts: the outer body (1) weighing 2.5kg shown in Figure(3.2C), and the inserted steel rod (7) weighing 2kg shown in Figure (3.2B), the total weight of the rammer is 4.5kg, the diameter of the rammer is 50.8mm, to reduce the weight of the rammer the rod must be unscrewed, thus reducing the total weight to 2.5kg, the diameter of the rammer is 50.8mm. The rammer is contained in a cage (2) that consists of three rods, also the function of the cage is to guide the rammer to fall on the mould.

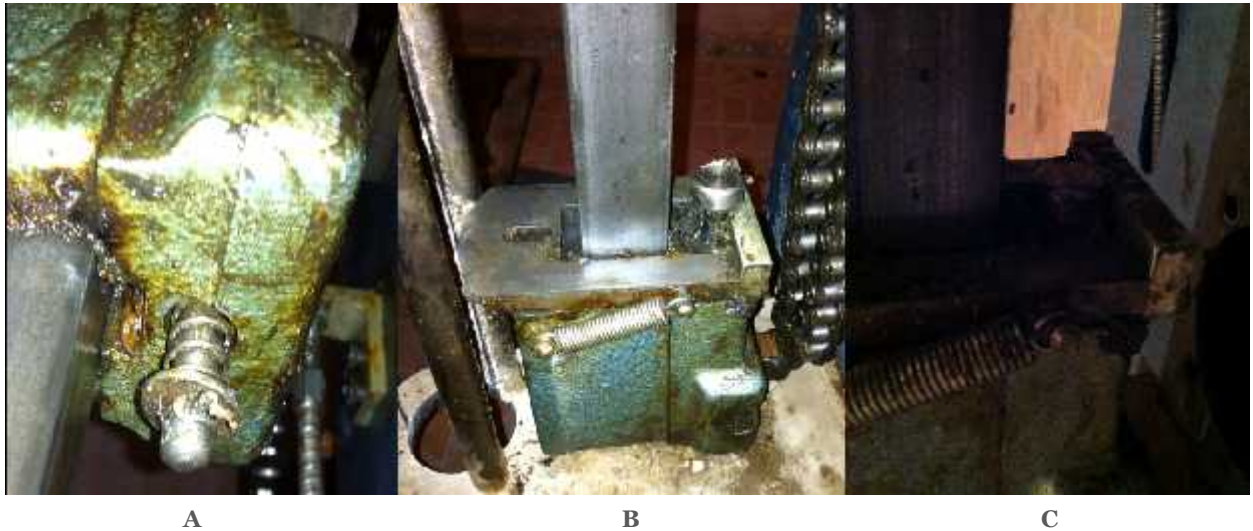


**Fig (3.2):** Photo for the rammer of the current machine

- **The Lifting Yoke**

The yoke (5) shown in the Figure (3.3) is the part that lifts the rammer to the specified height. The height is adjusted from a thumb button (10) where fixing it to the lower position pushes out a knockout block (19) at a height of 300mm, and putting it to the uppermost position conceal the lower block, so the upper knockout block at a height of 450mm is left. The role of the block is to pull the yoke which is connected to a tension spring that keeps it in position attached to the rammer as shown in

Figure(3.3B), whenever the yoke is pulled by any of the blocks a lever is pulled by a compression spring shown in Figure(3.3A) to lock the yoke in the reverse position as shown in Figure(3.3C) so the rammer is left to drop down, again whenever the yoke reaches the base; the lever is pushed up by the base so the spring can pull again the yoke to catch the rammer and lift it up until reaching the knockout block.



**Fig (3.3):** Photos for the yoke of the compaction machine

- **The motor**

The motor shown in Figure (3.4A) is a single phase AC motor used in the machine with the rated values shown in Table (3.1), attached to the motor is the gearbox that works to reduce the angular speed of the motor by a ratio of  $a=2.5$  , therefore increasing the torque.

**Table 3-1:** The motors plate

Norman Electrical Co. Ltd.	50Hz	Frame R12 B42
SER. NO B137385H	2.0 A	186 WATT
240V	142 rpm	SINGLE PHASE



The motor is used to rotate the chain (15) that is connected to the yoke, also attached to the chain is a roller that works to operate a lever (11), this in turn acts upon a special cable (12) which passes down to the base (9) around the table operating ratchet shown in Figure (3.4B)



**Fig (3.4):** Photo of the machine’s motor and ratchet.

The operation of the lever pulls up the cable that causes the ratchet to rotate in counter clockwise direction, but the counter clockwise rotation of the ratchet does not affect the mould; the rotation of the mould is caused by the attached spring which acts to rotate the ratchet in clockwise direction. At the time when the motor is lifting the yoke and the rammer, the load torque ( $T_L$ ) reaches its maximum value, according to equation (3.1), Figure (3.5) shows the load characteristics of the existing motor.

$$T_L = mgR_p \quad (3.1)$$

Where  $m$  = the mass of the rammer + the mass of the yoke

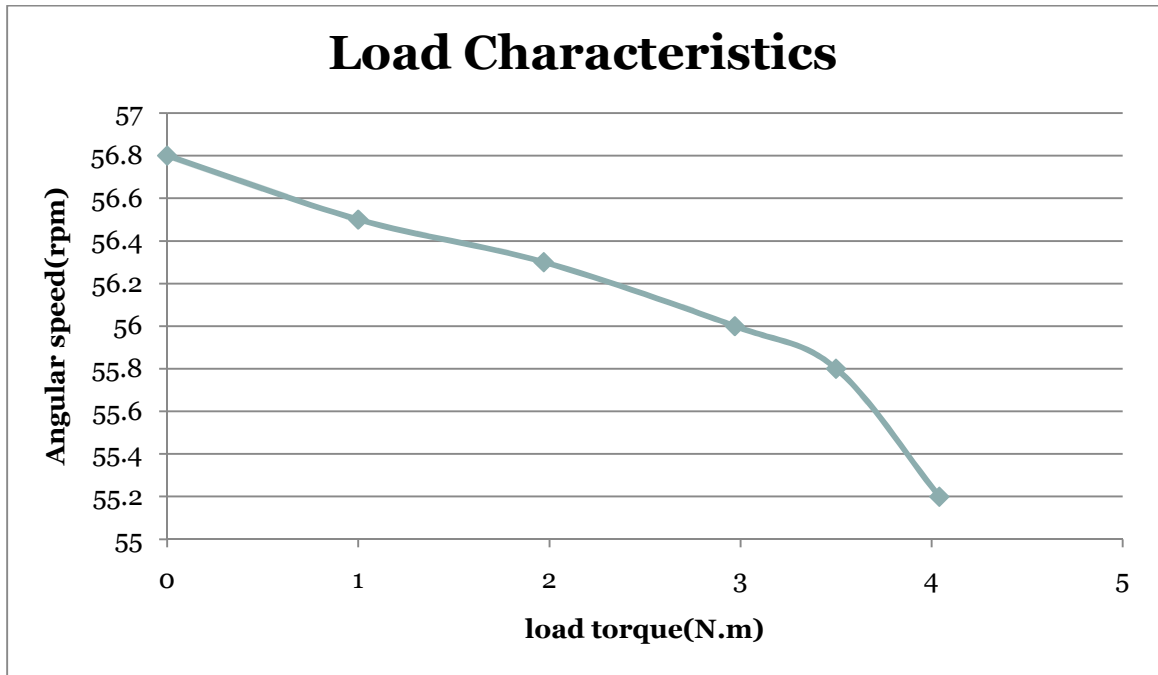
Note: (that the effect of friction is neglected because of lubrication which gives a small friction force compared to the mass)

$g$  = the gravitational acceleration (9.81N/kg)

$R_p$  = the pitch radius of the pinion attached to the motor (5.5cm)

**Table (3-2):** torque and speed values of the motor

	0 N.m	1N.m	1.97N.m	2.97N.m	3.5N.m	4.04N.m
	56.8rpm	56.5rpm	56.3rpm	56rpm	55.8rpm	55.2rpm



**Fig (3.5):** load characteristics curve of the motor

The lifting speed of the yoke ( $V_y$ ) is described in equation (3.2), N is multiplied by  $\frac{2\pi}{60}$  to convert the angular speed from rpm to rad/s.

$$V_y = R_p N \left( \frac{2\pi}{60} \right) \quad (3.2)$$

$$V_y = 0.055(56) \left( \frac{2\pi}{60} \right) = 0.32 \text{ m/s}$$

According to equation (3.2) the speed of the yoke when lifting the 2.5kg rammer is  $V_y = 0.32 \text{ M/s}$ , therefore the time required to lift the yoke to a height of 300mm is calculated as follows:-

$$t = \frac{H}{V_y} = \frac{0.3}{0.32} = 0.93 \text{ s}$$

- **The Cage**

Rammer is contained in the cage (2) shown in Figure (3.6) which has two main roles in the compaction operation, the first is to guide the rammer through falling down, the second is to change the direction of the rammer to the outer or inner diameter.



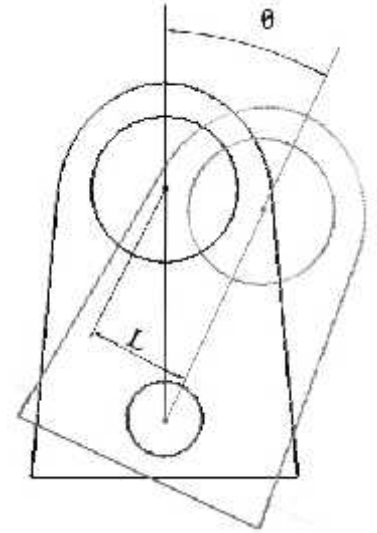
**Fig (3.6):** Photo of the machine's cage

The motion of the cage is caused by two linear actuators shown in Figure (3.7) that operates under AC 220V at 50Hz, for proctor test the inner diameter is used where 25 blows is delivered to mould, but when compacting for CBR the inner and outer diameter is used such that 17 blows are delivered to the outer diameter and 8 blows are delivered to the outer diameter. When the machine is set to CBR a mechanical interlock (16) is automatically activated to prevent a built up of soil around the sample. Each blow is detected by a limit switch (18) which gives pulses to the counter to increment the number of blows. For safety of the machine and users the guard (6) under the cage should be used to close the cage and prevent the rammer from falling down when changing the rammer or any other maintenance.



**Fig (3.7):** Electrical linear actuators

The angle  $\theta$  shown in figure (3.8) which is the angular displacement of the cage can be calculated using the equation (3.3) depending on the length  $L$  where it is the variable distance the rammer must travel to reach the end of the mould without pass on its edge so the rammer won't fall on the edge and break it.  $L$  is measured from the center of the rammer at the origin to the center of the rammer at the final position.



**Fig (3.8):** Sketch for movement of the cage

$$\theta = \cos^{-1} \left( \frac{L^2 - R^2}{2R^2} \right) \quad (3.3)$$

$$L = R_m - r \quad (3.4)$$

Where  $R$  = the distance between the center of joint to the center of rammer.  
 $R_m$  = the radius of the mould.  
 $r$  = the radius of the rammer.

- **The Control Panel**

For the user to control the machine a control panel with three switches and a counter is attached to the device, so the user can start/stop the machine, determine the test and number of blows as seen in Figure (3.9).



**Fig (3.9):** Control panel

Blows number can be set on the counter (3), but the counter does not count down but stays at number set for next operation. The green and red switches (4) are to start or stop the operation of the machine, the switch in middle (8) is to determine the test whether proctor (to the left) or CBR (to the right)

# CHAPTER 4

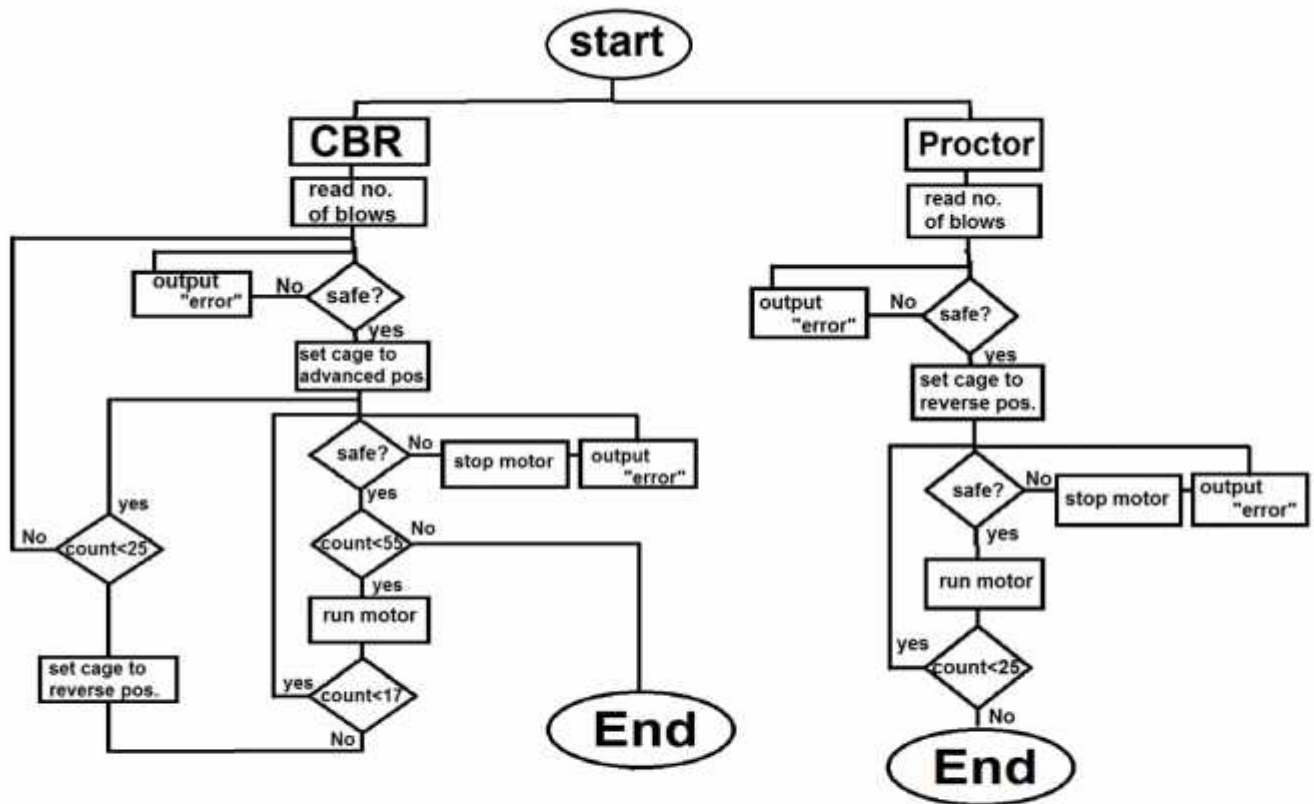
## System control Architecture for the Developed Compaction Machine

### 4.1 Introduction

In this chapter the block diagram for all components of the developed compaction machine will be discussed, to integrate all components with each other, this will include the mechanical device that will be fixed, the sensors, the interface device, and the controlling unit, as discussed in the previous chapter the user's interface was so as well as the controlling unit which relies completely on contactors that makes it hard to troubleshoot for any malfunction and takes a large space, never forget that the current controlling unit is malfunctioned.

### 4.2 Flow Chart and Block Diagram of the Machine

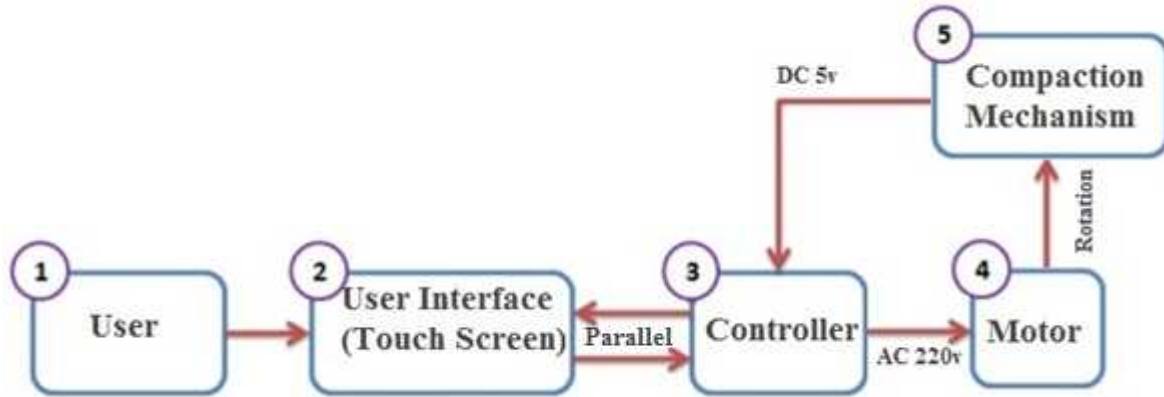
The flow chart in Figure (4.1) shows in general the steps and principles of operation of the developed compaction machine. As the machine start the two options are available the CBR and Proctor, when choosing CBR the number of blows will be defined, then safety measures will be checked such as: door is closed and the correct mould is inserted, if the safety measures are all done the system will continue, otherwise an output error will be displayed on the screen to notify for the error in the operation, whenever all are checked the system will set the cage to the advanced position using the linear actuator described in chapter 3, after that another check for safety measure will be performed, if the safety is insured the motor will run after checking the counter so it is less than the specified number of blows, the activation of the motor lifts the rammer and release it at the specified height.



**Fig (4.1):** General flow chart for operation of the developed compaction machine

Whenever the rammer is dropped a limit switch will be set to increment the counter in the controller, followed by the system will check whether the counter have reached the defined number of blows for outer diameter which is 17 for CBR, as long as the motor is running a safety check will be performed so any mistake will interrupt the operation of the motor, where at the same time another check is performed to make sure that the counter is less than 17. The motor will keep going until reaching 17, which causes the system to stop the motor for a short period of time until the cage is set to the reverse position at the inner diameter, then proceeds with the same steps, until 8 blows are done on the inner diameter, after that the system will go back to the first step to perform another 25 blows as described before to reach 55 blows.

Now for choosing Proctor the operation is simpler than in CBR where the first step is reading the number of blows then checking for safety, if everything were set the system will proceed to set the cage at the inner diameter and check again for safety, followed by the motor will start running and so the counter will be increased with the number of blows until reaching 25 blows, then the system will shut down.



**Fig (4.2):** Block diagram of the developed compaction machine

As mentioned in the block diagram in figure (4.2), the user who is the decision maker contact directly with the user interface through the touch screen to give the orders to the machine where there is a serial communication between the touch screen and the controller where the controller starts reading the commands of the user and then it starts the process. The controller must give signals to activate the motor which is the part that drive the compaction mechanism, other signals are sent to the linear actuators that rotate the cage. Attached to the mechanical part of the machine are limit switches that detect each drop of the rammer, and others that detect the status of the door, the mould, and the cage.

### 4.3 System Control Architecture

In the System control Architecture the parts of the developed machine will be explained and described each part by itself.

#### 4.3.1 User Interface

For the user interface two options are available to do, that are related to type of microcontroller that will be used in the project, the first choice is the Schneider Electric touch screen that can establish a serial communication with Shneider Electric PLC that is relatively too expensive, the other option is the use of a tablet with Google developed software Android which can make serial communication with IOIO microcontroller.



- **Schneider Electric Touch screen Panel**



**Fig (4.3):** Schneider Magelis touch screen panel

The Schneider touch screen is a very good choice except that it is much more than needed in the project. It provides two monochrome versions that are available with three-color backlighting for high visibility and status indication. The display features 16 levels of grey, numerous character fonts, and the Vijeo Designer extensive object library, which all help to provide best in class, high-quality graphic animations. Vijeo Designer supports 38 alphabets and enables the HMI to manage up to 15 languages simultaneously. And Ease of maintenance:-

- 1-No batteries required, backlit by LEDs
- 2-Removable power connector
- 3-USB memory stick for application downloads

- **Viewsonic Tablet**

A tablet computer, or simply tablet, is a one-piece mobile computer. Devices typically offer a touch screen, with finger (or stylus) gestures acting as the primary means of control, though often supplemented by the use of one or more physical context sensitive buttons or the input from one or more accelerometers; an on-screen, hideable virtual keyboard is generally offered as the principal means of data input. Available in a variety of sizes, tablets customarily offer a screen diagonal greater than 7 inches (18 cm), differentiating themselves through size from functionally similar smart phones or personal.



**Fig (4.4):** Viewsonic tablet

**Specifications:-**

Processor & Memory:-

- 1GHz NVIDIA Tegra 2 - Dual-core ARM Cortex-A9 CPU
- 512MB DDR

Storage:-

- 16GB internal memory (expandable up to 32GB via SD Card)

Operating System:-



- Google Android 2.2

Screen:-

- 4.5" TFT-LCD with LED driver system and 1024x600 resolution

Battery & Power

- 3650mAh Li-ion polymer battery
- 12V Power adapter

Networking & Communications:-

- Bluetooth: Bluetooth 2.1 + EDR
- USB 2.0 (Slave)
- Mini USB 2.0 (Host)
- Micro SD
- DC Input (Power)

- **2.8" TFT Touch Shield**

2.8" TFT Touch Shield is an Arduino UNO/ Mega compatible multicolored TFT display with a 4-wire resistive touch-screen. It is available in an Arduino shield compatible pin out for attachment. The TFT driver is based on ST7781R with 8bit data and 4bit control interface. The 2.8" TFT Touch Shield will be used in the project for its easiness in communication with the Arduino mega.



**Fig (4.5):** 2.8" TFT Touch Shield

**Table 4-2:** Specification of 2.8" TFT Touch Shield

Item	Min	Typical	Max	Unit
<b>Voltage</b>	4.5	5	5.5	VDC
<b>Current</b>	/	/	250	mA
<b>LCD Panel Size</b>		2.8		inch
<b>View angle</b>		60~120		Deg
<b>Resolution</b>		320x240		/
<b>LCD color</b>		65k		/
<b>Backlight Type</b>		LED		/
<b>LCD driver IC</b>		ST7781R		/
<b>Interface Type</b>		10±2		g
<b>Interface Type</b>		Parallel port ( 8bit Data + 4bit Control )		/
<b>Touch Screen</b>		4-Wire resistive touch screen		/
<b>Active area</b>		43.2*57.3		mm
<b>ESD contact discharge</b>		±4		KV
<b>ESD air discharge</b>		±8		KV
<b>Dimension</b>		72.5x54.7x18		mm
<b>Weight</b>		24±2		g

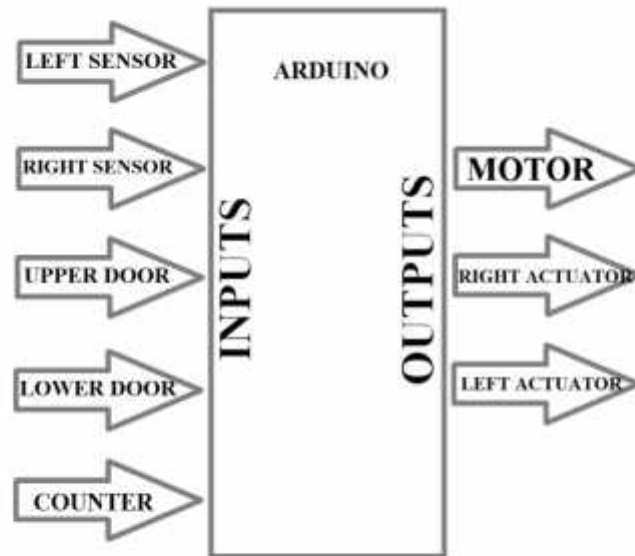
# CHAPTER 5

## Software Design

### 5.1 Introduction

To complete the desired procedures of the machine rules must be done to lead the operation of compaction; these rules can be implemented using a micro controller, and to make it work according to the rules; the rules must be translated to a language that can be understood by the used controller which is in this case Arduino Mega 2560, the Arduino Mega is programmed in C language with the Arduino software that is open source, which means a development model that promotes universal access via free license.

In any system there must be outputs and inputs to interface with the mechanical structure, in the project the outputs and inputs are illustrated in (figure 5.1).



**Fig (5.1):** Input and outputs of the system

According to the figure above these are the inputs and outputs of the system:-

- **Inputs**

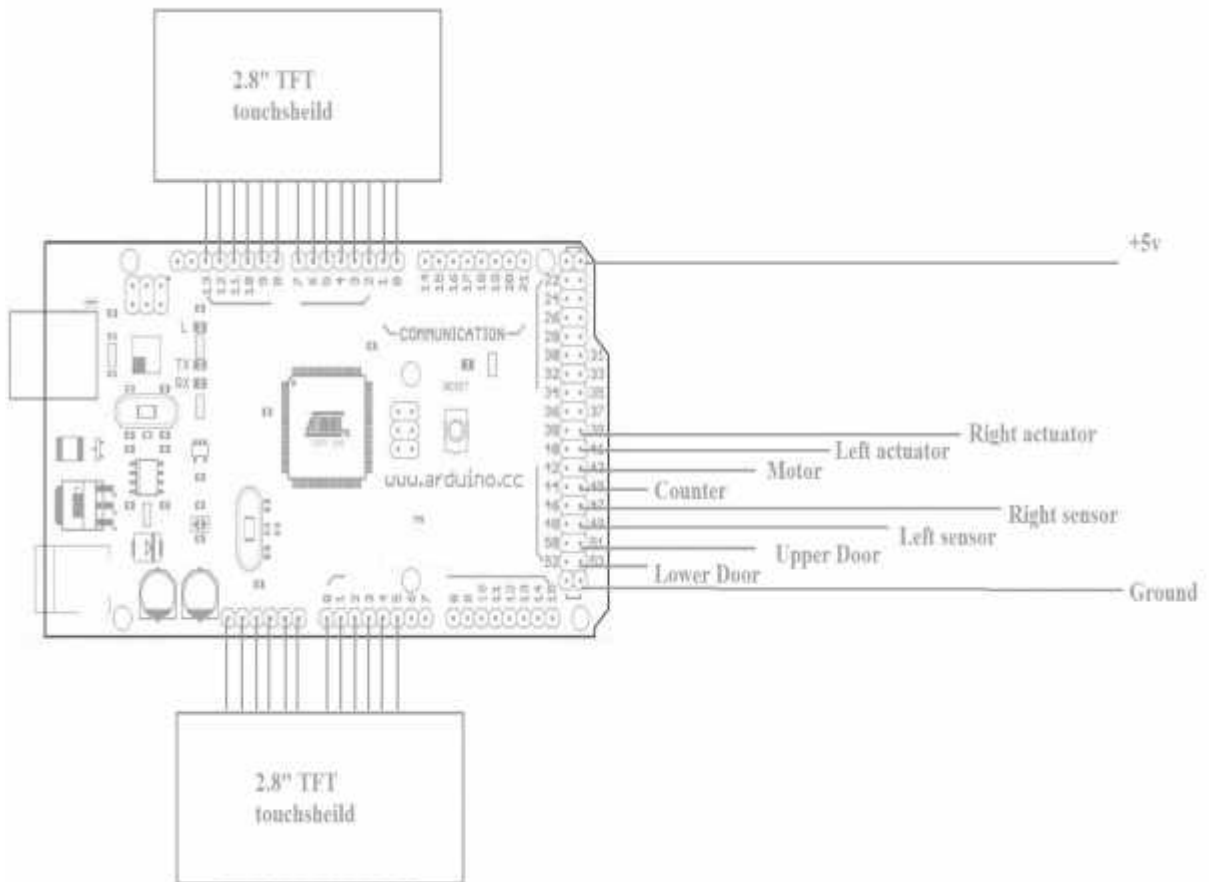
- Left sensor: that is the limit switch that detects the motion of the cage to the left side.
- Right sensor: that is the limit switch that detects the motion of the cage to the Right side.
- Upper door sensor: that is also a limit switch that detects if the upper door is opened or closed.
- Lower door sensor: detects if the lower door where the compaction occurs is opened or closed.
- Counter: is the part that detects the number of strikes.

- **Outputs**

- Motor: is electric machine that drives the machine.
- Right Actuator: is the actuator that moves the cage to the right.
- Left Actuator: is the actuator that moves the cage to the left.

## 5.2 Pin mapping

To interface with the outputs as denoted the address of each part must be known and named in the software, the following figure (figure5.2) shows the wiring of the peripherals and their pin number.



**Fig (5.2):** Pin mapping of peripherals to the Arduino

As shown in figure 5.2, following is the pin mapping of the machine's peripherals as input and outputs, and figure 5.3 shows the definition in the C language.

- **Inputs**

1. Left sensor (Pin 47)
2. Right sensor (Pin 49)
3. Upper door sensor (Pin 51)
4. Lower door sensor (Pin 53)
5. Counter (Pin 45)

- **Outputs**

1. Motor (Pin 43)
2. Right Actuator (Pin 39)
3. Left Actuator (Pin 41)

```

const int Counter      = 45;
const int upperdoor    = 51;
const int lowerdoor    = 53;
const int rightS       = 49;
const int leftS        = 47;

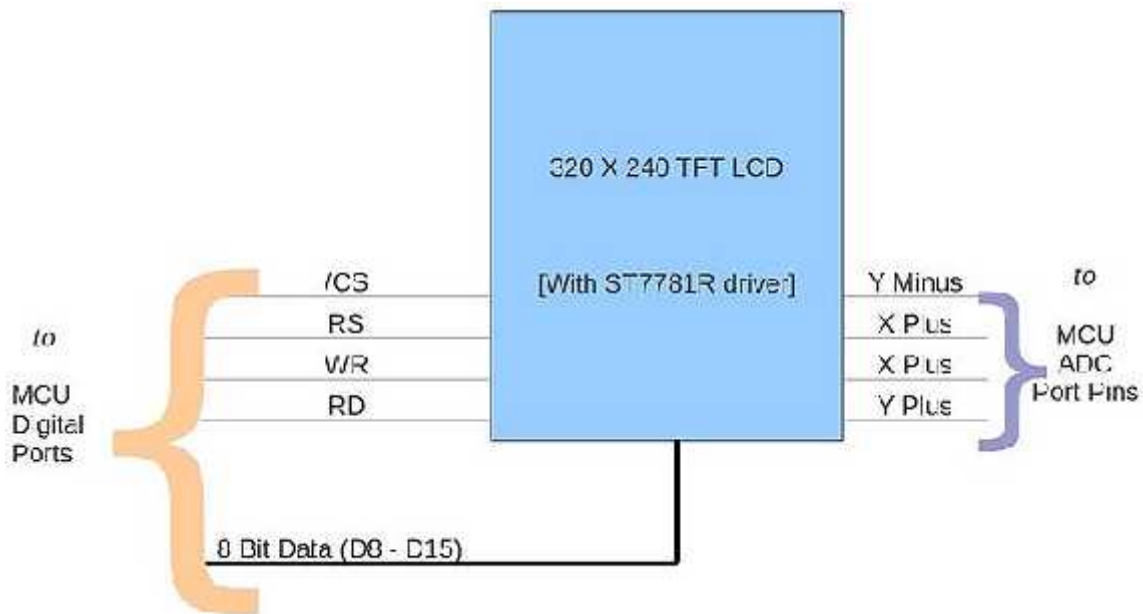
const int motor        = 43;
const int rightA       = 39;
const int leftA        = 41;

void setup()
{
  //OUTPUTS
  pinMode(motor, OUTPUT);
  pinMode(rightA, OUTPUT);
  pinMode(leftA, OUTPUT);
  //INPUTS
  pinMode(counter, INPUT);
  pinMode(rightS, INPUT);
  pinMode(leftS, INPUT);
  pinMode(upperdoor, INPUT);
  pinMode(lowerdoor, INPUT);
}
  
```

**Fig (5.3):** Definition of inputs and outputs

The program shown in figure 5.3 defines the peripherals of the machine, where at first the constant integers are defined with the number of pin and named according to the function, the constants are defined so they can be used with the Arduino library that deals with every pin according to its number. The second part is where each pin is defined as input or output using the function pinMode (pin number, output or input), all the peripherals are defined here except the counter that will be discussed in the next chapter.

- **Touch screen**



**Fig (5.4):** Touch shield pins usage on Arduino

**D0** - Unused.

**D1** - Unused.

**D2** - LCD data bit 8.

**D3** - LCD data bit 9.

**D4** - LCD data bit 10.

**D5** - LCD data bit 11.

**D6** - LCD data bit 12.

**D7** - LCD data bit 13.

**D8** - LCD data bit 14.

**D9** - LCD data bit 15.

**D10** - LCD CS pin, active low.

**D11** - LCD RS pin.

**D12** - LCD WR pin.

**D13** - LCD RD pin.

**D(A0)** - Touch Screen Y-.

**(A1)** - Touch Screen X-.

**(A2)** - Touch Screen Y+.

**(A3)** - Touch Screen X+.

**(A4)** - Unused.

**(A5)** - Unused.



### 5.3 Counter

To detect every pulse the counter's switch makes without any effect of the delay in the program timer 5 is used, this timer is built in the Arduino which is programmed and configured according to the datasheet of the micro-controller used in the Arduino board that is Atmega2560, there are several registers to configure the timer as desired, but for using it to count pulses from an external source 2 registers are needed (TCCR5B) and (TCNT5), the following table (table 5.1) is taken from the datasheet, and it shows the bits that must be defined.

**Table (5.1):** Clock selection bit description in register (TCCR5B)

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{ICL}/1$ (No prescaling)
0	1	0	$clk_{ICL}/8$ (From prescaler)
0	1	1	$clk_{ICL}/64$ (From prescaler)
1	0	0	$clk_{ICL}/256$ (From prescaler)
1	0	1	$clk_{ICL}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

The pulses in the machine are taken from the limit switch attached to the machine; therefore it is an external source so the bit must be configured as either (1, 1, 0) or (1, 1, 1), the choice is left to whether the counter must increment on the falling or rising edge, in this case it doesn't matter, so the (1, 1, 1) on rising edge is chosen, following is how to configure the bit in software using bitSet(register , bit) or bit Clear(register , bit) functions :-

```
bitSet (TCCR5B , CS52);  
bitSet (TCCR5B , CS51);  
bitSet (TCCR5B , CS50);
```

The value of the counter is held in register (TCNT5), which can be used directly in the program; the next section will discuss how it was used in the program for the CBR and Proctor choices.

## 5.4 CBR and Proctor

In this section the selection of the CBR or Proctor test will be discussed, as well as how each function works, in this program 2 functions from Arduino library were used, `digitalWrite` (pin number, high or low) and `digitalRead` (pin number).

```
void loop()
{
  TCNT5 = 0;           // reset counter
  digitalWrite(motor,LOW);
  digitalWrite(rightA,LOW); // to make sure that the motors are turned off before
  digitalWrite(leftA,LOW); // selection of the test
  while(finish==0){
    if ( digitalRead(upperdoor) && digitalRead(lowerdoor) ) // check if doors are closed
      {
        if (test==proctor)           // choose the test
          {
            .....
          }
        if(test==CBR)
          {
            .....
          }
      }
    else { digitalWrite(motor,LOW);} // if a door was opened turn off the motor
  }
}
```

In the previous program the tests are chosen, but first the motors were turned off and the counter was reset immediately when entering the void loop, after that an if-statement was initialized to make sure that the doors are closed, and then continue to choose the test. Next each function will be explained the CBR and Proctor.

- **Proctor**

After choosing the proctor test on the touch screen as discussed earlier the following code will begin functioning.

```
if (TCNT1<=25)                \\ if counter is less than 25 compact
{
    digitalWrite(rightA,LOW); \\ move the cage to the left
    digitalWrite(leftA, HIGH);
    if (digitalRead(leftS))    \\ if cage arrived to the left start compacting
    {
        digitalWrite (motor,HIGH);    \\ turn on motor
    }
    else
    {
        digitalWrite(motor,LOW);      \\ turn off all motors
        digitalWrite(rightA,LOW);
        digitalWrite(leftA,LOW);
        finish=1;                    \\ the compaction is done
    }
}
```

- **CBR**

```

if(TCNT1<=40) \\ if the counter is less than 40 then compaction must be done on
{
    \\ outer diameter.
    digitalWrite(leftA,LOW);
    digitalWrite(rightA,HIGH); \\ set cage to right
    if(digitalRead(rightS)) \\ if cage arrived start compaction
    {
        digitalWrite(motor,HIGH); \\ turn on motor
    }
    Else \\ if the cage didn't arrive turn off motor
    digitalWrite(motor,LOW);
}

```

```

if (TCNT1>40) \\ if the counts are more than 40 start compacting the inner diameter
{
    delay(400); \\ delay for less than 0.5s till the rammer doesn't contact the sample
    digitalWrite(motor,LOW); \\turn off the motor
    digitalWrite(rightA,LOW); \\ set cage to the left
    digitalWrite(leftA,HIGH);
    if (digitalRead(innerS))
    {
        If (TCNT1<=55)
        {
            digitalWrite(motor,HIGH);
        }
        else
        {
            delay(400); \\wait till the rammer detach from the sample
            digitalWrite(motor,LOW); \\ turn off the motors
            digitalWrite(outer,LOW);
        }
    }
}

```

```

        digitalWrite(inner,LOW);
        digitalWrite(error,HIGH);
        finish=1;           \\ the compaction is done for CBR
    }
}
else                       \\ if the door is open turn off the motor
digitalWrite(motor,LOW);
}

```

The above program sets the steps of compaction for the CBR test after choosing the CBR test if the number of counts is less than 40 then the machine will start compacting on the outer diameter of the mold until reaching 40 strikes, after that the motor is stopped and the cage is moved to inner diameter to compact for another 15 strikes, whenever the compaction is done the motors will stop and the finish variable is set to high to return to the menu.

## 5.5 TFT Touch shield

The TFT library provides the following Application Programming Interfaces (API). The library makes use of direct access to PORT registers instead of Arduino APIs. This is to increase the speed of communication between MCU and TFT. The library supports Arduino Mega, here are the functions used in the program for the TFT screen:-

- **paintScreenBlack(void)**  
The TFT RAM can only be initialized in display-off mode, so this function only for clearing screen with black colour.
- **setXY(unsigned int poX, unsigned int poY)**  
Sets the cursor position to (poX,poY). This function is internally used by other graphics APIs.

- **setPixel(unsigned int poX, unsigned int poY, unsigned int color)**  
Sets the (poX,poY) pixel to color color. This function is internally used by other graphics APIs.

## Lines

- **drawLine(unsigned int x0, unsigned int y0, unsigned int x1, unsigned int y1, unsigned int color)**  
Draws a line from pixel (x0,y0) to pixel (x1,y1) with color color.
- **drawVerticalLine(unsigned int poX, unsigned int poY, unsigned int length, unsigned int color)**  
Draws a Horizontal Line of length length with color color starting from pixel (poX,poY).
- **drawHorizontalLine(unsigned int poX, unsigned int poY, unsigned int length, unsigned int color)**  
Draws a Vertical Line of length length with color **color** starting from pixel (poX,poY).

## Rectangle

- **drawRectangle(unsigned int poX, unsigned int poY, unsigned int length, unsigned int width, unsigned int color)**  
Draws a rectangle starting from (poX,poY) of length length, width width and color **color**.
- **fillRectangle(unsigned int poX, unsigned int poY, unsigned int length, unsigned int width, unsigned int color)**  
Draws a filled rectangle starting from pixel (poX,poY) of length length, width width and color **color**.

## Circle

- **drawCircle(int poX, int poY, int r, unsigned int color)**  
Draws a circle at (poX,poY) of radius **radius** and color **color**.
- **fillCircle(int poX, int poY, int r, unsigned int color)**  
Draws a filled circle at (poX,poY) of radius **radius** and color **color**.

## Text

- **drawChar(unsigned char ascii,unsigned int poX, unsigned int poY,unsigned int size, unsigned int fgcolor)**

Draws a character starting from (poX,poY) using inbuilt font of size **size** and with color fgcolor. This function is used by drawString() function.

- **drawString(char \*string,unsigned int poX, unsigned int poY,unsigned int size,unsigned int fgcolor)**

Draws a string of text starting from (poX,poY) using inbuilt font of size **size** and with color fgcolor.

TFT Touch Shield uses the Adafruit Touch Screen Library. A 4-wire resistive touch screen provides two voltage divider each for X and Y axis. By applying proper voltages for each axis and scanning the ADC values the position of the touch can be detected. These values are always prone to noise; hence a digital filter is used. To use the TouchScreen Library first a TouchScreen object must be created by

```
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);
```

Where XP, YP, XM and YM are ADC port pins connected to XPlus, YPlus, XMinus and YMinus pins of Touch Screen. 300 is the resistance across X plates. After creating the touchscreen object the Raw ADC value must be read using:

```
Point p = ts.getPoint();
```

The Raw ADC value has to be converted to Pixel Coordinates. This is done with map function. This mapping changes for v0.9 and v1.0.

```
p.x = map(p.x, TS_MINX, TS_MAXX, 240, 0);
```

```
p.y = map(p.y, TS_MINY, TS_MAXY, 320, 0);
```

# CHAPTER 6

# Design of Electronic Circuits

## 6.1 Introduction

An electronic circuit is composed of individual electronic components, such as resistors, transistors, capacitors, inductors and diodes, connected by conductive wires or traces through which electric current can flow. The combination of components and wires allows various simple and complex operations to be performed: signals can be amplified, computations can be performed, and data can be moved from one place to another. Circuits in the project are constructed of discrete components connected by individual pieces of wire. In this chapter the electrical parts and circuits are discussed, to explain the necessary resistance needed for the circuit protection and explaining the circuits of each function such as counter, relays, limit switches, and power supply, these functions are divided into two groups the inputs and outputs as discussed in the previous chapter.

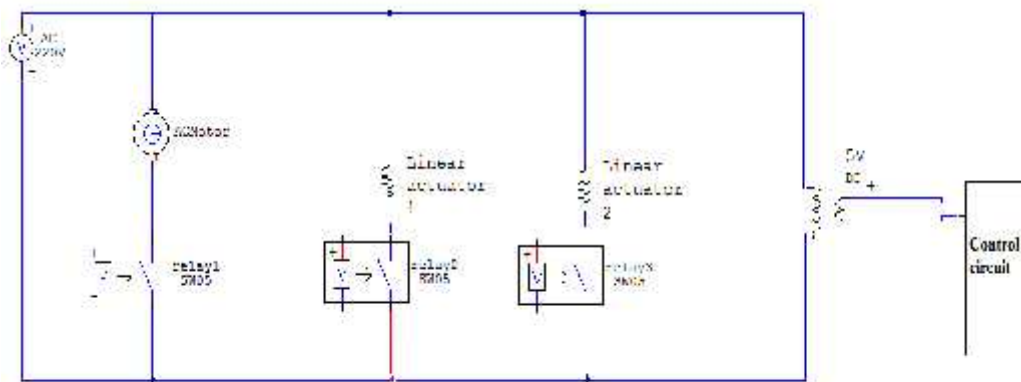


## 6.2 Power Supply

A power supply is a device that supplies electric power to an electrical load, a regulated power supply is one that controls the output voltage or current to a specific value; the controlled value is held nearly constant despite variations in either load current or the voltage supplied by the power supply's energy source. Two types of power sources are needed to power up the project, 220v AC to power up the electrical motor and the electrical linear actuators, the other type is 5v to power up the control circuits such as the micro controller.

- **220V AC power Supply**

There are three parts that need to be powered up with a 220V AC power source: the motor and the two linear actuators, figure 6.1 shows the connection circuit of these parts.



**Fig (6.1):** 220v AC powering circuit

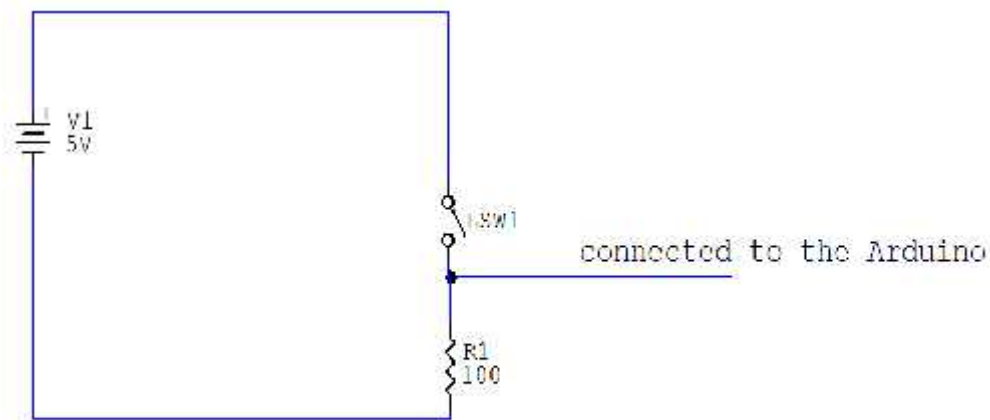
As the figure above shows the AC parts are connected directly to the power source and are controlled using relays that are connected to the Arduino which is powered with a 5v DC source and will be discussed later in section 6.4, an adapter is connected to provide the system with DC 5v source as shown in the circuit.

### 6.3 Inputs

This section discusses the inputs of the project that are four limit switches, and a counter.

- **Limit switch**

The limit switch is activated by physical contact, and must give a value 0.3v if opened and 5v if closed to do so the following circuit is derived so the Arduino can receive a logic order of 0 or 1.



**Fig (6.2):** limit switch connection

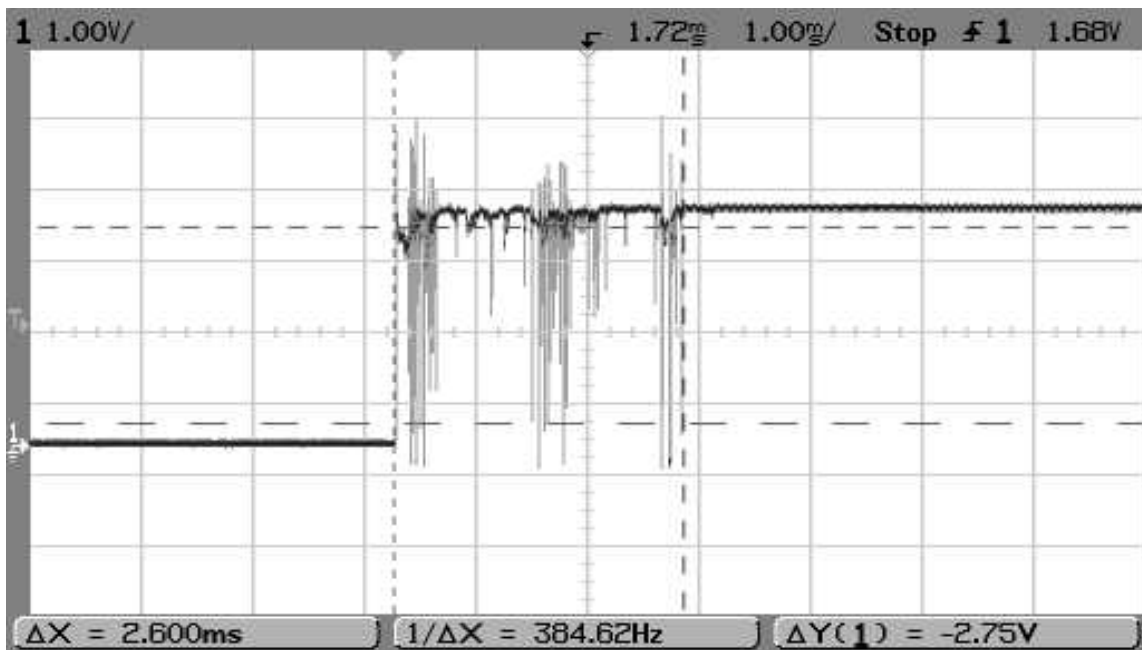
The resistor R1 is used to protect the micro-controller where the maximum current that can pass through the Arduino is 200mA; the following equation shows the required resistance.

$$R = \frac{V}{I} = \frac{5v}{0.2A} = 25\Omega$$

The minimum resistance is 25 to give 200mA, in the project a resistance of 100 is used where the current is 50mA which still can activate the Arduino and less risk can be achieved to guarantee a more durability for the Arduino.

- **Counter**

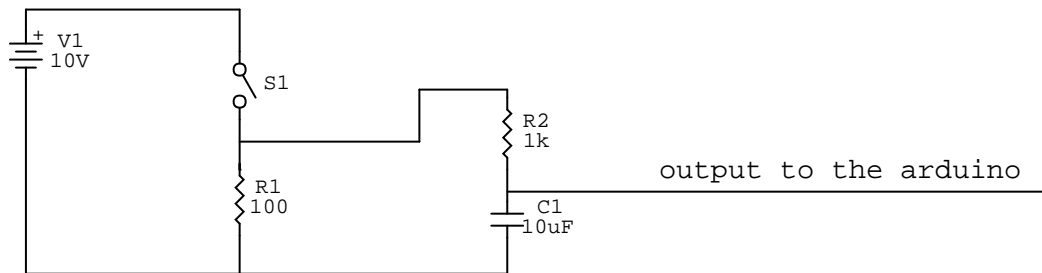
In digital logic and computing, a counter is a device which stores the number of times a particular event or process has occurred, in this case the counter is needed to store the number of strikes on the soil sample, to do so a limit switch is used so whenever the rammer reaches upward the switch is triggered, and the rising edge is detected to increment the counter as discussed in the previous chapter. The electronic circuit used in the counter is the same as the limit switch circuit in figure 6.2, but one problem is caused when using this circuit which is bouncing phenomenon.



**Fig (6.3):** Snapshot of switch bounce on an oscilloscope

The Bouncing phenomenon is a common problem with mechanical switches and relays. Switch and relay contacts are usually made of springy metals, when the contacts strike together their momentum and elasticity act together to cause them to bounce apart one or more times before making steady contact. The result is a rapidly pulsed electric current instead of a clean transition from zero to full current as shown in figure 6.3. The effect is usually unimportant in power circuits, but causes problems in logic circuits that respond fast enough to misinterpret the on-off pulses as a data stream, in counters the rapid pulses caused by bouncing are detected as

strikes, so the number of counts will be wrong and much more than the actual counts. Therefore the effects of bouncing can be eliminated by use of low-pass filter, which is a filter that passes low frequency signals and attenuates (reduces the amplitude of) signals with frequencies higher than the cutoff frequency. In the project a simple low-pass filter is used with RC circuit as shown in figure 6.4.



**Fig (6.4):** The counter circuit with low-pass filter

The frequency of operation in the machine is almost 1Hz therefore any frequency higher than 1Hz can be consider as disturbance or bouncing, but the frequency of bouncing is very high, therefore a cut off frequency  $f_c$  of 16Hz can solve the problem, following is the calculation of the required resistance when using 10 $\mu$ F capacitor:-

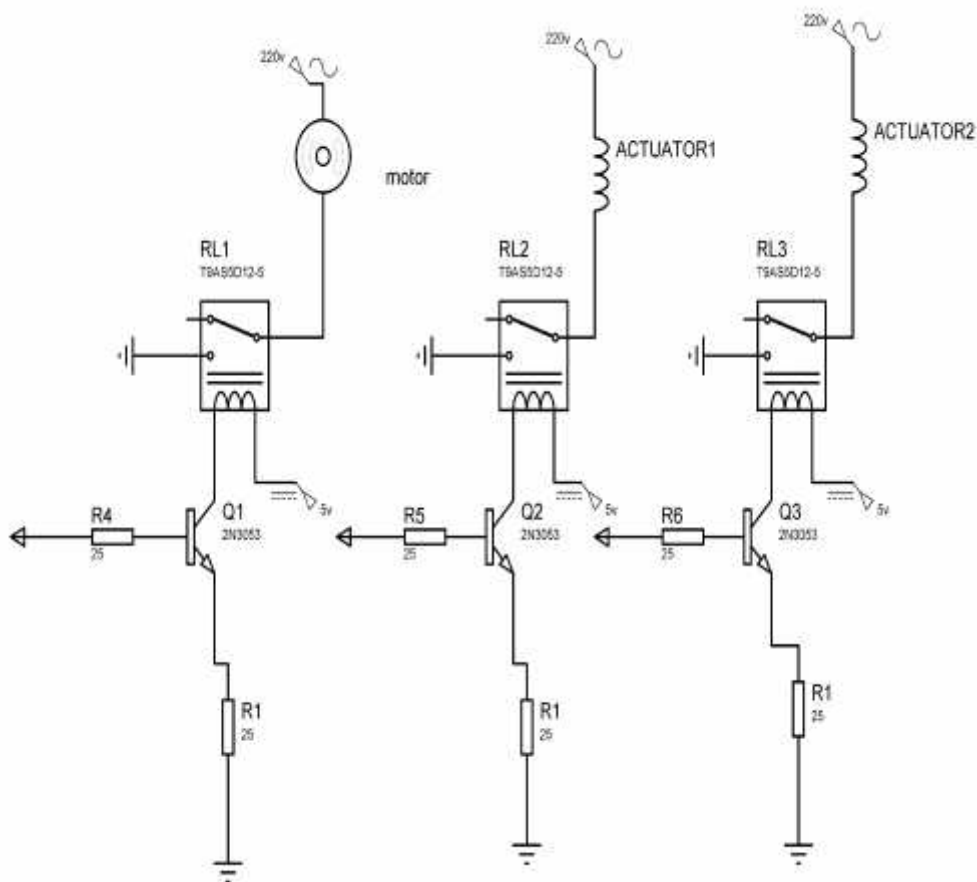
$$R = \frac{1}{2\pi f_c C} = \frac{1}{2 \times 16 \times \pi \times (10 \times 10^{-6})} = 996\Omega$$

Therefore a resistor of 1k is used.

## 6.4 Outputs

The control and activation of the outputs of the machine are discussed in this section, which are the electrical circular motor and two linear actuators, as mentioned before both the motor and actuator operate under a 220v AC voltage and are considered as electric machines. Electric machines are devices that convert electrical energy into mechanical energy, the circular motor gives a circular motion, where linear actuators are actuators that create motion in a straight line, in contrast to the circular motor.

The output voltage of the Arduino is DC 5v which is not enough to power up the actuators; the following circuit is required (figure 6.5).



**Fig (6.5):** Control circuit of the outputs

As shown in the previous figure the output taken from the Arduino is connected to a transistor to a relay and then to the electric machine, the relay can be activated with 5v at a minimum current of 127mA and it is used as a switch to turn on the electric machine, but the problem is that the output power from the Arduino is very low and cannot provide 127mA according to the datasheet, so that is why a transistor is used as switch to activate the relay, a transistor can be trigger at a low current (50mA), so whenever the Arduino gives order to turn on the machine, sequential operation is occurred where the transistor is activated to allow the current to flow through the relays' coil so the relay can connect the electric machine to its power source, a 25 resistor is used according to the following calculation:-

The needed current to activate the relay is 127mA

$$R = \frac{V}{I} = \frac{5}{0.127} = 50\Omega$$

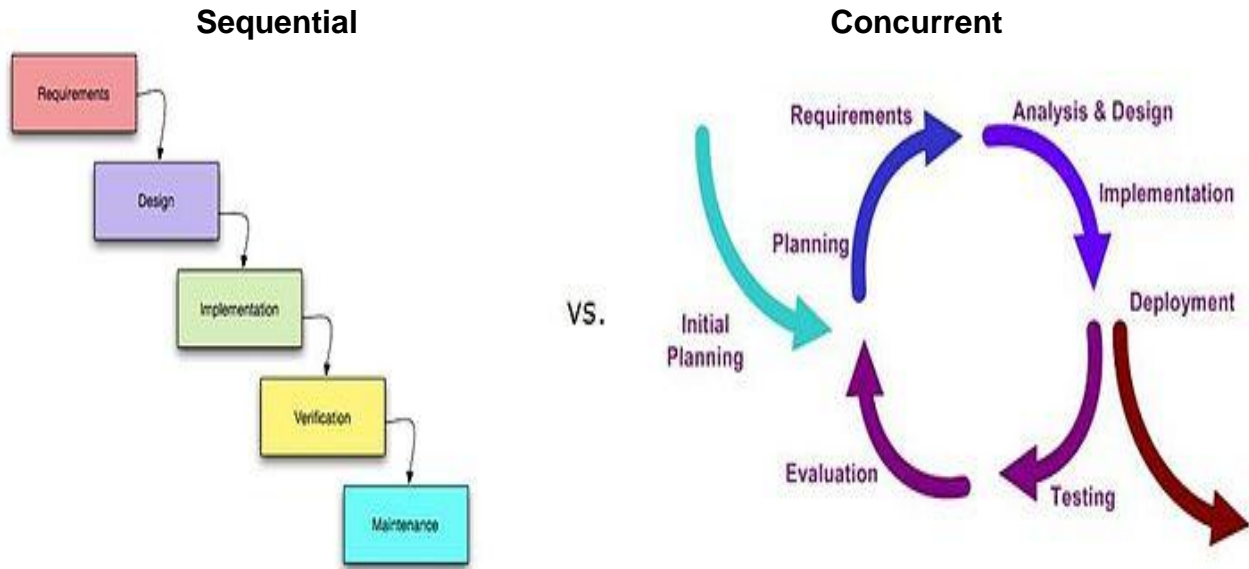
The minimum resistance needed so the relay can be activated is 50 according to the previous calculation, but in practice the 50 didn't allow a 127mA to pass and that's way a 25 resistance was used.

# CHAPTER 7

# Building and Testing

## 7.1 Introduction

In this chapter building and testing process of the machine will be discussed, in which there are two methods to accomplish the process of designing: sequential and concurrent design, figure 6.1 gives a hint about the sequential and concurrent design, as shown in the figure the Sequential Design is the term used to describe the method of designing in a linear format. The different steps are done one after another, with all attention and resources focused on that one task. After it is completed it is left alone and everything is concentrated on the next task. In concurrent engineering, different tasks are tackled at the same time, and not necessarily in the usual order. This means that info found out later in the process can be added to earlier parts, improving them, and also saving a lot of time. Concurrent engineering is a method by which several teams within an organization work simultaneously to develop new products and services and allows a more stream lined approach; in this project the two methods were used.



**Fig(7.1):**sequential design vs. concurrent design

The sequential method is used to design each part or function alone, where each function was built alone and tested by itself without the effects of other functions, and concurrent method is used in the design of each part where the electronic circuit and controlling program were done at the same time implemented, tested, and developed. after all parts are tested alone , all functions are added gradually to the system and after adding each function the system is tested to see if there is any effect.

### 7.2 limit switches

One limit switch was connected and tested on a LED and it worked, after that all limit switches were built with a software that turns on a LED for each limit switch, and all switches worked properly.

### 7.3 Counter

The electrical circuit of the counter was built first as a limit switch and the software was designed as discussed in the previous chapters, then it was implemented on the arduino, and when testing the function the bouncing problem was detected, and therefore the low pass filter was used and moderated to compensate for the problem until it worked.



## 7.4 Outputs

At first only relays were used to activate the outputs, except that after testing the relays were not able to be activated by 5v voltage from the arduino, that is due to the low current, where the minimum current to activate a relay 127mA, therefore transistors were used, because transistors can be triggered with a very low current, and so the transistors were used to activate the relays as shown in figure 6.5.

## 7.5 Assembly

After all parts were built and tested alone, the parts are assembled gradually and tested, successfully all parts worked on the prototype without connecting the functions to the machine, after testing the prototype for any problem that was expected to occur no problems were faced, and so the electrical circuit was soldered and connected to the machine, after that the function of the machine was tested and a problem was faced, that is caused by the short time when switching the cage from side to side , where both actuators are activated together for a short time and caused a very high noise also such a problem might decrease the life cycle of the actuators and electrical parts, the problem was solved by software where a delay must be done between turning off an actuator and turning on the other one, after that the machine were cleaned and prepared for the real test with soil samples, and successfully all functions worked properly and the machine was effective.

# References

- [1] American Association of State Highway and Transportation Officials (1967). Manual of Foundation Investigations, National Press Building, Washington, D.C.
- [2] Braga M. Das, Principles of Geotechnical Engineering 7th.
- [3] Cubrinovski, M., and Ishihara, K. (1999). Soils and Foundations.
- [4] Liao, S., and Whitman, R. V. (1986). "Overburden Correction Factor for SPT in Sand", Journal of Geotechnical Engineering.
- [5] Wikipedia, <http://www.wikipedia.org>.
- [6] Touch Display Research, [www.seeduino.com](http://www.seeduino.com)
- [7] Arduino, [www.arduino.com](http://www.arduino.com)
- [8] Alibaba website, [www.alibaba.com](http://www.alibaba.com)
- [9] Microchip Technology, [www.microchip.com/downloads/en/devicedoc/39632c.pdf](http://www.microchip.com/downloads/en/devicedoc/39632c.pdf).
- [10] Spark fun Electronics, <https://www.sparkfun.com/products>.

### 4.3.2 Controller

- **Microchip PIC4550 microcontroller**

The PIC4550 has relatively low price compared to other microcontrollers, but still one of the best controllers used for small application in Mechatronics field.

A microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, RAM, ROM, I/O ports, and timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O, or timer to it.

The fixed amount of on-chip ROM, RAM, and number of I/O ports in microcontrollers makes them ideal for many applications in which cost and space are critical.



**Fig (4.6):** PIC4550 microcontroller

**Table (4.2):** Specification of PIC4550

Program memory	Data memory		I/O
	SRAM	EEPROM	
32k	2048b	256b	35

Software used for programming the PIC microcontroller is Microchip Mplab, where the PIC is programmed with C language.

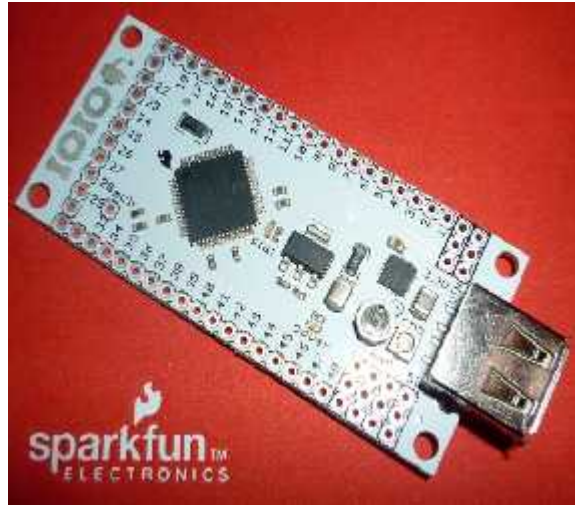
MPLAB Integrated Development Environment (IDE) is a free, integrated toolset for the development of embedded applications on Microchip's PIC and dsPIC microcontrollers.



**Fig (4.7):** Mplab IDE

- **IOIO-OTG microcontroller**

The IOIO (pronounced "yo-yo") is a board specially designed to work with Android 1.5 and later device. The board provides robust connectivity to an Android device via a USB or Bluetooth connection and is fully controllable from within an Android application using a simple and intuitive Java API - no embedded programming or external programmer will ever be needed. The IOIO board contains a single MCU that acts as a USB host and interprets commands from an Android app. In addition, the IOIO can interact with peripheral devices in the same way as most MCUs. Digital Input/output, PWM, Analog Input, I2C, SPI, and UART control can all be used with the IOIO. Code to control these interfaces is written in the same way as an Android app with the help of app-level library. In other words, one can combine the computing power, Internet/Bluetooth connectivity, touch screen, and a variety of sensors from the Android device with the ability to easily add peripheral devices to interact with the outside world. Also, using the IOIO does not require any hardware or software modifications to the Android device.



**Fig (4.8):** IOIO board

The software used in IOIO for programming is the IOIO manager; IOIO Manager is an Android application which lets you install new firmware on your IOIO board. The application is open-source and is available for download free of charge from the Android Market, using the links above. In order to understand the usage of the IOIO Manager, some key concepts need to be introduced first.



**Fig (4.9):** IOIO manager

- **Arduino Mega**



**Fig (4.10):** Arduino Mega

The chosen micro-controller is ARDUINO Uno as shown in Figure 3.6. It has many advantages that make it the preferred choice such as: user friendly, open source codes which are available for many applications and can be downloaded from the company's website, the simulation program, the available shields for many components and one of them is suitable for the wireless modules which allows the connection of modules easily. Other advantages are the Upgradability, the modularity and the last important thing is that, it does not need a dedicated programmer as it can be programmed through the same cable that is used for the PC connection. The Arduino Mega is a microcontroller board based on the ATmega1280. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 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 AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

**Table (4.3):** Specification of Arduino Mega

<b>Microcontroller</b>	ATmega1280
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limits)</b>	6-20V
<b>Digital I/O Pins</b>	54 (of which 15 provide PWM output)
<b>Analog Input Pins</b>	16
<b>DC Current per I/O Pin</b>	40 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	128 KB of which 4 KB used by bootloader
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Clock Speed</b>	16 MHz

### 4.3.3 Sensing elements

- **Limit switches**

A limit switch is a switch operated by the motion of a machine part or presence of an object. They are used for control of a machine, as safety interlocks, or to count objects passing a point.



**Fig (4.10):** Limit switch with roller

**Table (4.4):** Specification of the limit switch

Durability	15*10 <sup>6</sup> cycles
Rated insulation voltage	300v
Rated frequency	50/60Hz