Palestine Polytechnic University

College of Engineering & Technology

Electrical and Computer Engineering Department

Graduation Project

Smart Farm

Project Team

Mohammad M. Alsharawi *Shareef A. Suliman*

Tariq M. Abu Zahra

Project Supervisor

DR. Murad Abusubaih ***Eng. Nassem Iqtait***

Hebron - Palestine

Jun, 2013

جامعة بوليتكنك فلسطين

الخليل – فلسطين

كلية الهندسة و التكنولوجيا

دائرة الهندسة الكهربائية و الحاسوب

اسم المشروع

Smart Farm

أسماء الطلبة

شريف سليمان　　　　طارق ابو زهرة　　　　محمد الشعراوي

بناء على نظام كلية الهندسة و التكنولوجيا و إشراف و متابعة المشرف المباشر على المشروع وموافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية و الحاسوب وذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة أنظمة الحاسوب.

توقيع المشرف

....　　　　د. مراد ابو صبيح

توقيع اللجنة الممتحنة

....　　　　........................　　　　........................

توقيع رئيس الدائرة

....

Abstract:

There are many deferent system to control the eggs chicken farms, whether its traditional or modern. But it takes large time and huge effort and high coast. So we aim to design a new system to monitor and control the eggs chicken using the latest technology in the communication world. In order to monitor the internal conditions like temperature and humidity level. Beside to lighting an ventilation.

Controlling system will controls the feeding system and eggs collecting system and the cleaning inside the farm.

الفكره:

هناك عدة انظمه للتحكم بمزارع الدواجن فمنها ما هو تقليدي ومنها الحديث ولكنها تحتاج الى وقت وجهد وتكاليف عاليه.في مشروعنا سنعمل على تصميم نظام للمراقبة والتحكم بمزارع الدواجن باستخدام تكنولوجيا الاتصالات المعروفه ب الواي فاي. نظام المراقبه يتمثل بمراقبه الظروف الداخليه للمزرعة بما فيها الحرارة والرطوبة والإضاءة والتهوية المناسبة للمزرعة. اما نظام التحكم فيتمثل بعملية التحكم بالنظام الغذائي بشقيه الطعام والماء , ونظام تجميع البيض , وكذلك نظام التنظيف

Dedication

To Our Beloved Palestine

To our parents and families.

To all our teachers.

To all our friends.

To all our brothers and sisters.

To Palestine Polytechnic University.

May ALLAH bless you.

And for you we dedicate this project

The Work Group

Acknowledgments

We could not forget our families, who stood by us, with their support, love and care for our whole lives, they were with us with their bodies and souls, believed in us and helped us to accomplish this project.

We would like to thank our amazing teachers at Palestine Polytechnic University, to whom we would carry our gratitude our whole life. Special thanks to our **Supervisor DR.Eng. Murad Abusubaih** for his wise and amazing supervision, and **Eng. Nassem Qutait** supervisor of industrial automation.

Table of contents

List of Tables

Table of Figures

1

Chapter One

Introduction

1.1 Overview

Monitoring and controlling the "Chicken Eggs Farm" in the old traditional methods is a large dilemma; this is due to large time delay, huge effort, Non reliable and non efficient as the productivity percent of eggs is low.

We aim to design and implement the "optimal farm system", a kind of systems which really could solve the previous problems; by enabling farmers from monitoring and controlling their farms from a distance in a full automatic way, That could be achieved by using both: a wireless system based on temperature\humidity sensors and live imaging cameras for the monitoring process. And another wireless automatic control system based on conveyers, gears, fans, frequency converters and PLC units for the controlling process.

1.2 Project objectives

1. Design a monitoring system, to monitor the temperature \ humidity level using especial sensor, also to monitor the overall status of the farm using imaging cameras.
2. Design a control system, to carry out the usual functions like: cleaning, feeding, temperature \ humidity controlling and eggs collecting, using electrical and mechanical automation components like gear, conveyers, frequency converters, fans and PLC units.
3. Design interfacing systems, to connect the system entities together. Sensors, cameras, microcontrollers, and PLCs using either direct connection or Wi-Fi network using Wi-Fi transceiver.
4. Design a mobile application using the Android language as the human interface to the system.

1.3 Project Motivations

From the workers side; they will be able to do their work more efficiently, and more accurate; due to the easier and more reliable method the system provides.

From the owners (supervisors) side, they will be able to observe, supervise and control the conditions related to the farm in the most effective ways; in order to achieve maximum profit, this causing no time or effort waste.

Lastly, from our side as project designers, we see that our project will improve our nation and our future state, according to vision that the economy development is one of the most vital Pillars to do so. Moreover this project is expected to be very useful for improving the chances for establishing new jobs for both our colleagues and us.

1.4 Project idea and Approach

As previously mentioned, the main idea of this system is to monitor and control the "Chicken Eggs Farm" from a distance. The system expects to perform these functions as follows:

Monitoring:

The basic conditions that this system monitors are as the following:

1- Chicken, eggs ,food, water, and cleaning condition:-

The previous parameters can be perfectly monitored by using this system, which provides acknowledgment about the life status of the farm. Using cameras which are connected via Wi-Fi network that provides live images about what is happening at the moment in the farm to the manager side.

2- Temperature and humidity levels:-

Using this system will allow the manager to get an accurate measurement for the temperature and humidity status on the mobile, temperature \ humidity sensors covers the farm connected via Wi-Fi network will make it possible.

Controlling:

In order to get the functions cared out effectively, it is required to have a good controlling system that controls of:

1- Eggs: this system allows collecting the eggs automatically when it's ready to be collected, without the need to get inside the farm, eggs travels out the farm over special conveyors to the collecting room, were workers can arrange it in boxes.

2- Food: from its store, food will reach its own place in front of the chicken in an automatic way under the manager supervision, without human interfering.

3- Temperature and humidity: after receiving the temperature and humidity measurement you may need to change it to a cretin standard value, by using special fans for both reducing the amount of humidity and decreasing the temperature levels.

4- Cleaning: using rotated belts with its own motors shaping moving area under the chicken; in order to get rid of the rubbish in a good healthy way from the farm.

1.5 Time schedule

The time planning for the project is shown in the following table distributed on the weeks during working on the project introduction:

	Time	Activity
T1	1st SEP 2012 – 26th SEP 2012	Collecting information
T2	10th SEP 2012 – 26th SEP 2012	Preparing references
T3	27th SEP 2012 – 5th OCT 2012	Draw the general block diagram and understanding the system's protocols
T4	6th OCT 2012 – 30th OCT 2012	Analyzing the block diagram
T5	1st NOV 2012 – 15th NOV 2012	Preparing and buy the needed equipment
T6	5st NOV 2012 – 20th NOV 2012	Learning software languages for microcontroller, modems and database software
T7	1st DES 2012 – 1th DES 2012	Documentation For Supervisor
T8	1th DES 2012 – 10th DES 2012	Documentation For the Electrical Dept.
T9	1st Mar 2013 – 30 th Mar 2013	Make the hardware available
T10	15th Mar 2013 – 10 th Apr 2013	Build up the software
T11	20th Apr 2013 – 15 th May 2013	Testing the system and Writing Documentation

Table1.1: Time planning part A.

The time of the project is scheduled over 16 weeks. Table 1.2 shows how the work was scheduled over the time:

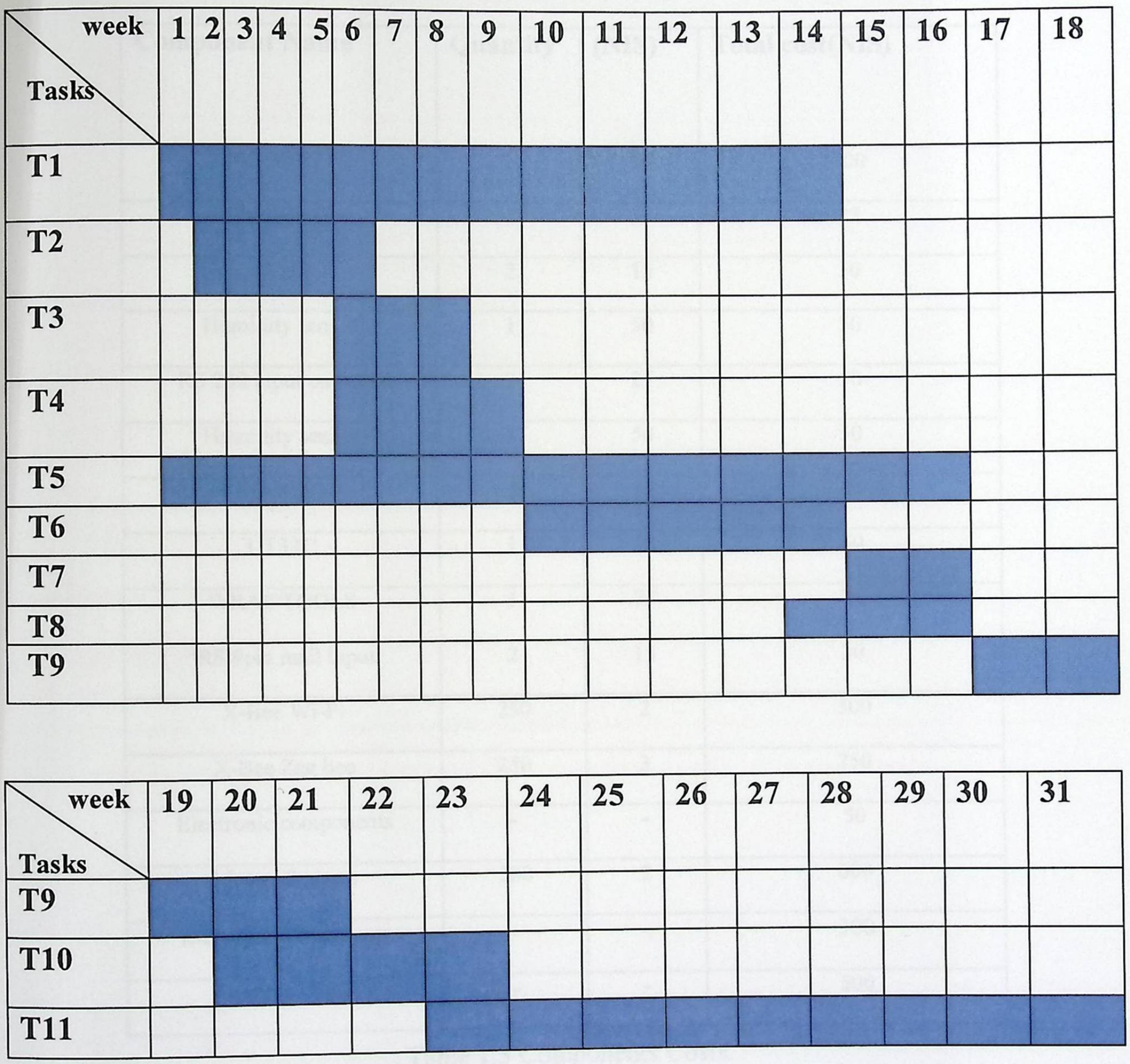


Table1.1: Time planning part B

1.6 Estimated Cost

1.6.1 Hardware Resources

The hardware components that we used in this project and its corresponding costs are shown in table1.3.

Component Name	Quantity	(NIS)	Total cost(NIS)
PIC16F877	2	50	100
PIC16F688	1	25	25
LM35	2	15	30
Humidity sensor	1	50	50
RS 232 input on board	2	25	50
Humidity sensor	1	50	50
MAX232	2	15	30
UT33C	1	50	50
WRAP TOOLS	1	75	75
RS 9pin mail input	2	15	30
X-Bee Wi-Fi	250	2	500
X-Bee Zeg bee	250	3	750
Electronic components	-	-	50
Motors 3 phase	300	2	600
Electrical components	-	-	300
Demo module	-	-	500

Table 1.3 Components Costs.

1.6.2 Human Resources

Type of resources	Cost NIS
Transportation	200
Printing supplies	300
Human resources 500 NIS	

Table 1.4 Components Costs.

1.6.3 Total costs

Resources	Cost $
Hardware resources	3320
Human resources	500
Total costs 3800 NIS	

Table 1.5 Total Costs.

2

Chapter Two

Theoretical Background

2.1 Overview

Each part in this system can be easily designed and implemented in many different ways. Sensors, microcontrollers, wireless technology, electrical/mechanical and even the interfacing technique all have a large rang of component categories and algorithms. So, in this chapter we intend to study and analyze each part of this project in intensive way showing the advantages and disadvantages for each component; in order to decide the best option that we could have.

2.2 Microcontrollers

Microcontroller is a small computer on a single integrated circuit contains central processing unit CPU, memory (RAM/ROM) for data and programs, serial and parallel I/O and timers, as shown in figure 2.1.

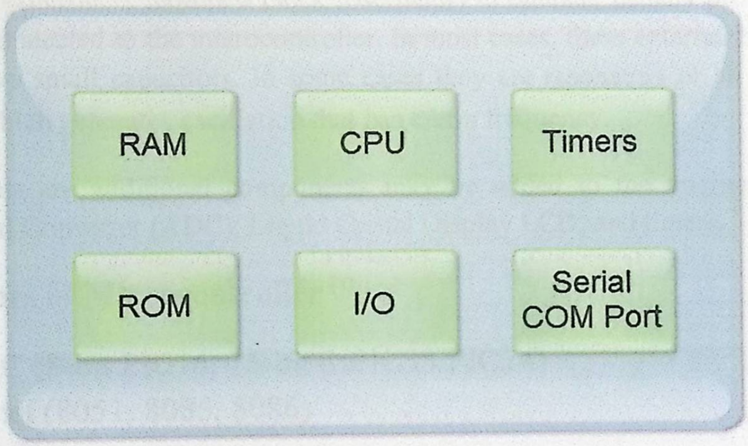


Figure 2.1: Microcontroller Component.

The microcontroller called embedded system; because it contains in single chip the CPU in addition to fixed amount of RAM, ROM, I/O ports, and timers. [1]

Microcontrollers were designed to do a very specific task, controlling a particular system automatically. Such as Power tools, mechanical device, critical medical device, remote controls, and baby toys. Were cost and size are critical in such application. Therefore microcontroller has all the support chips incorporated inside its single chip.

Central Processing Unit (CPU) is the main part of microcontroller where all of the arithmetic and logic operation are performed.

Input ports allow microcontroller to read data from the out world, while the output ports allow the microcontroller to control other systems, on other hand microcontroller also contains Universal Asynchronous Receiver/ Transmitter (UART). It is a type of "asynchronous receiver/transmitter", a computer hardware that translates data between parallel and serial forms.

Microcontrollers can be programmed using either the assembly language or using a high-level language such as BASIC, PASCAL, or C++. Although assembly language is fast in decoding and fetching processes but it has basic disadvantage that any assembly program consists of mnemonics, which makes learning and maintaining a program written using the assembly difficult. Also, microcontrollers manufactured by different firms have different assembly language, so each microcontroller requires the user to learn a new language and that's abet difficult. So programming microcontroller with high level language can be easier to learn by user than assembly languages.

Microcontrollers are classified by several parameters, number of bits they process it's the main one. Microcontrollers with 8 bits are the most suitable for most microcontroller-based small applications. In other hand Microcontrollers with 16 and32 bits are much more powerful and complex, but they are usually more expensive and it's suitable for medium and large size general purpose applications that call for a powerful microcontrollers.

All microcontrollers require a clock (oscillator) to operate, usually provided by external timing devices connected to the microcontroller. In most cases, these external timing devices are a crystal plus two small capacitors. In some cases they are resonators or an external resistor-capacitor pair, which generates oscillation that has cretin frequency.

Peripherals are additional components may be added to the microcontroller such as Analog-to-Digital Converter (ADC), Liquid Cristal Display LCD, and timers.

Common Types of Microcontroller [2]

1- PIC (8-bit PIC16, 16-bit dsPIC33/PIC24)
2- Intel (8051, 8085, 8086)
3- PLC
4- ATmega
5- MIPS
6- Arduino
7- ARM processors

2.3 Sensor

In this kind of systems temperature and humidity degrees must be monitored all the time, due to its large effect on the chickens lives, sensors can provide us an accurate measurement for both temperature and humidity degrees.

The large farm size; requires large numbers of temperature and humidity sensors, which are distributed regularly all around the farm, in order to get measurement from it; these sensors must be connected to the main system interface, wireless network is the most suitable form of connection, to do so sensors are connected to the interfacing system using a Wi-Fi network. Wireless sensor node contains: sensors, analog circuit, microcontroller, battery and radio connection, as shown in figure 2.2.

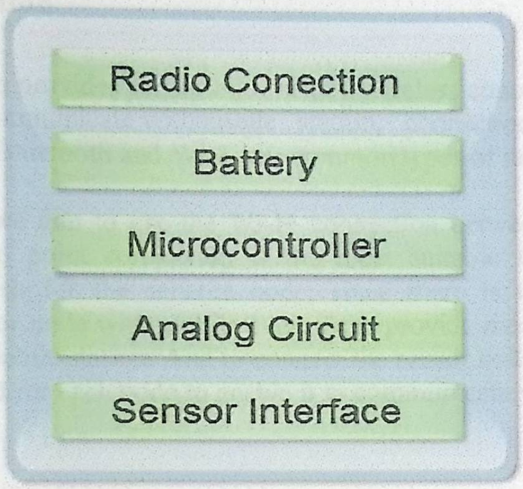


Figure 2.2: Sensor Node Architecture.

2.3.1 Sensor

The simplest devices in the sensor node, it gathers measurement about environment physical parameters such as temperature, pressure and humidity levels. Different types of sensors can be used, from simple temperature and humidity monitoring sensors to complex digital sensors which are capable of performing some processing, collect measurement and communicating with other nodes in the network. Sensor node components should be small size consume extremely low energy; in order saving battery life.

2.3.2 Analog Circuit

Sensors output signals in analog form, but microcontrollers accept digital signals as input; so we need to convert the analog sensors output signals to a digital signals that microcontroller can accept and process. Analog circuit acts like interface between sensors and microcontroller which contains analog-to-digital converter (ADC) that converts sensors output signals to a microcontroller input digital ones with 10 bit resolution.

2.3.3 Battery

Battery is power supply for the sensor node, it provides all sensor node components with a constant Dc voltage, and deferent types of batteries can be used in sensor node, also we can use an Ac voltage by converting it to a Dc voltage using a Dc power supply circuit as permanent supply without the need of replacing the batteries

2.3.4 Radio Connection

Radio connection provides sensor node with a wireless communication technique, which allows sensor node to communicate withmobile station, connection can be implemented in deferent ways; ZigBee, Bluetooth and Wi-Fi are common types of radio connection.

In this system, we aim to use the Wi-Fi connection between the sensors nods and the mobile station, point to point connection is the most suitable form to do so, no network configuration is establish for the sensors node; since there is no need to do so, a direct connection to each sensor node with mobile station can provide measurements transferring from the sensor node to the mobile station. And to connect the sensor node to the Wi-Fi network a Wi-Fi transceiver is required for each node to enable it to communicate via the network.

2.4 Camera

Similarly, the camera node consists of several parts: camera device, analog circuit, power supply and the radio connection as shown in figure 2.3.

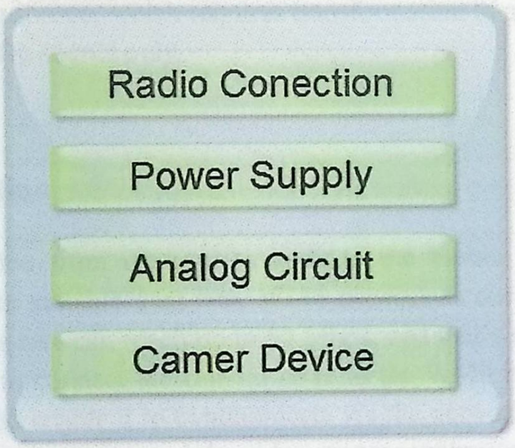


Figure 2.3: Sensor Node Architecture.

2.4.1 Camera device

A video camera is a camera used for electronic motion picture acquisition, initially developed by the television industry but now common in other applications as well. Video cameras are used primarily in two modes. The first, characteristic of much early broadcasting, is live television, where the camera feeds real time images directly to a screen for immediate observation. A few cameras still serve live television production, but most live connections are for security, military/tactical. In the second mode the images are recorded to a storage device for archiving or further processing; for many years, videotape was the primary format used for this purpose.

We intent to use the camera device in order to monitor the farm condition like: eggs, feeding, cleaning and to monitor the chicken condition if there is a dead chicken in order to get rid of it quickly; to prevent the infection to other chicken.

2.4.2 Analog circuit

Camera output signal is in the analog form, those we need to convert this signal to a digital one, analog to digital converter ADC can convert the camera output signal to a digital signal that the PIC microcontroller can understand, the ADC must be fast as the camera signal data rate; to overcome delay and to do so the operating frequency of the ADC must be greater than signal data rate. This can be done by using a 20 MHz oscillation crystal.

2.4.3 Power supply

Camera node must be on all the time; which requires permanent power source, thus, we must connect it to an Ac source, which requires transformation to a Dc voltage; since monitoring camera operates with a Dc source only.

2.4.4 Radio connection

Transferring video from the camera node to the mobile station wirelessly requires a wireless connection with suitable data rate, Wi-Fi network is our choice; sense Wi-Fi network can support us with data rate can reach up to 54 Mbps, and that's more than enough to carry out video signals, in order to connect the camera node to the Wi-Fi network a Wi-Fi transceiver is required for each node to enable it to communicate via the network, also Wi-Fi transceiver must have at least 1-2 Mbps data rate to send the video streams without any kind of delay.

2.5 Wi-Fi network

Wi-Fi: wireless local area network (WLAN) is a popular technology that allows an electronic device to exchange data wirelessly using radio wave, including high speed connections. Products that are based on the (IEEE) 802.11 standards.[3]

A device that can use Wi-Fi (such as a personal computer, tablet, or digital audio player) can connect to a network resource such as the Internet via a wireless network access point. Such an access point (or hotspot) has a range of about 20 meters (65 feet) indoors and a greater range outdoors. Hotspot coverage can comprise an area as small as a single room with walls that block radio waves or as large as many square miles.

Advantages of Wi-Fi [4]

- Mobility
- Ease of Installation
- Flexibility
- Cheep cost
- Reliability and Security
- Use unlicensed part of the radio spectrum 2.4 GHz
- Speed up to 54 Mbps

Limitations of Wi-Fi [5]

- Interference
- Degradation in performance
- High power consumption
- Limited range

Why using Wi-Fi technology?

Wi-Fi has many advantages: high data rate connection up to several tens MHz, it's good for live video transmitting, also Wi-Fi support mobility, for that Wi-Fi forms good solution for our system; we need to send a live video from the monitoring camera to mobile station, and this require high data rate which Wi-Fi can support.

The mobility in Wi-Fi allows the manager to monitor and control farm from different areas using a mobile station which can communicate wirelessly throw Wi-Fi network with the monitoring and controlling system.

A Wi-Fi transceiver is transmitting receiving devices that connect peripheral to Wi-Fi network, which allow devices to communicate throw the wireless channel.

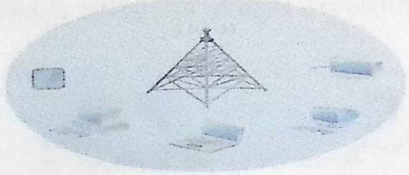


Figure 2.4: Wi-Fi Network.

2.6 Electrical and Mechanical Components

2.6.1 Squirrel Cage Induction Motors

The types of electrical motors, we have classified induction motors based on the type of rotor used. They are

1. Squirrel Cage Induction Motors
2. Slip-Ring or Wound-rotor Induction Motors

Squirrel cage Induction motors rotors consists of a cylindrical laminated core with parallel slots for carrying the rotor conductors, which are not wires because of the type of rotor used in these motors named Squirrel cage, almost 95% of the induction motors used is of squirrel cage type [6]

Construction of Squirrel cage induction motor

Any induction motor has a stator and a Rotor. The construction of Stator for any induction motor is almost the same. But the rotor construction differs with respect to the type which is specified above.

The rotor [7]

This kind of rotor consists of a cylindrical laminated core with parallel slots for carrying the rotor conductors, which are not wires, there are end rings which are welded or electrically braced or even bolted at both ends of the rotor.

The stator [7]

The stator is the outer most component in the motor, the three phase windings are placed on the slots of laminated cylindrical core and these windings are electrically spaced 120 degrees. These windings are connected as either star or delta depending upon the requirement as shown in figure 2.5.

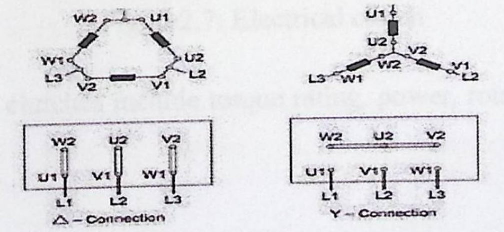


Figure 2.5: Stator Connection.

The equivalent electrical circuit of the induction motor as shown in figure 2.6:

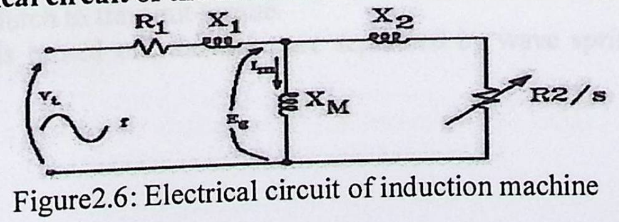


Figure2.6: Electrical circuit of induction machine

The squirrel cages have been classed under 6 different Categories [7]:

1. Class A: The squirrel cage rotor has relatively low resistance & reactance.
2. Class B: These motors can be started at a full-load while developing normal starting torque.
3. Class C: These are double squirrel cage motors. These double squirrel rotors combine high starting torques with low starting currents
4. Class D: These motors have high starting torques which is achieved by design of the rotor slots. These rotor slots have thin rotor bars
5. Class E: These motors have relatively very low slip at rated load. For motors above 5 KW rating, the starting current may be sufficiently high as to require a compensator or a resistance starter.
6. Class F: these motor combine low starting torque with low starting current and may be started with full voltage.

2.6.2 Electrical clutch

Electric clutches are equipment drive assemblies that contain electrically actuated components for connecting two shafts so that they can either be locked together and spin at the same speed, or decoupled and spin at different speeds. Engaging the clutch transfers power from an engine to devices such as a transmission and drive wheels [9].
Disengaging the clutch stops the power transfer, but allows the engine to continue turning.

Figure2.7: Electrical clutch

Specifications for electric clutches include torque rating, power, rotational speed, and operating voltage [8].

How it work:

Electrically engaged clutches or brakes require that electricity be supplied to a coil for engagement. While electricity is being supplied, a coil generates a magnetic field and the unit is engaged. The magnetic field is used to pull on an end plate, which squeezes a set of friction discs together allowing the clutch to transmit torque.
When the electricity is turned off, the discs are separated by wave springs, and the clutch is disengaged [9].

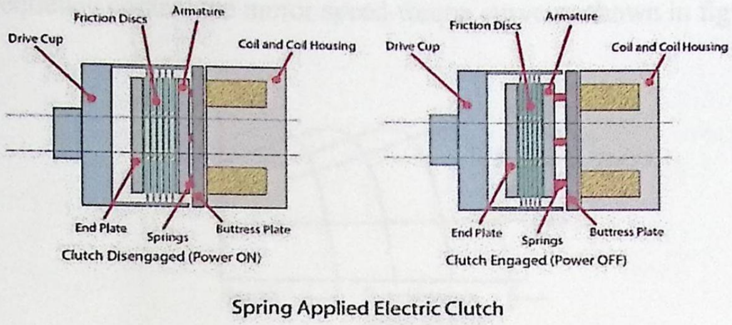


Figure2.8: Electrical clutch engaged and disengaged

2.6.3 Frequency converter:

Variable frequency drive is an electronic speed control for an alternating current (AC) motor as shown in figure2.9. Variable frequency drives are also known by other names such as adjustable frequency drives (AFDs), AC drives, inverter drives, inverters, variable speed drives (VSDs) and adjustable speed drives (ASDs) [10].

These terms are generally applied to just the electronic control unit, but are sometimes applied to the entire drive system including the control unit, motor, operator control interface and other related equipment.

Reason for using adjustable speed drives:

1- Adjustable speed drives can be means of saving energy, E.g. the fans and pump are the most common energy saving application
2- provide smother operation
3- provide acceleration control
4- use different operating speed for each process
5- allow slow operation for setup purpose
6- allow accurate positioning
7- control the torque

Figure 2.9 Frequency converters.

AC drives –frequency control:

Adjustable frequency control changes the frequency of the power supply to change the motor synchronous speed.

With adjustable frequency control the motor speed-torque curve as shown in figure 2.10.

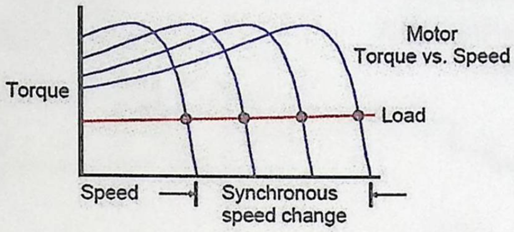


Figure 2.10: Motor Speed-Torque Curve with Adjustable Frequency.

In order for the motor to have similar torque VS speed characteristic over a range of operating frequencies the applied voltage must be proportional to the applied frequency V/F =K

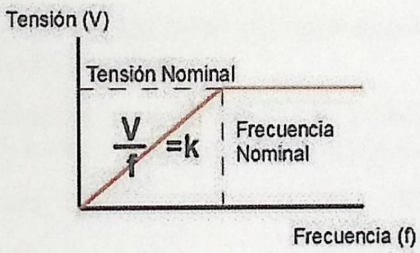


Figure 2.11: Speed –Torque characteristics with V/F control

Electrical circuit:

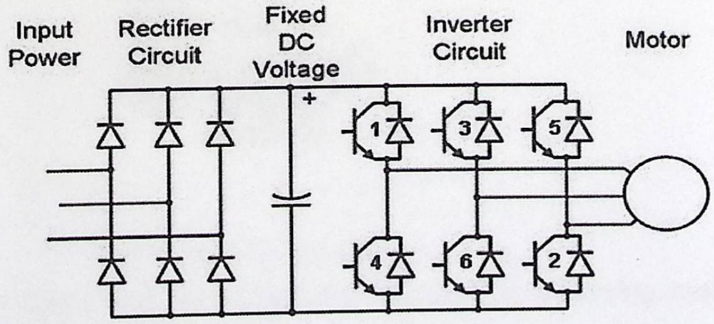


Figure 2.12: AC drives Electrical Circuit.

The frequency converter work as shown in figure2.13:

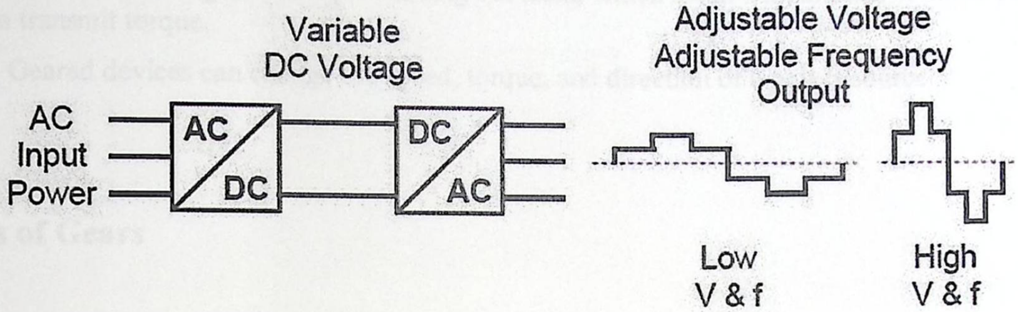


Figure 2.13: Frequency converter work

2.6.4 Rotary Valves [11]

Rotary valves: are commonly used in industrial and agricultural applications as a component in a bulk or specialty material handling system.

Rotary feeders are primarily used for discharge of bulk solid material from hoppers, receivers, and cyclones into a pressure or vacuum-driven pneumatic conveying system.

Components of a rotary feeder include a rotor shaft, housing, head plates, and packing and bearings.

Rotors have large vanes cast or welded on and are typically driven by small internal combustion engines or electric motors as shown in figure 2.14.

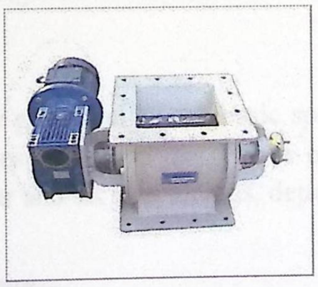


Figure 2.14: Rotary valve.

Rotary airlock feeders/ rotary airlock valves are used in pneumatic conveying systems, dust control equipment, and as volumetric feed-controls.

Type's Rotary valve:

1. Rotary airlock feeders.
2. Blow-thru rotary airlock feeder.
3. Easy clean rotary feeder.
4. Filter valve.
5. Knife rotary feeder.

2.6.5 Gears [12]

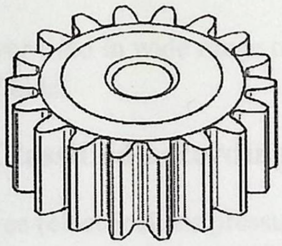
Gear is a rotating machine part having cut teeth, which mesh with another toothed part in order to transmit torque.

Geared devices can change the speed, torque, and direction of a power source.

Types of Gears

1. A spur gear

Cylindrical in shape, with teeth on the outer circumference that are straight and parallel to the axis (hole).
There are a number of variations of the basic spur gear including pinion wire, stem pinions, rack and internal gears



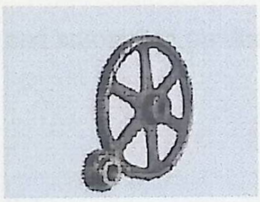


Figure 2.15: Spur Gear

2. Rack gear

It's yet another type of spur gear. Unlike the basic spur gear, racks have their teeth cut into the surface of a straight bar instead of on the surface of a cylindrical blank as shown in figure 2.16. Rack is sold in two, four and six foot lengths, depending on pitch.

Figure 2.16: Rack Gear.

3. Sprocket gears

A sprocket-wheel is a profiled wheel with teeth, even sprockets hat mesh with a chain, track or other perforated or indented material Sprockets are used in bicycles, motorcycles, cars, tracked vehicles, and other machinery either to transmit rotary motion between two shafts where gears are unsuitable or to import linear motion to a track.

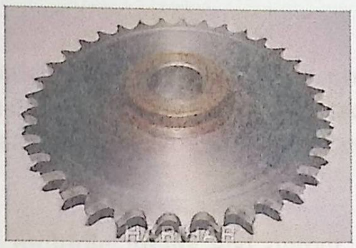


Figure2.17: Sprocket Gear

2.6.6 Conveyors [13]

Equipment intended for carrying material or persons transported from one place to another continuous flow or a specific time interval. Belt conveyors or continuous transport machines has the possibility of loading and unloading of non-stop of the carrier.

The conveyor used in wide range of industry and automated production processes in various industrial fields.

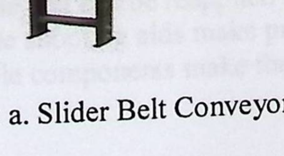
The conveyor Classified according to:

1. Driving force (electrical, air pressure, hydraulic).
2. Type of material
3. The direction of motion (level, horizontal, vertical)
4. Situation at the facility (fixed, movable).
5. The carrier element of the movement (belts, chains), The most dominant category of conveyors

The conveyor according to carrier elements

Handling belt conveyor systems: The most common transport machines in farming and industrial areas, food and service with varying productivity up to 20000 tons / hour, and transmission distance of up to 4500 meters. Also used at various speeds up to7m/ s.

General material handling belt conveyor systems are typically comprised of one of two designs – **roller bed belt conveyors**, in which the belt is supported and carried by rollers as shown in figure 2.18b, or **Slider bed belt conveyors**, in which the belt is supported and carried over a continuous metal bed surface as shown in figure 2.18 a.



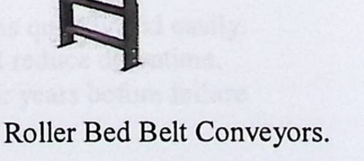


Figure 2.18: a. Slider Belt Conveyor

b. Roller Bed Belt Conveyors.

This type of conveyor will be used in our project, using slider belt conveyor for cleaning and roller bed belt conveyor for eggs collection.

2.6.7 PLC

A programmable logic controller (PLC) or programmable controller is a digital computer used for automation of electromechanical processes.
The advent of the PLC began in the 1970s, and has become the most common choice for manufacturing controls.

PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact.

PLC consists of:

1. Central Processing Unit (CPU)
2. Power Supply Unit
3. Memory Unit
4. Input/output Interface
5. Programming Device

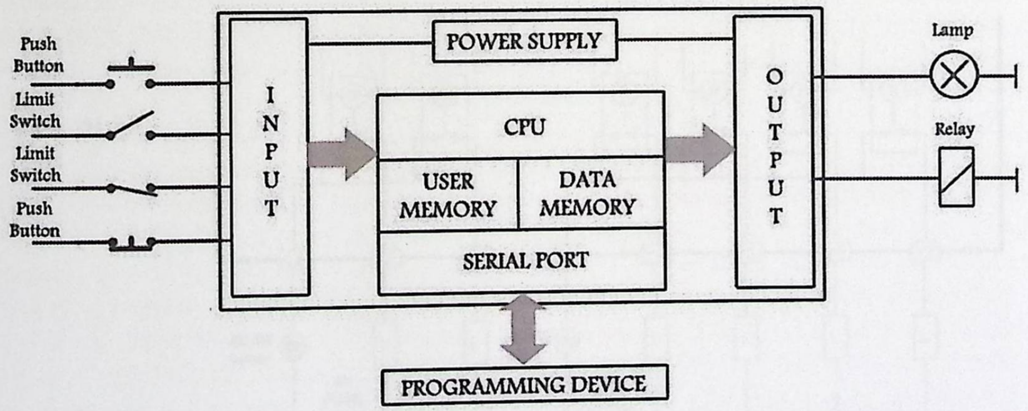


Figure 2.19: PLC components

PLCs have been gaining popularity on the factory and will probably remain predominant for some time.

The advantages of PLC:

1- Cost effective for controlling complex systems.
2- Flexible and can be reapplied to control other systems quickly and easily.
3- Trouble shooting aids make programming easier and reduce downtime.
4- Reliable components make these likely to operate for years before failure.

The PLC inputs are isolated from the output as shown in figure 2.20.

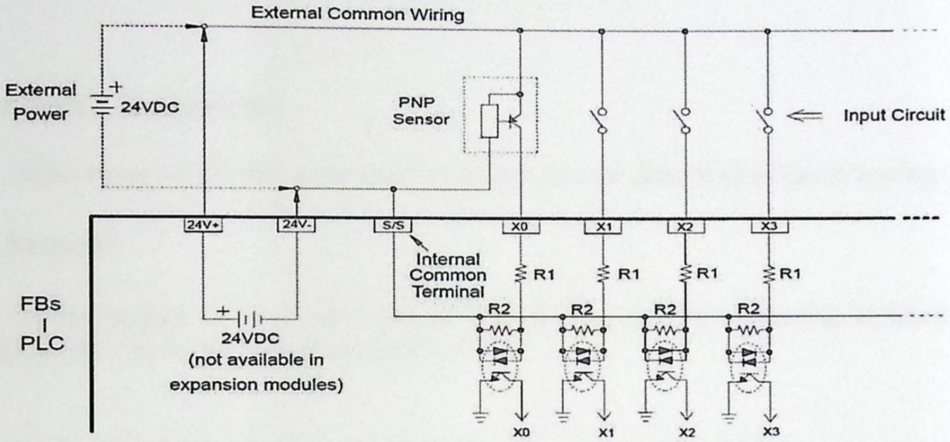


Figure2.20: Input PLC Connections

The PLC outputs as shown in Figure are the relay with no polarity. We can connect AC and DC loads with this type of PLC.

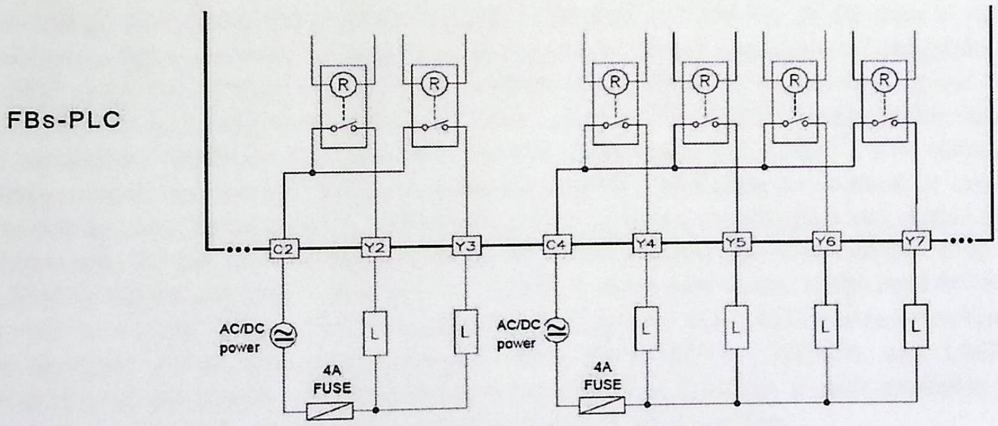


Figure2.21: PLC outputs connection

The PLC can be programmed using different languages, such as:

1. Ladder diagram
2. Functional block diagram
3. Instruction list

2.6.8 Fans

Farms needing to entrance a new oxygen to breath birds and exciting the second oxidized carbon(CO2), ammonia, Excess humidity and Excess heat. Ventilation is very important factor in chicken farms because chicken Exposed to poor ventilation diseases especially chronic respiratory. New air entering the new oxygen needed to breathe birds.

This several parameters (Excess humidity and Excess heat) can be controlled by using fans; a humidity suction fans and temperature decreasing fans.

2.7 Project Components

Main components that were used in this project are described as the following:

2.7.1 Sensors

In this project we used two types of sensors: first one for measuring temperature degree and the second one to measure the humidity level:

2.7.1.1 LM35 Precision Centigrade Temperature Sensor

General Description:

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features:

- Calibrated directly in ° Celsius (Centigrade).
- Linear + 10.0 mV/°C scale factor.
- 0.5°C accuracy guaranteeable (at +25°C).
- Rated for full −55° to +150°C range.
- Suitable for remote applications.
- Low cost due to wafer-level trimming.
- Operates from 4 to 30 volts.
- Less than 60 μA current drain.
- Low self-heating, 0.08°C in still air.

- Nonlinearity only ±1⁄4°C typical.
- Low impedance output, 0.1 W for 1 mA load.

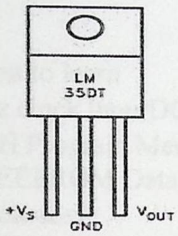


Figure 2.22: LM35 Temperature Sensor.

2.7.1.2 HS1101LF – Relative Humidity Sensor

General Description: [16]

Based on a unique capacitive cell, these relative humidity sensors are designed for high volume, cost sensitive applications such as office automation, automotive cabin air control, home appliances, and industrial process control systems. They are also useful in all applications where humidity compensation is needed.

Features:

- Lead free component.
- High reliability and long term stability.
- Patented solid polymer structure.
- Suitable for linear voltage or frequency output circuitry.
- Fast response time and very low temperature coefficient.
- Operating Temperature Ta -60 to 140 °C.
- Supply Voltage (Peak) Vs 10 Vac.
- Humidity Operating Range RH 0 to 100 % RH

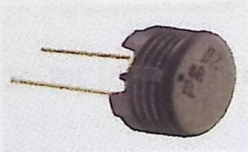


Fig 2.23: LM35 Temperature Sensor

2.7.2 Microcontroller

In this system, we used both **PIC16f877a** and **PIC16f688** microcontroller, it takes inputs from a verity sensors and devices, also it can easily give output signals to control other systems.

2.7.2.1 PIC16f877a Microcontroller [17]

General Description:

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- Operating speed: DC - 20 MHz clock inputDC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,Up to 368 x 8 bytes of Data Memory (RAM)Up to 256 x 8 bytes of EEPROM Data Memory
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Power saving SLEEP mode
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Low-power consumption:
  - < 0.6 mA typical at 3V, 4 MHz
  - 20 μA typical at 3V, 32 kHz
  - < 1 μA typical standby current

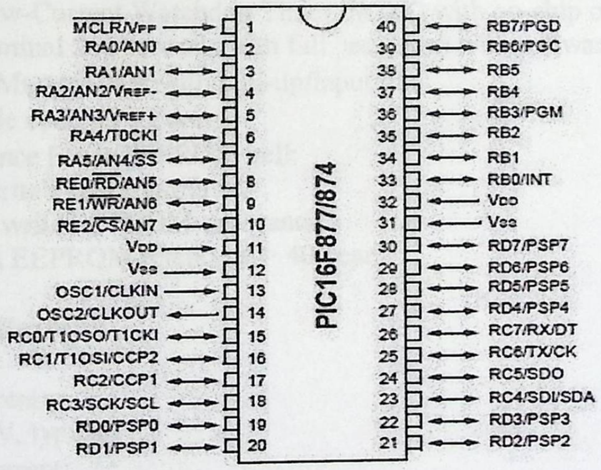


Fig 2.24: PIC16f877 Microcontroller.

2.7.2.2 PIC16f688 Microcontroller [18]

General Description:

High-Performance RISC CPU:

- Only 35 instructions to learn:
  - All single-cycle instructions except branches.
- Operating speed:
  - DC – 20 MHz oscillator/clock input.
  - DC – 200 ns instruction cycle.

- Interrupt capability.
- 8-level deep hardware stack.
- Direct, Indirect and Relative Addressing modes.

Special Microcontroller Features

- Precision Internal Oscillator:
  - Factory calibrated to ±1%.
  - Software selectable frequency range of 8 MHz to 31 kHz.
  - Software tunable.
  - Two-Speed Start-up mode.
  - Crystal fail detect for critical applications.
  - Clock mode switching during operation forpower savings.
- Power saving Sleep mode.
- Wide operating voltage range (2.0V-5.5V).
- Industrial and Extended temperature range.
- Power-on Reset (POR).
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST).
- Brown-out Detect (BOD) with software control option.
- Enhanced Low-Current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable.
- Multiplexed Master Clear with pull-up/input pin.
- Programmable code protection.
- High-Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance.
  - 1,000,000 write EEPROM endurance.
  - Flash/Data EEPROM retention: > 40 years.

Low-Power Features

- Standby Current:
  - 1 nA at 2.0V, typical.
- Operating Current:
  - 8.5µA at 32 kHz, 2.0V, typical.
  - 100µA at 1 MHz, 2.0V, typical.
- Watchdog Timer Current:
  - 1µA at 2.0V, typical.

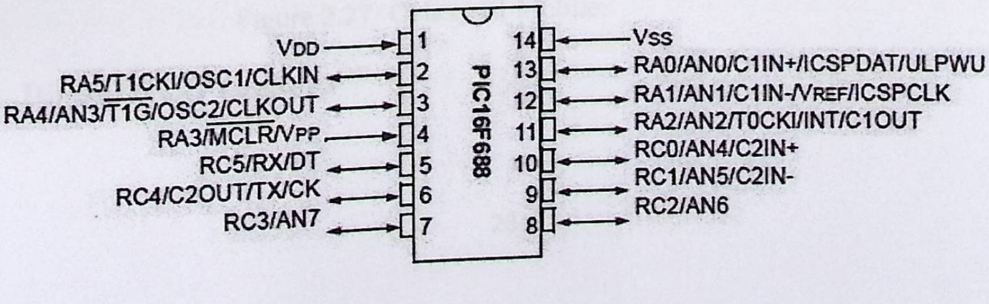


Figure 2.25: PIC16f688 Microcontroller.

2.7.3 PLC

In this project we intend to use delta PLC DVP-ES2/EX2 Series; because it's available in the market and easy to deal with and also it's not expensive.

ES2 series is a small PLC for basic sequential control. It is economical, highly efficient and functional.

According to the number of input/output of the delta PLC, the output method and the power supply we have of many kind of ES PLC series choose **32ES00R2** that have 100-220 VAC and relay output ,have 16 input and 16 output and have fuses 2A/250VAc.

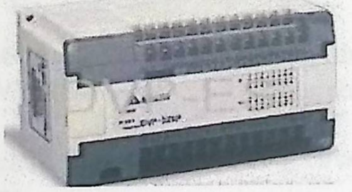


Figure 2.26: Delta PLC DVP-ES2/EX2

2.7.4 Mobile Station

The mobile device will be the interfacing ring between humans and the system, manager can easily monitor and control the system from different places inside the farm; due to mobility that both mobile station and Wi-Fi provides.

So we need to design an application that controls the system and install it on the mobile station, the application will be in android language operating on a device with android operating system.

Galaxy SII with an operating system android version 2.1.1will be our choice; a smart phone and an open source operating system allows the application to install on any device.

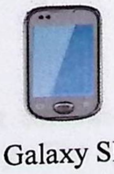


Figure 2.27: Galaxy SI Mobile.

2.7.5 Digital CCD Camera[20]

The Digital CCD Camera is a highly integrated serial camera module which can be attached to any host system that requires a video camera or a JPEG compressed still a

Camera for embedded imaging applications. The module uses an Omni Vision CMOS VGA color sensor along with a JPEG compression chip that provides a low cost and low powered camera system. The module has an on-board serial interface (TTL or RS232) that is suitable for a direct connection to any host micro-controller UART or a PC system COM port. User commands are sent using a simple serial protocol that can instruct the camera to send low

Resolution (160x120 or 80x60) single frame raw images for a quick viewing or high resolution (640x480 or 320x240) JPEG images for storage or viewing. The CCD Camera comes in a compact form factor with a built in lens and a 4-wire connector that provides easy access to both power and serial data.

Features:

- Small size, low cost and low powered camera module for embedded imaging applications.
- CAM-TTL: 3.3V DC Supply
- CAM-232: 5.0V DC Supply
- On-board EEPROM provides a command based interface to external host via TTL or RS-232 serial link.
- UART: up to 1.2Mbps for transferring JPEG still pictures or raw images.
- On board OmniVision OV7640/8 VGA color sensor and JPEG CODEC for different resolutions.
- Built-in down sampling, clamping and windowing circuits for VGA, QVGA, 160x120 or 80x60 image resolutions.
- Built-in color conversion circuits for 2-bit gray, 4-bit gray, 8-bit gray, 12-bit RGB, 16-bit RGB or standard JPEG preview images.
- No external DRAM required.

Figure 2.28: Digital CCD Camera

2.7.6 Wi-Fi Transceiver [21]

A transceiver is a device that contains a transmitter and a receiver, which are both combined and share common circuitry. The system needs Wi-Fi transceiver for sending and receiving data about the farm to mobile station. Transceiver is a two-way radio that combines a radio transmitter and a radio receiver for exchanging information.

The XBee Wi-Fi RF module provides wireless connectivity to end-point devices in 802.11 bgn networks. Using the 802.11 feature set, these modules are interoperable with other 802.11 b/g/n devices, including devices from other vendors. With XBee, users can have their 802.11 b/g/n network up-and running in a matter of minutes. The XBee Wi-Fi modules are compatible with other devices that use 802.11 b/g/n technologies.

General Specifications:

- Dimensions: 0.960 x 1.297 (2.438cm x 3.294cm).
- Operating Temperature: -40 to 85° C (Industrial).
- Antenna Options: RPSMA Connector, or Integrated Wire.
- Operating Current (Receive): 140mA.
- Deep Sleep Current: <2uA @25C.
- Frequency: ISM 2.4-2.5GHz.
- Number of Channels: 14.
- Adjustable Power: Yes.
- Interface immunity: 802.11 b, g, and n.
- Indoor/Urban Range: TBD.
- Outdoor RF LOS-sight Range: TBD.
- Transmit Power Output : >15dBm.
- RF Data Rate: 802.11 b 1, 2, 5.5, and 11Mbps.
  802.11 g 6, 9, 12, 18, 24, 36, 48, and 54 Mbps.
  802.11 n 6.5, 13, 19.5, 26, 39, 52, 58.5, and 65 Mbps.

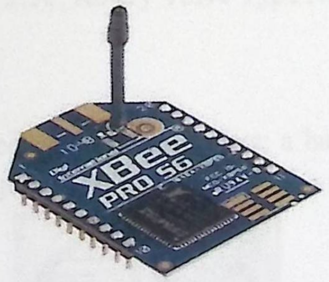


Figure 2.29: X-Bee Wi-Fi Transceiver.

2.7.7 Motors

Use 3 phase squarel cage induction motor for conveyor drive because it's easy to control using frequency converter and provide high torque we need for conveyor, Use 1 phase ac motor for valves.

2.7.8 Conveyers

Using for cleaning conveyor use slider bed belt conveyor because its supported and carried over continues metal bed surface, and use apron conveyor for the main egg collection conveyor and slider bed for sub conveyor.

2.7.9 Frequency converters

The specific types of frequency converter determined by calculate the power of each motor in my project use frequency converter VFD-L series the Specifications shown in appendix

2.7.10 Gears

Use a spur and rack gears for eggs conveyor to be move up and down and also use sprocket gears to connect cleaning conveyor with each other.

2.7.11 Rotary Valves

Rotary valve type NS

PAM rotary type NS is design for conveying of almost all free flowing non –abrasive powders and granulates, the housing and the covers are made of heavy cast iron [21].

Figure 2.30: Rotary Valve Type NS

2.7.12 Fans

Two types of fans are required in this project; a humidity suction fans and temperature decreasing fans, are shown in figure 2.31.

Figure 2. 31: Fan

3

Chapter 3

Conceptual System Design

This chapter presents the design concepts of the system. It describes the design objectives, illustrating how the system entities operate.

3.1 System operation

System main block diagram figure 3.1 demonstrating how the system entities operates:

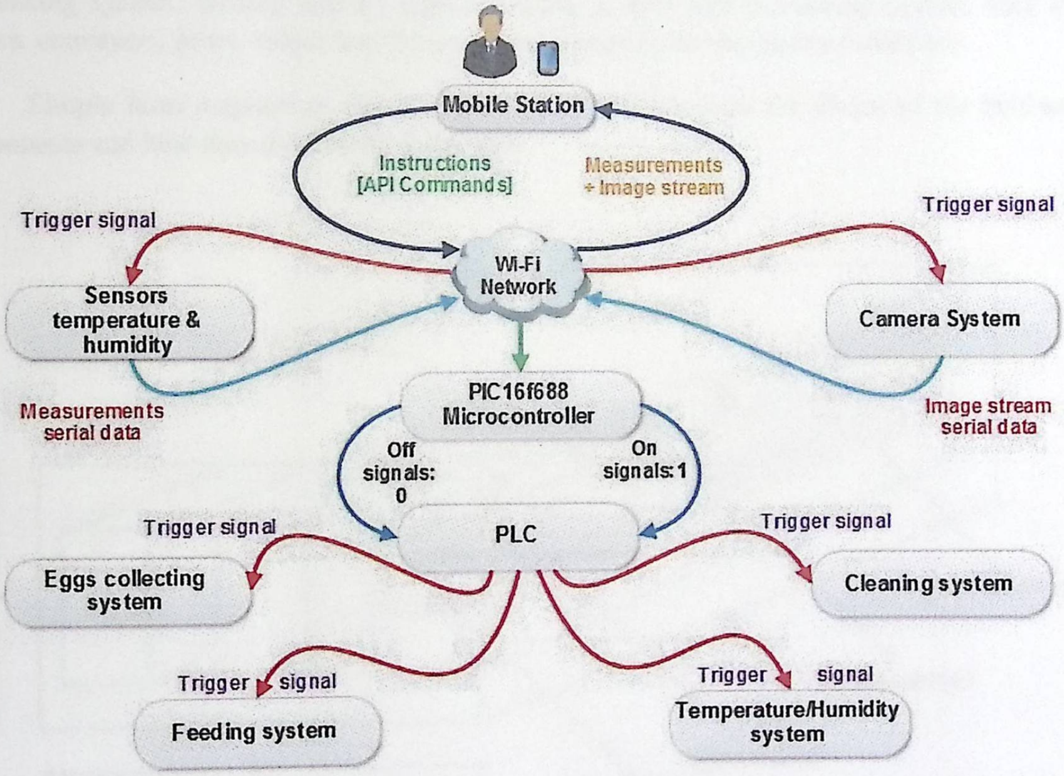


Figure 3.1: System Main Block Diagram.

The manager requests measurements from the sensors, sensors start sensing the status of the farm in terms of temperature and humidity levels, these measurements are processed using a microcontroller (processing unit) to evaluate temperature in Celsius and humidity in percentage, in order to send it to the mobile station.

Just as sensors: camera captures live images about the farm status in general, for chickens, eggs, water, and cleaning conditions. Then the images sent to the PIC control unit to be processed and to be sent to mobile station using Wi-Fi network.

From the mobile station side, the manager observes data and decides the best action to do -cleaning, feeding, eggs collecting and humidity/ temperature controlling- Actions translate to functions by a mobile application. Then the PIC receives functions from the mobile station via Wi-Fi network. The PIC unit processes the received functions and translates it to On/Off signals, which PLC unit can deal with.

PLC control unit: is the head of the automation system, it controls the cleaning, feeding and eggs collecting sub systems. It receives the control-On/Off- signals from the PIC unit, with these control signals, PLC unit controls of the automation system components to do the required tasks.

Automation system contains different subsystems with different components, subsystems as cleaning system, feeding system, eggs collecting system and monitoring system, they use motors, conveyers, gears, valves and frequency converters to do the required functions.

Simple farm diagram as shown figure 3.2, which illustrate the places of the hardware components and how they distributed in the farm.

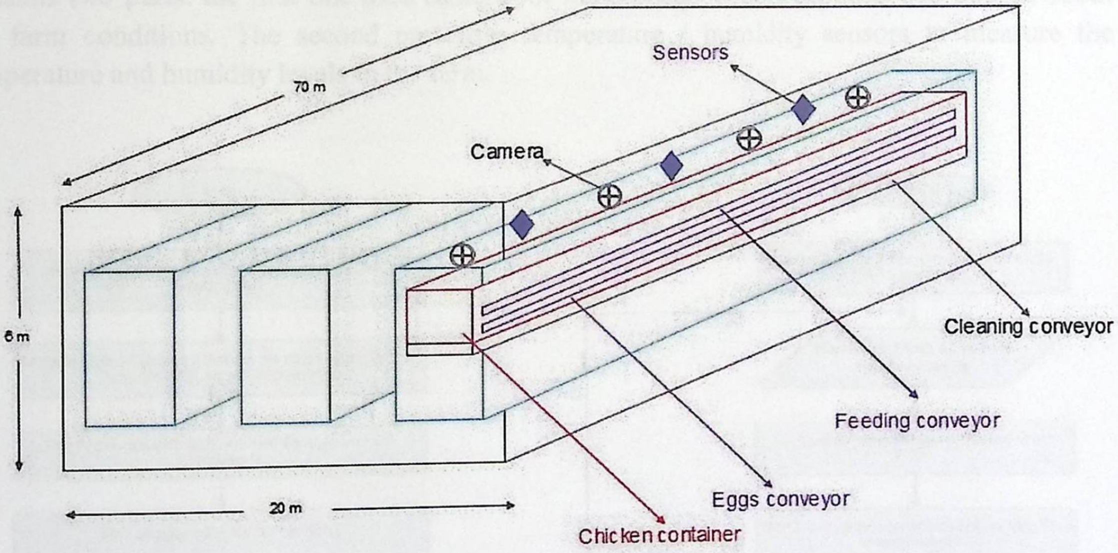


Figure 3.2: Simple Farm Diagram.

System organizational chart figure 3.3, which presents the system main functional parts, is shown in next page.

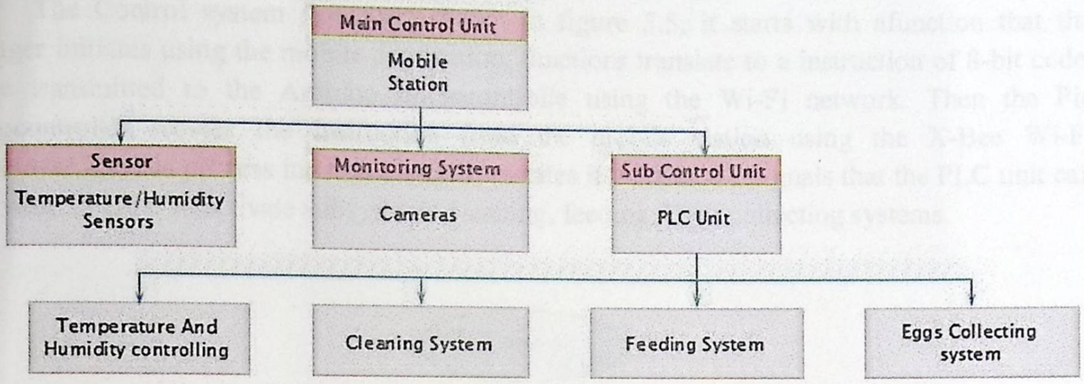
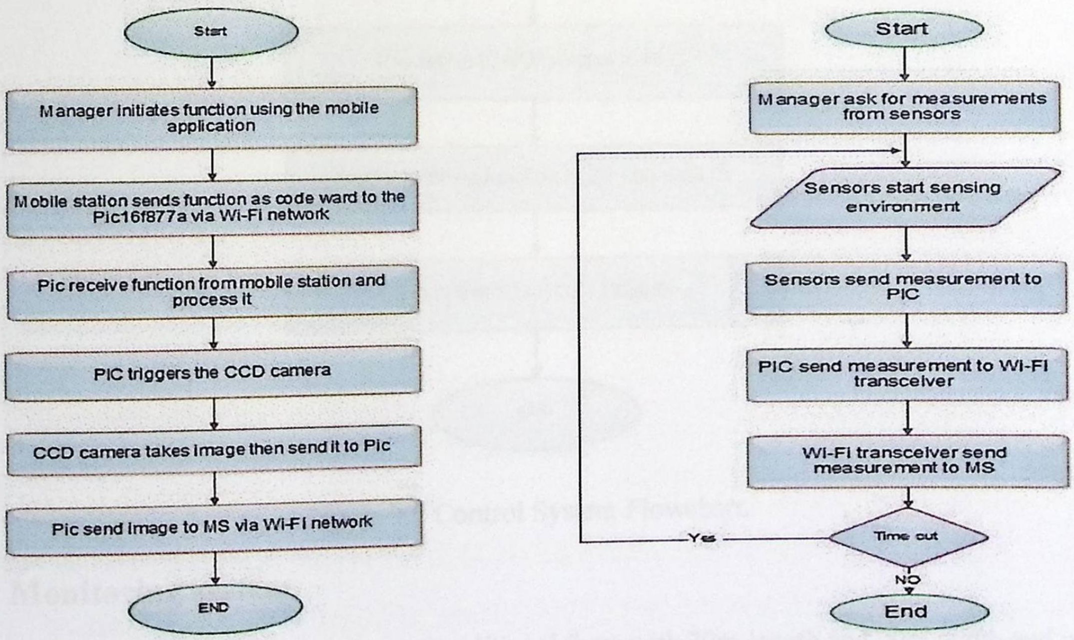


Figure 3.3: System Functional Chart

The monitoring system flowcharts are shown in figures 3.4; the monitoring system contains two parts: the first one uses camera for monitoring, which captures live images about the farm conditions. The second part uses temperature / humidity sensors to measure the temperature and humidity levels in the farm.



a- monitoring using camera

b- monitoring using sensor

Figure 3.4: Monitoring System Flowcharts.

The Control system flowchart shown in figure 3.5, it starts with afunction that the manager initiates using the mobile application, functions translate to a instruction of 8-bit code, to be transmitted to the Arduino microcontrolle using the Wi-Fi network. Then the Pic microcontroller recvies the instruction from the mobile station using the X-Bee Wi-Fi transciever, the Pic process instruction and translates it to ON/OFF signals that the PLC unit can deal with, in order to activate subsystems-cleaning, feeding, Eggs collecting systems.

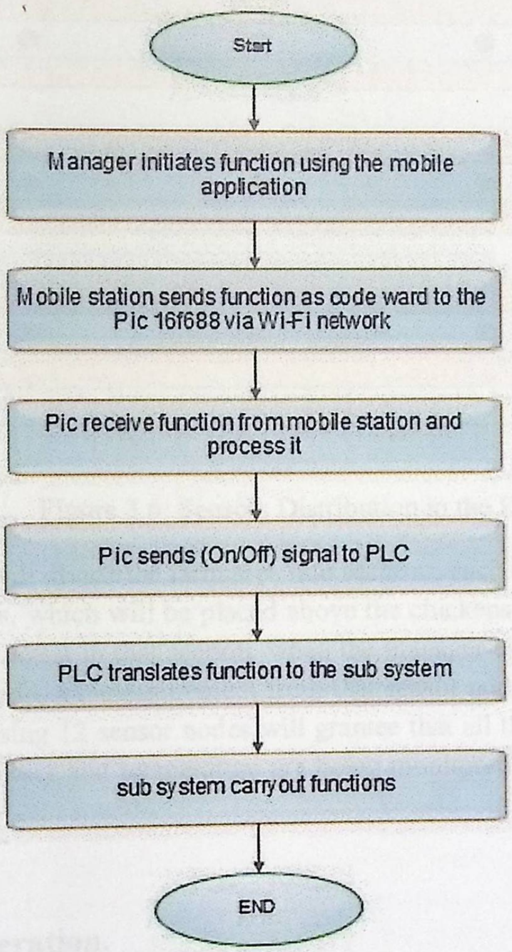


Figure 3.5 Control System Flowchart.

3.2 Monitoring system

This system is designed for real traditional farm with 70m length and 20m width and with 6m height.

Main monitoring system contains two monitoring subsystems, first subsystem designed for monitoring temperature and humidity levels using LM35 and HS1101LF Temperature and humidity and sensors.

The second monitoring subsystem, designed for monitoring the live status of the farm, using the CCD digital camera that capture images all the time and all around the farm.

3.2.1 Sensor system

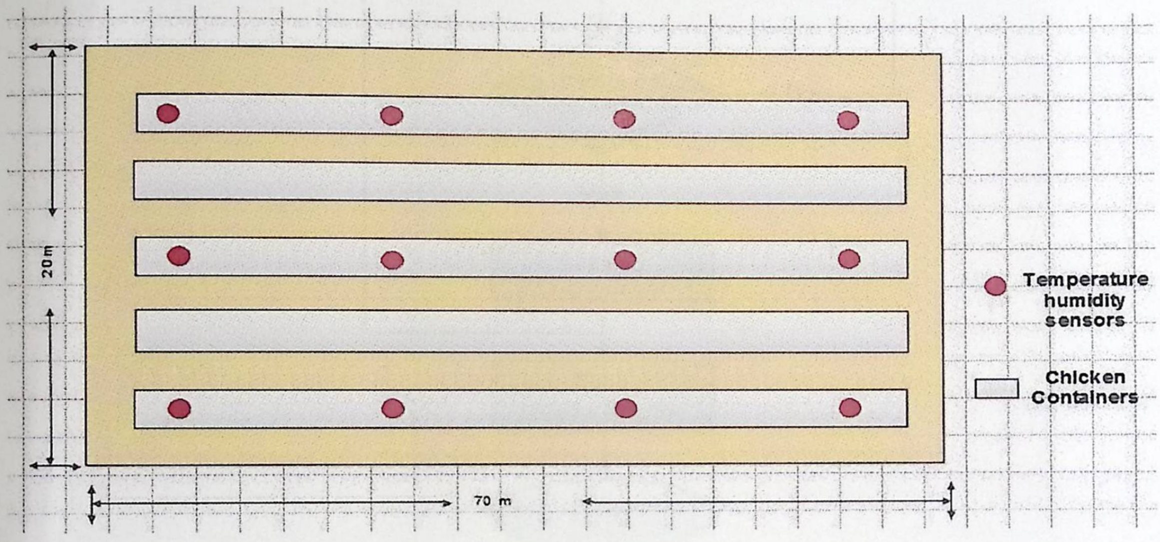


Figure 3.6: Sensors Distribution in the Farm.

In this design, we will divide the farm area into sections, each section contains (humidity-temperature) sensors nodes, which will be placed above the chickens containers; to measure the humidity and temperature levels in that section, when the manager asks for these measurements from a particular sensor node, the measurement from that sensor node sent to the mobile station via the Wi-Fi network. Using 12 sensor nodes will grantee that all the farm area being covered and the two levels of humidity and temperature are being monitored all around the farm all the time.

3.2.1.1 Sensor operation.

Sensors at the end points operate only when the MS request measurements. So, the normal mode is sleeping mode; which helps saving power.

Sensors output is normally in analog form. Thus, we need analog to digital converter (ADC) to convert sensors analog output to digital input for the PIC controller.

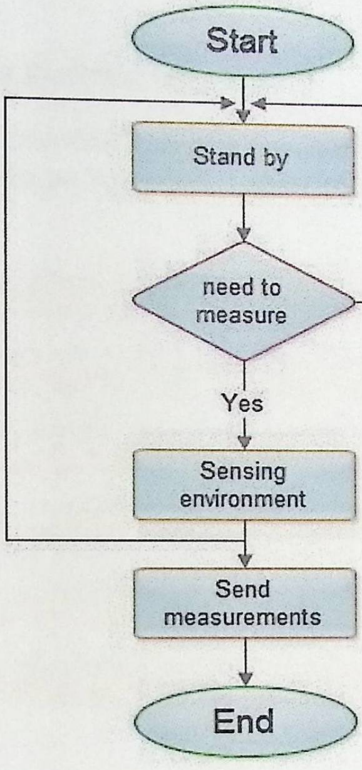


Figure 3.7: Sensors System Analysis.

3.2.1.2 Sensor node design:

Sensor must be interfaced to microcontroller to form a complete system. The sensor circuits which are known as sensing subsystem are connected to the PIC16F877a microcontroller. In this system, LM35 and HS1101LF sensors are the temperature and humidity sensor, the PIC contains A/D convertor, the microcontroller received the analog output by the analog input pins, so calculation is needed to get a real temperature and/or humidity value.

After receiving measurements from LM35 and HS1101LF the PIC16F877a microcontroller start processing measurements, the PIC passes it to the X-Bee Wi-Fi transceiver in order to transmit it to the mobile station when it requested, figure 3.8 in next page shows the sensors node component and how they are connected.

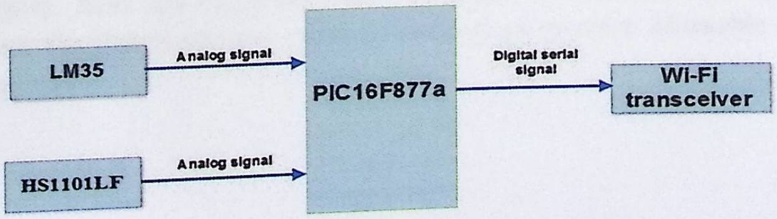


Figure 3.8: Sensors Node Component Analysis.

3.2.2 Camera system

3.2.2.1 Cameras distribution

For the second monitoring system to be implemented we can distribute the cameras around the farm in two different ways. So we have two different designs: the first one is shown in figure 3.9:-

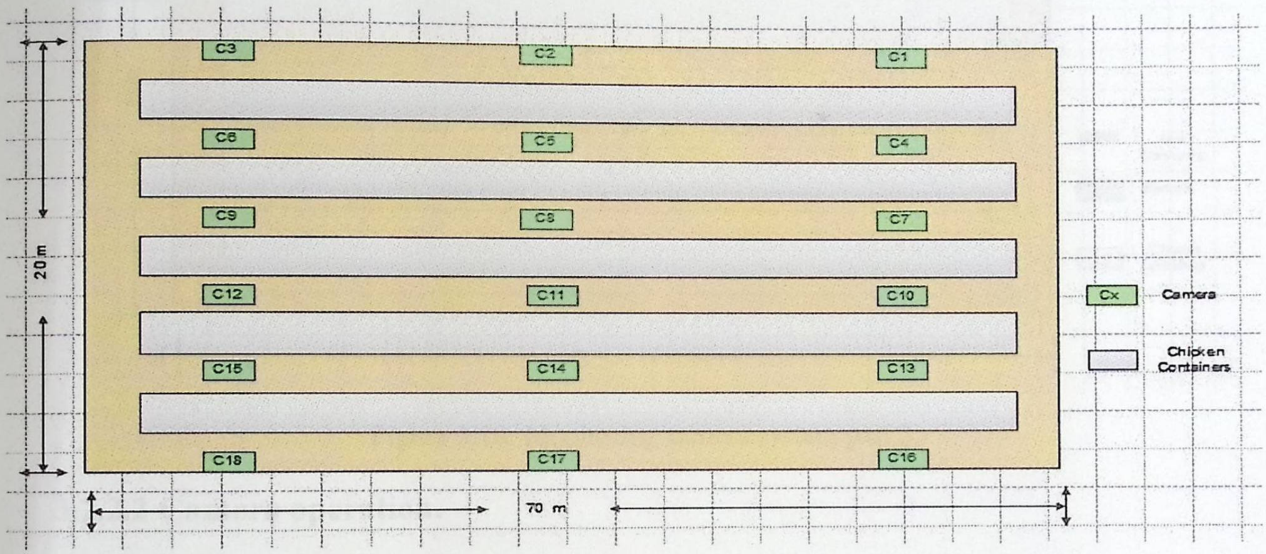


Figure 3.9: Monitoring Camera System Part 1.

First design method figure 3.9. Digital CCD Camera is the type of the used camera, it covers about 20m width; so three fixed Digital CCD monitoring camera are suitable to cover a 60m chicken containers length, thus; farm requires 18 camera of them.

The second method for setting up the monitoring system is shown in figure 3.10.In this method; we use a six Digital CCD monitoring camera along with 60m line conveyors for each to construct the monitoring system, cameras moves on the top of the line conveyors in bidirectional way forward and backward.

But this method is more expensive than the first one; duo to the large cost of the large length line conveyors. Also this method is more complex from the implantation side. So we intend to implement the first mentoring system; which gives us more adjustable margin in the implementation strategy.

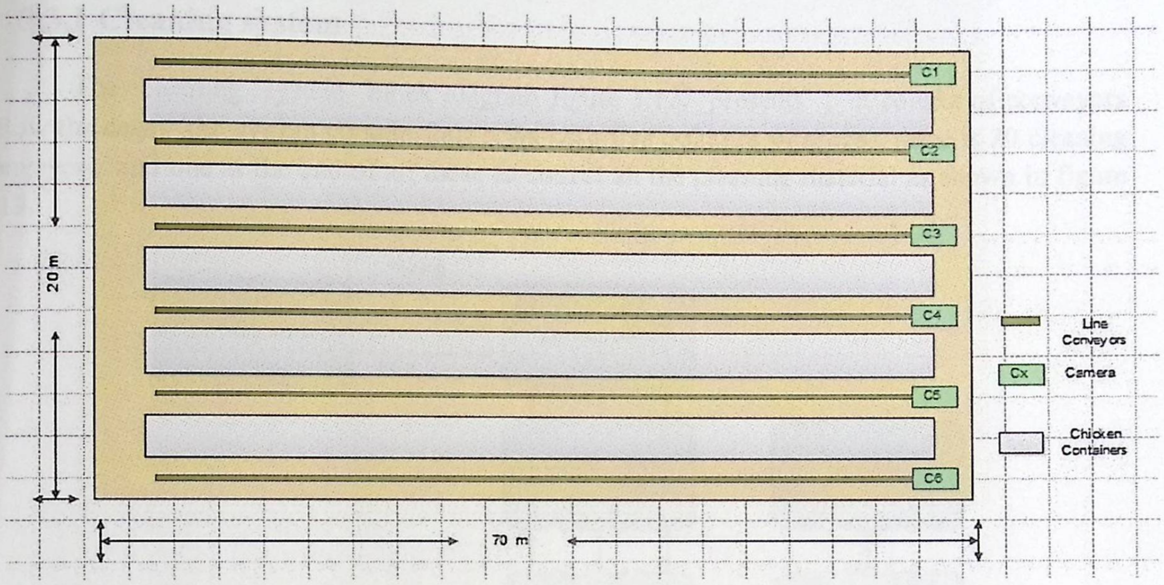


Figure 3.10: Monitoring camera system part 2.

3.2.2.2 Camera operation.

In this system, we need camera to show the situation of chicken, food, water, cleaning and eggs if it's ready to be collected. Also camera helps to detect if there is any dead chicken in order to get rid of it quickly to protect the farm from diseases as shown in figure 3.11.

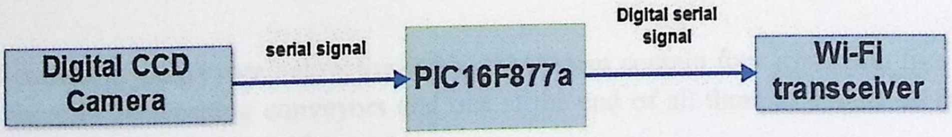


Figure 3.11: Camera System.

Camera output is in the serial digital form and so that PIC can understand it. Using RS232 as the interface between PIC and the camera on the 9600 baud rate, then PIC16F877a processes the image and to be sent throw the Wi-Fi transceiver, but to make the PIC fallows the high data rate of the image streaming, an 20MHz crystal are required to provide an a suitable oscillation for the PIC to deal with the image stream. Also the Wi-Fi transceiver must have a suitable transmitting rate over a 300-400Kbps, the X-Bee Wi-Fi transceiver has up to 16 Mbps, and that should do the required task.

3.3 Controlling system

3.3.1 Cleaning system

The cleaning system block diagram figure 3.12: presents ; It consist of conveyors below the cages ,the system contain four rows with five columns of cages , there is 20 cleaning conveyors and one at the end of all these to collect all the cleaning material as shown in figure 3.13.

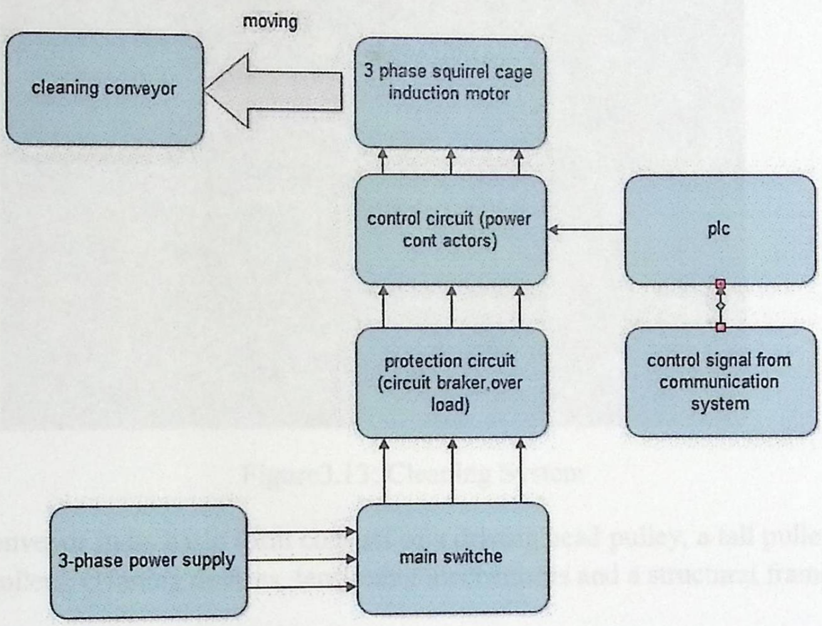


Figure 3.12: Cleaning System Block Diagram

It consist of conveyors below the cages, the system contain four rows with five Colums of cages, there is 20 cleaning conveyors and one at the end of all these to collect all the cleaning material as shown in figure 3.13.

Conveyors design

Using roller bed belt conveyors type , which have The carrier element handling belt conveyor systems which The most common transport machines in farming and industrial areas in which the belt is supported and carried by rollers because the cleaning material can be moist or wet, so need the belt to be supported and carried over a metal .

Roller bed belt conveyor used rather than slider bed belt because we have a large area of conveyor so need a large area of supported metal of steal or sometime Wood.

At the end of each these conveyors Use the carrier element handling belt conveyor systems in our design use slider bed belt conveyors because the cleaning material can be moist or wet.

The cleaning material of four conveyors collected on this conveyor; so we need the belt to be supported and carried over a continuous surface This surface can be either steel or a specialty material such as UHMW (Ultra High Molecular Weight Polyethylene).

Need a slant piece of iron at the end of the conveyor to remove the cleaning material from the conveyors under the cage to collect it on the main conveyor as shown in figure 3.13.

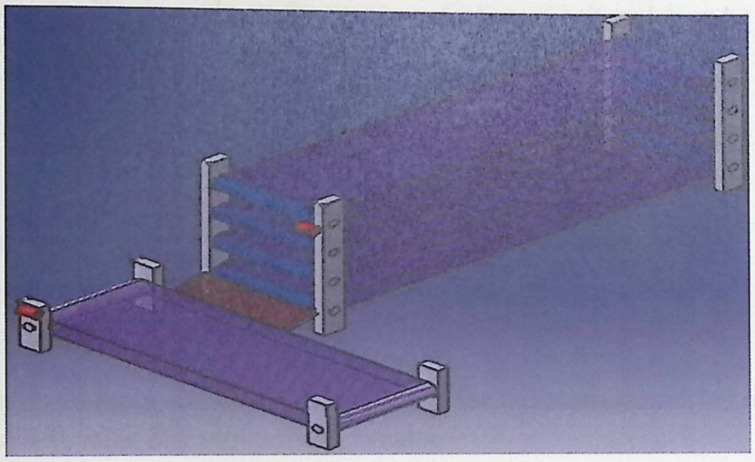


Figure3.13: Cleaning System

The conveyor in its basic form consists of a driving head pulley, a tail pulley, the moving belt, support rollers, cleaning devices, tensioning mechanisms and a structural frame.

How a Belt Conveyor Works

An electric motor and gearbox turn the head pulley, belt is pulled tight to produce friction between it and the head drum, only friction is used to drive the belt. If the friction falls the belt will slip or stop moving even though the head pulley keeps turning.

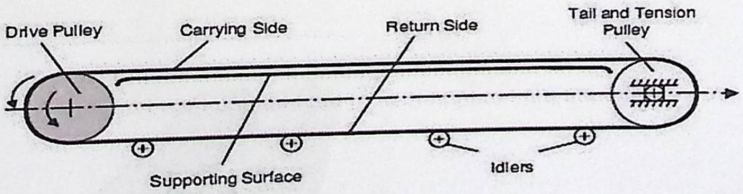


Figure3.14: Conveyor Constructor.

The pulley

Using pulley at the head and tail of conveyors to make movement duo to the friction between the belt and pulley connecting the conveyor using the motor where the motor connect to the shaft of pulley by sprocket gear.

The diameters of standard pulleys are: 200, 250, 315, 400, 500, 630, 800,1000, 1250, 1400 and 1600 mm.

The height of the crown is usually 0.5% of the pulley width, but not less than 4 mm. The pulley diameter Dp depends on the number of plies of belt and may be also be determined from the formula

$$D_p > Ki(mm) \ldots\ldots \text{Eq [3 -1]}.$$

Where

Dp : The pulley diameter

K: afactor depending on the number of plies (125-150).

i: number of plies in the conveyor belt .

In our design use Spun end curve crown pulley for the cleaning conveyor as shown in figure 3.15, with standard value 200 mm pulley diameter.

Figure 3.15: Spun end curve crown pulley.

Idler pulley

Idlers used in conveyor as supported metal that support the belt and make movement easy, providing low rolling resistance, suitable for high-speed belt conveyors used in long conveyor.

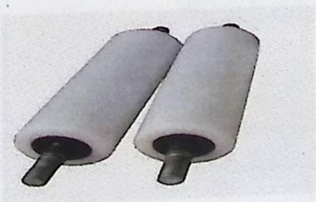


Figure 3.16: Idlers.

The supporting or slider surface can be made from various materials with different friction values e.g. steel, wood or synthetic.

The best selection depends on acquiring complete operating and environmental information. It is best to get this information immediately and before any calculations is attempted, this information should include:

1. Carrying Surface. State if load surface of belt is supported by flat or troughed idlers or type of flat slider bed surface. State angle of troughed idlers.
2. Drive Data.
3. Environment. Temperature, chemicals, oils, and any special conditions.
4. Height. Vertical difference of head and tail (terminal) pulleys, elevation
5. Length. Distance between head and tail pulley.
6. Loading Rate. Tons/hour.
7. Material. Type, temperature, size and percentage of maximum lumps.
8. Pulley Diameter. In addition to tail and head pulley, and carry idlers, any pulley that changes belt direction (identify each--include in system sketch).
9. Speed. (m./minute) of belt.
10. Take-up. Type (mechanical screw or automatic), location, and total amount of movement (included in sketch).
11. Width of Belt. (m) Include width of pulley face width, if known.

3.3.2 Eggs collecting system

The block diagram: the Eggs collecting system block diagram shown in Figure 3.17, It consist of conveyors beside the cages ,the system contain eight rows with five columns of cages , there is 40 eggs conveyors and one at the end of all these to collect all the eggs.

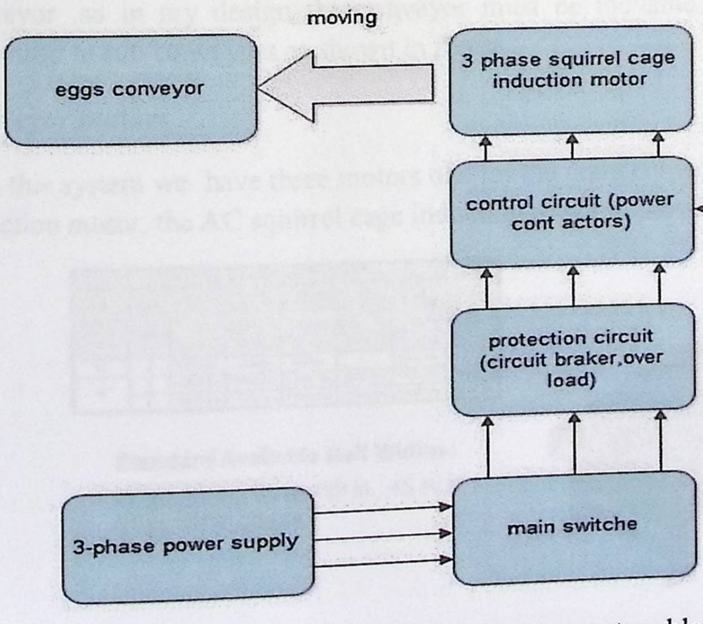


Figure 3.17: Eggs collecting system block diagram

General Description:

The cage designed to be with slop where Eggs roll at low speed, over a floating wire-mesh floor with slope about 10°, onto the collecting belts without crowding. In my designs need to collect the egg at specific location where The system consist of tow type of conveyors, first conveyor designed to be in front of the cage which the eggs collect there.

Figure3.18-a: Egg Conveyor

The eggs transport belt is made of woven polypropylene, all these conveyors spreaded in front of cages collects the egg on a main conveyor.

The main eggs conveyor is the type of roller without belt to increase the friction between the egg and conveyor , the space between the rod is small in order to save the eggs from fall or broken these type Generally used for transporting materials pieces with speed of 3.5 m/min.

Main conveyor will be movable up and down to be at the same surface row of sub conveyor .so in my design the conveyor must be movable to four level according to eggs collection in sub conveyors as shown in figure .

Conveyor motors

In this system we have three motors one for the main conveyor which is 3 phase squirrel cage induction motor, the AC squirrel cage induction motor to move the main conveyor up and down.

LOAD CAPACITY CHART

30 FPM Roller Speed

Diameter Drive Pulley	HP	B/F WIDTH 27" to 31"		B/F WIDTH 33" to 51"		B/F WIDTH 57" to 61"	
		50' OAL	100' OAL	50' OAL	100' OAL	50' OAL	100' OAL
12"	2	8000 lbs.	6000 lbs.	600 lbs.	--	5000 lbs	--
16"	3	15000 lbs.	12000 lbs.	12000 lbs.	8000 lbs.	12000 lbs.	6000 lbs.
	5	18000 lbs.	17000 lbs.	17000 lbs.	12000 lbs.	16000 lbs.	11000 lbs.

Standard Available Belt Widths:

27 in., 31 in., 33 in., 39 in., 45 in., 51 in., 57 in., 61 in.

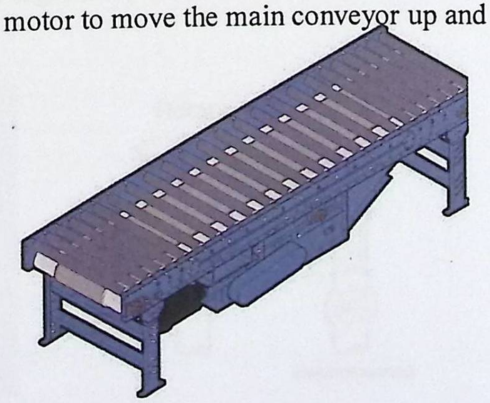


Figure3.18-b: Egg Conveyor

Need another motor to move the sub conveyor and collect the eggs to main one, Length=60m, Width=0.15 m, Speed=9m/min.

3.3.3 Feeding system

Block diagram: The system contain mainly main tank have the feed and then push the feed using pump throw a pipe which have screw inside it to make the move easily through the pipe. Figure3.19: presents the feeding system block diagram:

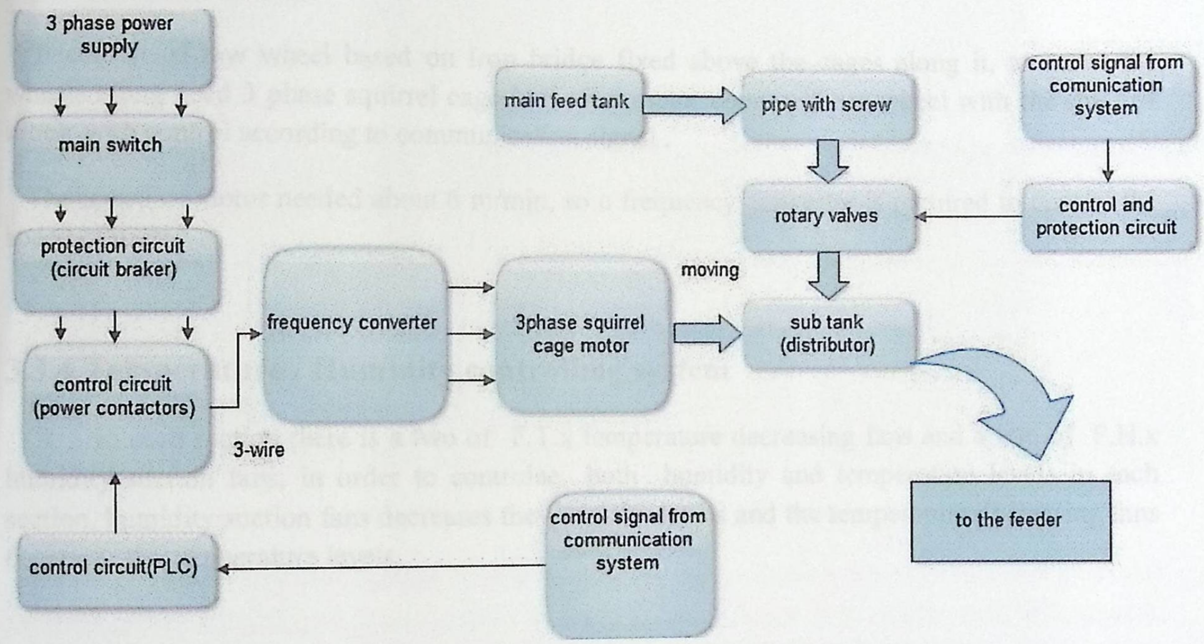


Figure 3.19: Feeding system block diagram

The valves system

The system contain 10 valves in the pipe to distribution the feed on the sub tank or the distributer, there are 10 distributer, every side of cages have one, and everyone one have valve above it as shown in figure3.20.

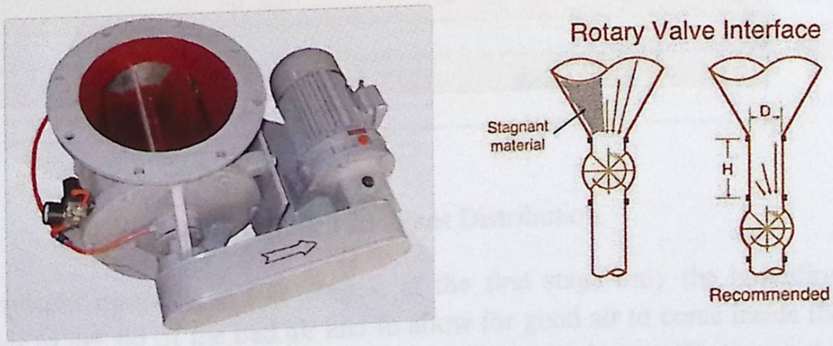


Figure 3.20: Valve Device.

When the control signal come from communication system to the control circuit of the valve motor, the Blades in valve body connects to the motor shaft and the material go through the distribution pipe to distributor.

Every sub tank or distributor distributes the feed to one Column in one side of cage. The distributer will be movable along the cage Back and forth.

The shape designed to allow the feed follow in the Feedlots easily and also to follow in four Feedlots at the same time.

It consist of tow wheel based on Iron bridge fixed above the cages along it, to move the wheels along need 3 phase squirrel cage induction motor connect these wheel with the shaft of motor with control according to communication signal .

The speed of motor needed about 6 m/min, so a frequency converter is required to control the speed of motor.

3.3.4 Temperature / Humidity controlling system

In each section there is a two of F.T.x temperature decreasing fans and a two of F.H.x humidity suction fans; in order to controlee both humidity and temperature levels in each section, humidity suction fans decreases the humidity levels and the temperature decreasing fans decreases the temperature levels.

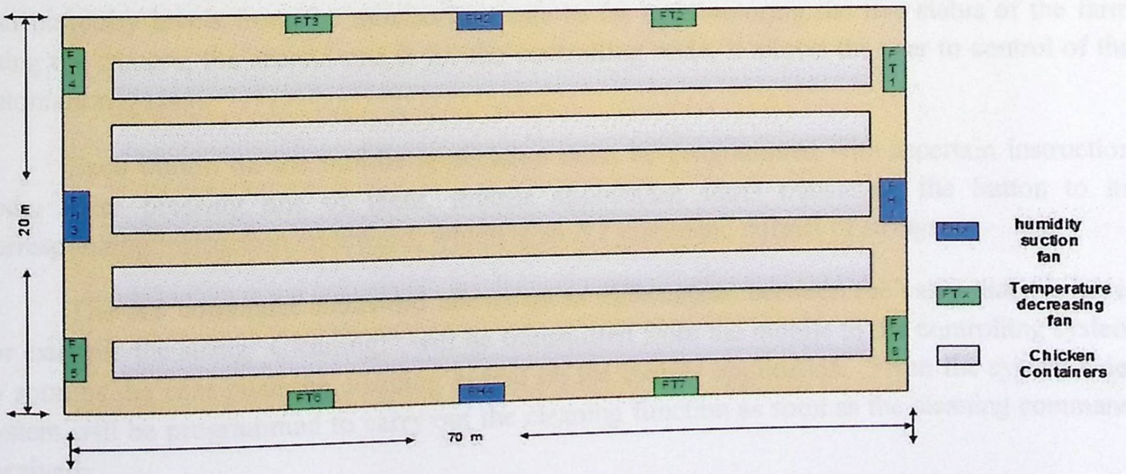


Figure 3.21: Fans Distribution.

This system operates as two stages: at the first stage only the humidity suction fans works; in order to get rid of the bad air and to allow for good air to come inside the farm, and to minimize the humidity level in the farm. Then, in the second stage the temperature decreasing fans operates; to minimize the temperature level in a short time after the humidity level been corrected.

3.4 Human interfacing system

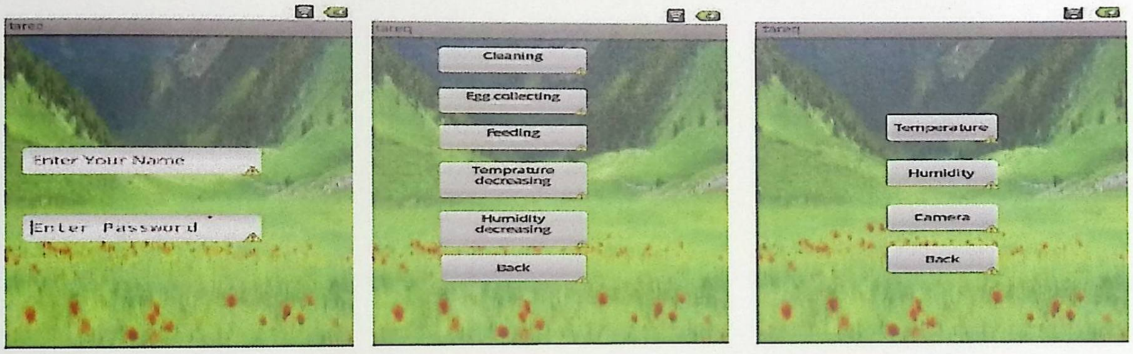


Figure 3.22: Human Interfacing System.

The human interface system as shown in figure 3.22, which is a mobile application based on the Android operating system contains two parts, the first one is for monitoring temperature and humidity levels from the monitoring node as far as monitoring the live status of the farm using the camera, the second one is for the controlling node, it allows the user to control of the automation system.

Each button on the mobile application must be programmed with ascertain instruction code. After pressing one of them, mobile application starts translating the button to its corresponding instruction codeword; which is an AT command consist of strings.

The AT command codeword allows us to differentiate between the cases that we have, for example the string" Cleaning!" will be transmitted from the mobile to the controlling system as soon as the user push the cleaning button on the mobile application. From the system side, system will be programmed to carry out the cleaning function as soon as the cleaning command received,

Table 3.1 illustrates the required AT command that are corresponding to the functions that the mobile application contains. Also, the monitoring and controlling system must be programmed based on these AT commands.

Signal	Function	Corresponding AT command
1	On/OFF Cleaning system.	"Cleaning!"

2	On/OFF Eggs Collecting system.	" Eggs!"
3	On/OFF Feeding system.	" Feed!"
4	On/OFF Decreasing Humidity system.	"Humidity!"
5	On/OFF Decreasing Temperature system.	" Tempretur!"
6	On/OFF Monitoring Camera system.	"ReedCamera!"
7	On/OFF Monitoring Humidity system.	"ReedHumidity!"
8	On/OFF Monitoring Temperature system.	"Reed tempretur!"

Table 3.1: Signals and the Corresponding Code.

4

Chapter FOUR

Detailed Design

4.1 Introduction

In this chapter we will describe the hardware and interfacing detailed design, showing how the connection between each components in this system.

4.2 Hardware design

4.2.1 Monitoring system

The monitoring system as we illustrated in chapter three, consist of three parameter that we want to monitor, temperature, humidity levels and the overall status of the farm.

Figure 4.1 presents the monitoring node components. It contains the humidity and temperature sensors in addition to the camera as the terminal device. And in order to control and connect these terminals together we need to a connection system, and that can be achieved if we used some kind of microcontroller device.

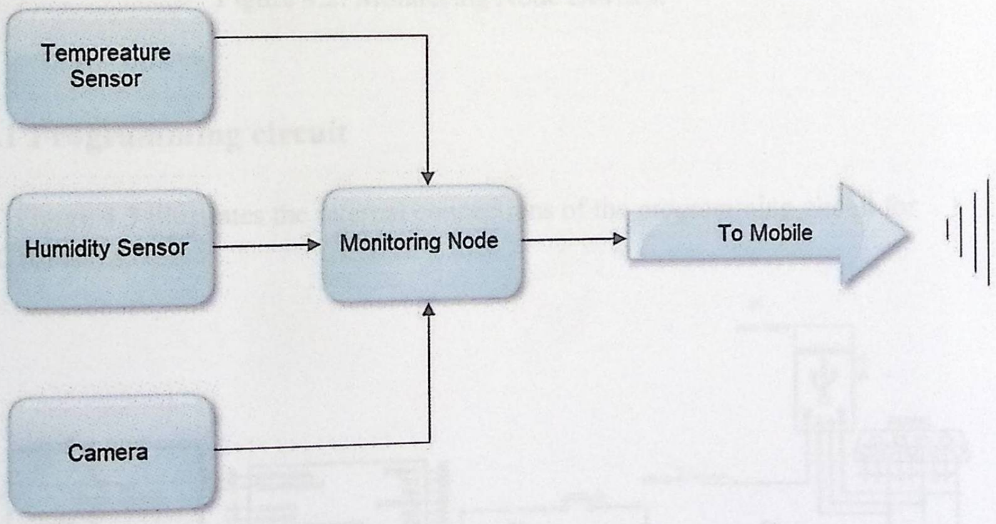


Figure 4.1: Monitoring node components.

The microcontroller that we used in this node is the PIC16F877a; duo to its high capability to receive large number of deferent signal from multiple device such as sensors and other terminals. Also this kind of microcontroller can be programmed easily using a small simple circuit that we can implement using available electronics, which will offered us flexibility, save our time and our effort to compile and test our cod on the PIC.

The last component in the monitoring node is the wireless communication device, it responsible of the communication process between the monitoring system and the mobile station, and figure 4.2 next page represent the main components in the monitoring node.

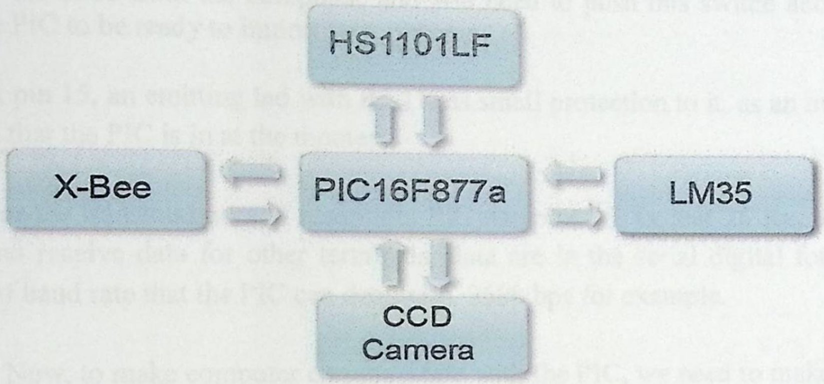


Figure 4.2: Monitoring Node Devises.

4.2.1.1 Programming circuit

Figure 4.3 illustrates the internal connections of the programming circuit for the PIC16F877a.

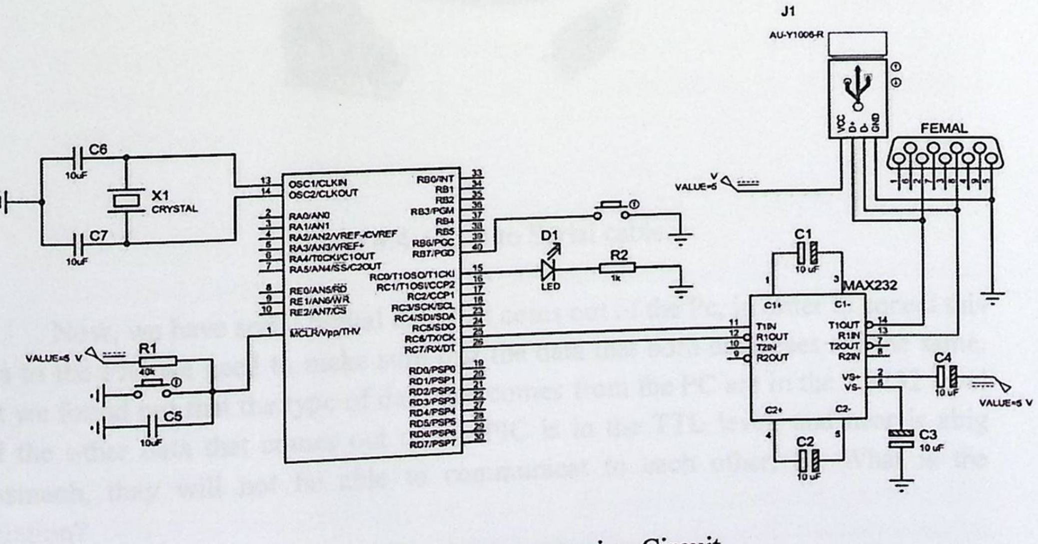


Figure 4.3: PIC16F877a Programming Circuit.

The first thing that this PIC required is the oscillation circuit, crystal of 4MHz and another two of 10nF filtration capacitors, connected to pins 13 and 14 as in figure 4.3. Now in order to rest the PIC, the MCLR active low pin1 is connected to switch and other capacitor C5 -to filter the signal that comes from the switch- as only one pulse not more- and an protection resistor R1 to the a 5 voltage supply.

At pin 40, there is another switch, the usage of this one is to make PIC ready to receive the code from the computer, and you need to push this switch about 10s so that the PIC to be ready to import your program.

At pin 15, an emitting led with R2 1K as small protection to it, as an indicator to the stat that the PIC is in at the moment.

For the transmission and reception, PIC use pins 25 Tx and 26 Rx, in order to send and receive data for other terminals, data are in the serial digital form with a range of baud rate that the PIC can deal with, 9600 bps for example.

Now, to make computer communicate with the PIC, we need to make sure that the two devices understand each other, understanding mean the same baud rate and the same level of voltage for both, we can use the Universal Serial Bus USB to connect the Pc Via a USB mail connector, but there is no USB in the PIC side; So we can use the USB to Serial cable figure 4.3 in the next page to overcome this problem.

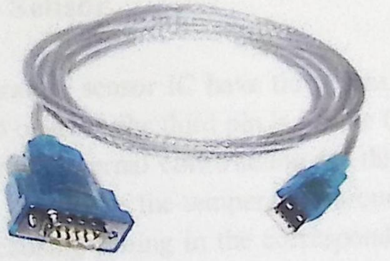


Figure 4.4 : USB to Serial cable.

Now, we have serial digital data that coms out of the Pc, in order to conect this data to the PIC we need to make sure that the data that both deviceces are the same. But we found out that the type of data that comes from the PC are in the RS232 level and the other data that comes out of the PIC is in the TTL level, and hear is abig missmach, thay will not be able to communicat to each other, So What is the soluation?

After asmall diging we found out that the MaxIC can do conversion from the TTL level to the RS232 level and vise versia.

At pin 40, there is another switch, the usage of this one is to make PIC ready to receive the code from the computer, and you need to push this switch about 10s so that the PIC to be ready to import your program.

At pin 15, an emitting led with R2 1K as small protection to it, as an indicator to the stat that the PIC is in at the moment.

Now, for the transition and reception, PIC use pins 25 Tx and 26 Rx, in order to send and receive data for other terminals. And the connection of this Ic is demonstrated in figure 4.3, the pin 12 Tout of the MAX232 must be connected to pin 26Rx on the PIC and pin 11Tin on the MAX232 to the pin25 TX so that the data out from the PIC are entered the MAX and the data out of the max entered the PIC.

Finely, in the programming circuit we need to make sure that the PIC and the MAX are in the same voltage level, and that can be accrued if we connect the PIC to +5 volt to both VDD at pins 11 and 32, and the 0 volt (GND) at both VSS pin 11 and pin 31 according to the datasheet of the PIC16F877a.

4.2.1.2 LM35 and HS1101LF interfacing

4.2.1.2.1 LM35 Sensor

The LM35, temperature sensor IC have three pins, pin one is for the voltage supply +5 volts in order to operate, the third pin is for the GND. The middle one pin 2 is for output signal, the main internal construction for this sensor is a grope of heat transistor, after feeding it in voltage the temperature around it affect on the behavior of the internal transistor causing aching in the corresponding output voltage at pin 2 according to the ratio:

$$1\ ^{\circ}C \text{ ------------- } 10\ mV \quad \ldots\ldots\ldots\ldots\ldots (4.1)$$

The output signal of the LM35 is in the analog form, so PIC16F877a can deal with it as an analog input signal without any problem, except that the maximum amplitude of this signal must not exceed the saturation level of the input analog pin in the microcontroller and which is around 12Vp-p.

PIC16f877a can reed eight deferent analog signals via eight pins from AN0 to AN1, also this PIC can carry out the analog to digital conversion (ADC), this conversion results in a corresponding 10-bit digital number. The A/D module has high and low-voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3. The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D clock must be derived from the A/D's internal RC oscillator.

Figure 4.5 represent how we can connect the LM35 temperature sensor to PIC16F877a, the pin 2 from the sensor to pin ANO at the PIC, and for the (ADC) we can esiely do it from the software part in the PIC program.

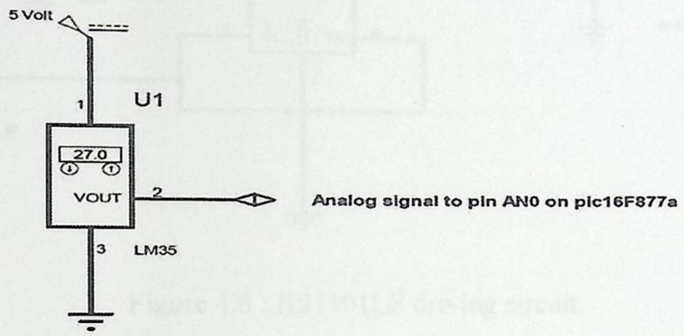


Figure 4.5 : LM35 IC Connections.

4.2.1.2.2 HS1101LF sensor

The HS1101LF humidity sensor is mainly a capacitor, its capacitance change due to the change in the permittivity of the material between its plats. And we all know that the humidity is the amount of water vapor in the air, now in the normal ideal mode there is no humidity so there is nothing between the two plats of the capacitor except the free space and it has the permittivity eqall, but in reality, humidity exist that's mean there is an amount of water vapor in air or between the two plat of the capacitor sensor. That will lead to a corresponding changing in the amount of the capacitance in the variable capacitor sensor.

But here we need to express this changing in the capacitance as changing in some electrical quantity like voltage or current or any other electrical ones. However; fortunately our sensor com from its company with its own driving circuit, in order to convert the changing in capacitance as a changing in frequency, this circuit is shown in figure 4.6.

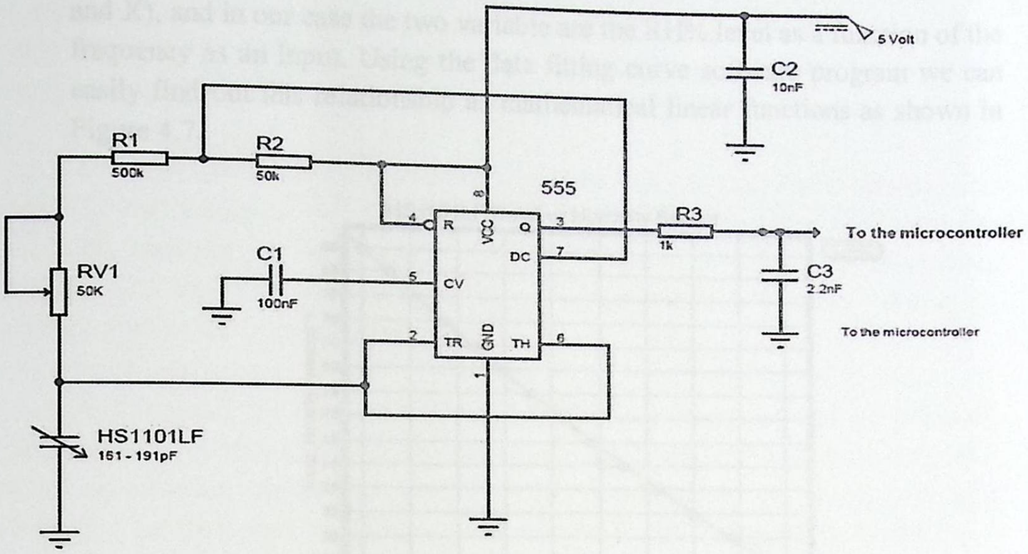


Figure 4.6 : HS1101LF driving circuit.

The main function of this circiut is to expres the changing in the capacitance in the humidity vareable capacitor sensor in terms of frquancy. In avery simpel basic explenation, the 555 timer IC charged the HS1101LF sensor throw R2 and R3 from the DC and VCC pin 7 and 8, after that this timer receives the discharge signal from the sensor through pin 2(TR), then the timer 555 put out the signal on pin 3.

From other hand; the company gives the following table[] which represents an eighteen measurement for frequency with its own crossholding RH% level, that's mean if we have the frequency we will get the RH% level and that only for an eighteen cases. So the reading that we will have will not be acierate. So how we will overcome this problem?

# of Reading	RH%	F in KHz
1	10	7.155
2	15	7.080
3	20	7.010
4	25	6.945
5	30	6.880
.	.	.
.	.	.
18	95	6.210

Table 4.1:Typical Response Look-Up Table -Humidity Output

To overcome this problem; shall use the numerical analysis, interpolation particularly. To find out the relationship between two variable(Y and X), and in our case the two variable are the RH% level as a function of the frequency as an input. Using the data fitting curve software program we can easily find out this relationship as mathematical linear functions as shown in Figure 4.7.

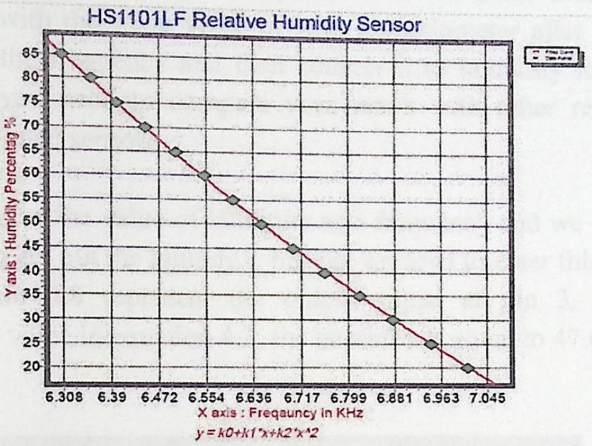


Figure 4.7: HS1101LF Frequancy-RH% Level curve.

The relationship between the frequency at X-axis and the humidity at Y-axis is almost negative liner. According to the next equation

$$Y_{(RH\% \; at \; X \; frequance \; in \; KHz)} = K_0 + K_1 * X + K_2 * X^2 \; .... (4.2)$$

Were $Y : RH\%.$
$X : frequancy \; in \; KHz.$
$K_0 = 1738.41353819.$
$K_1 = -415.21980003.$
$K_2 = 24.26410259.$

Note that this function is from the second order, also we can find other linear functions with higher order to be more accurate, but if we chose a function with order three and higher it will form to us a big problem in the time consuming in PIC16F877a since it doesn't contain a hardware multiplayer. It carries out the mathematical multiplication operation as an software iteration and that will led to a large time operation which could led the PIC to waste a large time and that eventually affect on the stability of the PIC, also the second order linear equation is very close to the higher order ones with a small error less the 0. 01%.

Yet, the HS1101LF sensor needs to be calibrated, the calibration method is not complicated, first thing is to measure the humidity with other sensor in a specific place and convert it to frequency, and then you need to measure out the frequency at the output of the HS1101LF sensor driver circuit at pin 3, after that you need to change the value of the potentiometer tell you get the same frequency that you were convert it from the reading of the first sensor, mark out the value of potentiometer.

To make sure that your calibration exact, you need to take a new measurement using HS1101LS sensor with the same value of your potentiometer after your first calibration, measure out the frequency and then convert it to humidity level using equation 4.2. And then you need to compare your result with other reading for humidity from other celebrated sensor.

At this point, we have the value of humidity as a frequency and we can easily substitute the frequency to get out the humidity, but yet we need to enter this signal to the microcontroller, figure 4.8 represent the output signal at pin 3, and after substituting the frequency value in equation 4.2, the humidity is equal to 47.650 % .

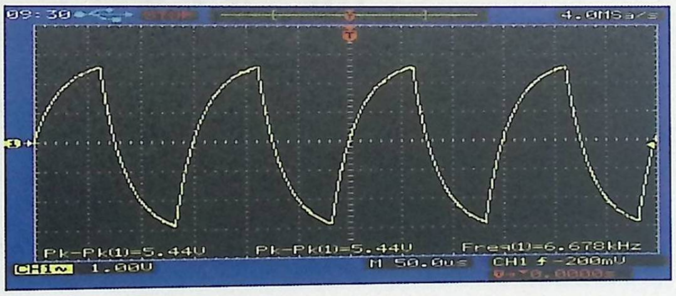


Figure 4.8 : HS1101LF Driver Circuit Output

Later on, we need to enter this signal to the PIC16F877a, and hear we gave to option; the first one is to enter the signal via CAPTURE/COMPARE/ (CCP) at pins 16 or 17 on the pic16F877a and we can measure the frequency, either that or we can convert this signal to square wave and the enter it as a digital input to the PIC at the digital input pins and then we can set a timer 0 for example at the moment that one pulse is coming in, and then we can measure the time to the second pulse to come, hear we have the half time of the duty cycle, times two and one over your answer you will get your frequency in your little microcontroller, with a little mathematical manipulation inside your PIC you will have the value of the humidity that your sensor measure.

So, to be accurate enough, we shall use the second method, converting our signal to a square wave signal and to be entered to the PIC. Yet the PIC16F877a can read the time between the pulses, and our signal is in the range of 6 to 7 KHZ and that's equal to 1.67 to 1.43 ms. And at this point, from the practical aspect we discovered that this kind of PIC have no stability or in other word no enough sensitivity in measuring time, after 3 KHZ and our measurement will be wrong, t o solve this problem our engineering thinking starts from the fact we need to maximize the range of time, so that the timers inside the PIC can read it, and this can be happened by minimizing the frequency at least to forth of the original one, to become at the maximum of 1.8 KHz and hear the PIC don't have any issues with reading it.

Yet, how we could reduce the frequency? This is our mater now, but in fact we can use tow JK flip-flop to do so, as figure 4.9 which represent the circuit connection for the frequency divider system that we designed using the Proteus software program.

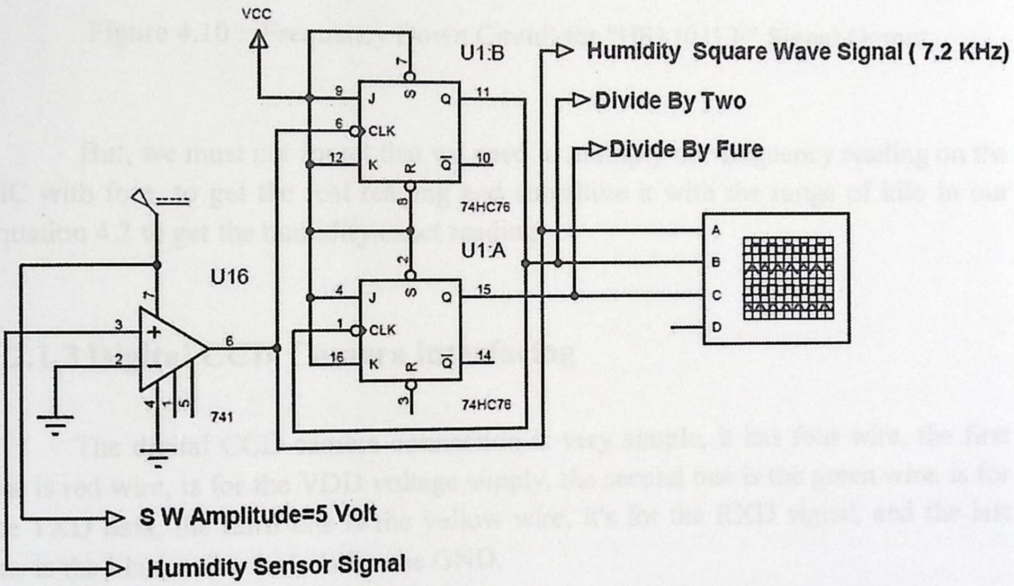


Figure 4.9: Frequancy Down Circuit for HS1101LF signal.

In the first stage the LM741 Op-Amp is acting as a comparator. When there is a positive signal level it gives the 5 volt as an output on pin 6, and if the input signal is in the negative, the output of the comparator will be equal to zero and by this way we make sure that our humidity signal is a square wave signal.

The aim of exciting a two JK flip-flop is to divide the frequency two times. In order to get our maximum frequency lowest than 3 KHz, So that our smart PIC can deal with it without any barriers, and the output of the frequency divider circuit is illustrated in figure 4.10 in the next page; were channel one - yellow - is the input humidity signal after the comparator, channel two - Blue - represent the signal after we minimize its frequency to half, and channel three - red - is the output signal after we minimize its frequency to the fourth.

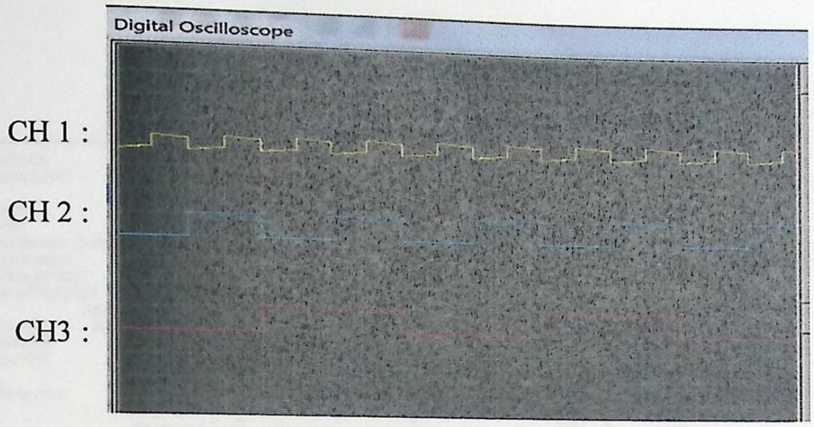


Figure 4.10 : Frequancy Down Circuit for "HS1101LF" Signal Output.

But, we must not forget that we need to multiply our frequency reading on the PIC with fore, to get the real reading and substitute it with the range of kilo in our equation 4.2 to get the humidity exact reading.

4.2.1.3 Digital CCD Camera interfacing

The digital CCD camera connection is very simple, it has four wire, the first one is red wire, is for the VDD voltage supply, the second one is the green wire, is for the TXD data, the third one is the yellow wire, it's for the RXD signal, and the last one is the black wire, and it's for the GND.

Figure 4.11 in the next page, shows how the CCD camera is connected to the PIC 16F877a, the TXD and RXD on the camera are serial data with 5volt reference, so we need to connect it to pins with the serial type on the PIC, but the PIC 16F877a contains only one TX and another RX that comes previously defined from the company, but from the programming said we can set any two pins as an TX an RX, choosing for example pins 37 and 38 to do so.

This kind of ZM CCD serial camera has its own interface to the standard of RS232/485 and the TTL input /output signal and can connect it to Pc or any other device through one of its own serial interfaces.

So, our connection will be throw the RS232, using the MAX232 IC because its already used on our programming circuit, after connecting the CCD camera to the PIC 16F877a, the PIC will be as the host terminal, it controls the camera from many sides; the time to take picture or the size of it or the resolution of it and the baud rate.

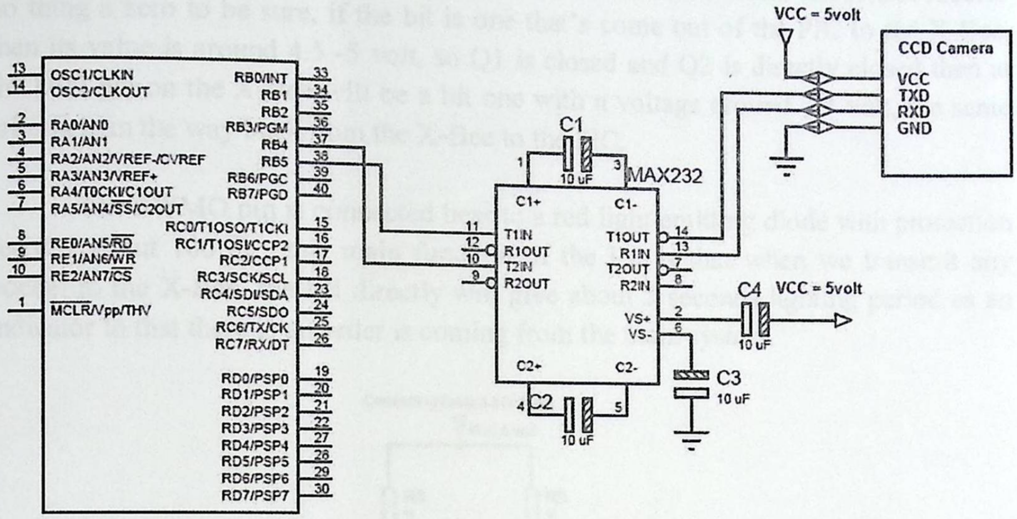


Figure 4.11 : CCD Camera Connection to PIC16F877a.

4.2.1.4 X-Bee Wi-Fi interfacing

The monitoring node need to communicate with the main system; in order to start working and transmitting data and measurement that the manager asked for from the mobile station, that can be done using one of wireless communication system like X-Bee.

Therefore; hear in the monitoring node the X-Bee represent the communication path between the manager and the farm, the data that the X-Bee transmit and receive are digital serial data, and we also know that the PIC16F877a has its own serial digital data pins, so it's very simple to connect the X-Bee to the PIC through the pins DOUT and DIN from the X-Bee side to pins RX and TX from the PIC side.

Yes, it seems very easy, but a new obstacle just shows up, thing that we discovered that we were from the beginning connecting our PIC to a 5 volt power supply and our little X-Bee were connected to 3.3 volt, that's mean the level of the data transferred between the two devices are deferent, we need to make a matching circuit so that the component can understand each other. The matching circuit is shown and cleared in figure 4.12.

Figure 4.12 represent the matching circuit that we designed to overcome the mismatching problem between the PIC and the X-Bee, in a very simple explanation to

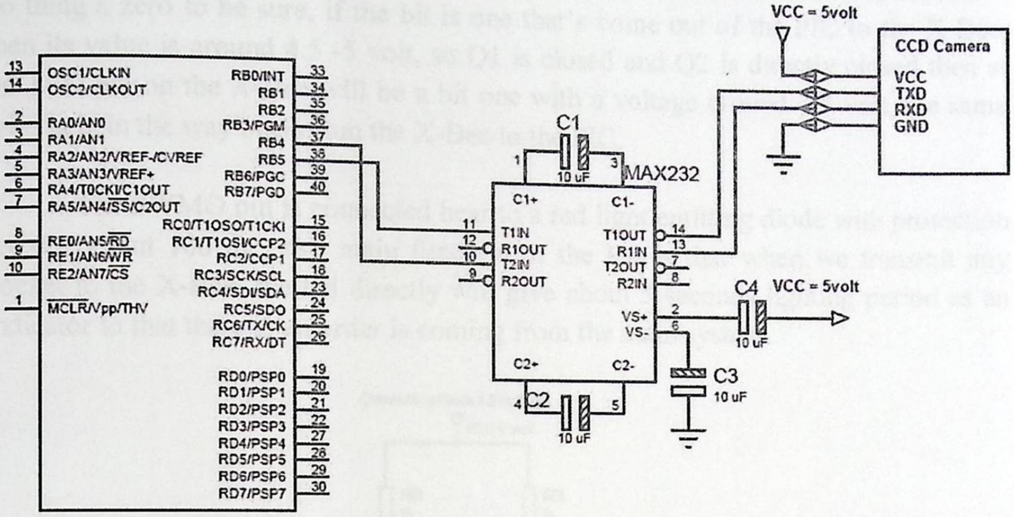


Figure 4.11 : CCD Camera Connection to PIC16F877a.

4.2.1.4 X-Bee Wi-Fi interfacing

The monitoring node need to communicate with the main system; in order to start working and transmitting data and measurement that the manager asked for from the mobile station, that can be done using one of wireless communication system like X-Bee.

Therefore; hear in the monitoring node the X-Bee represent the communication path between the manager and the farm, the data that the X-Bee transmit and receive are digital serial data, and we also know that the PIC16F877a has its own serial digital data pins, so it's very simple to connect the X-Bee to the PIC through the pins DOUT and DIN from the X-Bee side to pins RX and TX from the PIC side.

Yes, it seems very easy, but a new obstacle just shows up, thing that we discovered that we were from the beginning connecting our PIC to a 5 volt power supply and our little X-Bee were connected to 3.3 volt, that's mean the level of the data transferred between the two devices are deferent, we need to make a matching circuit so that the component can understand each other. The matching circuit is shown and cleared in figure 4.12.

Figure 4.12 represent the matching circuit that we designed to overcome the mismatching problem between the PIC and the X-Bee, in a very simple explanation to how this circuit work, the data that comes out of the PIC to the X-Bee if there is bit zero that's mean no voltage so Q1 doesn't work on. And that means the X-Bee receive no thing a zero to be sure, if the bit is one that's come out of the PIC to the X-Bee, then its value is around 4.5 -5 volt, so Q1 is closed and Q2 is directly closed then at the DIN pin on the X-Bee will be a bit one with a voltage around 3.3 volt, the same criteria is in the way backroom the X-Bee to the PIC.

The PWMO pin is connected hear to a red light emitting diode with protection resistor about 100 Ω , this main function of the led is that when we transmit any pocket to the X-Bee, the led directly will give about 5 seconds lighting period as an indicator to that there is an order is coming from the main system.

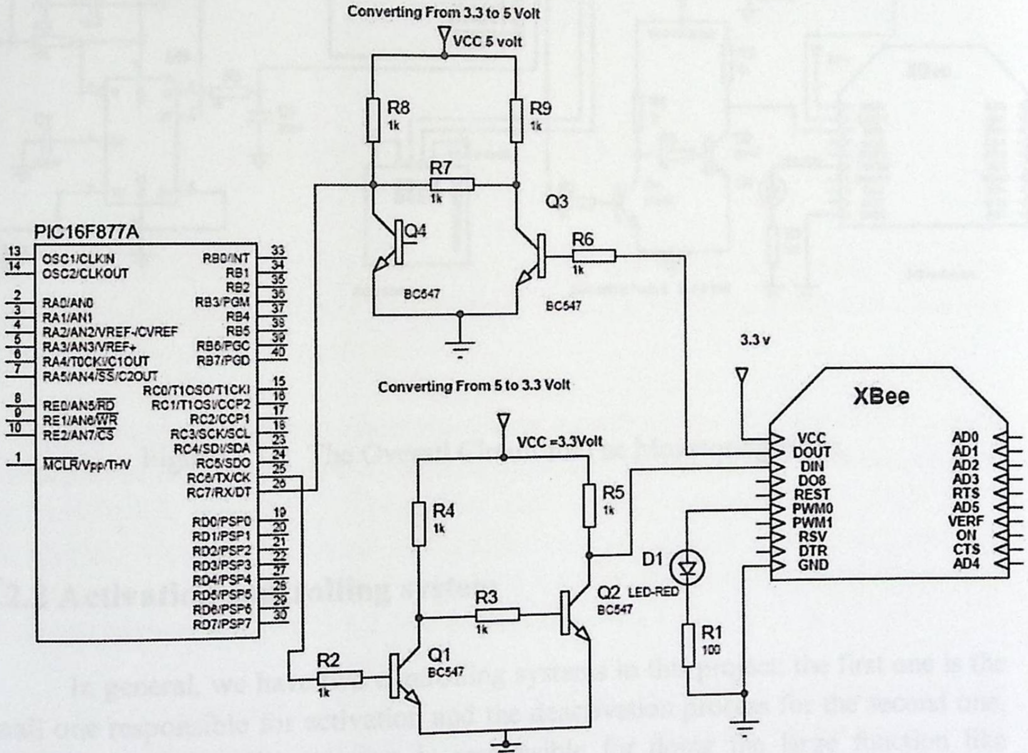


Figure 4.12: PIC-X-Bee matching circuit.

The next figure 4.12 shows the overall component in the monitoring node and its internal connection.

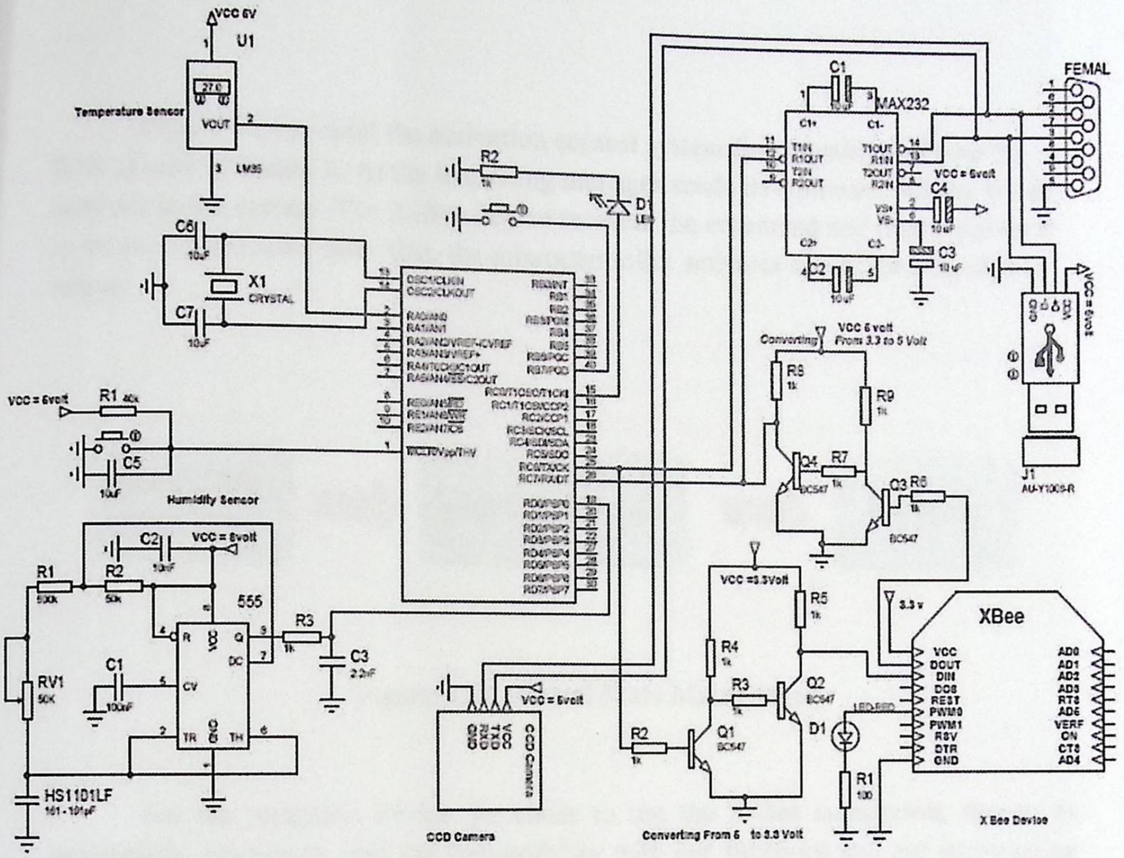


Figure 4.13: The Overall Circuit forThe Monetoring Node.

4.2.2 Activation controlling system

In general, we have two controlling systems in this project; the first one is the small one responsible for activation and the deactivation process for the second one, and the other controlling system is responsible for doing the large function like feeding, cleaning and so on.

At this point, we shall start building the activation control system, this controlling system is the connector between the human interfacing system and the real practical system. So we need an electronic component to receive command signals from the human interfacing system, and that requires some kind of microcontrollers to decide the corresponding related signal to be initiated. Now we have our chosen signal but how can we transfer it to the main controlling system?

The answer cannot be simpler from this, using the electrical relays, we can simply send our signal to activate and deactivate the main controlling system.

Figure 4.14 present the activation control system functionality and how the flow of control inside it. At the beginning manager sends its command via the Wi-Fi network to the system. The X-Bee device receives the command and then it passes it to the microcontroller after that, the microcontroller activates one of the responsible relays.

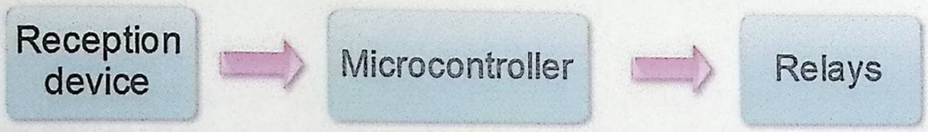


Figure 4.14 Control Node Main Stages.

For the reception device we chose to use the X-Bee component, due to its availability, simplicity, and the compatibility with our functions and our surrounding environment from the distance side.

Now, for the microcontroller part, hear we will use the PIC16F688 as the mind of this part of the system, that we already illustrated its characteristics in chapter two, this kind of PIC is extremely cheap, can receive serial data from the terminal and it has fourteen pin, and that comes along with our needs to have a microcontroller that can read serial data from the X-Bee and it has at least other five pins; in order to initiate deferent five signal to do deferent five functions.

The last part of the activation control system is the Relays. It's responsible for starting or ending any function that the manager wants to do, it's the output o the activation control system to be as an input to the main control system, figure 4.14 shows the containing devices in the activation control system.

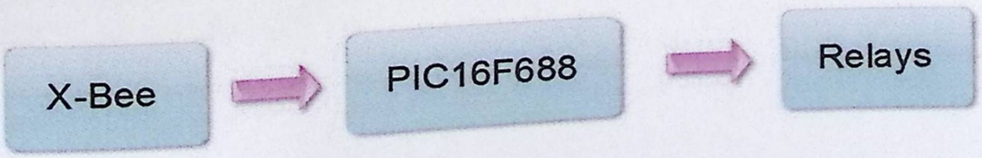


Figure 4.15 Activation Control system Components.

4.2.2.1 X-Bee Wi-Fi interfacing

Now, from the connections sides, first thing is to connect the X-Bee to the PIC16F688, note that both components operates on the same voltage level, so we don't have to do a matching circuit between them figure 4.16 shows how the things goes.

Pins connection as the fallowing, DOUT pin from the X-Bee is to the RX PIN on the PIC, pin DIN on the X-Bee to the TX on the PIC, but why we do as this? The answer is because sometimes the PIC sends feedback that the data was received correctly or not.

LED D1 with its own protection resistor indicates that there is a packet arrived from the transmitter.

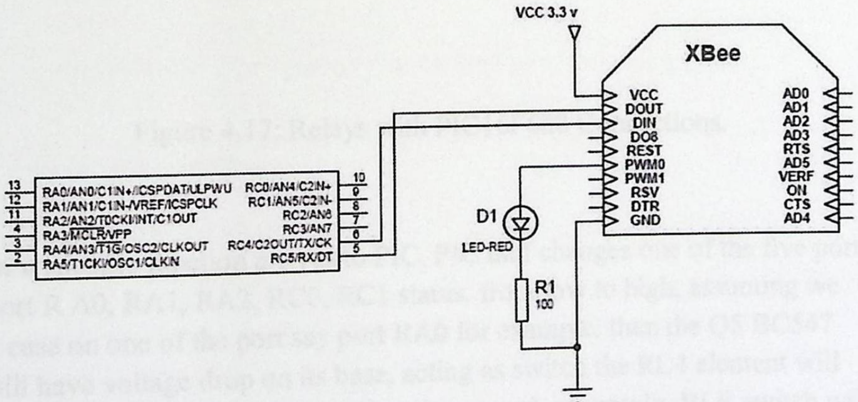


Figure 4.16: X-Bee with PIC16F688 Connections.

4.2.2.2 Relays connections

As we illustrated previously, we need relays to transfer the PIC signal to the main automation system to do the several functions that we talked about like feeding, eggs collecting and so on, figure 4.17 presents how the connections between the PIC16F688 and the relays are.

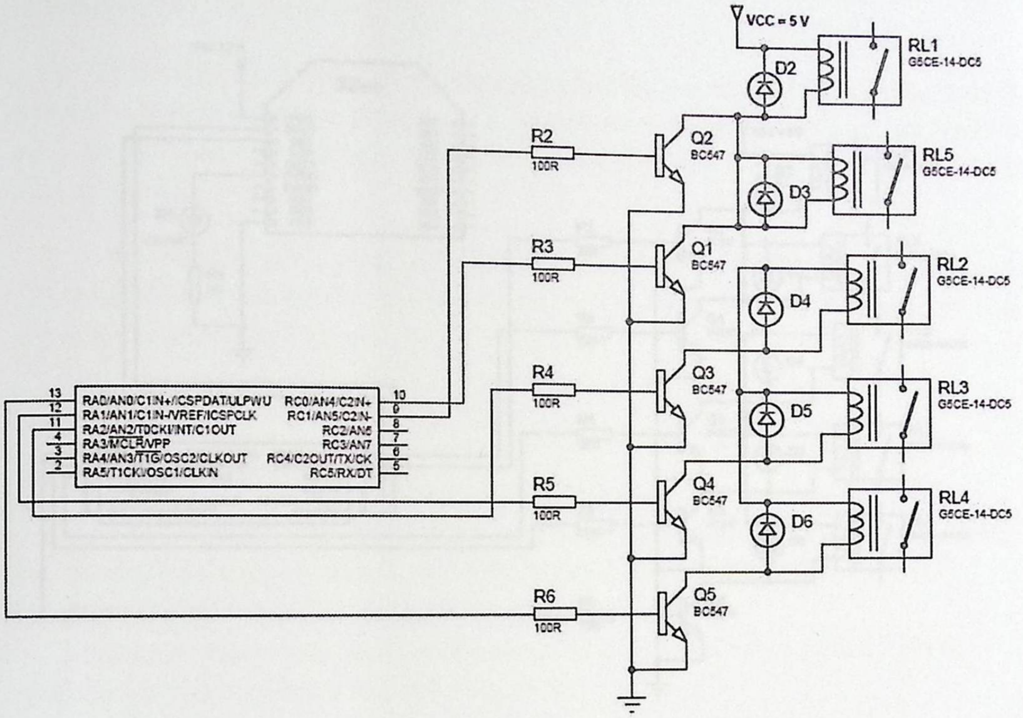


Figure 4.17: Relays with PIC16F688 Connections.

After a specific function arrives to PIC, PIC unit changes one of the five port pins from port R A0, RA1, RA2, RC0, RC1 status, from low to high, assuming we have a high case on one of the port say port RA0 for example, then the Q5 BC547 transistor will have voltage drop on its base, acting as switch the RL4 element will experience a 5 voltage drop from the VCC to the ground, eventually RLE switch will close, and from there we have our automatic remotely switch that can be used for any sub system you want to turn it on or off.

As you can see in the circuit, we connect the light emitting diode by the reverse bias on the terminal of the relays, and that's because when the electrical signal go off, a reveres current will appear due to the magnetic flux inside the coil, according to Linz low, so the LED come in handy to prevent this type of current from back to PIC and damage it, by transferring it to light.

In the next page, we present the full circuit diagram for the overall activation controlling system, figure 4.18 shows that clearly.

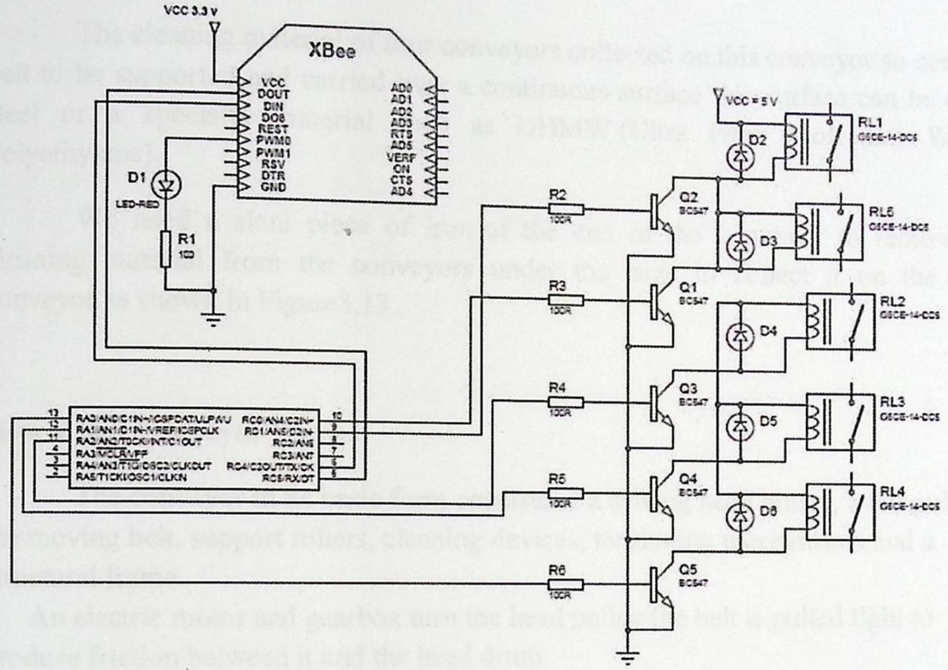


Figure 4.18: The Overall Circuit forThe Controlling Node.

4.2.3 Cleaning system design

4.2.3.1 General description

It consist of conveyors below the cages, the system contain four rows with five Columns of cages, there’s 20 cleaning conveyors and one at the end of all these to collect all the cleaning material.

4.2.3.2 Conveyors design

Using roller bed belt conveyors type, which has the carrier element handling belt conveyor systems it’s the most common transport machines in farming and industrial areas, where the belt is supported and carried by rollers because the cleaning material can be moist, so need the belt to be supported and carried over a metal.

Roller bed belt conveyor used rather than slider bed belt because we have a large area of conveyor so need a large area of supported metal of steal or sometime Wood.

At the end of each these conveyors Use the carrier element handling belt conveyor systems, in our design use slider bed belt conveyors because the cleaning material can be moist and wet.

The cleaning material of four conveyors collected on this conveyor so need the belt to be supported and carried over a continuous surface this surface can be either steel or a specialty material such as UHMW (Ultra High Molecular Weight Polyethylene).

We need a slant piece of iron at the end of the conveyor to remove the cleaning material from the conveyors under the cage to collect it on the main conveyor as shown in Figure3.13.

How a Belt Conveyor Works

The conveyor in its basic form consists of a driving head pulley, a tail pulley, the moving belt, support rollers, cleaning devices, tensioning mechanisms and a structural frame.

An electric motor and gearbox turn the head pulley the belt is pulled tight to produce friction between it and the head drum.

Only friction is used to drive the belt. If the friction falls the belt will slip or stop moving even though the head pulley keeps turning.

The pulley

Using pulley at the head and tail of conveyors to make movement because the friction between the belt and pulley make the movement and connect the conveyor using the motor where the motor connect to the shaft of pulley by sprocket gear.

In our design use Spun end curve crown pulley for the cleaning conveyor, with standard value 200 mm pulley diameter.

Pulleys shafts are made from carbon steel/EN-8 and are designed for torque and bending moment.

Idler pulley

Idlers used in conveyor as supported metal that support the belt and Mack movement easy. Providing low rolling resistance, suitable for high-speed belt conveyors used in long conveyor.

The supporting or slider surface can be made from various materials with different friction values e.g. steel, Galvanized Iron, wood or synthetic.

Conveyor belt material

The belt consists of one or more layers of material. They can be made out of rubber .

Many belts in general material handling have two layers. An under layer of material to provide linear strength and shape and an over layer called the cover. The under layer is often a cotton or plastic. The cover is often various rubber or plastic compounds specified by use of the belt.

4.2.3.3 The motor design:

Using 3 phase squirrel cage induction machine for each four conveyor

Horse power calculation [22]

$$T_E = T_x + T_y \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4.3).$$

Where:

T_E: *the effective tension of the belt*

T_x: *the tension component of the empty belt or the force to move it empty* .

T_y: *the tension or force required to move the load over the conveyor length*

$$T_x = 9.81 p L_1 F_B \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4.4).$$

Where:

p: *the weight of moving part* $(\frac{kg}{m})$

L_1: *the conveyor length*(m).

F_B: *friction coefficients for empty belt*

The weight of moving parts its value is to be taken from catalogue of companies.

The friction coefficient for empty belt generally taken to be around 0.03

$$T_Y = 9.81 L_1 F_L M \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4.5).$$

Where:

M: *the weight of material load* $\left(\frac{N}{m}\right)$.

F_L: *friction coefficients for load.*

The friction coefficient for the load is generally taken to be around 0.03

The weight of material load in kg/m is given by:

$$M = \frac{Q\left(\frac{ton}{h}\right)}{V\left(\frac{m}{s}\right)} * \frac{1000}{360} = \frac{Q}{0.36V}\frac{N}{m} \quad \text{..................................} (4.6).$$

Where

Q: *the conveyor capacity ton/h.*

V: *bels speed* $\frac{m}{s}$

The power required to drive the system:

$$power(KW) = \frac{T_E(N)*V\left(\frac{m}{s}\right)}{1000} \quad \text{..................................} (4.7).$$

Before calculate the motor power determined the cleaning conveyor data:

Cleaning conveyor under the cages calculation

Speed of conveyor =5m/s.

Length of belt = 60m

Width=2m.

The conveyor consists of tow pulleys and belt of EP Rubber material.

The weight of the belt part is given in MRF (Madras Rubber Factory) catalog as shown in the appendix A the nominal cascade weight =3.54Kg/$\mathbf{m^2}$.

$$\text{The area of one side of belt } = \text{width} * \text{length of belt} = 2\text{m} * 60\text{m} = 120\ m^2$$

$$the\ total\ area\ of\ belt = 2 * 120m^2 = 240m^2$$

$$\text{The weight of belt } = 240m^2 * \frac{3.54Kg}{m^2} = 849.6\ kg$$

The pulley dimension of eggs conveyors:

Diameter=0.2m.

Length=2m.

Using hollow cylinder with shaft at the center.

Inner radius=15cm.

Outer radius=20cm.

Shaft radius=5cm.

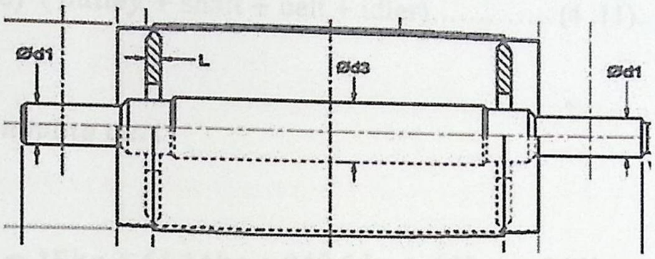


Figure 4.19: Pulley shape.

$V = \pi * h * (R^2 - r^2)$.................................... (4 .8) .

Where:

V : The pulley volume.

h: width of pulley

R: outer raduis.

r: inner raduis.

The shaft is from carbon steel with diameter of 3cm consider as cylinder

Shaft volume:

$$V = \pi * h * \frac{D^2}{4} \text{.................................. (4 .9).}$$

$$= 3.14 * 2 * \frac{0.05^2}{4} = 0.003925 m^3$$

$$density\ of\ carbon\ steel(\rho) = \frac{7.84gm}{cm^3} = 7840Kg/m^3$$

$$\text{Weight} = \rho * v \text{.................................. (4 .10).}$$

From eq(4 .10)

$$\text{Weight} = 7840 * 0.003925 = 30.7kg$$

Use every 6 m one idler, we need 10 idler up and 10 down the conveyor .

idler pulley weight $=$ number of idler $*$ idler weight.

$$= 20 * 2\text{kg} = 40\text{kg per conveyor}$$

There's tow symmetry shaft

$$\text{the weight} = 30.7 * 2 = 61.4\text{kg}$$

$MW(Kg) = weight\ of\ (pulley + \text{shaft} + \text{belt} + \text{idler})$............ (4 .11).

Where:

MW: *the weight of moving part*

From eq (4 -11)

$$MW(Kg) = 15kg + 61.14kg + 849.6\ kg + 40kg\ = 966kg$$

$MW\left(\frac{Kg}{m}\right) = \frac{MW}{L}$..(4 .12).

$$MW\left(\frac{Kg}{m}\right) = \frac{966kg}{60m} = 16.1\left(\frac{Kg}{m}\right)$$

P=16.1 $\left(\frac{Kg}{m}\right)$

$F_B = 0.03$

$F_L = 0.03$

The operation time of 30 min every day and the maximum total cleaning material =360Kg [1]

$$Q = \frac{360/1000}{30/60} = 0.72\ ton/h$$

From eq (4 .6)

$$M = \frac{Q\left(\frac{ton}{h}\right)}{V\left(\frac{m}{s}\right)} * \frac{1000}{360} = \frac{0.72 * 1000}{5 * 360} = 0.4\frac{N}{m}$$

From eq (4 .4)

$$T_x = 9.81pL_1F_B = 9.81 * 16.1 * 60 * 0.03 = 284.293\ N$$

These value taken from real farm and is maximum value [1]

From eq (4 -5).

$$T_Y = 9.81 L_1 F_L M = 9.81 * 60 * 0.03 * 0.4 = 7.0632N$$

From eq (4 .3)

$$T_E = T_x + T_y = 284.293 + 7.0632 = 291.356N$$

From eq (4 .7)

$$power(KW) = \frac{T_E(N) * V\left(\frac{m}{s}\right)}{1000} = \frac{291.356 * 5}{1000} = 1.456KW$$

The 4 conveyors connected at the same motor the power= $1.456\ Kw * 4 = 5.824KW$

Main cleaning conveyor calculations

The main cleaning conveyor data:

Length=20m

Width=1.5m

Speed=5m/s

The operation times of 30 min every day and the total cleaning material 360kg of each conveyor.

Each four conveyor connect together the total cleaning material:

the total cleaning material on main cleaning conveyor $= 4 * 360 = 1440\ Kg$

Conveyor capacity (ton/h)

$$Q = \frac{1440/1000ton}{0.5h} = 2.88ton/h$$

The weight of the belt part is given in MRF company catalog as shown in the appendix A the nominal cascade weight =3.54Kg/m^2 .

The area of belt =$20m * 1.5m = 30\ m^2$.

$total\ belt\ area = 30\ m^2 * 2 = 60m^2$.

$$\text{The weight of belt} = 60m^2 * \frac{3.54Kg}{m^2} = 212.4\ kg$$

The weight of pulleys =15Kg

$$\text{Weight of idelers} = 3 * 2 = 6\text{kg}$$

The supported surface for main cleaning conveyor

L=20m

h=1.5m

Thickness=1mm

The density of Galvanized Iron is no different from other Steels, and is generally taken as7850 $\frac{Kg}{m^3}$.

$v = l * h * thick$................................... (4 .13).

$$V = 20m * 1.5m * 0.001m = 0.03m^3$$

$$mass = \text{density} * \text{volume} = \frac{7850\text{kg}}{\text{m}^3} * 0.03 = 235.5\text{kg}$$

Shaft volume:

From Eq (4 .9).

$$= 3.14 * 1.5 * \frac{0.05^2}{4} = 0.00294m^3$$

$$density\ of\ carbon\ steel(\rho) = \frac{7.84gm}{cm^3} = 7840Kg/m^3$$

From eq(4 .10)

$$\text{Weight} = 7840 * 0.00294 = 23.079kg$$

$$\text{Weight of Tow shafts} = 2 * 23.079 = 46.158\text{Kg}$$

From eq (4 .11)

$$\text{The net weight of conveyor} = 212.4 + 15 + 6 + 235.5 + 46.158 = 515.058\text{Kg}$$

From eq (4 .12).

$$MW\left(\frac{Kg}{m}\right) = \frac{515.058Kg}{20m} = \frac{25.75kg}{m}$$

$$p = \frac{25.75kg}{m}$$

From eq (4 .6)

$$M = \frac{Q(\frac{ton}{h})}{V(\frac{m}{s})} * \frac{1000}{360} = \frac{2.88 * 1000}{5 * 360} = 1.6N/m$$

From eq (4.4)

$$T_x = 9.81pL_1F_B = 9.81 * 25.75 * 20m * 0.03N = 141.84\ N$$

From eq (4 .5)

$$T_Y = 9.81L_1F_LM = 9.81 * 20 * 0.03 * 1.6 = 9.41\ N$$

From eq (4 .3)

$$T_E = T_x + T_y = 141.84 + 9.41 = 151.25N$$

From eq (4 .7)

$$power(KW) = \frac{T_E(N) * V\left(\frac{m}{s}\right)}{1000} = \frac{151.25 * 5}{1000} = 0.75625\ KW$$

4.2.3.4 Control of cleaning conveyor motor:

In the control circuit using plc and protection circuit (over load, emergency, circuit breaker, on/off switches).

The controls signals come from communication system according to the sensor reading replace the control switch just on/off switch.

The main conveyor has its only squirrel cage motor calculated according to previous equation.

The motor forbidden to move the conveyor under the cage if the main conveyor doesn't operate in order to avoid Accumulation of material.

The speed of the conveyor not exceeds 5 m/s so we need gear ratio to reduce the speed as we need.

4.2.3.5 The gear design for cleaning conveyor

Sing sprocket type QD sprocket gears as shown in figure4.9 to connect the motor shaft using chain between the gear on shaft and gear on the pulley.

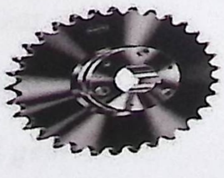


Figure 4.20: QD sprocket gear

Use double Strand sprocket gear in order to connect four conveyor under each other at the same motor and because need of auxiliary gear as shown in figure4.23.

Figure 4.21: Double Strand sprocket gear

We have four conveyor under each other ,so we need to QD sprocket gears and six double Strand sprocket gear as shown in figure4.24, where the first one connected to the motor shaft and connect to first gear to control the speed (reduce) by teeth ratio calculated below.

Second gear connects to first conveyor and connects to auxiliary gear to reverse the direction of motion to transfer the motion of motor to another conveyor at the same speed and the same direction.

In the design use this method to reduce the number of motors and reduce the cost in cleaning conveyor.

The speed of the four conveyors the same of the motor speed so the gear ratio will be one after reduce the motor speed as we need.

The linear speed of conveyor is $5m/s$ and diameter of pulley is 0.2m

$$v = r * \omega \text{.................................. (4 .14).}$$

$$\omega = \frac{v}{r} = \frac{5m/s}{0.1m} = 50\ rad/s$$

$$\omega = \frac{2\pi n}{60} \text{.................................. (4 .15).}$$

$$n = \frac{60 * 50/s}{2 * \pi} = 477.707rpm$$

Now find the speed ratio using gear ratio

$$N_1 V_1 = N_2 V_2 \text{.................................. (4 -16).}$$

Where:

N_1: $number\ of\ teeth\ of\ first\ gear$

V_1: $the\ speed\ of\ gear\ connect\ on\ shaft.$

N_2: $number\ of\ teeth\ of\ second\ gear$

V_2: $speed\ on\ the\ second\ gear$

We have:

$N_1 = 20\ teeth$ From appendix B.

$V_1 = 1500\ rpm$

$V_2 = 477.707\ rpm$

$$N_2 = \frac{V_1 N_1}{V_2} = \frac{1500 * 20}{477.707} = 62.8\ teeth$$

From appendix B and C of sprocket gear choose double strand type with teeth=60 and Outside diameter=24.9936cm

Figure4.22 show the gear system of cleaning system to connect 4 cleaning conveyor at the same motor.

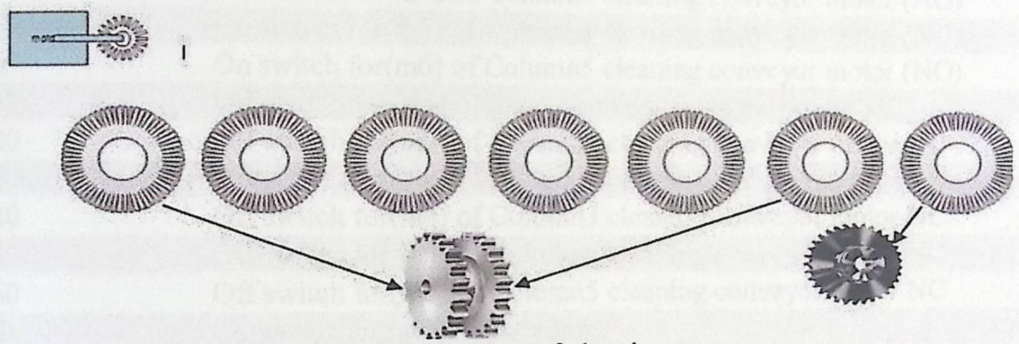


Figure 4.22: Gear system of cleaning conveyors

Sprockets should be large as possible given the application. The larger a sprocket is, less working load for a given amount of transmitted power allowing the use of a smaller-pitch chain [23].

Gear ratio of main cleaning conveyor

$N_1 = 20\ teeth$

$V_1 = 1500\ rpm$

$V_2 = 477.707\ rpm$

From Eq (4.16)

$$N_2 = \frac{V_1 N_1}{V_2} = \frac{1500 * 20}{477.707} = 62.8\ teeth$$

To connect the gears and Transfer the movements to another conveyor use steal chain as shown in figure4.23.

Figure 4.23: Steal chain

Common diametric pitches are 12 (big teeth), 24, 32, 48, 64 (fine teeth) [24].

4.2.3.6 Electrical design for cleaning system:

Allocation table of cleaning system show input and output parameters as shown in table 4.2.

Symbol	Function
Em	Emergency(NC) of cleaning unit
S0	Off switch (NC) of cleaning conveyors
S1	On switch for (m1) of main cleaning conveyor motor (No)
S2	On switch for (m2) of Column1 cleaning conveyor motor (NO)
S3	On switch for(m3) of Column2 cleaning conveyor motor (NO)
S4	On switch for(m4) of Column3 cleaning conveyor motor (NO)
S5	On switch for(m5) of Column4 cleaning conveyor motor (NO)
S6	On switch for(m6) of Column5 cleaning conveyor motor (NO)
S10	Off switch for(m1) of main cleaning conveyor motor NC
S20	Off switch for(m2) of Column1 cleaning conveyor motor NC
S30	Off switch for(m3) of Column2 cleaning conveyor motor NC
S40	Off switch for(m4) of Column3 cleaning conveyor motor NC
S50	Off switch for(m5) of Column4 cleaning conveyor motor NC
S60	Off switch for(m6) of Column5 cleaning conveyor motor NC
K1	Contactor for (m1) main motor
K2	Contactor for (m2) coulumn1 of conveyors
K3	Contactor for (m3) coulumn2 of conveyors
K4	Contactor for (m4) coulumn3 of conveyors
K5	Contactor for (m5) coulumn4 of conveyors
K6	Contactor for (m6) coulumn5 of conveyors

Table 4.2: allocation table for cleaning system

Communication control signal needed to control the cleaning system as shown in table 4.3.

S1	**On switch for (m1) of main cleaning conveyor motor (No)**
S2	On switch for (m2) of Column1 cleaning conveyor motor (NO)
S3	On switch for(m3) of Column2 cleaning conveyor motor (NO)
S4	On switch for(m4) of Column3 cleaning conveyor motor (NO)
S5	On switch for(m5) of Column4 cleaning conveyor motor (NO)
S6	On switch for(m6) of Column5 cleaning conveyor motor (NO)
S10	Off switch for(m1) of main cleaning conveyor motor NC
S20	Off switch for(m2) of Column1 cleaning conveyor motor NC
S30	Off switch for(m3) of Column2 cleaning conveyor motor NC
S40	Off switch for(m4) of Column3 cleaning conveyor motor NC
S50	Off switch for(m5) of Column4 cleaning conveyor motor NC
S60	Off switch for(m6) of Column5 cleaning conveyor motor NC

Table 4.3: communication control signal needed

4.2.3.6.1 Plc connection

Show the schematic diagram of PLC connection of the feeding system as shown in figure 4.24.

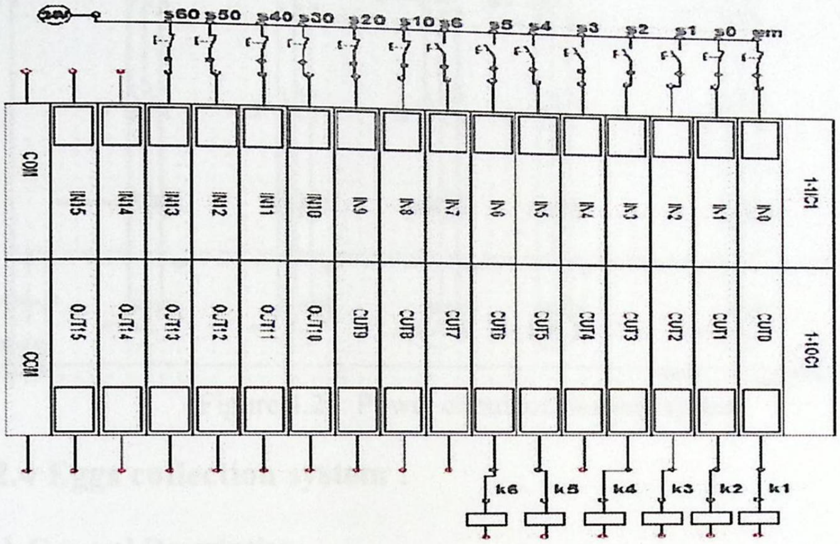


Figure 4.24: plc connection of cleaning system

4.2.3.6.2 Power circuit

Power circuit of cleaning system: shows the motor distribution and control devices used and the protection devices used (fuses, over load, circuit breaker) as shown in figure 4.25.

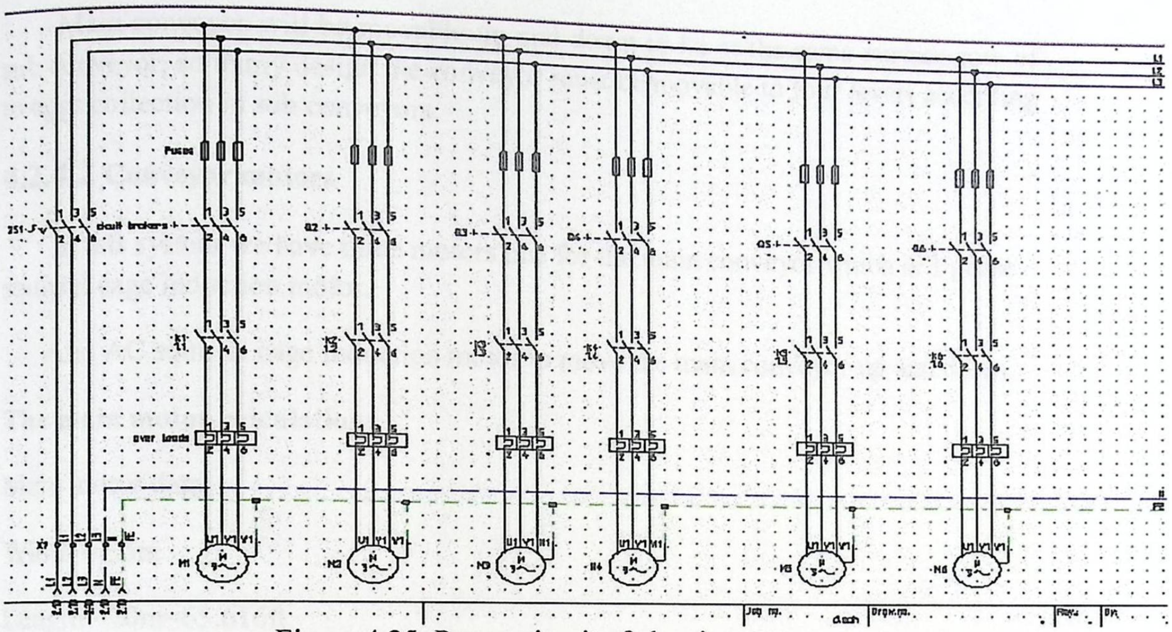


Figure 4.25: Power circuit of cleaning system

4.2.4 Eggs collection system :

4.2.4.1 General Description:

The cage designed to be with slop where Eggs roll at low speed, over a floating wire-mesh floor with slope about 15°, onto the collecting belts without crowding.

In my design we need to collect the egg at specific location where the systems consist of tow type of conveyors.

First a conveyor designed to be in front of the cage which the eggs collect there.

The eggs transport belt is made of woven polypropylene.

Figure 4.26: Woven polypropylene

All these conveyors spread in front of cages collects the egg on a main conveyor.

The main eggs conveyor is the type of roller without belt to increase the friction between the egg and conveyor, the space between the rods is small in order to save the eggs from fall or broken, these types generally used for transporting materials pieces with speed of 3.5 m/min.

Main conveyor will be movable up and down to be at the same surface row of sub conveyor, so in my design the conveyor must be movable to four levels according to eggs collection in sub conveyors.

4.2.4.2 Conveyor motors

In this system we have three motors one for the main conveyor which is 3 phase squirrel cage induction motor.

And AC squirrel cage induction motor to move the main conveyor up and down.

The main motor calculation:

Main motor data:

Width =1.5m

Length= 20m=65.616ft

Speed of conveyor=3.5m/min.

Material weight using 8 kitchens each $1m^2$.

$$\frac{area\ of\ cages * number\ of\ kitchen\ per\ 1m^2 * weight\ of\ egg}{1000} = \frac{60m^2 * 8 * 60}{1000}$$

$$= 28.8\ Kg\ at\ one\ side\ of\ cages$$

For 10 sides the egg weight =288kg at a time the main conveyor carry as maximum value.

$$the\ weight\ of\ eggs\ per\ unit\ area = \frac{total\ weight\ of\ eggs}{area} = \frac{288Kg}{1.5 * 20\ m^2}$$

$$= 9.6Kg/m^2$$

From appendix E tack the standard value of Belt Driven Live Roller Heavy Duty Conveyor at load of 600lb maximum as shown in figure Figure4.27.

Speed=30fpm=9.144m/min Length=61''=1.55m

LOAD CAPACITY CHART

30 FPM Roller Speed

Diameter Drive Pulley	HP	B/F WIDTH 27" to 31"		B/F WIDTH 33" to 51"		B/F WIDTH 57" to 61"	
		50' OAL	100' OAL	50' OAL	100' OAL	50' OAL	100' OAL
12"	2	8000 lbs.	6000 lbs.	600 lbs.	--	6000 lbs.	--
10"	3	15000 lbs.	12000 lbs.	12000 lbs.	8000 lbs.	12000 lbs.	6000 lbs.
	5	18000 lbs.	17000 lbs.	17000 lbs.	12000 lbs.	16000 lbs.	11000 lbs.

Standard Available Belt Widths:

27 in., 31 in., 33 in., 39 in., 45 in., 51 in., 57 in., 61 in.

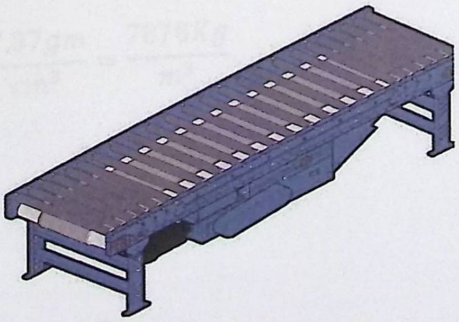


Figure4.27: Roller heavy duty conveyor

Need another motor to move the sub conveyor and collect the eggs to main one.

Approximate calculation of Sub egg conveyors:

Length=60m.

Width=0.15 m.

Speed=3.5m/min.

Consider that every $1m^2$ have 8 kitchens there *60* $1m^2$ every side so we have 480 kitchens, the maximum productivity 480 eggs.

The average weight =60 gm.

$$\text{Material (eggs) weight} = 0.06\text{kg} * 480 = 28.8\text{kg}.$$

The conveyor consist of tow pulleys and belt of EP Rubber

The weight of the belt part is given in MRF company catalog as shown in the appendix A the nominal cascade weight =3.54Kg/ $1m^2$

$$\text{The area of sub conveyor belt } = 0.15\text{m} * 60\text{m} = 9\ 1m^2$$

$$The\ total\ belt\ area = 2 * 9m^2 = 18m^2$$

$$\text{The weight of sub eggs conveyor belt} = 18m^2 * \frac{3.54Kg}{m^2} = 63.72kg$$

The pulley dimension:

Diameter=0.1m.

Length=0.15m

From eq(4.7)

$$\text{The volume (v) } = \pi * \text{h} * (\text{R}^2 - \text{r}^2) = 3.14 * 0.15\text{m} * (0.1^2 - 0.07^2)/4$$
$$= 0.001275\text{m}^3$$

$$density\ of\ iron(\rho) = \frac{7.87gm}{cm^3} = \frac{7870Kg}{m^3}$$

From eq (4.9)

$$\text{Weight} = \rho * v = 7870 * 0.001275 = 5kg$$

There's tow symmetry pulleys so

$$\text{the weight } = 5 * 2 = 10\text{kg}$$

We use every 5m one idler to support the belt; there are 24 idlers with 0.5Kg each

$$Idlers\ weight = 24 * 0.5 = 12Kg$$

From eq(4.10)

$$MW(kg) = 63.72kg + 10 + 12 = 85.72Kg$$

From eq(4.11).

$$the\ weight\ of\ moving\ part\left(\frac{Kg}{m}\right) = \frac{85.72kg}{60m} = 1.428$$

$$p = \frac{1.428Kg}{m}$$

$F_B = 0.03$

$F_L = 0.03$

The egg collections time of 30 min every day and the total cleaning material Kg 28.8[2]

$$Q = (28.8/1000)/(30min/60) = 0.0576ton/h$$

From eq(4.6)

$$M = \frac{Q\left(\frac{ton}{h}\right)}{V\left(\frac{m}{s}\right)} * \frac{1000}{360} = \frac{0.0576 * 1000}{0.1524 * 360} = 1.04\frac{N}{m}$$

From eq(4.4)

$$T_x = 9.81 * 1.428 * 60m * 0.03N = 25.215\ N$$

From eq(4.5)

$$T_Y = 9.81 * 60 * 0.03 * 1.04 = 18.364\ N$$

From eq(4.3)

$$T_E = 25.215 + 18.364 = 43.579\ N$$

From eq(4.7)

$$power(KW) = \frac{43.579 * 0.0583}{1000} = 0.002547\ KW$$

We have symmetry 10 conveyor at the same motor

$$power(KW) = 10 * 0.002547 = 0.02547KW$$

These value taken from real farm and is maximum value [2]

4.2.4.3 Gear design of egg collection conveyors

The main egg conveyors motion can be achieved by using gears, rack and pinion gears as shown in figure4.28 connected to the motor shaft to move up and down linear motion.

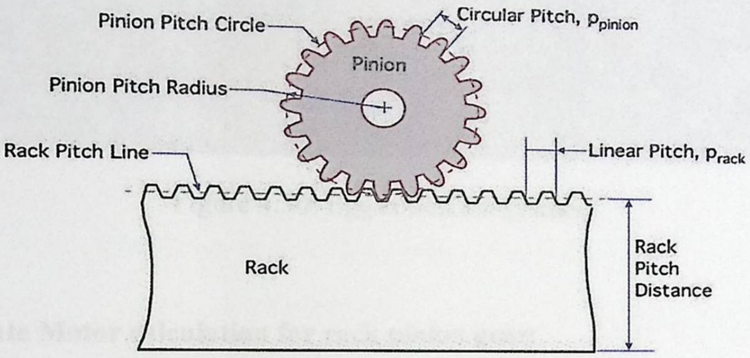


Figure4.28: Rack pinion gear

For a rack and pinion to mesh together properly, the pitch of rack and pinion must be equal $p_{pinion} = p_{rack}$

Note that the number of teeth on the rack is not relevant to the velocity ratio. The linear speed of the rack is simply a function of the pinion pitch radius and the angular velocity of the pinion.

The conveyor distributed in the farm in front of cages (sub egg conveyors) moved using motor and use hobbing spur gear as shown in figure 4.30 to move five sub conveyors at the same surface and at the same time.

Figure4.29: Hobbing spur gear

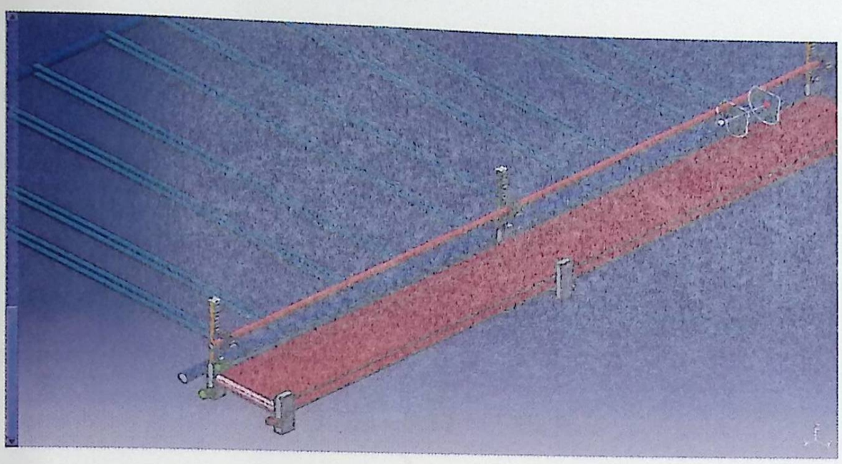


Figure 4.30: Egg collection system

Approximate Motor calculation for rack pinion gear:

Linear speed=5m/min=0.083 m/s

From eq(3.15)

$$\omega = \frac{v}{r} = \frac{0.083}{0.09144} = 0.911 \frac{rad}{s}$$

Choose standard dimension of spur gear from appendix D

70 teeth and 7in pitch radius=17.78 cm

Approximate weight of main conveyor =1200Kg

The coefficient of friction between the gear rack B and the horizontal surface is (μ) = 0.3[25].

$$\sum F_y = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(4 -17).}$$

$$F_{friction} = mg * \mu_s \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(4 -18).}$$

Where:

$F_{friction}$: $friction\ force$.

μ: $friction\ coefficient.$

m: $total\ mass\ of\ body.$

g: $gravitational\ constant$ (9.81)

$$T = F_{friction} * r \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(4 .19).}$$

Where:

$T: torque.$

$r: radius\ of\ wheel$

From eq(4.18)

$$F_{friction} = 1200 * 9.8 * 0.3 = 3528\ N$$

From eq(4.19)

$$T = 3528 * 0.17 = 599.76N$$

Eq (4 .16)

$$P = T * \omega = 599.76 * 0.911 = 0.54638\ KW$$

$$= \frac{546.38136}{746} = 0.732\ Hp$$

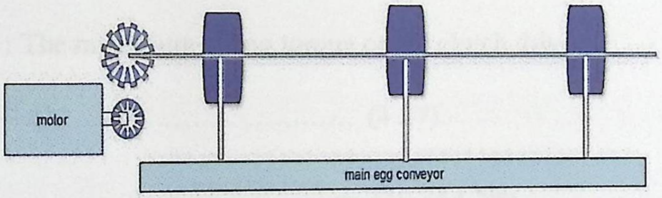


Figure4.31: Main egg conveyor.

Gear design for sub egg conveyors motor

In my design I need to move the sub conveyor to collect on the main eggs conveyor the motor ,each row of sub egg conveyor collect together when the main conveyor at the same surface .

In my design use one 3phase squirrel cage induction motor with electrical clutch system.

The system consists of 4 electrical clutches connected to 4 rods and the rods connect to pulley of sub egg conveyor, each 5 conveyor at the same row connect at the same rod and the clutch connects to the motor shaft as shown in figure4.23.

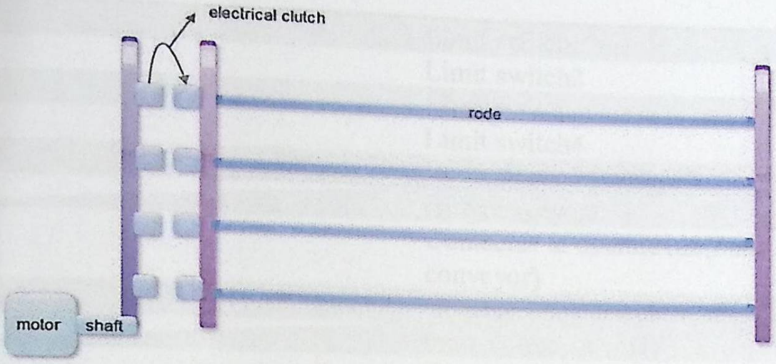


Figure4.32: Gear design for sub egg conveyors motor

Motor data connected to their clutch gear

Speed=3.5m/min=0.05833.

Motor required to moving the sub conveyor =0.01577KW

Motor power need to rotate the load =0.01371KW.

Torque Rating : The maximum rating torque of the clutch drive.

$Torque = \frac{power}{\omega}$.................................. (4 .17).

$$= \frac{0.01577*1000}{0.058333} = 270.344 \text{ Nm}$$

Operating Voltage: the input voltage range for an electrically-operated clutch=220Vac.

An input current to the stator coil creates a magnetic field that lines-up the particulate to provide load engagement (clutches) or slowing (brakes). The coupling torque is controlled directly by the magnetic field that is created by the input current. Magnetic particle brakes provide good control and a wide range of holding torque at any speed.

4.2.4.4 Electrical design of egg collecting system

Allocation table Allocation table show the input and output parameter needed to control the feeding system using PLC as shown in Table 4.4.

Symbol	Function
Em	Emergency switch(NC)
S0	Off switch(NC)
S1	On switch m1 (No)
S2	On switch m2(No)
S3	On switch m3(No)
S10	Off switch of m1(NC)
S20	Off switch of m2(NC)
S30	Off switch of m3(NC)

LS1	Limit switch1
LS2	Limit switch2
LS3	Limit switch3
LS4	Limit switch4
K1	Contactor to operate m1 (sub conveyor)
K2	Contactor to operate m2(main egg conveyor)
K3	Contactor to control the motor move the main conveyor (m3)
K4	Contactor to reverse the direction of (m3)
C1	Electrical clutch1
C2	Electrical clutch2
C3	Electrical clutch3
C4	Electrical clutch4

Table 4.4: allocation table for cleaning system

Signal taken from communication and replace the switches as shown in table 4.5

Signal	Function
S0	Off switch(NC)
S1	On switch m1 (No)
S2	On switch m2(No)
S3	On switch m3(No)
S10	Off switch of m1(NC)
S20	Off switch of m2(NC)
S30	Off switch of m3(NC)

Table 4.5: Signal taken from communication

4.2.4.4.1 Plc connection

Show the schematic diagram of PLC connection of the feeding system as shown in figure 4.33

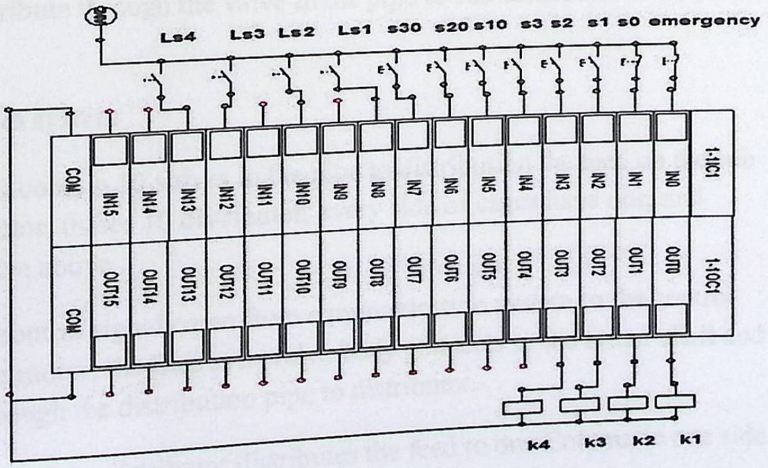


Figure 4.33: PLC connection of egg collection system.

4.2.4.4.2 Power circuit of egg collection system

Power circuit of egg collection system shows the motor distribution and control devices used and the protection devices used (fuses, over load, circuit breaker) as shown in figure 4.34.

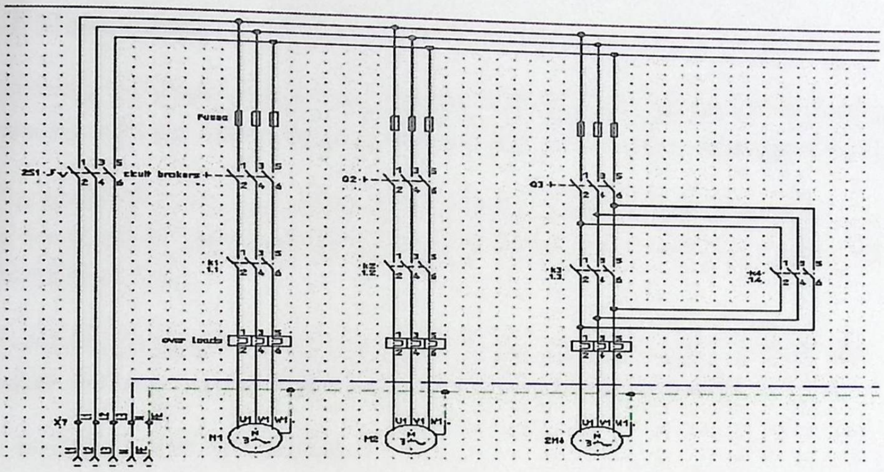


Figure 4.34: Power circuit of egg collection system

4.2.5 The feeding system:

4.2.5.1 General description:

The system consist of main hopper (tank), pipes, valves, sub tank (distributor), motor.

The Main tank contain the feed and then push the feed using pump through the pipe which have screw inside it to make the move easily through it.

Then the feed distribute through the valve in the pipe to sub distributor to distribute when needed.

4.2.5.2 The valves system

The system contain 10 valves in the pipe to distribution the feed on the sub tank or the distributor, theses 10 distributor, every side of cages have one, and everyone have valve above .

When the control signal come from communication system to the control circuit of the valve motor, the Blades in valve body connects to the motor shaft and the material go through the distribution pipe to distributor.

Every sub tank or distributor distributes the feed to one Column in one side of cage; the distributor will be movable along the cage Back and forth.

The shape designed to allow the feed follow in the Feedlots easily and also to follow in four Feedlots at the same time.

To move the distributor use tow wheel based on Iron Bridge fixed above the cages along it as shown in figure 4.35.

4.2.5.3 Motor design

To move the wheels along need 3 phase squirrel cage induction motor connect these wheels with the shaft of motor with control according to communication signals

The speed of motor needed about 10 m/min, so need frequency converter to control the speed of motor and the speed when the distributor empty is more than when it's full of food.

In my design need 10 motor Ac squirrel cage induction motor to move the distributors we can connect every tow motor on frequency converter.

Motor calculation for distributor

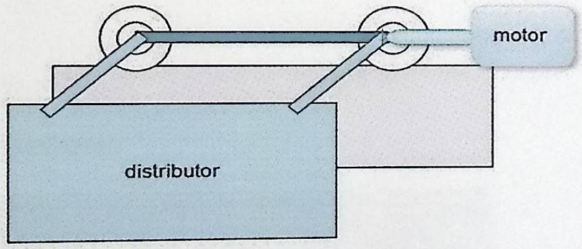


figure4.35: distributor system

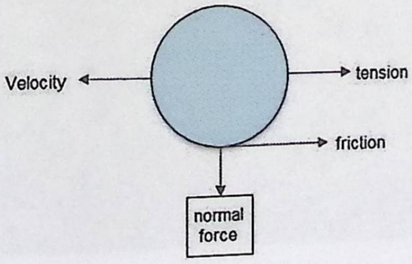


Figure4.36: wheel forces Analysis

Distributor data:

Speed=10m/min=0.16666$\frac{m}{s}$.

Weight=140 kg food+100kg structure.

Using 2 wheels to carry.

Diameter of wheel=30cm=0.3m

From force analysis on the wheel as shown in figure4.29 can calculate the motor power to move the wheels.

From Eq(4 .14)

$$\omega = \frac{v}{r} = \frac{0.16666}{0.15} = \frac{1.111rad}{s}$$

From Eq (4. 19).

$$F_{friction} = 240 * 9.8 * 0.2 = 470.4\, N$$

From Eq(4 .17).

$$T = 470.4 * 0.15 = 70.56 N.m$$

The weight distribute on the tow wheel, so the mass=mass/2

$$T = \frac{70.56}{2} = 35.28\; N.m$$

From Eq(4.17)

$$P = T * \omega = 35.28 * 0.6666 = 39.198\; w$$

Figure 4.37 show the feeding distributor

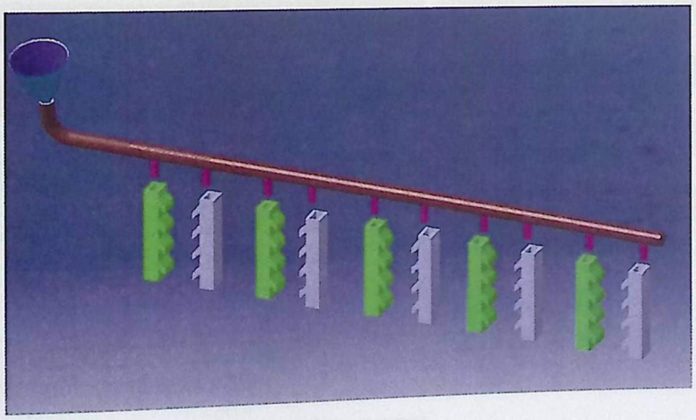


Figure 4.37: Feeding system

4.2.5.4 Electrical design of feeding system

Allocation table show the input and output parameter needed to control the feeding system using PLC.

Symbol(input)	Function
Emergency	Emergency switch(NC)
S0	Off switch(NC)
S1	On switch m1 (No)
S2	On switch m2(No)
S3	On switch m3(No)
S4	On switch m4(No)
S5	On switch m5(No)
S10	Off switch of m1(NC)
S20	Off switch of m2(NC)
S30	Off switch of m3(NC)
S40	Off switch of m4(NC)
S50	Off switch of m5(NC)
LS1	Limit switch1
LS2	Limit switch2
LS3	Limit switch3
LS4	Limit switch4
LS5	Limit switch5
LS6	Limit switch6
LS7	Limit switch7
LS8	Limit switch8
LS9	Limit switch9
LS10	Limit switch10

Table 4.6-a

Symbol(output)	Function
K1	control m1
K2	reverse m1
K3	control m2
K4	reverse m2
K5	control m3
K6	reverse m3
K7	control m4
K8	reverse m4
K9	control m5
K10	Reverse m5
Y1	Valve1
Y2	Valve2
Y3	Valve3
Y4	Valve4
Y5	Valve5

Table 4.6-b

Signal taken from communication

S1	On switch m1 (No)
S2	On switch m2(No)
S3	On switch m3(No)
S4	On switch m4(No)
S5	On switch m5(No)
S10	Off switch of m1(NC)
S20	Off switch of m2(NC)
S30	Off switch of m3(NC)
S40	Off switch of m4(NC)
S50	Off switch of m5(NC)

Tabel 4.7: Signal taken from communication

4.2.5.4.1 Plc connection of feeding system

Show the schematic diagram of PLC connection of the feeding system as shown in figure4.38

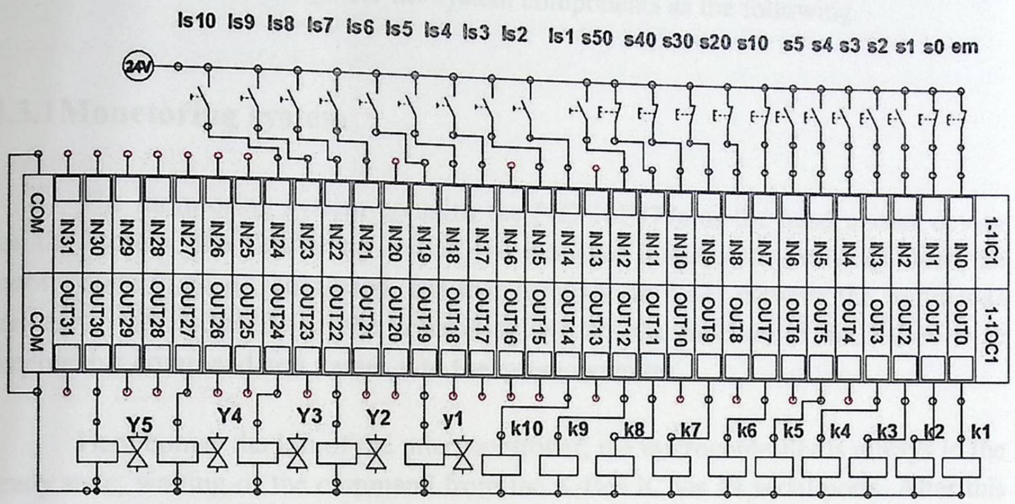


Figure 4.38: PLC connection

4.2.5.4.2 Power circuit of feeding system

The power circuit shows the motor distribution and control devices used (contactors switches) and the protection used (fuses, over load, circuit breaker) as shown in figure4.39

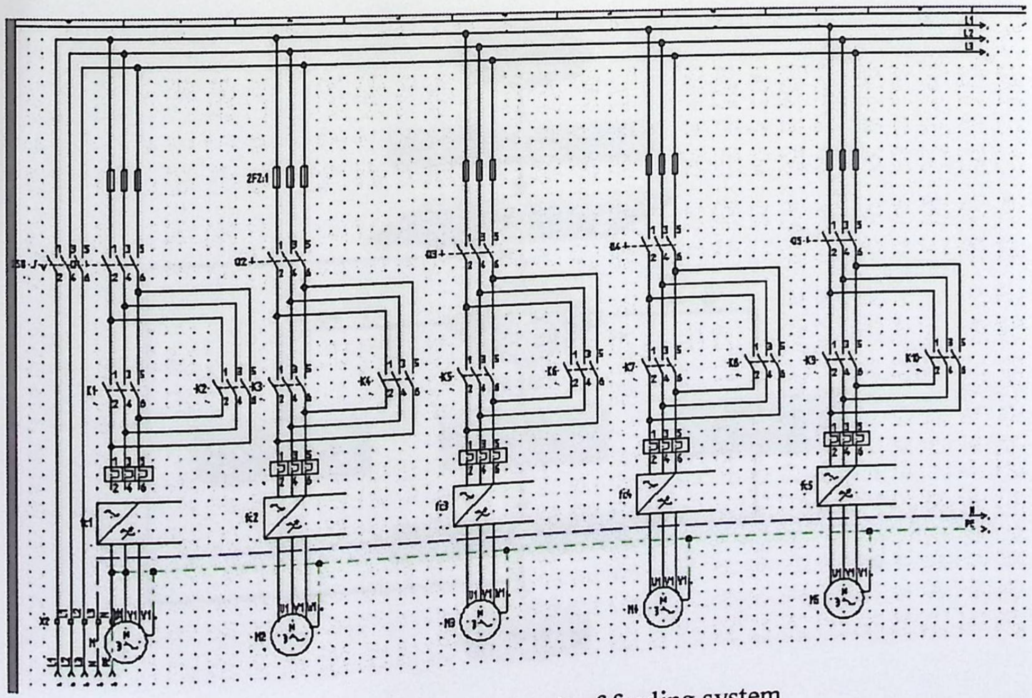


Figure 4.39: Power circuit of feeding system

4.2.5.4.1 Plc connection of feeding system

Show the schematic diagram of PLC connection of the feeding system as shown in figure4.38

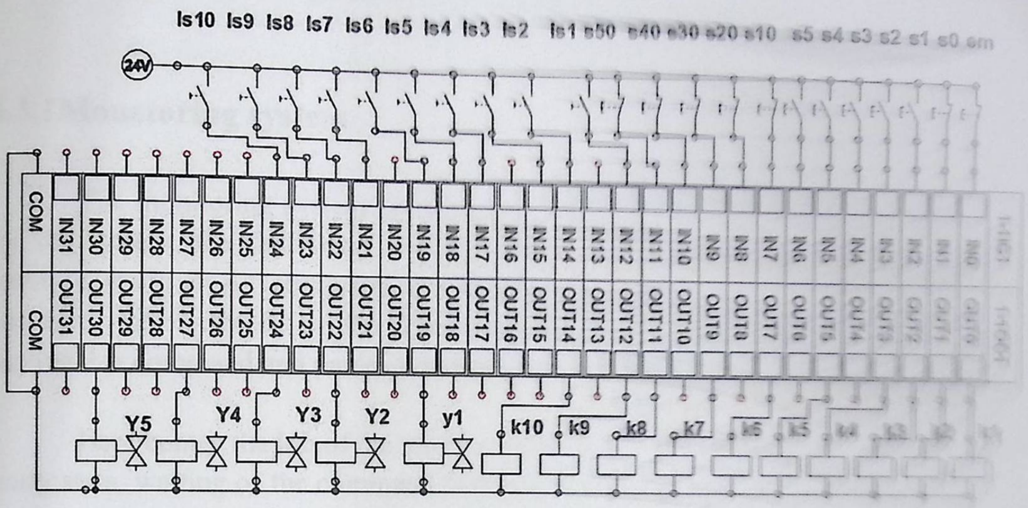


Figure 4.38: PLC connection

4.2.5.4.2 Power circuit of feeding system

The power circuit shows the motor distribution and control devices used (contactors, switches) and the protection used (fuses, over load, circuit breaker) as shown in figure4.39

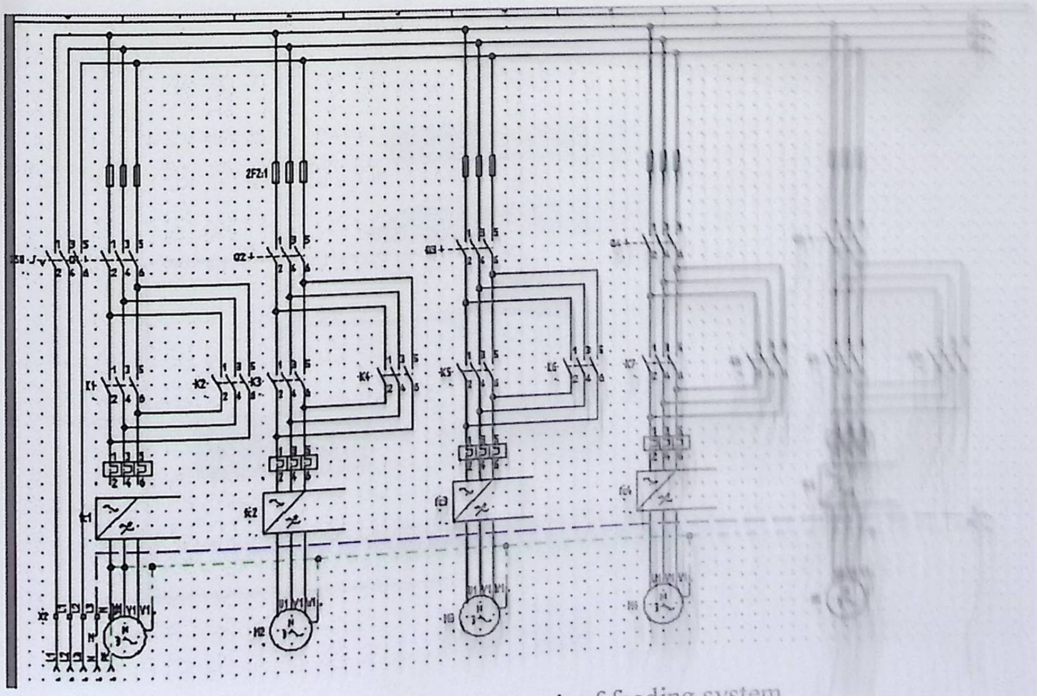


Figure 4.39: Power circuit of feeding system

4.3 Software design

In this section, and due to our Shortage in programming skills, we will present a little about software design for the system components as the following.

4.3.1Monetoring system

The monitoring system contains the PIC 16F877a as the head master device for the system, the first thing is that the manager initiates its functions from the mobile node as an AT command, and then through the Wi-Fi network the commands arrives to the X-Bee IC on the controlling system, the function of the X-Bee is to receive this command and passes it to the microcontroller.

Here comes the job of the microcontroller; the microcontroller is always in the ready state, waiting or the command from the X-Bee IC via its serial ports. After this point microcontroller starts the comparing process, it compares the strings that arrived from the AT command to its own predefined strings, if the strings are equal then a cretin function will start working.

The microcontroller has three strings to compare with and another three corresponding functions to carry out. Figure 4.19 shows the flowchart witch presen the way of how the flow of control in programming side in the controlling node.

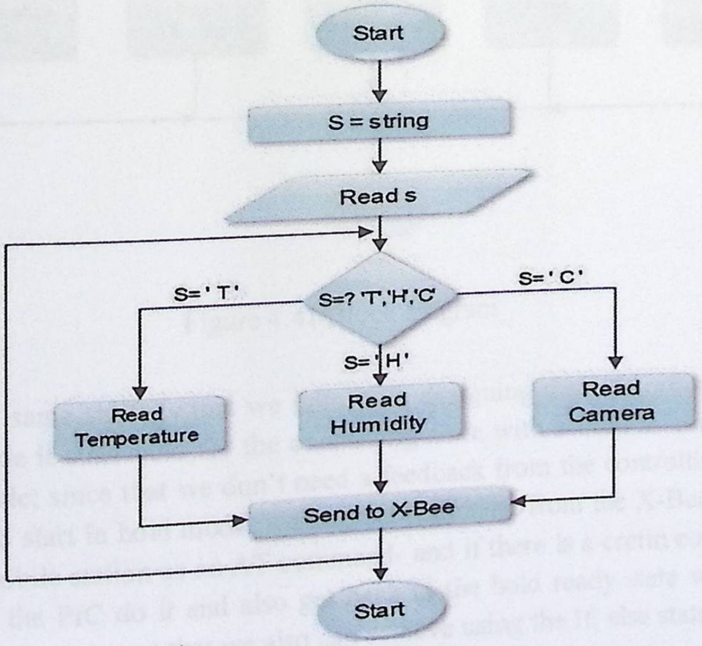


Figure 4.40: Logic design for the monitoring system

Taking in concern that the microcontroller PIC16F877a programming code must have the ability to accept two or more commands at the same time; due to the fact that there is a multiple function need to be caring at the same time, hence, to make sure that we can use the If, else statement which allows us to carry out a first command for example and ignoring others since that there is only one character on the command, and then to go back to ready state for waiting the next command and so on .

4.3.2 Controlling system

Figure 4.20 shows the block diagram for the logical software design for the controlling node, as the same of the monitoring node the controlling nod contains a microcontroller which is PIC16F 688 smaller then the PIC 16F877a due to its smaller function that need to be take caring.

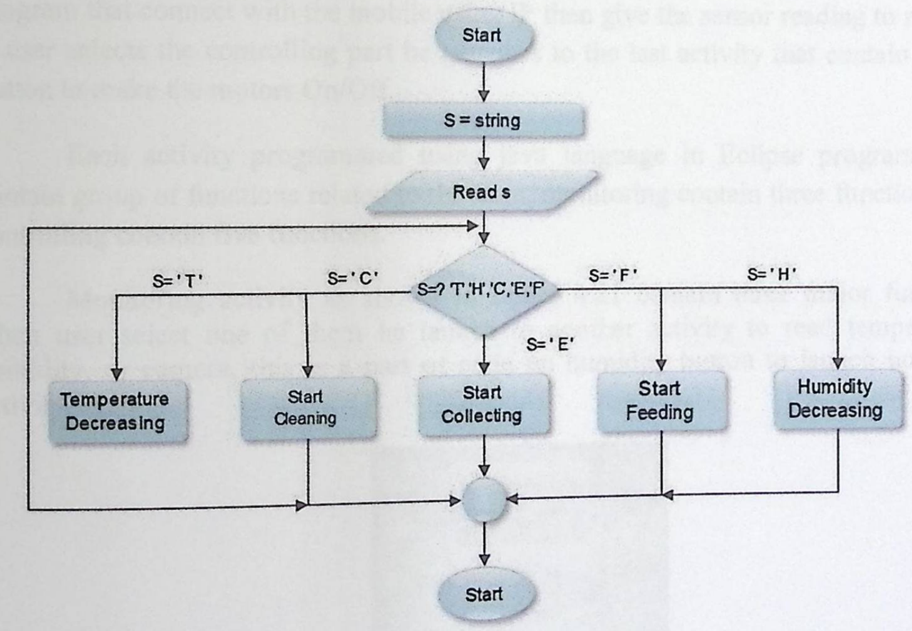


Figure 4.41 Block diagram

The same strategy that we fallows in designing the mentoring node from the software side it stills hold for the controlling node with a little minimizing form the feedback side; since that we don't need a feedback from the controlling nod . So the PIC 16F688 start in hold mode, waiting the command from the X-Bee -which comes from the mobile station as an AT command- and if there is a cretin command need to be execute the PIC do it and also get back to the hold ready state waiting the next command to comes, and that we also can achieve using the If, else statement.

4.3.3 Human interfacing system

In the android mobile application, we have five activities on the mobile for monitoring and controlling the farm, monitoring includes the temperature, humidity, and camera reading. User requests readings by the android application. But in the controlling part user make the motors of feeding, cleaning, collecting eggs and suction fan on/off.

The first activity is the main activity contain user name and password, after programming the two button in this activity for user name –PPU- and password -PPU- .after entering the correct user name and password you move to the second activity, this activity contain two choices monitoring or controlling. if you select monitoring you launch the third activity, this activity contain three choices for monitoring (Temperature reading, Humidity reading, and camera reading).user select the function he want to do. The reading of sensors came from laptop (server) using net beans [] program that connect with the mobile using IP then give the sensor reading to mobile. If user selects the controlling part he launches to the last activity that contain toggle button to make the motors On/Off.

Each activity programmed using java language in Eclipse program, each contain group of functions related to the farm, monitoring contain three functions and controlling contain five functions.

Monitoring activity as shown in figure 4.21 contain three major functions when user select one of them he launch to another activity to read temperature, humidity, or camera, this is a part of code on humidity button to launch humidity activity.

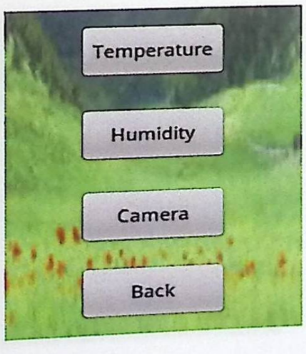


Fig 4.42: Monitor Manu on Mobile Application.

```
*********************************************************************
android:onClick="humidity"
android:text="Humidity" />  .

\\ the function in the humidity activity is

android:onClick="trigger"
```

```
android:text="Humidity Reading" />
```

Code to get data from server:

```
case R.id.humiditySensBtn:
try {
URL url  = new
URL("http://192.168.0.109:8092/SmartFarmProject/getData?x=Humidity");HttpURL
Connection con = (HttpURLConnection) url.openConnection();
String result = readResult(con.getInputStream());
TextView txt = (TextView) findViewById(R.id.humidityTxt);
txt.setText(result);
                    }
 catch (MalformedURLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
catch (Exception e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
```

****

The controlling activity figure 4.22 contains five toggles to make the motors (feeding, cleaning, egg collecting, temperature or humidity decrease) On/Off.

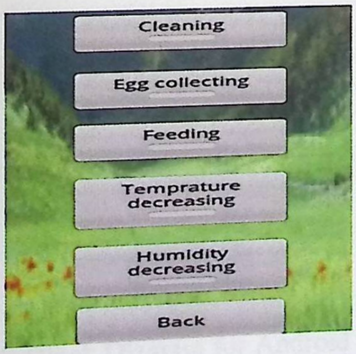


Figure 4.43 Control Manu on the Mobil Application

This is a part of the Java code to activate each toggle:

****

```
public void cont(View v){
setContentView(R.layout.control);
```

```
public void sendData(View v){

String data="";
switch(v.getId()){
case R.id.cleaningTgl:
data = "cleaning";
break;
```

****

Figure 4.23 shows the flowchart of the android applications that the user can do and how the way of doing it:

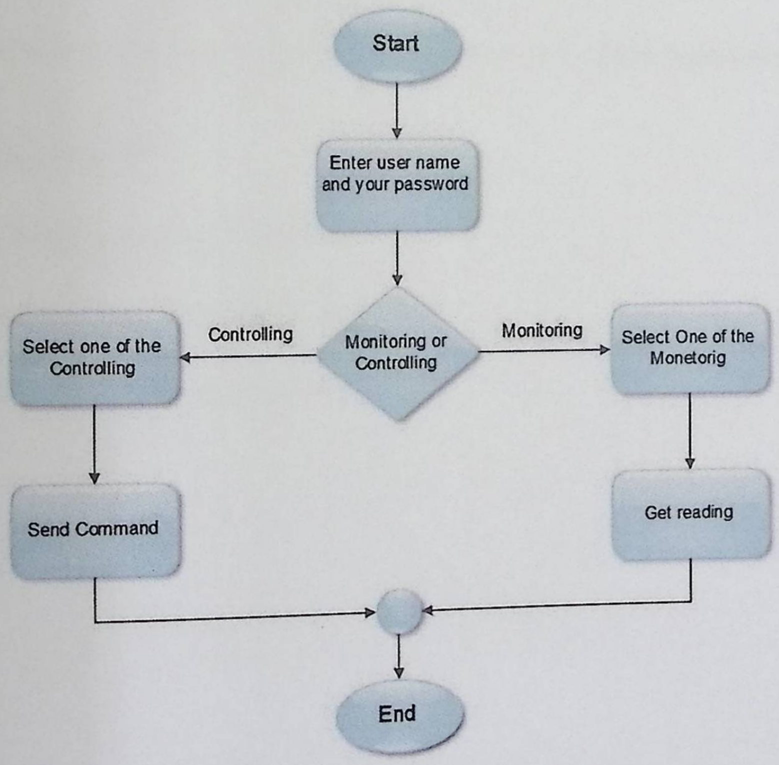


Figure 4.44: Block Diagram for Android Application.

5

Chapter Five

Testing performance

5.1 Introduction

In this chapter we will present how we implement the system components together and the testing operation for all system components.

5.1 Monitoring node Testing

As we explained in the previous chapter the monitoring node contains three main devices; LM35 temperature sensor, HS11011LS humidity sensor and the CCD camera, here are the testing process for each device.

5.1.1 LM35 Temperature sensor Testing

Figure 5.1 present the real testing for measurement that the LM35 gives as voltage. The reading of the voltmeter is 0.27 volt, that present the output of the sensor in term of voltage and relating this reading to equation 4.1 the temperature r in centigrade will be equal to 10* the voltmeter reading which equal to 27 C.

Figure 5.1 LM35 reading.

After taking the temperature level as a voltage, we entered it to PIC 16F877a. figure 5.2 in next page shows the simulation for the PIC 16F877a on Proteus software, after doing the analog to digital conversion, the PIC sends the reading to the serial port on order to get to the X-Bee.

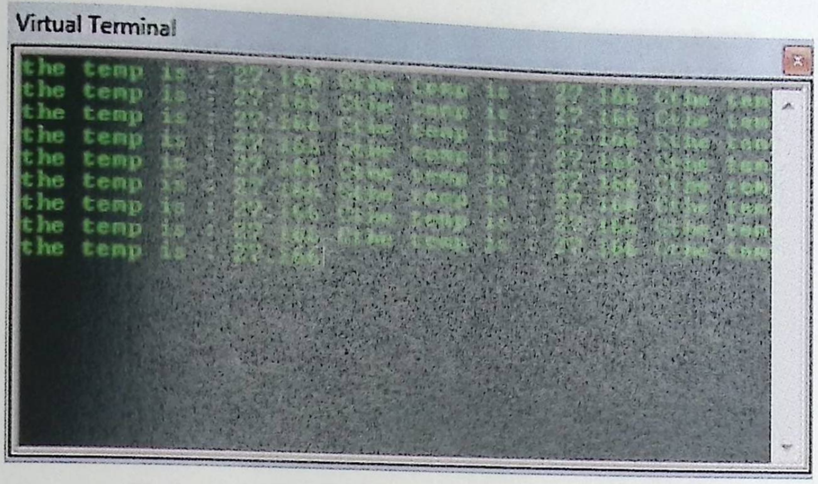


Figure 5.2: Temperature level at the PIC16F877a output. .

5.1.2 HS1101LS Humidity sensor Testing

Figure 5.3 present the real implementation circuit for the HS1101LS sensor, the figures shows the output reading for the amplitude of the humidity signal on the UT-33C device which equal to 2.31V, and the reading of its frequency on the GDM-396 device which equal to 6.55KHz.

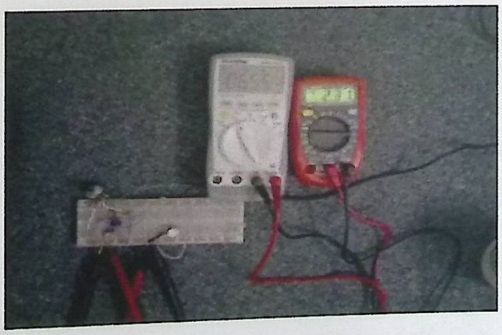


Figure 5.3 :Humidity circuit testing

On other hand figure 5.4 next page present the humidity signal on the oscilloscope that we used in the university, this humidity signal is not a square signal we need to square it using the LM741 as a comparator in order to get it into the PIC, figure 5.5 present the humidity converted signal to the square one, but also the problem that we are facing hear is that our PIC cannot read frequency up to this range, and the maximum value that the PIC can handle is2.3KHz, so we convert this humidity signal frequency to a low frequency one. Using the JK flip flop.

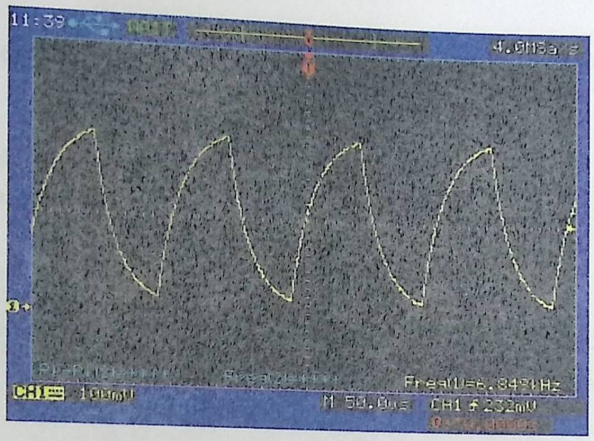


Figure 5.4 :Humidity circuit testing on the oscilloscope.

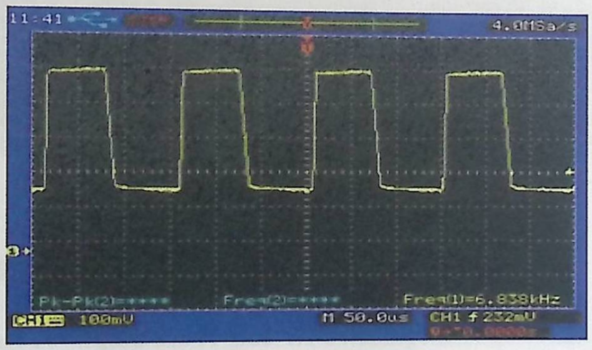


Figure 5.5:Humidity Square signal.

Now, to get the square low signal into the PIC 16F877a, we used the CCP PINS, on PIC software side we can manipulate mathematically according to equation 4.2 to get out the value of humidity and sending it to the serial port to X-Bee. Figur5.6 shows the output at the serial port of the PIC when the humidity need to be reed.

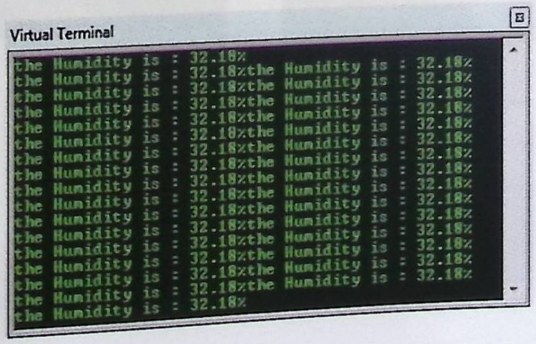


Figure 5.5:Humidity Reading in PIC signal.

5.1.3 CCD camera Testing

Figure 5.7 present the output of the CCD camera at the output of the PIC16F877a, it's about serial HEX characters, need to be convert via Java language, that we used Java program to do so.

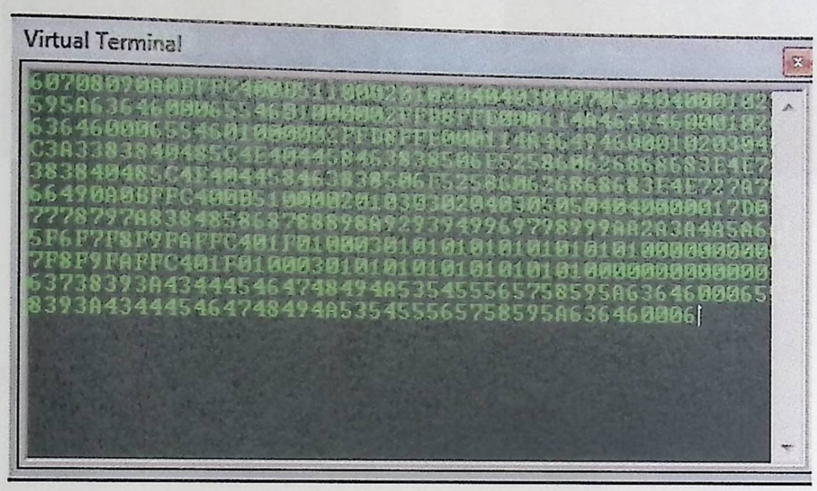
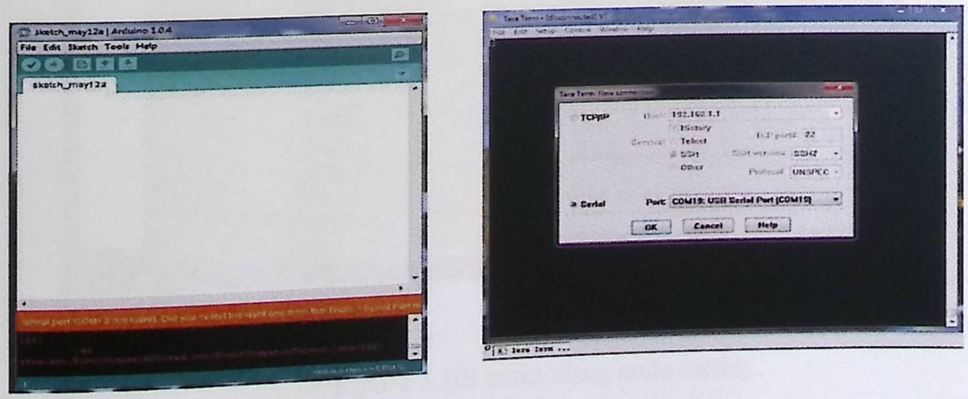


Figure 5.7: CCD Camera Output

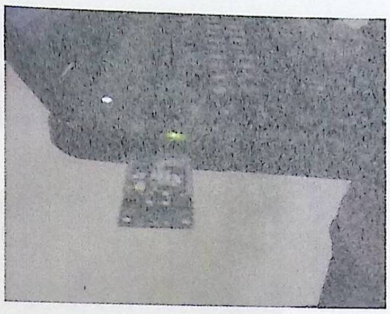
5.1.4 X-Bee Wi-Fi Testing

When we start configure the X-Bee Wi-Fi components, we discovered that our X-Bee adapter doesn't support it, due the fact the X-Bee need at least 766 mA at stating, and the adapter has its own limitation on current, current should not exceeds 500mA, so we tried to use deferent software in order to over con this problem but we didn't succeed, figures 5.8 shows deferent types of software that we tried using it to get access to the X-Bee Wi-Fi.



Figures 5.8: Used to configure the X-Bee.

so we replace the Wi-Fi with Zig-Bee, figures 5.9 present the zig-bee testing.

Figures 5.9: Testing Zig-Bee.

5.2 Controlling node testing

In controlling node we used the PIC16F688 due to reasons that we explained previously. But while implementing the project component the PIC16F688 was damaged, and for the shortage in time we decide to use PIC16F877a due to the fact it exist in the local market. Figur 5.10 present the testing for the controlling node, according to the terminal texts as you can see the switch are on and thay are off also according to the case of the latter.

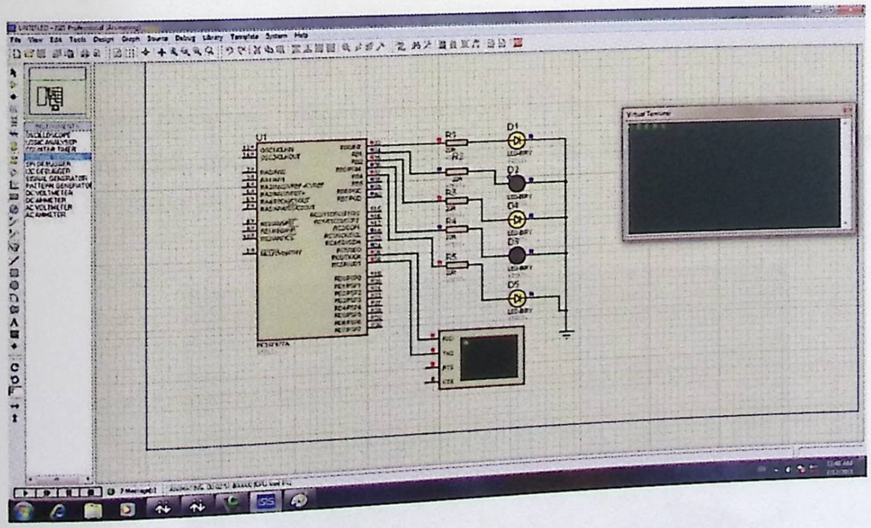


Figure 5.10: controlling node testing.

5.3 Hardware implementation

Figure 5.11 Represent the demo design in general shows the operation needed, cleaning, eggs collection, feeding system, and humidity system.

Figure 5.11 : demo frame

5.3.1 Cleaning system:

The figure show the cleaning conveyor in the project demo with 3 phase induction motor and use gear box to reduce the speed suitable for cleaning application. The conveyor be under the cage demo

Figure 5.12 : Cleaning conveyor

5.3.2 Eggs collecting system

The figure 5.13 Show a conveyor beside the cage used to collect the eggs, the cage slant with a degree allows the egg to collect on the egg conveyor.

Use gear box with ratio 1:50 and tow sprocket gears (N1=16 teeth, N2=25 teeth) to reduce the speed as needed about 16 rpm, where the speed control is very important.

Figure 5. 14:Eggs collecting conveyor

5.3.3 Feeding system

The figure 5.15 :show the feeding system consist of hopper as the distributor connected with wheels carry the hopper connected with a 3 phase induction motor through Screw

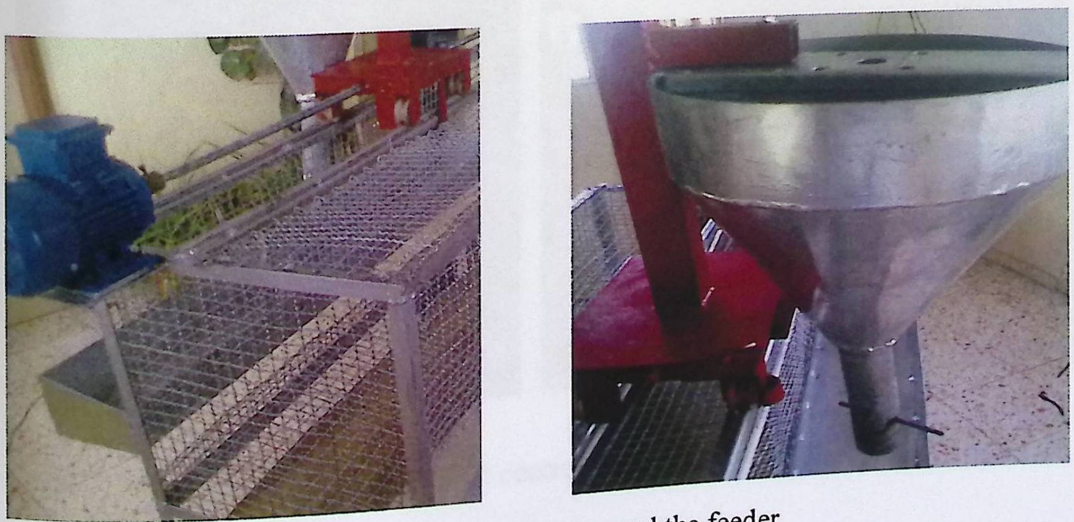


Figure 5. 15: Hopper with motor and the feeder

5.3.4 Humidity system

In figure 5.16 Represent the suction fan in the demo as a humidity controlling

Figure 5.16 : Suction fan

5.3.5 Control room

Contain the circuit barkers, over loads, contactors, relay, switches, PLC ,frequency converter.

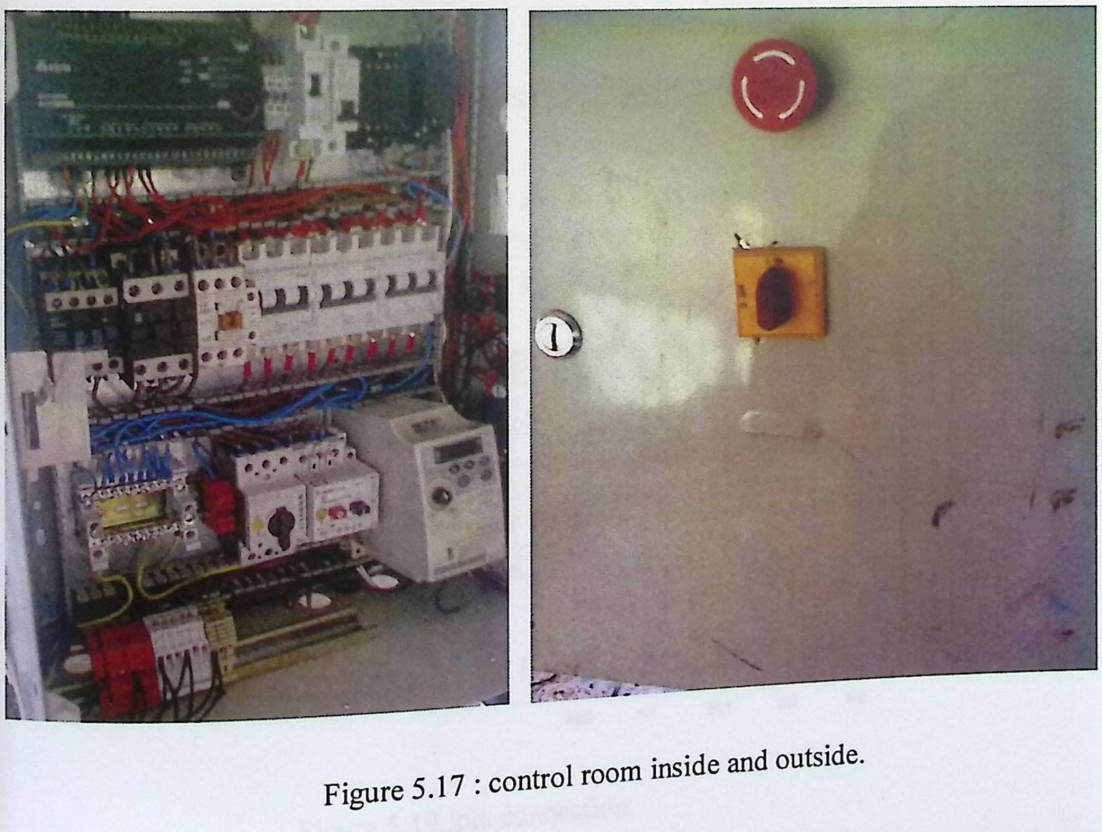


Figure 5.17 : control room inside and outside.

Allocation table:

Input table shows the input parameters of PLC as shown in table 5.1

input	comment
S0	Emergency switch
S1	On/off switch for M1
S2	On/off switch for M2
S3	On/off switch for M3
S4	Pushbutton (NO) for the fan
S5	Pushbutton (Nc) for the fan
S6	Limit switch (Ls2)
S7	Limit switch (Ls1)

Table 5.1 :input of PLC

output table shows the output parameters of PLC as shown in table 5.2

output	comment
K1 (y4)	Contactor for motor 1
K2 (y1)	Contactor for motor 2
K3(y2)	Contactor for motor 3
R1(y3)	Relay for fan
R2(y5)	Relay to reveres M1

Table 5.2 : output of PLC

5.3.6 Electrical design of demo

Power circuit contain the electrical distribution of motors as shown in figure 5.

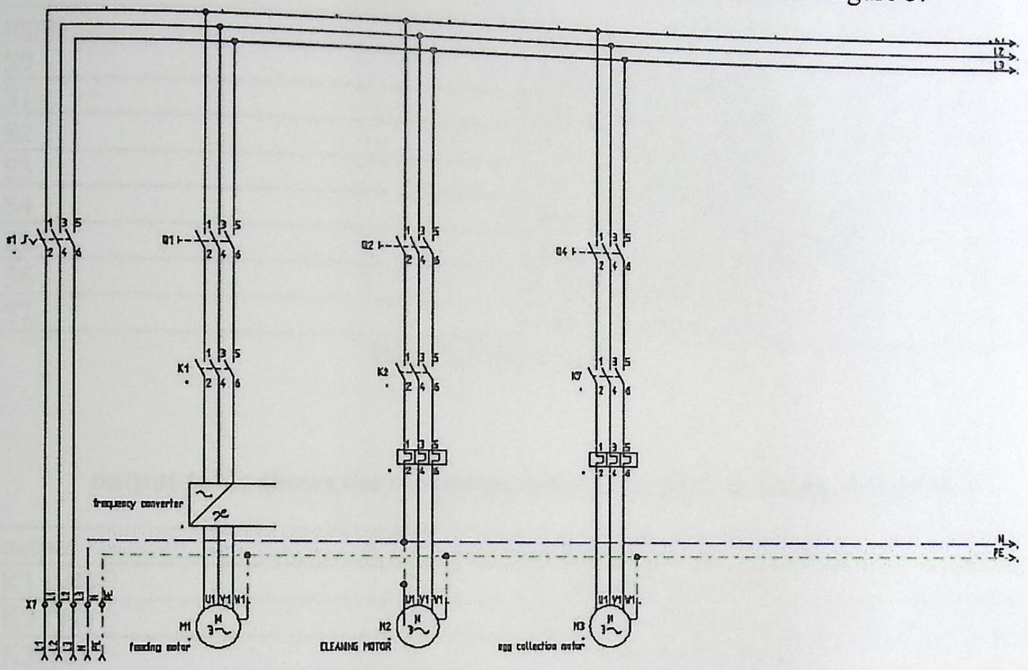


Figure 5.18 :power circuit of demo

PLC connection of demo cantains the inputs and outputs as shown in figure 5. :

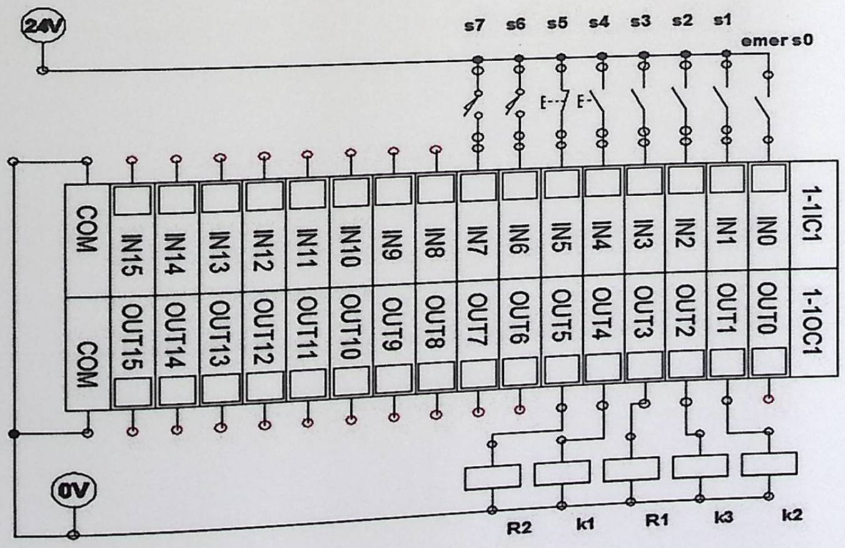


Figure 5.19 :plc connection

Allocation table:

Input table shows the input parameters of PLC as shown in table 5.1

input	comment
S0	Emergency switch
S1	On/off switch for M1
S2	On/off switch for M2
S3	On/off switch for M3
S4	Pushbutton (NO) for the fan
S5	Pushbutton (Nc) for the fan
S6	Limit switch (Ls2)
S7	Limit switch (Ls1)

Table 5.1 :input of PLC

output table shows the output parameters of PLC as shown in table 5.2

output	comment
K1 (y4)	Contactor for motor 1
K2 (y1)	Contactor for motor 2
K3(y2)	Contactor for motor 3
R1(y3)	Relay for fan
R2(y5)	Relay to reveres M1

Table 5.2 : output of PLC

6

Chapter Six

Recommendation and Conclusion

1.1 Introduction

Our project was mainly designed for improving the monitoring and controlling traditional farms inside Palestine, in order to help in the developing process for the animal production - Eggs production- inside our country.

Meanwhile, we have some suggestions and recommendations for helping on the future work, and that what the next sections presents.

1.2 System achievements

At this point, we can say that nearly all the aims of our system have been achieved; and these achievements are listed in the next few points.

- ✓ Building mobile application to control all over the system using the android language, as a human interface to the system.
- ✓ Building a full automatic system to carry out all functions that the manager usually does in his farm, like cleaning, feeding, and eggs collecting.
- ✓ Implementing remotely controlling system, which allows the manager to activate subsystems remotely.
- ✓ Implementing a wirelessly monitoring system that can monitor humidity and temperature levels, besides monitoring the surrounding area via cameras.

1.3 Real learning outcomes and conclusion

During the executing process of "Smart Farm" project, we faced a lot of challenges, challenges that we have manage to gain a lot of helpful outcomes from it, and eventually we gain a large experience in the following points:

- The team work of the project put the aims of the project and studied the theoretical part of the project (theories and laws). The team proved that the theoretical methods can be executed in real world and they can be applicable.
- Learn how to work as team and how to distribute duties for the team members.
- Learn how to programming a PIC16F877a microcontroller using PIC C software tool.
- Learn how to programming a PIC16F688 microcontroller using Micro C software tool.

- Learn how to download the codes on PIC using PC and homemade programming circuit.
- Learn how to design mobile application using Android language.
- Learn how to configure X-Bee Zig Bee device using the X-CTU program.
- Learn how to interface the CCD camera to the PIC 16F877a.
- Learn how to interface the LM35 temperature sensor to microcontroller.
- Learn how to interface and celebrate the HS1101LS humidity sensor.
- Learn how to programming the PLC Unit using the delta programming software.
- Learn how to implement the electrical circuits on board using wire Raping tool.
- Learn how to deal with deferent types of motors and there circuit.
- Learn how to use mechanical workshop in order to build the project demo.

1.4 Recommendation

As a project designer, after implementing this system, we have a several recommendations that we advice for whom intend to go ahead with this project in order to get a good performance.

- This system can be improved by making the monitoring node moving all around the farm using a conveyor in order to minimize the number of monitoring nodes.
- Also we recommend to get the X-Bee Wi-Fi especial board in order to configure the X-bee Wi-Fi, to allow the Mobile station directly communicate with the system through the IP not via the PC,
- As advantages, this system any one can use the controlling node to control in any other systems, and in very easy method you can increase the numbers of relays so that you can control in many subsystems..
- Also, this monitoring system can be used to monitor any other place for the temperature and humidity levels besides that this system can be also upgrade by using other types of sensors in order to monitor other parameters.
- The eggs collecting system can also be upgraded to be fully automated ones so that you can gave up the human interference in the arranging process of eggs.

References:

internet	[3] http://computer.howstuffworks.com/wireless-network1.htm [4] http://ipoint-tech.com/wireless-networking-wi-fi-advantages-and-disadvantages-to-wireless-networking/ [5] http://wifinotes.com/wifi-limitations.html [6] http://www.brighthubengineering.com [8] http://www.globalspec.com [9] http://www.cjmco.com/technical-info [11] http://www.aquamisr.com [12] http://www.rotaval.co.uk [14] http://www.HS1101LF.com [15] http://www.PIC16f877a.com [16] http://www.PIC16f688.com [17] http://www.delta.com.tw [18] http://www.digitalcamera.com [19] http://www.wifitranceiver.com [21] http://www.dillin.net [25] http://www.vskills.in
Books	[1], [2] pic microcontroller and embedded system Muhammad ali mazidi 2008, 822. [7] electric machinery fundamentals, by stevin J.chapman , 3rd edition [13] Mechanical Engineering Design Richard.G budynas and J.Keith Nisbet 9th edition [20] Design Of Machine Elements 3E, V. B. BHANDARI [22] Applied Mechanics, A K Hosking, M R Harris, H & H Publishing. [23] [24]GearsBy Jim McGinn & Kristoph Minchau.
Paper	[10] Bose, Bimal K. (1980). *Adjustable Speed AC Drive Systems*. New York: IEEE Press. ISBN 0-87942-146-0.

Appendix 1

Mobile application programming code:

Main activity.java

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

import android.app.Activity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }


  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    //Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
  }

  public void fn1(View v){

      EditText e1,e2;
      e1=(EditText)findViewById(R.id.username);
```

```
            e2=(EditText)findViewById(R.id.password);
            String user,pwd;
            user=e1.getText().toString();
            pwd=e2.getText().toString();
            //Toast.makeText(this, "w2",Toast.LENGTH_SHORT).show();
            if(user.equals("PPU")&&pwd.equals("PPU")){
                    setContentView(R.layout.monitoring_controlling_form);
                    //Toast.makeText(this, "Soun will
play",Toast.LENGTH_SHORT).show();

                    MediaPlayer m =MediaPlayer.create(this, R.raw.alrighty);
                    m.start();
                    //Toast.makeText(this, "Sound
Played",Toast.LENGTH_SHORT).show();

                    }
            else{
                    Toast.makeText(this, "wrong
username/pwd",Toast.LENGTH_SHORT).show();
            }
    }
    // this constructor used to open Control Activity
    public void cont(View v){
        setContentView(R.layout.control);
    }
    public void mon(View v){
        setContentView(R.layout.monitoring);
    }
    public void Temp(View v){
        setContentView(R.layout.sensors);
    }
    public void humidity(View v){
        setContentView(R.layout.humidity);
    }
    public void camera(View v){
        setContentView(R.layout.camera);
}
    public void back(View v){
        setContentView(R.layout.monitoring_controlling_form);
    }
    public void OK(View v){
        setContentView(R.layout.monitoring_controlling_form);
    }


        public void sendData(View v){
                String data="";
                switch(v.getId()){
                case R.id.cleaningTgl:
```

```
				data = "cleaning";
				break;

			case R.id.humidityTgl:
				data = "humidity";
				break;

			case R.id.tempratureTgl:
				data = "temprature";
				break;

			case R.id.feedingTgl:
				data = "feeding";
				break;

			case R.id.eggTgl:
				data = "eggCollection";
				break;
		}


			try {
				URL url  = new
URL("http://192.168.0.109:8092/SmartFarmProject/control?para="+data);
				HttpURLConnection con = (HttpURLConnection)
url.openConnection();
				String result = readResult(con.getInputStream());
				//Toast.makeText(this, result, Toast.LENGTH_LONG).show();


			}catch (Exception e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
			}
		}


	private String readResult(InputStream is) {
		String result="";
		try {
		InputStreamReader isr = new InputStreamReader(is);
		BufferedReader br = new BufferedReader(isr);

		String line;

			while((line = br.readLine())!=null)
			{
				result = result + line;
			}
```

```java
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }

                return result;
        }
        public void trigger(View x){
                switch(x.getId()){

                case R.id.TempSensBtn:
                        try {
                                URL url =
newURL("http://192.168.0.109:8092/SmartFarmProject/getData?x=Temprature");
                                HttpURLConnection con = (HttpURLConnection)
url.openConnection();

                                String result = readResult(con.getInputStream());
                                TextView tempratureTxt = (TextView)
findViewById(R.id.tempratureTxt);
                                tempratureTxt.setText(result);
                        } catch (MalformedURLException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }catch (Exception e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
                        break;

        case R.id.humiditySensBtn:
                try {
                        URL url  = new
URL("http://192.168.0.109:8092/SmartFarmProject/getData?x=Humidity");
                        HttpURLConnection con = (HttpURLConnection)
url.openConnection();
                        String result = readResult(con.getInputStream());
                        TextView txt = (TextView) findViewById(R.id.humidityTxt);
                        txt.setText(result);
                } catch (MalformedURLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }catch (Exception e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                break;
        case R.id.cameraSensBtn:
                try {
```

```
        android:hint="Enter Password" >
        <requestFocus />
    </EditText>

    <EditText
        android:id="@+id/username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/password"
        android:layout_marginBottom="55dp"
        android:ems="10"
        android:hint="Enter Your Name" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/button1"
        android:layout_alignLeft="@+id/password"
        android:layout_marginBottom="32dp"
        android:onClick="fn1"
        android:text="OK" />

</RelativeLayout>
```

Monitoring_Controlling.xml:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/s3"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="70dp"
        android:onClick="cont"
        android:text="controlling" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignRight="@+id/button2"
```

```
    android:hint="Enter Password"

    <requestFocus />
  </EditText>

  <EditText
    android:id="@+id/username"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/password"
    android:layout_marginBottom="33dp"
    android:ems="10"
    android:hint="Enter Your Name" >

  <Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button"
    android:layout_alignLeft="@+id/password"
    android:layout_marginBottom="32dp"
    android:onClick="fn1"
    android:text="OK" />

</RelativeLayout>
```

Monitoring_Controlling.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:background="[illegible]"
  android:orientation="[illegible]" >

  <Button
    android:id="[illegible]"
    android:layout_[illegible]
    android:layout_[illegible]
    android:layout_[illegible]
    android:layout_[illegible]
    android:layout_[illegible]
    android:onClick="[illegible]"
    android:text="[illegible]" />

  <Button
    android:id="[illegible]"
    android:layout_width="[illegible]"
    android:layout_height="[illegible]"
    android:layout_align[illegible]
    android:layout_align[illegible]
```

```
    android:layout_marginTop="119dp"
    android:onClick="mon"
    android:text="monitoring" />

</RelativeLayout>
```

Monitoring.xml:

```
<Button
    android:id="@+id/humidityBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button2"
    android:layout_alignParentTop="true"
    android:layout_alignRight="@+id/button2"
    android:layout_marginTop="193dp"
    android:onClick="humidity"
    android:text="Humidity" />

<Button
    android:id="@+id/backBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/cameraBtn"
    android:layout_alignRight="@+id/button2"
    android:layout_below="@+id/cameraBtn"
    android:layout_marginTop="18dp"
    android:onClick="back"
    android:text="Back" />

<Button
    android:id="@+id/cameraBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/humidityBtn"
    android:layout_alignRight="@+id/button2"
    android:layout_below="@+id/button2"
    android:layout_marginTop="22dp"
    android:onClick="camera"
    android:text="Camera" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/humidityBtn"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="72dp"
    android:onClick="Temp"
    android:text="Temperature" />
```

```
</RelativeLayout>
```

Sensors.xml:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/s3"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tempratureTxt"
        android:layout_width="169dp"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/TempSensBtn"
        android:layout_alignBottom="@+id/TempSensBtn"
        android:layout_toRightOf="@+id/TempSensBtn" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/TempSensBtn"
        android:layout_alignRight="@+id/TempSensBtn"
        android:layout_below="@+id/TempSensBtn"
        android:layout_marginTop="117dp"
        android:onClick="back"
        android:text="Back" />

    <Button
        android:id="@+id/TempSensBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="139dp"
        android:onClick="trigger"
        android:text="Temp.reading" />

</RelativeLayout>
```

Humidity.xml:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/s3"
    android:gravity="top"
    android:orientation="vertical" >
```

```
<TextView
    android:id="@+id/humidityTxt"
    android:layout_width="245dp"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/humiditySensBtn"
    android:layout_alignBottom="@+id/humiditySensBtn"
    android:layout_toRightOf="@+id/humiditySensBtn" />

<Button
    android:id="@+id/humiditySensBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button1"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="149dp"
    android:onClick="trigger"
    android:text="Humidity Reading" />

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/humiditySensBtn"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="128dp"
    android:layout_toLeftOf="@+id/humidityTxt"
    android:onClick="back"
    android:text="Back" />

</RelativeLayout>
```

Camera.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:background="@drawable/s3"
  android:orientation="vertical" >

  <Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/cameraSensBtn"
    android:layout_alignRight="@+id/cameraTxt"
    android:layout_below="@+id/cameraSensBtn"
    android:layout_marginTop="85dp"
    android:onClick="back"
    android:text="Back" />
```

```
<TextView
    android:id="@+id/cameraTxt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/cameraSensBtn"
    android:layout_alignTop="@+id/cameraSensBtn"
    android:layout_toRightOf="@+id/cameraSensBtn" />

<Button
    android:id="@+id/cameraSensBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="75dp"
    android:layout_marginTop="148dp"
    android:onClick="trigger"
    android:text="Camera Reading" />

</RelativeLayout>
```

Control.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:background="@drawable/s3"
  android:orientation="vertical" >

  <ToggleButton
    android:id="@+id/eggTgl"
    android:layout_width="134dp"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/cleaningTgl"
    android:layout_alignRight="@+id/cleaningTgl"
    android:layout_below="@+id/cleaningTgl"
    android:layout_marginTop="20dp"
    android:onClick="sendData"
    android:text="ToggleButton"
    android:textOff="Egg collecting"
    android:textOn="Egg collecting" />

  <ToggleButton
    android:id="@+id/tempratureTgl"
    android:layout_width="119dp"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/cleaningTgl"
    android:layout_alignRight="@+id/feedingTgl"
```

```
    android:layout_below="@+id/feedingTgl"
    android:layout_marginTop="16dp"
    android:onClick="sendData"
    android:text="ToggleButton"
    android:textOff="Temprature decreasing"
    android:textOn="Temprature decreasing" />

<ToggleButton
    android:id="@+id/feedingTgl"
    android:layout_width="132dp"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/cleaningTgl"
    android:layout_alignRight="@+id/eggTgl"
    android:layout_below="@+id/eggTgl"
    android:layout_marginTop="14dp"
    android:onClick="sendData"
    android:text="ToggleButton"
    android:textOff="Feeding"
    android:textOn="Feeding" />

<Button
    android:id="@+id/button1"
    android:layout_width="132dp"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/humidityTgl"
    android:layout_alignParentBottom="true"
    android:layout_alignRight="@+id/humidityTgl"
    android:layout_marginBottom="72dp"
    android:onClick="back"
    android:text="Back" />

<ToggleButton
    android:id="@+id/humidityTgl"
    android:layout_width="107dp"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/tempratureTgl"
    android:layout_alignRight="@+id/tempratureTgl"
    android:layout_below="@+id/tempratureTgl"
    android:layout_marginTop="16dp"
    android:onClick="sendData"
    android:text="ToggleButton"
    android:textOff="Humidity decreasing"
    android:textOn="Humidity decreasing" />

<ToggleButton
    android:id="@+id/cleaningTgl"
    android:layout_width="136dp"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
```

```
        android:layout_marginTop="27dp"
        android:onClick="sendData"
        android:text="ToggleButton"
        android:textOff="Cleaning"
        android:textOn="Cleaning" />

</RelativeLayout>
```

Code convert HEX. Code to picture:

```
import java.io.FileOutputStream;
import java.util.logging.Level;
import java.util.logging.Logger;


public class Main {

        /**
         * @param args
         */
        public static void main(String[] args) {
                convertToByte(
"FFD8FFE000104A46494600010200000100010000FFFE00042A00FFE2021C49434
35F50524F46494C450001010000020C6C636D73021000006D6E747252474220585
95A2007DC0001001900030029003961637370415050504C0000000000000000000
000000000000000000000000000000F6D6000100000000D32D6C636D7300000000
000000000000000000000000000000000000000000000000000000000000000000
0000000000000000A64657363000000FC0000005E637072740000015C0000000B
777470740000016800000014626B70740000017C000000147258595A000001900000
00146758595A000001A40000000146258595A000001B80000001472545243000001C
C0000004067545243000001CC0000004062545243000001CC000000406465736300
000000000000363320000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000074657874000000000464200005
8595A2000000000000000F6D6000100000000D32D58595A20000000000000000316000
00333000002A458595A2000000000000006FA2000038F50000039058595A20000000
00000062990000B785000018DA58595A2000000000000024A000000F840000B6CF
00000062990000B785000018DA58595A200000000000000024A000000F840000B6CF
637572760000000000000001A000000CB01C903630592086B0BF6103F15511B3421
637572760000000000000001A000000CB01C903630592086B0BF6103F15511B3421
F1299032183B92460551775DED6B707A0589B19A7CAC69BF7DD3C3E930FFFF
F1299032183B92460551775DED6B707A0589B19A7CAC69BF7DD3C3E930FFFF
FFDB004300080606070605080707070909080A0C140D0C0B0B0C1912130F141D1
A1F1E1D1A1C1C20242E2720222C231C1C2837292C30313434341F27393D38323
C2E333432FFDB0043010909090C0B0C180D0D1832211C213232323232323232
3232323232323232323232323232323232323232323232323232323232323232323
2323232323232FFC2001108012F00CD03002200011101021101FFC4001B0000020
301010000000000000000000000203010405000607FFC4001901000301010100000
0000000000000000000103020405FFC400190100030101010000000000000000000
000000103020405FFDA000C03000001110211000001F522491E6A0650B43D0C77
74A12D53DA816F27462D56E7BDAD3C3B332C26F2F3A9ACABDD11CB69D7F
4F80A0818AD2CED016ED67D6E4EECA2A96F0810F431A40C421E8B0C238345
65D80625AA196DD7B3191A5EA6C6E1CD17BAF2A22C8F4B86BDD8B72DDFC
```

DCFB31EAA9729DC32287240DAA721162B590360310B8286575D85B2A0390D5
924CC68F750D68509B25871C508CEAFE4FD1FA1CC772A5A1C268D44F5EC79
680F4B670B7126B52E05F4C8241AB6565585683292454BD5631AD47562E3B584
51CDEDE6C2F49E67D26C3CA554CEAD58ADA29D7A9EA3073AA97A8DE78D
8B393A33D9DA83AE120E06AB2AD080932429B66E4F70A29E5B55CED4CCF5
BCFF0037AB937616515FD79DA74ABB2561C1F4159AF34D7BE91EDEA729DC
60377258B3816732C5C1C057E25F3D79A05CF50C9DCC8F4B8FCA7A5F37ECD
3C7D5CDA72BFA2C7682A57B4FBE940362749CDD1B949136D7579E9C356104
2435F770911CDE6B4ADD5E54B197A943AF9FC6FB5F39E96B8C41D3A73A1D
EAC73E82B95540E90EC36FA4C7D9BF3C3547A91094855ABA0B1E4F69510AA
721CB71916E36559EBACBCA7ACA37FAA4B4B493C78D3A53B35D5DF8AC4A
3D0B95C360DE0A681A193DC25A9A039816066D2F418DCF595B550A0C4F6959
5B97D32A7D62B6F372D059DE30F27D7642A55F5BF3CDC31EA42BB18712B4D
A4B21144C2692E10752B6BCBCE4BABF2744CA9AF3692E5DE734AFD3D2B77
B3B4693E4D90D3F2D75421E8085A8351883244D1312038594A09640D232F7336
3405A2EE1BD4C0DA3AF614CAFB38DB3499410331EADAAFA7A5A399A8942
DB5D9618B241C7165A0F859005021939462B3472E34717768209E0ABAD9BA94
C128C74B16BAE5BB7BDE63585A8A07238EA3C19C5C98A6C20256D21241E00
845B565AA726FE36F8AEF02BB5AE526283ADA3CE1A0877F433F61E5933202E
55B47714262B7280494D641742212DAE9FCF3D0F9D8C53DD965680B42C60706
D525CD33E6C645AD2DBC2DB797B299E35A4624D70943122CAA0366B8859C
AA3E39BF57E4D7295CE594ECCF45E61A8D7D1C4D79D536904F19C5AAD6A9
DC999D048549D9B5899B59FD03BE7537E7FA152F1151BF51E5D600621080931
13CD2C9D59C0439D5CBECBF59D81A52B5CEAC03B55EA51D634AA65269374
01562521C290E58142E18D01809908439F58F3BB12B2CE9B1123289E4589AF675
9ECAD7CDA4EA9AE41E4B21CC0F0B83A0020D69F74723BA25ABC9DACCCE
C09469B0C19973C423190815BAF6A95E54A23906DAEE09898648742222413E8
B1BA8C2F45B4F9BFFFC4002A1000020201030304030003010100000000102000
311101231042132132022412333420514304315FFDA0008010000010502D3AB6C9
D470DE15F9369F46B6AA002C955B880E6594775B327B5808ED60DC959DD5D5
C74F0FEE1C4632EF2D4707C5393A6278CB13316D578ACD590D99656AE37354
7B58A460D5D9D7B5B5F6B6CFDCBC18DDE5A7F2EA3C4F8D7C9D4CC4C89B
7014ED81A30EDB4D67B5AAC36DADDAEE2FBBCD3C4C699CBEA38FE6BF23
0686621E3254EE0D06E105B89CCFB216D170C423F2BD26C82A00132E7C01E5
A8F1FE6BE4C1EC3C110F009581B33389BF300CCDB9006B4359BAE83C86A38
FE539307B0C31B8890E62F66513333AAFA74AD8C1997CB5DEA2060427260F6
1D1B895C221111FE21B5372094FED117CB21437550BB185B115DD4D5D56187
CD47B0E8DC448618BD99607CB9DC594775F31370596F526D2263B2D1BE59D9
8094B9560C61EA176D5BB6C3A36890C32B4CCC6211F90FEC3D9E0E2FB77B4
51994D4CC5ABD95DC20ED0012A6CCAEBD4CC42B99E998A9B66D9B62FC53
77CDBC9C76B3F646B1BFD7F4889E936EA680A36812CEE2D49B713BB914946
C607B31A1D0E9FF00B379D83E17794404D748CB85101D5D44357AEDD32EC27
065376FD3EBD867D18BC8F23FB1876B4674A2B07A75ED6AC2768B3AC668AE
C2057BE0181B40244E973FED4FAF619F4601386FFD0C613ED0616FFC6DEBDB
2AB1AF9E97709F24AF6A43C6FED5F4FE822D6AEA6985187B0CFA3CD5E2C7
1079347F35ED609D7062456A56918B18662100961 8CCFAA86EB9FC578D310D6
A6356443EC3CFDBF8D9E5D501FF00D212DAB336626E0A4DD9989C13A74699
B1F81EC3398C81A3A14838CFCB998C0B3C2C851D7AF11B9B172A7B84301533
1DB3002C69AFD2ACC07D8D383C8C643D4522F73811E59E0D2CE909EAE1E62
D1BACB2964651885C6333A5A3609F7FD7F5A1E0F7034E46CD96478FE1FCC22
D1BACB2964651885C6333A5A3609F7FD7F5A1E0F7034E46CD96478FE1FCC22

32E94AF7B2B5B03F46C2580D53A335B5BA7D9E4F0A7B6741ECB4778F1FC57
C21E4E95FEB8D3ABA77D35F1D13D9BBD5303E498C7E038D043CCFA6F943D
8B9F916ECBD961EEDF4F3A73F866211DFD3287A22037D81A1E3D9F660E3EE
D58CDB6DC030B05D3EE3F8D0729A19D4281786D96839238E3DA60E5B81C7F
52FE9F74F53F2E9FD4613A46F90D3EBAAED68618E9DB20706083DC3B0D072
F484BF41A3C418EA04FA13AEC654AECA0E6E1391C45E3333A111634DB3131
1D438DBB27D0D0F09FB47633EFAE6F9AB616BBD12C5399F6C20F68F269969F
2986855A1439F6229F57EA1E7AC2A7A8ACA88CBEAAD0DF8D6CC98090C1A6
756F23A6664CC3186BF6D641234FABCE6EADF12A605826DB8A66069F6BEC6
F71329EA05D3D369F35999520D0CC6D2FB4D8840349F9080F6C4099331ABF19
F61221BEA582749D66ED0095F11A3427B83F2E9D87A8341F2807B3B88C7720D
09C4B7FC8515B755D7611EC67304E97A8F5533887A9305AF12FCCB3838CFDD
0DF93E9AF512BEA040411A9E1B4679D57575D4ACF93EC0FB0D5D4FAD4D5F
284614982DCD248C8466888CA7B9D7D4D913AD103061B84C89BBBF57FE4155
5EC7B0CC42B338D04FB07695EA4E77A95F4F3055022898D77010BC670A1FAB
113ABB567FB574FF6AE8589F66E99F7D36293A66661686D86D8F79019CB9D73
A6666675CCCCFAFA1AD7D4BA45BD6C9BA178D6A88FD5AC6EA5DA649F76
75CFB01820E071AF04DA673130259FB20F6E7FE226741ECCC0DD81978C5914
CCFB0C061F73D7B6CF60D72206104EA38F791A1D6BA2CB655D05216B50CD7
26C61A8D3131A0E2F7FF8661CE95D4D69AFFC78408A02EC13FFC4002211000
10304030101010100000000000000000100021103102031122130044122251FFDA00080
10211013F01C41841D2B56AB4F92220DBF7C03FFDBD5A7C941D2141C5111E1
283D35D2A2D378CDA60D9FF00471D5C045B16246F1689282FA04390DA6B254
22254770AA338EB165BE80BF535D089941A542AC7A8C59AB571D2FD436BA0
A54AA87FAC5BAB541211DD98FB39F1931D63807944E4D3DA95389CC3BAC0
EEC7C0627D4F9F0444236E0729BB7B4E64A14906816706F80E90758B823514F9
39C7D4F842FFC400231100020201040301000300000000000000000102111003122
02104303151224161FFDA000801011013F01E2D264A34563475B67D14AD5AC4
B0F94B4BF0712A8D1D6D8FF00C372AB27E4C109D8F9D263D2FC270DACB79
79BE0B1A91B58D3F1372B6F0CBA13B2ACEFE0B86ACA963C595C063950D917
46EE88BEF8EB63C37D3431C6CE97D1B459A7C75BE891E23FE4324CA1AC417
5C67F71A2EA586550C8C62C2E0BA1DF2BA9753B20D429B95A6A637174EE51
EA3847A2B8D48A4203280723C7A4C22C912855BEFA32808DF3B6868E9F50816
8F808699A0737D1466F13C3E2F41466C992F080178D6263BE50A47193C3C48FF
C46C7F3AD05465152983B2703E235097CA91A5B68CD651946DFE28360ECA58
2DEADE34065DD6FA04709DB6EBCE137AE7B21E237D7CC14A83BABEA4630
D770BCEB79CB8BE51B7AF0B6FE1017DA90AEB7B69DCD3078A6FE8576CDE
60BCC1018F44A14B9586492BA7C4B9D8052E7126B85DE61486ECBCC543BE51
5B7F0B642C696B950E6FC290672C8A6E02E938DFF00E949CB20DD126CE42AE
9DC05E756532AE6B20A8788EEA419152B078577F3C2973893DF42421880FC2D
5B8868561451357F931BFE9567E05B160EA553494E170A24BC3B14246885061A
62FCCCC8F42521FC97C492DBC0D46616DCCCF22CF68983FF00823C80F32133
77A10532CE8428CEDDB2ED1640E97912E3418544C6AE5CA1428B5125128C38
D257A092B2567905C5A28D3834DA2064378F22B210A89DBF65016C78E711127
6F8E6E13E11A2043881E07FC171A998CFA785B189C9A2837A45A42D78252DF
022239C090D49AC27484B9E619C7B56CB727D74271C557E38CD70AFEAE1E05
D0C8B035686D2CB2BC9AC2F57443312D5E824B843F542174D7B3067EE188DB
3464F8B5068C4490C47204D7A157E59EA83E0350161C8E758D605A211EB8CD6
82E0EAF763FE732EB996B4F10580B68E1704556B72FC5865A50978625665D734
50021C54B598FC40A026018232DF4A7FFFD9");

```
    }
    public static void convertToByte( String hex){


        int len = hex.length();
        byte[] data = new byte[len / 2];
        for (int i = 0; i < len; i += 2) {
            data[i / 2] = (byte) ((Character.digit(hex.charAt(i), 16) << 4)+
Character.digit(hex.charAt(i+1), 16));
        }

            try {
                FileOutputStream fileOuputStream =
                    new FileOutputStream("C://picture1.jpg");
                fileOuputStream.write(data);
                fileOuputStream.close();

                System.out.println("Done");

            } catch (Exception ex) {

            } finally {

            }
        }
}
```

R.jave:

```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.

 * This class was automatically generated by the

 * aapt tool from the resource data it found.  It

 * should not be modified by hand.

 */

package com.example.tareq;

public final class R {

   public static final class attr {

   }
```

```java
public static final class dimen {
    /**  Default screen margins, per the Android Design guidelines.
    Customize dimensions originally defined in res/values/dimens.xml (such as
    screen margins) for sw720dp devices (e.g. 10" tablets) in landscape here.
    */
    public static final int activity_horizontal_margin=0x7f050000;
    public static final int activity_vertical_margin=0x7f050001;
}
public static final class drawable {
    public static final int ic_launcher=0x7f020000;
    public static final int s2=0x7f020001;
    public static final int s3=0x7f020002;
    public static final int s5=0x7f020003;
    public static final int ss2=0x7f020004;
    public static final int ss3=0x7f020005;
    public static final int ssss=0x7f020006;
}
public static final class id {
    public static final int TempSensBtn=0x7f090011;
    public static final int action_settings=0x7f090012;
    public static final int backBtn=0x7f09000e;
    public static final int button1=0x7f090000;
    public static final int button2=0x7f090003;
    public static final int cameraBtn=0x7f09000f;
    public static final int cameraSensBtn=0x7f090004;
    public static final int cameraTxt=0x7f090005;
    public static final int cleaningTgl=0x7f090007;
```

```
    public static final int eggTgl=0x7f090006;
    public static final int feedingTgl=0x7f090009;
    public static final int humidityBtn=0x7f09000d;
    public static final int humiditySensBtn=0x7f09000c;
    public static final int humidityTgl=0x7f09000a;
    public static final int humidityTxt=0x7f09000b;
    public static final int password=0x7f090001;
    public static final int tempratureTgl=0x7f090008;
    public static final int tempratureTxt=0x7f090010;
    public static final int username=0x7f090002;
}
public static final class layout {
    public static final int activity_main=0x7f030000;
    public static final int camera=0x7f030001;
    public static final int control=0x7f030002;
    public static final int humidity=0x7f030003;
    public static final int monitoring=0x7f030004;
    public static final int monitoring_controlling_form=0x7f030005;
    public static final int sensors=0x7f030006;
}
public static final class menu {
    public static final int main=0x7f080000;
}
public static final class raw {
    public static final int alrighty=0x7f040000;
    public static final int music=0x7f040001;
}
```

```
public static final class string {
    public static final int Button=0x7f060007;
    public static final int action_settings=0x7f060001;
    public static final int app_name=0x7f060000;
    public static final int choice=0x7f060003;
    public static final int hello_world=0x7f060002;
    public static final int pwd=0x7f060005;
    public static final int sensors=0x7f060006;
    public static final int user=0x7f060004;
}
public static final class style {
    /**
    Base application theme, dependent on API level. This theme is replaced
    by AppBaseTheme from res/values-vXX/styles.xml on newer devices.

        Theme customizations available in newer API levels can go in
        res/values-vXX/styles.xml, while customizations related to
        backward-compatibility can go here.

    public static final int AppBaseTheme=0x7f070000;
All customizations that are NOT specific to a particular API-level can go here.
    public static final int AppTheme=0x7f070001;
}
}
```

Control.java in net beans program:

```java
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class control extends HttpServlet {
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    //response.getWriter().write(request.getParameter("para"));
    System.err.println(request.getParameter("para"));

  }
protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
  }
}
```

GetData.java from net beans program:

```java
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class getData extends HttpServlet {
 protected void doGet(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {
      String requiredData;
      requiredData= request.getParameter("x");
      if(requiredData.equals("Humidity")){
         File f = new File ("c:\\humidity.txt");
         FileInputStream fis = new FileInputStream(f);
         InputStreamReader isr = new InputStreamReader(fis);
        BufferedReader br = new BufferedReader(isr);
       String humidity="", line;
        while((line = br.readLine())!= null){
           humidity = humidity + line;
        }
        response.getWriter().write("Humidity : " +humidity);
        }
else if(requiredData.equals("Tempreture")) {
        File f = new File ("c:\\temprature.txt");
        FileInputStream fis = new FileInputStream(f);
        InputStreamReader isr = new InputStreamReader(fis);
       BufferedReader br = new BufferedReader(isr);
       String humidity="", line;
       while((line = br.readLine())!= null){
      humidity = humidity + line;
       }
```

```
        response.getWriter().write("Tempreture : " +humidity);
    }else if(requiredData.equals("Picture")) {
    }
    }

protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
  }
```

Appendix 2

PIC16F877a programming "Controlling Node"

```
#include "C:\Users\shareef suliman\Desktop\shareef\c.h"
#include <math.h>
#include <string.h>
char c=0;
void main()
{
setup_adc_ports(NO_ANALOGS);
setup_adc(ADC_OFF);
setup_psp(PSP_DISABLED);
setup_spi(FALSE);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);
// TODO: USER CODE!!
SET_TRIS_B(0x00);
delay_ms(100);
output_b(0x00);
delay_ms(1);
while(1){
restart_wdt();                          // reed the command.

if(c=='C'){                             // cleaning on.
output_high(PIN_b0);
} else if(c=='F')  {                    // Feeding is on.
output_high(PIN_b1);
}
else if(c=='E'){                        // Eggs collecting is on .
output_high(PIN_b2);
}
else if(c=='T'){                        // Temperature decreasing on.
output_high(PIN_b3);
}
else if(c=='H'){                        // Humidity decreasing on.
output_high(PIN_b4);
}
else if(c=='c'){                        // Cleaning is off.
output_low(PIN_b0);
}
else if(c=='f'){                        // Feeding is off.
output_low(PIN_b1);
}
else if(c=='e'){                        // Eggs collecting is off.
output_low(PIN_b2);
}
else if(c=='t'){                        // stop decreasing temperature.
output_low(PIN_b3);
}
```

Appendix 3

PIC16F877a programming "Monitoring Node"

```
#include <math.h>
#include <string.h>
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,restart_wdt,stream=Xbee)
#use
rs232(baud=9600,parity=N,xmit=PIN_b0,rcv=PIN_b1,bits=8,restart_wdt,stream=Camera)
long Value;
float Voltage=0;
int8 duty;
int8 pr2_value;
int16 isr_ccp_delta;
float frequency;
int16 current_ccp;
int16 old_ccp = 0;
float k0=213.41353819;
float k1=-323.21980003;
float k2=434.26410259;
float HM;
char c=0;
char c1=0;
#int_RDA
RDA_isr()
{ c=getc(xbee);
} #
int_ccp1
void ccp1_isr(void)
{ current_ccp =
CCP_1;
isr_ccp_delta = current_ccp - old_ccp;
old_ccp = current_ccp;
} void main()
{
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,restart_wdt,stream=Xbee)
#use
rs232(baud=9600,parity=N,xmit=PIN_b0,rcv=PIN_b1,bits=8,restart_wdt,stream=Camera)
setup_adc_ports(AN0);
setup_adc(ADC_OFF);
setup_psp(PSP_DISABLED);
setup_spi(FALSE);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
setup_timer_2(T2_DISABLED,0,1);
setup_ccp1(CCP_CAPTURE_RE);
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);
enable_interrupts(INT_RDA);
enable_interrupts(INT_CCP1);
enable_interrupts(GLOBAL);
```

```
clear_interrupt(INT_CCP1);
// TODO: USER CODE!!
while(1){                                    // start: ready to get data .
c=getc(Xbee);
if(c=='L'){                                  // temperature value.
set_adc_channel(0);
delay_ms(20);
value=read_adc();

Voltage=value*0.49;
printf("The Temp is %1f C",Voltage);
} else if(c=='M'){                           // humidity value.
clear_interrupt(INT_CCP1);
frequency = (1000000/isr_ccp_delta);
frequency=4*frequency/1000;
HM=k0+k1*(frequency)+k2*(frequency*frequency);
fprintf(Xbee,"xbeeThe Freq is %2f \n\r the HM is %2f \n\r",frequency,HM);
set_timer1(0);
}
else if (c=='N'){                            // camera reading.
fprintf(Camera,"");
while(c!='n'){
c=fgetc(Camera);
fprintf(Xbee,"%u",c);
}}
```

Appendix 4

Appendix A: conveyor belt catalog from MRF Company

CONVEYOR BELTS

TECHNICAL DATA – EP BELTS

Belt Designation EP or PN (Polyester - Nylon)	Recommended Maximum Belt Tension (RMBT)	Nominal Carcass Thickness	Nominal Carcass Weight	Maximum belt width (mm) for satisfactory load support for material bulk density upto (t/m^3)			Minimum Belt Width (mm) for Adequate Troughing of Empty Belt		
* Rating	(kN/m)	(mm)	(kg/m^2)	1.0	1.6	2.5	20° idlers	30°/ 35° idlers	45° idlers
200/2	20	2.2	3.54	650	500	400	250	300	350
250/2	25	2.6	3.90	800	650	500	300	350	400
315/2	31	2.6	4.00	800	650	500	300	350	400
400/2	40	3.2	4.96	1000	800	650	350	400	450
500/2	50	3.8	5.28	1000	800	650	350	400	450
630/2	63	4.6	6.20	1200	1000	800	400	500	600
250/3	25	2.8	3.79	1000	800	650	350	400	450
315/3	31	2.8	3.93	1000	800	650	350	400	450
400/3	40	3.4	4.29	1000	800	650	350	450	500
400/4	40	3.9	5.24	1200	1000	800	400	500	600
500/3	50	3.4	4.44	1200	900	800	350	450	500
500/4	55	4.7	5.72	1400	1200	1000	400	500	600
630/3	63	4.0	5.07	1200	1000	800	400	500	600
630/4	70	4.7	5.92	1400	1200	1000	450	550	650
630/5	70	6.0	7.15	1800	1600	1400	500	650	750

Appendix B: sprocket gear type QD catalog from U.S. Tsubaki, company

Appendix 4

Appendix A: conveyor belt catalog from MRF Company

CONVEYOR BELTS

TECHNICAL DATA – EP BELTS

Belt Designation EP or PN (Polyester – Nylon) * Rating	Recommended Maximum Belt Tension (RMBT) (kN/m)	Nominal Carcass Thickness (mm)	Nominal Carcass Weight (kg/m²)	Maximum belt width (mm) for satisfactory load support for material bulk density upto (t/m³)			Minimum Belt Width (mm) for Adequate Troughing of Empty Belt		
				1.0	1.6	2.5	20° idlers	30°/35° idlers	45° idlers
200/2	20	2.2	3.54	650	500	400	250	300	350
250/2	25	2.6	3.90	800	650	500	300	350	400
315/2	31	2.6	4.00	800	650	500	300	350	400
400/2	40	3.2	4.96	1000	800	650	350	400	450
500/2	50	3.8	5.28	1000	800	650	350	400	450
630/2	63	4.6	6.20	1200	1000	800	400	500	600
250/3	25	2.8	3.79	1000	800	650	350	400	450
315/3	31	2.8	3.93	1000	800	650	350	400	450
400/3	40	3.4	4.29	1000	800	650	350	450	500
400/4	40	3.9	5.24	1200	1000	800	400	500	600
500/3	50	3.4	4.44	1200	900	800	350	450	500
500/4	56	4.7	5.72	1400	1200	1000	400	500	600
630/3	63	4.0	5.07	1200	1000	800	400	500	600
630/4	70	4.7	5.92	1400	1200	1000	450	550	650
630/5	70	6.0	7.15	1800	1600	1400	500	650	750

Appendix B: sprocket gear type QD catalog from U.S. Tsubaki, company

No. Teeth	Outside Diameter	Catalog Number	List Price	Bushing	Max. Bore	Type	Hub Dia.	LTB	X	Y	L	F	Wt. Lbs.
					40 QD* 1/2" Pitch								
15	2.652	40JA15	$37.80	JA	1 1/4	B	2 1/16	9/16	1	1	19/32	2	.30
16	2.814	40JA16	39.60	JA	1 1/4	B	2 1/16	9/16	1	1	19/32	2	.40
17	2.974	40JA17	43.20	JA	1 1/4	B	2 3/16	9/16	1	1	19/32	2	.50
18	3.136	40JA18	46.20	JA	1 1/4	B	2 1/4	9/16	1	1	19/32	2	.50
19	3.292	40JA19	49.40	JA	1 1/4	B	2 1/4	9/16	1	1	19/32	2	.60
20	3.457	40SH20	53.40	SH	1 11/16	B	2 3/4	13/16	1 3/8	1 3/8	29/32	2 11/16	.60
21	3.618	40SH21	57.80	SH	1 11/16	B	2 3/4	13/16	1 3/8	1 3/8	29/32	2 11/16	.80
22	3.778	40SH22	65.00	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.10
23	3.938	40SH23	69.40	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.10
24	4.098	40SH24	73.00	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.20
25	4.258	40SH25	77.40	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.30
26	4.418	40SH26	81.00	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.40
27	4.578	40SH27	82.80	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.50
28	4.738	40SH28	85.40	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.60
30	5.057	40SH30	88.20	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.70
32	5.376	40SH32	97.40	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	1.90
35	5.856	40SH35	100.60	SH	1 11/16	B	3	13/16	1 3/8	1 3/8	29/32	2 11/16	2.20
36	6.015	40SDS36	101.00	SDS	2	B	3 1/2	3/4	1 7/16	1 7/16	31/32	3 3/16	2.30
40	6.653	40SDS40	111.20	SDS	2	B	3 1/2	3/4	1 7/16	1 7/16	31/32	3 3/16	2.80
42	6.972	40SDS42	113.40	SDS	2	B	3 1/2	3/4	1 7/16	1 7/16	31/32	3 3/16	3.00
45	7.450	40SDS45	115.60	SDS	2	B	3 1/2	3/4	1 7/16	1 7/16	31/32	3 3/16	3.40
48	7.928	40SDS48	120.60	SDS	2	B	3 1/2	3/4	1 7/16	1 7/16	31/32	3 3/16	3.90
54	8.884	40SDS54	126.80	SDS	2	B	3 1/2	3/4	1 7/16	1 7/16	31/32	3 3/16	4.70
60	9.840	40SDS60	132.60	SDS	2	B	3 1/2	3/4	1 7/16	1 7/16	31/32	3 3/16	5.80
70	11.433	40SK70	134.80	SK	2 5/8	B	4 1/4	1 1/4	2 1/16	2 1/16	1 17/32	3 7/8	9.00
72	11.752	40SK72	135.80	SK	2 5/8	B	4 1/4	1 1/4	2 1/16	2 1/16	1 17/32	3 7/8	9.10

Appendix C: sprocket gear type double strand catalog from U.S. Tsubaki, company

Dimensions			Double Strand						Triple Strand					
No. Teeth	Outside Diameter	Plain Bore	Catalog Number	List Price	Hub Dia.	LTB	†Max. Bore	Wt. Lbs.	Catalog Number	List Price	Hub Dia.	LTB	†Max. Bore	Wt. Lbs.
11	2.003	1/2	D40B11	$68.60	1 7/16▲	1 1/2	3/4	.60	T40B11	$103.20	1 7/16▲	2 1/8	3/4	.80
12	2.166	1/2	D40B12	69.00	1 9/16▲	1 1/2	7/8	.70	T40B12	104.60	1 9/16▲	2 1/8	7/8	1.10
13	2.328	1/2	D40B13	69.80	1 1/2	1 1/2	1	.80	T40B13	105.00	1 1/2	2 1/8	1	1.20
14	2.490	1/2	D40B14	71.20	1 21/32	1 1/2	1 1/8	1.00	T40B14	105.00	1 21/32	2 1/8	1 1/8	1.50
15	2.650	1/2	D40B15	71.20	1 13/16	1 1/2	1 1/4	1.20	T40B15	105.80	1 13/16	2 1/8	1 1/4	1.70
16	2.814	5/8	D40B16	71.60	1 31/32	1 1/2	1 3/8	1.40	T40B16	108.60	1 31/32	2 1/8	1 3/8	2.00
17	2.974	5/8	D40B17	73.80	2 5/32	1 1/2	1 7/16	1.60	T40B17	113.80	2 5/32	2 1/8	1 7/16	2.40
18	3.136	5/8	D40B18	78.00	2 5/16	1 1/2	1 1/2	1.90	T40B18	120.60	2 5/16	2 1/8	1 1/2	2.70
19	3.292	5/8	D40B19	81.80	2 15/32	1 1/2	1 5/8	2.10	T40B19	126.40	2 15/32	2 1/8	1 5/8	3.10
20	3.457	5/8	D40B20	87.20	2 5/8	1 5/8	1 3/4	2.60	T40B20	134.00	2 5/8	2 1/4	1 3/4	3.70
21	3.618	5/8	D40B21	91.20	2 25/32	1 5/8	1 7/8	2.90	T40B21	142.00	2 25/32	2 1/4	1 7/8	4.10
22	3.778	5/8	D40B22	94.80	2 15/16	1 5/8	1 7/8	3.30	T40B22	149.00	2 15/16	2 1/4	1 7/8	4.60
23	3.938	5/8	D40B23	101.00	3 3/32	1 5/8	2	3.60	T40B23	156.60	3 3/32	2 1/4	2	5.10
24	4.098	5/8	D40B24	106.40	3 9/32	1 5/8	2 1/4	4.00	T40B24	164.60	3 9/32	2 1/4	2 1/4	5.60
25	4.258	5/8	D40B25	109.00	3 7/16	1 5/8	2 1/4	4.40	T40B25	168.60	3 7/16	2 1/4	2 1/4	6.20
26	4.418	5/8	D40B26	109.00	3 19/32	1 5/8	2 3/8	4.80	T40B26	168.60	3 19/32	2 1/4	2 3/8	6.80
30	5.057	11/16	D40B30	111.20	3 1/4	1 5/8	2 1/4	5.60	T40B30	174.00	3 1/4	2 1/4	2 1/4	8.00
35	5.856	11/16	D40B35	129.00	3 1/4	1 5/8	2 1/4	7.10	T40B35	191.80	3 1/4	2 1/4	2 1/4	10.50
36	6.015	15/16	D40B36	134.40	3 3/4	1 3/4	2 1/2	9.40	T40B36	210.40	3 3/4	2 3/8	2 1/2	12.20
40	6.653	15/16	D40B40	153.20	3 3/4	1 3/4	2 1/2	10.80						
45	7.450	15/16	D40B45	190.00	3 3/4	1 3/4	2 1/2	12.90						
48	7.928	15/16	D40B48	198.00	3 3/4	1 3/4	2 1/2	14.20						
54	8.884	15/16	D40B54	228.60	3 3/4	1 3/4	2 1/2	17.20						
60	9.840	15/16	D40B60	235.40	3 3/4	1 3/4	2 1/2	20.60						

† Dimensions shown allow for standard keyway with set screw at 90°

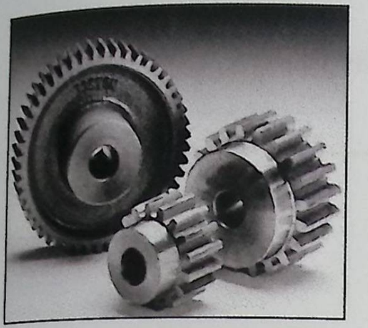
▲Has recessed groove in hub for chain clearance

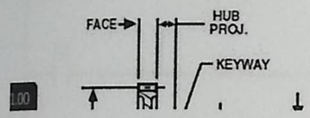
AppendixD:spur gears catalog from Boston gear company .

SPUR GEARS

10 AND 8 DIAMETRAL PITCH
CAST IRON, STEEL AND NON-METALLIC

$14\frac{1}{2}$° PRESSURE ANGLE
(Will not operate with 20° spurs)





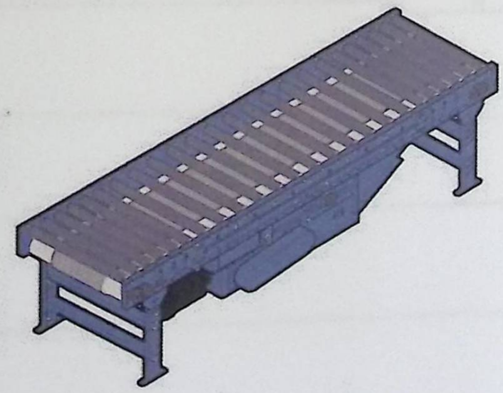
ALL DIMENSIONS IN INCHES
ORDER BY CATALOG NUMBER OR ITEM CODE

No. of Teeth	Pitch Dia.	Bore	Hub Dia.	Hub Proj.	Style See Page 150	Without Keyway or Setscrew: Catalog Number	Without Keyway or Setscrew: Item Code	With Keyway and Setscrew†: Catalog Number	With Keyway and Setscrew†: Item Code
10 DIAMETRAL PITCH						Face = 1.000" Outside Dia. = Pitch Dia. + .200" Overall Length = 1.000" + Hub Proj.			
CAST IRON									
40	4.000					NF40	10320	–	–
42	4.200					NF42	10322	–	–
45	4.500				B	NF45	10324	–	–
48	4.800					NF48	10326	–	–
50	5.000					NF50	10328	–	–
54	5.400		2.12		C	NF54	10330	–	–
55	5.500	.875		.88		NF55	10332	–	–
60	6.000					NF60	10334	–	–
64	6.400					NF64	10336	–	–
70	7.000					NF70	10338	–	–
72	7.200					NF72	10340	–	–
80	8.000					NF80	10342	–	–
84	8.400					NF84	10344	–	–
90	9.000		2.25		D	NF90	10346	–	–
96	9.600					NF96	10348	–	–

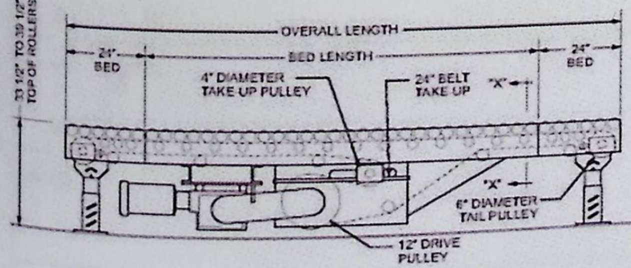
Appendix E:roller belt conveyor catalog from hytrol conveyor company

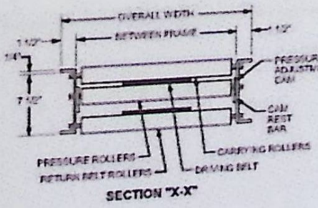
6 **Belt Driven Live Roller: Heavy Duty Conveyor**

LOAD CAPACITY CHART							
30 FPM Roller Speed							
Diameter Drive Pulley	HP	B/F WIDTH 27" to 31"		B/F WIDTH 33" to 51"		B/F WIDTH 57" to 61"	
		50' OAL	100' OAL	50' OAL	100' OAL	50' OAL	100' OAL
12"	2	8000 lbs	6000 lbs	600 lbs	--	5000 lbs	--
16"	3	15000 lbs	12000 lbs	12000 lbs	8000 lbs	12000 lbs	6000 lbs
	5	18000 lbs	17000 lbs	17000 lbs	12000 lbs	16000 lbs	11000 lbs



Standard Available Belt Widths:
27 in., 31 in., 33 in., 39 in., 45 in., 51 in., 57 in., 61 in.





Applications
- Heavy Duty Conveying
- Rugged Design
- Limited Accumulation
- Easy Installation and adjustment

Appendix: PLC program