The Islamic University of Gaza
Deanery of Graduate Studies
Faculty of Engineering
Electrical Engineering Department

الجامعة الإسلامية-غزة
عمادة الدراسات العليا
كلية الهندسة
قسم الهندسة الكهربائية

Design of an Automatic Synchronizing Device for Dual-Electrical Generators Based on CAN Protocol

By

Ahmad I. Abo Dabowsa

Supervisors

Prof. Dr. Mohammed Abdelati

&

Dr. Iyad M. Abu Hadrous

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

June
1432-2011

صفحة نتيجة الحكم على البحث (نتيجة الحكم من قبل لجنة المناقشة)

ABSTRACT

Automatic synchronizers are widely employed to connect more than one synchronous generator. The main task of a synchronizer is to capture the voltage, frequency and phase differences between the generators accurately and rapidly. This thesis introduces a new automatic synchronizer for dual generators based on CAN “controller area network” protocol. The device consists of three independent microcontroller modules connected with CAN protocol. One of them is called the circuit control breaker, it is responsible for closing the circuit breaker between the generators. The other two modules are assigned to capture the voltage, frequency and phase differences for each generator due to a reference signals, and control the governor of the generators to much the acceptable limits. Due to the fastness and error detection techniques of CAN protocol, the probability of false synchronization decision is minimized which is the main contribution of the thesis, therefore, the reliability of the synchronization is improved. The developed automatic synchronization unit is fast, cost effective, reliable and precise to be used for monitoring, measuring and parallel operations of the synchronous generators.

DEDICATION

To all my family members who have been a constant source of motivation, inspiration, and support.

ACKNOWLEDGEMENT

I thank Allah, the Lord of the worlds, for His mercy and limitless help and guidance. May peace and blessings be upon Mohammed the last of the messengers.

I would like to express my deep appreciation to my advisors Dr. Iyad Abu Hadrous and Prof. Dr. Muhammed Abdelati for providing advice, support and excellent guidance. The warm discussions and regular meetings I had with them during this research, contributed greatly to the successful completion of this research.

There are no words that can describe how grateful I am to my family specially my wife for her support and encouragement through the years. My deepest thanks go to my brothers, my sisters and my daughter for their patience and understanding during my busy schedule.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

AC	Alternating Current
A/D	Analog to Digital
ADC	Analog to Digital Converter
DC	Direct Current
GND	Ground
I/O	Input / Output
Op-Amp	Operational Amplifier
DSP	Digital Signal Processing
CAN	Controller Area Network
SG	Synchronous Generator
Profibus	Process Field Bus
Rpm	Revolutions Per Minute
Hz	Hertz
kV	Kilo Volts
MVA	Mega Volt- Amperes
RTR	Remote Transmit Request
CSMA	Carrier-Sense Multiple Access
CDAMP	Collision Detection with Arbitration on Message Priority
OSI	Open Systems Interconnection
DLL	Data Link Layer
LLC	Logical Link Control
MAC	Medium Access Control
PSL	Physical Signaling Layer
PMA	Physical Medium Attachment
MDI	Medium Dependent Interface
SOF	Start of Frame
RTR	Remote Transmission Request
EOF	end of frame
MCP	CAN Controller Chip
LCD	Liquid Crystal Display
FPGA	Field-Programmable Gate Array
PLC	Programmable Logic Control

PLD	Programmable Logic Device
CPU	Central Processing Unit
ISO	International Organization for Standardization
RS232	Recommended Standard 232
RS485	Recommended Standard 485
EMC	Electromagnetic Compatibility

CHAPTER 1 INTRODUCTION

Electrical power system mainly consists of a generator, transmission lines, and supplies large numbers of widely distributed loads. In many cases, there is a need to connect more than one generator to the system. Some of the benefits of operating multiple generators in parallel include increased reliability, expandability, flexibility, serviceability and efficiency. Parallel operation allows operating generators around their rated load resulting in operating with high efficiency [1, 2].

When connecting a generator to an interconnected system containing many other generators, the voltage, phase and frequency at its terminals should meet the operating ones. Severe damage to the generator as well as system disturbances may result if the generator is allowed to connect to the system outside of established safe levels. Therefore, the automatic synchronizing device plays an important role in the generator synchronizing. For many years, the researchers have paid great attention to develop the high performance-synchronizing device [3].

The main problems of connecting a synchronous generator to an electrical system to establish safe limits for each of the delta phase angle, the delta frequency, and the delta voltage magnitude are summarized as:

1) Delta Phase Angle

Connecting a generator to an electrical system with a high delta phase angle between the generator and the system causes a shock to the generator and system when the generator is forced to try to instantly synchronize with the system. This shock creates stress on the generator’s prime shaft, with sub sequential shaft exhaustion. The greater the phase angle difference, the more severe the shock will be. Several shocks or a single severe shock can lead to or cause shaft failure.

Excessive phase angle difference can also cause high stator winding currents, which could cause damage to the generator stator end turns [4].

2) Delta Frequency

Delta frequency refers to the relative frequency of the generator with respect to the system. Excessive frequency difference during synchronization will also cause a shock. However, this shock is not as severe as with delta phase angle but combined together can compound the negative effects of each parameter alone [5].

3) Delta Voltage Magnitude

Excessively low generator terminal voltage during the connection could cause stability problems due to the weak magnetic circuit between the generator and the system. On the other situation, high generator terminal voltage during the connection will create a large reactive power flow from the system to the generator which can damage the generator shaft and mechanical shock to the stator windings could result from this instantaneous reactive power flow.

Therefore, it is highly important to keep the three critical synchronizing parameters within acceptable limits. A protection should be provided to the generator during the synchronization process. This can be done by measuring the generator parameters before it is physically connected to the system. There are several methods available for generator parameters measurements; most of them can be categorized into either hardware based or software based methods [6].

This thesis propose a new automatic synchronization system and measuring circuits for frequency, voltage and phase angle based on PIC modules connected by CAN protocol, which it is chosen for serial communication to reduce the complexity of wiring and maintenance problems. Moreover, to get the speed synchronization for multi-generator with the real time data communication, this is the main goal of the thesis.

1.1 Motivation

There are intensive and chronicle problems in the power distribution in Gaza Strip. Many factors contribute to make the situation worse. One of the factors is the lack of sufficient power supplies due to the siege imposed on GAZA since 2006. To partially mitigate this problem, small size local electrical generators have been used when the electricity power is cut. In some cases more than one electrical generator are used to supply the same load. This is particularly important when a higher system reliability and flexibility is required. For example, the failure of one generator does not cause a total power failure to the load. Therefore, we can remove the failed generator for shutdown and preventive maintenance. Moreover, If only one generator is used and it is not operating at near full load, then it will be relatively inefficient. However, with several smaller generators, it is possible to operate only a fraction of them. The ones that do operate are operating near full load and thus more efficient.

For efficient multi-generator system, synchronization between them is necessary. The synchronization process usually involves a large amount of wiring. Reduce this complexity of wiring's disturbance is one of the main contributions of our thesis by using (CAN protocol). This allows for ease of cabling, ease of change of cabling and ease in adding controller modules.

1.2 Problem statement

Synchronization of two generators or more means that their characteristics should be matched as closely as possible before the generators are connected together. They may be rotating at different frequencies. This difference in rotation is called "slip frequency". It is also desired that, when the coupling circuit breaker is closed, the relative phase-angle difference between the two generators is at or near zero [7]. Further, when the Circuit breaker is closed as in Figure 1.1, the generator's phase-angle displacement will be taken to zero instantaneously and the generator's speed will be instantaneously matched to that of the other generator. The output voltages of the dual generator should be equal. The closer the speed is matched and the smaller the phase-angle and voltage difference, the less mechanical stress is placed on the generator.

If the slip frequency is too large, the phase angle difference is too wide and/or the voltage difference is too large, so the oncoming generator could slip a pole from, causing damage to the machine. Significant voltage difference between the two systems will result in reactive power flow and can lead to a loss of synchronization. Reactive power will flow from the generator with the higher voltage toward the generator with the lower voltage [8]. When this power flow occurs, the excitation system will quickly try to compensate by adjusting the field current.

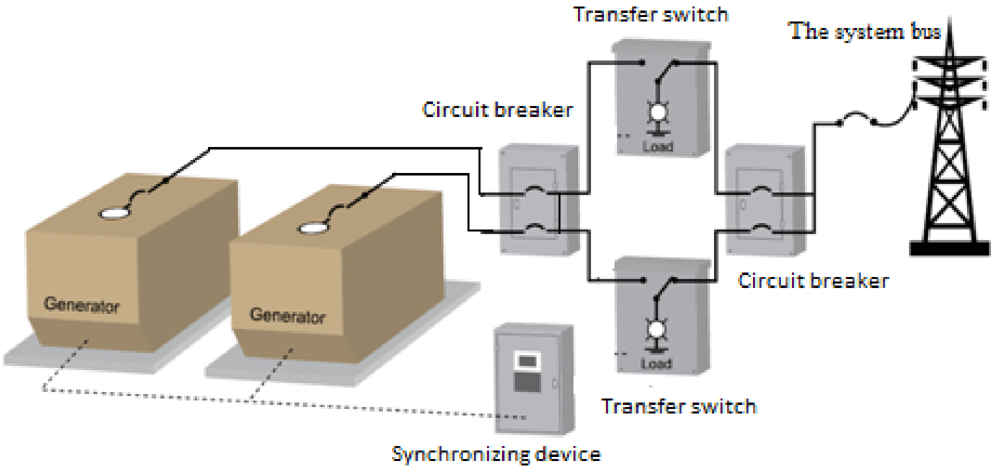


Figure (1.1): A wiring diagram for the parallel operation of the two generators

During this correction period, power swings could develop due to oscillations caused from the difference in mechanical power input and electrical power output. If the power swing becomes significant, the machine will continue to compensate and result in loss of synchronism. So when connecting two generators to one system, it is necessary to match the characteristics of the two generators. This is accomplished by minimizing the slip frequency, voltage and phase-angle difference between the two generators. The acceptance limits for each case are given as [3]:

- The voltage difference cannot exceed the rated voltage by more than 5% to 10%,
- The slip frequency should not exceed the rated frequency by more than 0.2% to 0.5%,
- The phase-angle difference is required to be zero.

The real time data communication is necessary in case of the fast synchronization for multi-generators. The complex signal lines are usually used to detect each generator characteristics. Many problems, in the view of reliability and economy, result from this classical implementation technique because of the amount and complexity of wiring, noise and maintenance problems, etc. These problems are serious especially when the generators controllers are far away from each other.

1.3 Methodology

In order to put a solution to the problems above, the automatic synchronizing device needs to perform the following operations:

First, the device monitors the voltage difference between the two generators and regulates the voltage amplitudes to reduce the voltage difference when it is not in the allowed scope.

Second, the device monitors the slip frequency between the two generators and regulates the frequencies to reduce the slip frequency when the slip frequency is not allowed.

Third, the device monitors the phase-angle difference between the two generators and regulates the phases to reduce the slip frequency when the slip frequency is not allowed.

Fourth, when the frequency, voltage, and phase-angle match the parallel requirements, the closing order is issued.

This thesis introduces the principles and the structure of a new automatic synchronizer. It also introduces the corresponding algorithm and some other key implementing techniques based on CAN protocol, which is the aim of the thesis, to achieve the previous operations with minimizing the wiring harassment.

The proposed model consists of three independent PIC modules, which are connected using CAN serial communication protocols. These PIC modules run independently and simultaneously. Two PIC modules monitor the voltage parameters of each generator. Only when both the PIC modules satisfy the synchronization conditions, the third PIC issues the paralleling instruction (a signal initiating the closing of generator breaker). Due to the independent hardware, the probability of a false paralleling instruction made by both PIC systems at a same time could be minimized. Therefore, the reliability of synchronization could be improved greatly by using the CAN protocol. Furthermore, the device also employed some other techniques to ensure the veracity and celerity of synchronization.

1.4 Literature review

Generator paralleling is a frequent and essential operation in the power system. In order to connect dual synchronous generators to the system (commonly referred to as paralleling operation), the two generators must first be synchronized by an automatic synchronizer [9] [11]. Severe damage to the generators as well as system disturbances may result if the generators are allowed to be connected to the system outside of established safe levels [6].

There are many methods available for generator's parameters measuring and synchronization; most of them can be categorized into either hardware based or software based methods [6]. Each method has its own characteristics. With hardware based methods, the generator parameters are measured by a special hardware circuit, while, with software based methods, the parameters are estimated by data acquisition and digital processing, the proposed device in the thesis is a combination between the hardware and software methods .

The hardware is now commonly used due to its simple principle and little calculation. However, it needs additional measuring circuit, and its measuring accuracy could be easily affected by noises and harmonics as well as other disturbances.

The software needs no special circuit; its measuring accuracy and synchronization performance mainly depend on its digital algorithm and accuracy of the analogue-to-digital converter.

Various studies related to synchronization operation are presented in the scope of this research. These studies include simulation, power control, synchronization and stability of power system including synchronized generators [12] [13].

The past philosophy of synchronization was provided by a solid-state check relay. The relay is set to prevent the generator breaker from closing unless the allowable delta phase angle and the delta voltage were in within the acceptable levels.

However, the synch check relay did not provide delta frequency protection. The protection for the delta frequency is done manually by an operator. The operator issues the connection command when he observes using the syncroscope that the frequency difference is within the accepted margin.

The main drawback of this method that it needs a well-trained staff and it is subjected to human errors. In [9], the author introduced microprocessor-based governor to replace the operator.

Nowadays, the designs of an automatic synchronizing device mainly adopt several kinds of controllers: PLC, singlechip and DSP etc. [14] [15]. Some researchers adopted double single chips to design the synchronizing device [16].

An automatic digital synchronization system has been proposed in [17]. Nevertheless, using sensors and PLC in the control unit increases the cost of the system[18].

In [19], microcontroller based on an automatic synchronization unit has been developed for the parallel operation of Synchronous Generators “SG”. The control unit reads calculates and evaluates the frequency, voltage, phase sequence of the received input signals and then provides the synchronization for the monitoring parallel connection conditions and parallel operation of generators. The program coded into the microcontroller was effectively developed to eliminate the interface electronic circuits from the system.

However, this method does not consider the delay caused by the separation of the two generators. PIC16F877 was used as the microcontroller and FPGA was used as the additional controller [3] but again it does not consider real time data communication.

An automatic synchronizer based on dual principles and dual microprocessors is developed in [6]. Both CPU modules monitor the voltage parameters of generators; only when the synchronization conditions are satisfied by both CPU modules, the closing of generator breakers command is issued with an appropriate advance time with respect to the occurrence of phase coincidence.

Like the previous approach, this method does not consider the time delay in the system proposed.

Researchers start to adopt the DSP controller. Using DSP can realize the high arithmetic rate, but the cost is expensive, and the developing period is long [20]. Besides, single-chip and DSP cannot realize a multitasking parallel processing. When the signal such as the generator frequency changes quickly, single-chip and DSP cannot measure the multipath signal at the same time.

The increasing demand for communication has lead to the specification and existence of various communication protocols. Since the mid- 1980s various field bus protocols and sensor/actuator protocols have been under design and/or available. Since the end of the 1980s the so-called "autobus" protocols have been in there final development or early production phase.

CAN protocol, being one of the most advanced autobus protocol in those days, was launched in 1989 as a standard product by Intel. CAN originally had been invented and driven by R. Bosch GmbH. Germany, at the beginning of the 1980s.

CAN was first applied in Mercedes S-class cars, launched in 1992, providing a high speed network for communication between engine controller, gear box controller, and dash board and a low speed network for distributed air conditioning control. Today CAN has been designed into many areas related to vehicles and industrial control.

In order to improve the celerity, veracity and security of the generator synchronizing operation, a new type of generator synchronizing device based on PIC16F4685 based CAN protocol will be addressed in this thesis.

1.5 Thesis Structure

This section outlines the overall structure of the thesis and provides a brief description for each chapter:

Chapter 2 provides some basic knowledge about general generators, synchronization and CAN bus. Chapter 3 addresses the sub system development which are the voltmeter, frequency counter, phase angel meter and governor control unit . Chapter 4 addresses the overall system and software that consists of three parts: the circuit breaker unit, the generator control unit and the data acquisition module. Finally, in Chapter 5, conclusions and suggestions for future work are summarized.

1.6 Summary

Chapter 1 has introduced the thesis. The motivation were described briefly in section one .The chapter also provided a problem statement in section two. Section three described the methodology of the thesis. The literature review relevant to the thesis listed in section four. Section five demonstrate the thesis structure

CHAPTER 2 THEORETICAL BACKGROUND

In this chapter, we introduce theoretical background of the main hardware elements and protocols used in the implantation of this thesis project. Firstly, we introduce the concept of the synchronous generator. The structure and theory of synchronization between generators in power systems are presented in section two. In the next section, we discuss the damage to the generator and system at asynchronization. In section four, we illustrate the synchronization methods. Then we describe Controller Area Network (CAN) bus, which is used to data transfer in this thesis.

2.1 Synchronous Generators

In this section, we explain the mechanical construction and electrical operation of an ac generator, which we refer to as an alternator or synchronous generator.

2.1.1 Definition

Alternating current (ac) generators are commonly referred to as synchronous generators or alternators.

A synchronous machine, whether it is a generator or a motor, operates at synchronous speed, that is, at the speed at which the magnetic field created by the field coils rotates. As shown in equation 2.1 an expression for the synchronous speed N, in revolutions per minute (rpm) as [21]

$$N = 120 f / P \qquad (2.1)$$

Where f is the frequency in hertz (Hz) and P is the number of poles in the machine. Thus, for a 4-pole synchronous generator to generate power at 50 Hz, its speed of rotation must be 1500 rpm.

On the other hand, a 4-pole synchronous motor operating from a 50-Hz source runs at 1500 rpm.

2.1.2 Construction of a Synchronous Machine

A synchronous machine consists of the stator, which houses the armature conductors, and the rotor, which provides the necessary field as in Figure 2.1.

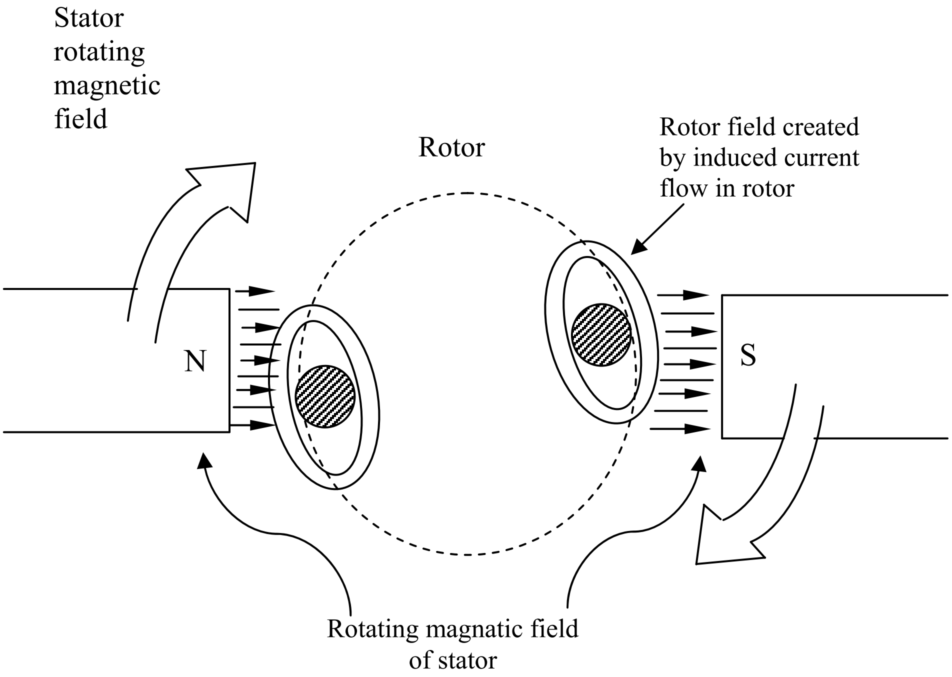


Figure (2.1): Pole pairs wound in a specific pattern to provide a north to south magnetic field

2.1.1.1 Stator

The stator or the armature consists of made of thin laminations of highly permeable steel in order to reduce the core losses as in Figure 2.2.

A stator frame holds and groups the stator laminations, it may be of cast iron or fabricated from mild steel plates. The frame is designed not to carry the flux but to provide mechanical support to the synchronous generator [21].

The slots inside the stator are to house thick armature conductors (coils or windings).

To form a balanced poly-phase winding, the armature conductors are symmetrically arranged.

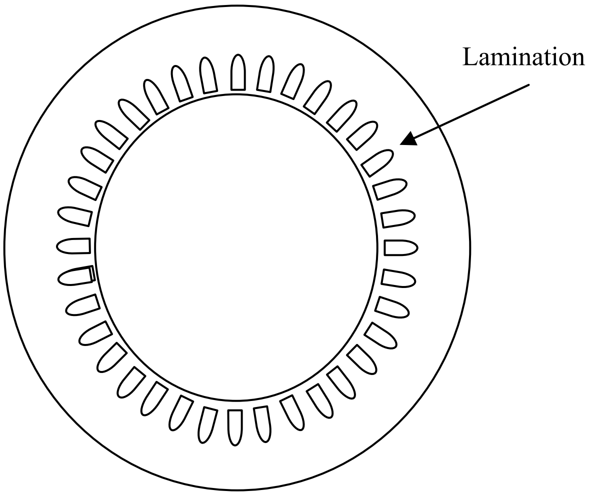


Figure (2.2): Stator construction

To this end, there are some notes must be taken into consideration [21]:

- The induced emf per phase in large synchronous generators is in kilovolts (kV),
- A power handling capacity in Mega Volt- Amperes (MVA).

The axial length of stator can be divided into two form depending on the speed of the generator. For slow-speed generators the axial length of the stator is short. These generators have many poles and are left open on both ends for self-cooling. They are installed at locations where hydroelectric power generation is possible.

On the other hand, for high-speed generator having two or four poles the axial length can be many times its diameter. Other than slow-speed generator, high-speed generators require forced air circulation for cooling and are totally enclosed. They are used when the rotors are driven by gas or steam turbines.

2.1.1.2 Rotor

The rotor consists of poles equal to the number of stator's poles, it houses the dc field winding which is usually receives its power from a 115- or 230-V dc generator.

The rotor is driven by a prime mover at its synchronous speed. The dc generator may be driven either by the same prime mover driving the synchronous generator or by a separate electric motor.

Therefore, to produce constant flux per pole the dc generator must has double-layer winding to carry dc current.

Rotor can be of either the cylindrical type or the salient-pole type [21]. Low- and medium-speed generators use a salient-pole rotor because the windage loss is small at these speeds. The salient-pole rotor consists of an even set of outward projecting laminated poles. Each pole is designed so that it fits into a wedge-shaped niche or is bolted onto a magnetic wheel called the spider. The field winding is placed around each pole, as indicated in Figure 2.3. The poles must alternate in polarity.

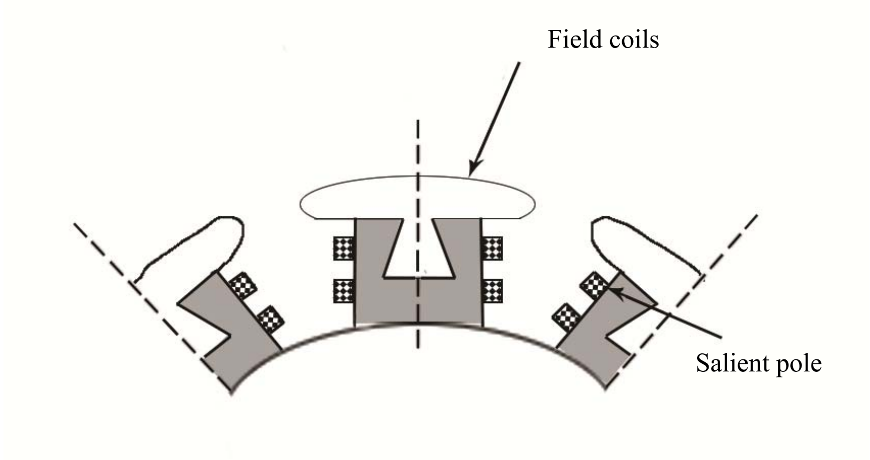


Figure (2.3): A salient pole rotor

On the other hand, high-speed turbo-generator use the cylindrical rotor. The cylindrical rotor is made of a smooth solid forged steel cylinder with a number of slots on its outer periphery. These slots are designed to accommodate the field coils, as shown in Figure 2.4. The cylindrical construction offers the following benefits [21]:

1. It results in a calm operation at high speed.
2. It provides better balance than the salient-pole rotor.
3. It reduces the windage loss.

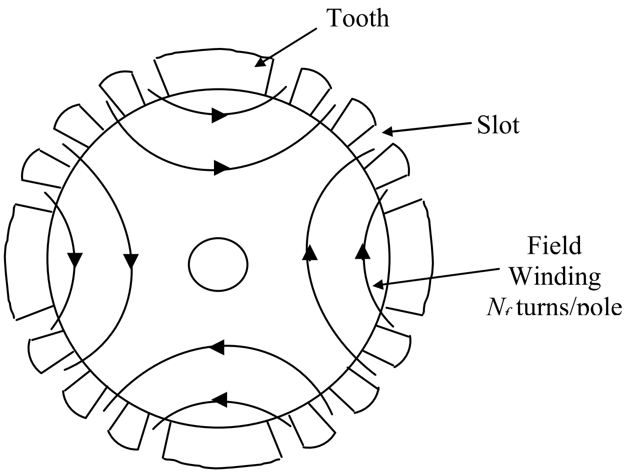


Figure (2.4): A 4-pole cylindrical rotor

2.2 Synchronization

Generators are removed or connected from service due to several factors such as variations in load, maintenance and emergency outages. Each time that a generator is connected to a power system, it must be synchronized with it before the interconnecting breaker can be closed.

Definition

Synchronizing, in its simplest form, is the process of electrically connecting and matching dual generators to each other as shown in Figure 2.5. To be precise, synchronizing is the act of matching the voltage magnitude, phase angle and frequency of the first generator to the second generator values.

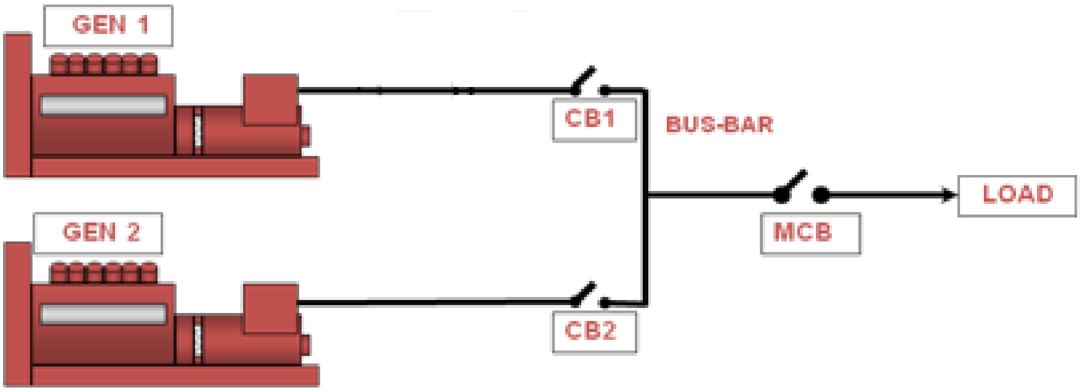


Figure (2.5): Connecting two generators

To illustrate the synchronizing definition, we consider the function of the volt component supplied by any generator, which is as follows:

$$V = A \cos(\omega t + \Theta) \qquad (2.2)$$

Let the voltage of the first generator is $A_1\cos(\omega_1 t + \Theta_1)$,And the volage of the second generator is $A_2\cos(\omega_2 t + \Theta_2)$, then, if we connect the two generators in one bus, the voltage components of each must be equal as mentioned in the definition above, this implies that:

$$A_1\cos(\omega_1 t + \Theta_1) = A_2\cos(\omega_2 t + \Theta_2) \qquad (2.3)$$

From equation 2.3, we get that

$A_1 = A_2$ the amplitude

$\omega_1 = \omega_2$ → $f_1 = f_2$ the frequency

$\Theta_1 = \Theta_2$ the phase

These confirm the condition of the definition above. To discuss the mismatch condition, Figure 2.6 is a vector representation of the variables associated with synchronizing. First generator voltage E_s and speed (frequency) ω_s are set by the power system. During the synchronizing process, the frequencies of the two generators can be different from each other.

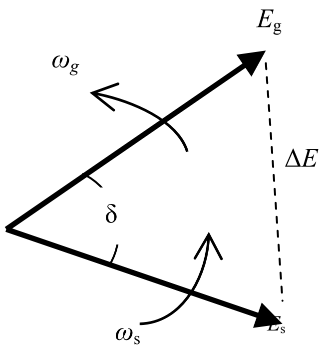


Figure (2.6): Synchronizing variables

If the second generator is assumed to be at a higher speed than the First generator, the second generator voltage, E_g, would be rotating about the fixed vector diagram of First generator voltage in a counterclockwise direction. Slip frequency (ω_s-ω_g) being the difference between the First generator, the second generator frequencies. The phase angle δ and voltage difference ΔE between the two generators will vary at slip frequency.

From the discussion above, we get that: ideally, the First generator and the second generator voltages on either side of the open synchronizing breaker should be equal in magnitude and frequency and in perfect phase alignment at the instant of breaker closure [22].

A perfect synchronizing would result in no electrical or mechanical system transients, hence no stress on the generators. In practice, both the electrical and mechanical systems are tolerant of small deviations from the ideal and synchronizing is often performed with an intentional mismatch of frequency and voltage [22].

2.3 Damage in case of asynchronization

A properly failure in synchronization can result from electrical and mechanical transients that can damage the generator, prime mover, generator step-up "GSU" transformer, and severely perturbate the power system [22]. We will only focus on the damages that occurred to the system and the generator.

2.3.1 System Problems

Along with the transient torques to the mechanical system, there will be electrical power oscillations [23]. These oscillations will relatively increase when the generator is synchronizing to a weak system. On the other hand, the generator constitutes a large dynamic source/sink for reactive power.

If the generator's voltage is lower than the system voltage, and the connected system cannot supply the reactive power to hold the voltage up until the generator increase its voltage, the generator's voltage can cause a voltage dip to the local power system.

The situation can actually be worse if the generator regulate its voltage during synchronization. As soon as the generator is synchronized to the system, the generator could immediately back off excitation to try to bring the voltage down to its set point, resulting in an extreme under excited condition. The weak magnetic field can result in the generator not pulling into synchronism or pulling back out of synchronism shortly after synchronization

2.3.1 Generator damage

When the generator is connected to the power system, the electrical and mechanical systems are tied together. Prior to closing the generator breaker during synchronizing, the angular velocity of the rotating magnetic field and therefore the frequency of the voltage induced in the stator are governed by the rotor speed [23].

When the breaker is closed the frequency of the power system govern the rotating magnetic field. So the rotor and prime mover will be forced to match their speed and position to be or become identical with the power system. If the speed and position of the rotor are closely matched at the instant the generator is connected to the power system, the transient torque required bringing the rotor and prime mover into synchronism is acceptable.

Two situations can happened if there is mismatch between the generator speed and the power system speed:

first If the position, as measured by the angular difference between the incoming and running voltages, is close and the angular velocity (frequency) is significantly off, as measured by the slip between the incoming and running voltages, there will be a large transient torque on the mechanical systems to accelerate or decelerate the rotating masses to match the power system angular velocity.

Second If the rotor position is also off (voltage phase angle difference is large), there can be an even higher transient torque required to snap the rotor and prime mover position into phase with the power system.

These transient torques can cause instantaneous and/or cumulative fatigue damage to the generator and prime mover over the life of the system

Note that the generator standards [24] [25], and [26] allow ± slip. However, from the mechanical perspective, it is desirable to limit synchronization from zero to positive slip to reduce shock in the mechanical system because of drive-train lash. There can be clearances in the mechanical drive train that cause a small amount of free play between forward and reverse torques.

When the prime mover is driving the generator prior to synchronization, the entire drive-train lash is made up in the forward direction. If the generator is running slightly faster than the system, the generator and prime mover will decelerate, and the lash is made up in the correct direction. If the generator is running slower than the system, it will have to accelerate, and the drive-train lash will now have to shift to the opposite direction.

Finally, the instantaneous current associated with a severely faulty synchronization can exceed the three-phase bolted fault duty that the generator and transformer must be designed to withstand. Large forces in the generator and transformer windings caused by the current surge can damage the windings and associated blocking, leading to catastrophic failure or reduced life.

2.4 Synchronizing methods

Synchronizing methods can be classified into two general categories, manual synchronizing and automatic synchronizing [22]. During a manual synchronizing, the operator has a full control over generator speed and voltage, and after meeting the synchronization conditions, he initiates the breaker closure command. In its simple form, manual synchronizing is completely performed by the operator. This type of synchronizing method is quite simple. However, the main disadvantage of this method is that it requires well trained operators at the controls to prevent costly damage to system components due to improper synchronizing command.

In many cases, the loads in the system increase based on random demand and they require immediate connection of the standby emergency generator sets. This demand for immediate attention excludes the use of operating personnel and manual synchronizing, which therefore leads us to automatic synchronizing.

With automatic synchronizing, the automatic monitors frequency, voltage and phase angle, provides correction signals for voltage matching and frequency matching, and provides the breaker closing output contact.

2.4.1 Manual Synchronizing

Synchronizing equipment has come a long way from the dark lamp synchronizer used in the early days of parallel generator operation.

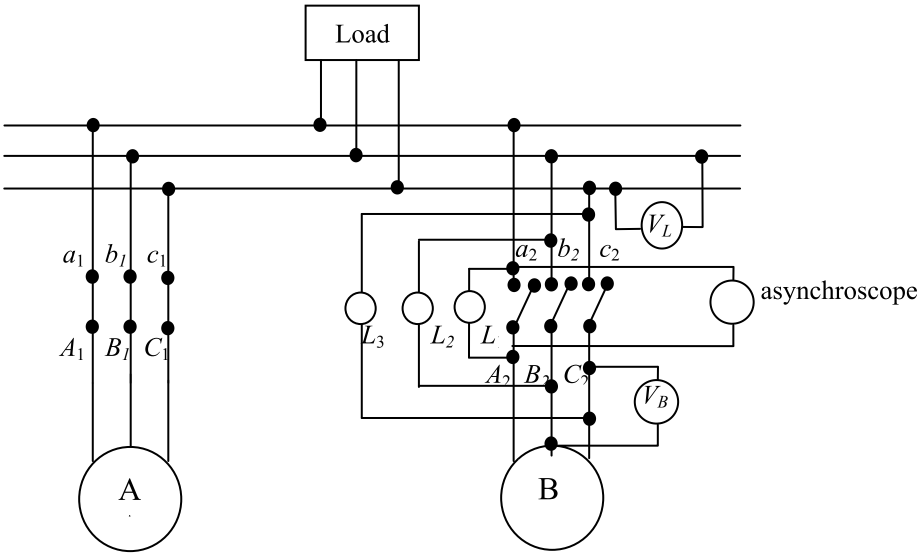


Figure (2.7): A wiring diagram for the parallel operation of the two alternators

This method uses three lamps connected across like phases of the open breaker, and two voltmeters one to measure the first generator voltage, and the other to measure second generator voltage to satisfy the first condition of paralleling as shown in Figure 2.7.

Satisfying other conditions of phase sequence, voltage opposition, and frequency may be determined by the use of the incandescent lamps. The lamp would be at maximum brilliance when the generators were completely out of phase (180 displacements) and completely extinguished when the two voltages were in phase (zero degree displacement) with identical magnitudes.

At any instant, it is seen that the voltage across the lamp is the sum of the individual phase voltages. The procedural steps for putting incoming generator in parallel with the running one are as follows:

Step1: The prime mover of the incoming machine starts, and the generator is brought up to near its rated speed.

Step2: By adjusting the field current, the terminal voltage of the incoming machine is made the same as that of the running generator. The lamp in the circuit will now flicker at a rate equal to the difference in frequency of the two generators. Correct connection of the phases result on synchronous brightening and blacking of the lamps. If this is not the case, then it means two of the lines are connected wrongly and they need to be interchanged.

Step3: Further adjustment of the incoming prime mover is now necessary, until the lamps flicker at a very low rate; the lamps pulsed as the generator voltage rotated with respect to the system voltage at slip frequency.

Step4: Final adjustment the operator would initiate a breaker close when the lamps were dark, indicating matching voltages and phase alignment.

Obviously, this system is not perfect. The whole operation of synchronization is subjected to the operator judgment or impatience. Moreover, there is a minimum voltage to the luminescence lamps, which means that “no light” does not happen exactly at zero voltage. There is also a delay between the initiation of the close signal and the actual breaker closure. The dark lamp method of synchronizing is certainly an inexpensive design. However, the potential for damage from a major out-of-phase closure or reduced service life due to repeated hard closures has led to the development of more secure synchronizing schemes.

The alternative practice is to supervise manual synchronizing with protective functions to prevent out-of-phase closures that would result from operator error. Sophisticated protective functions with settable parameters have become a necessary part of manual synchronizing scheme. A voltage is provided from step-down potential transformers (in high voltage applications) for the input signal to these devices.

Manual synchronizing equipment currently is depicted in Figure 2.8. Synchronizing meter panels are used to provide information to operators. The metering devices typically include individual bus and generator frequency meters for matching frequency, individual bus and generator a-c voltmeters for matching voltage, asynchroscope, and two indicator lamps.

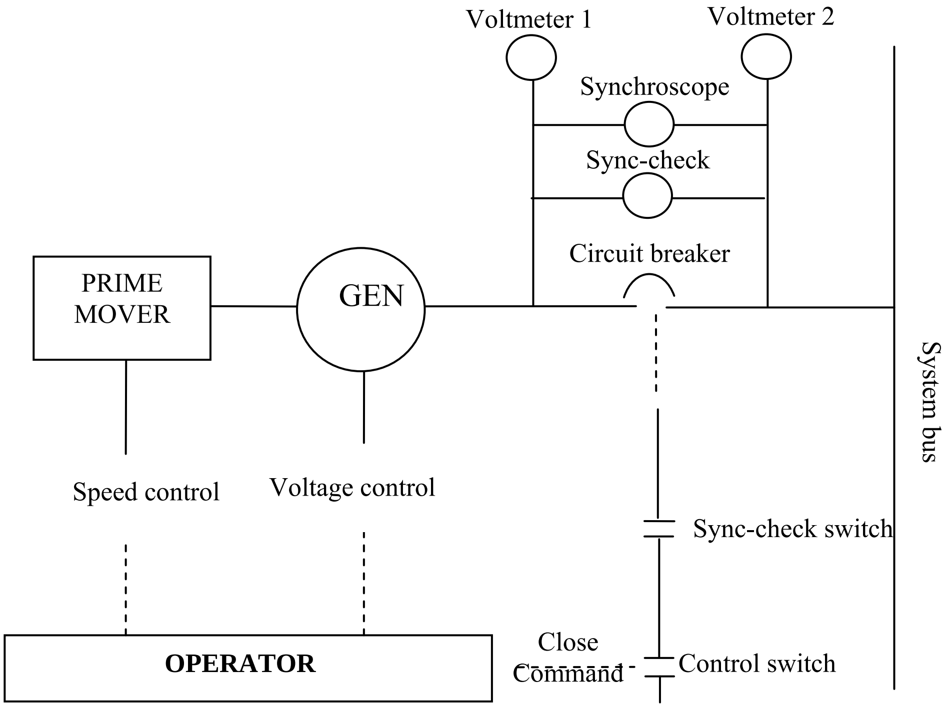


Figure (2.8): Manual synchronizing

The frequency and phase angle match between the two systems are now determined by observation of a synchroscope, which is shown in Figure 2.9.

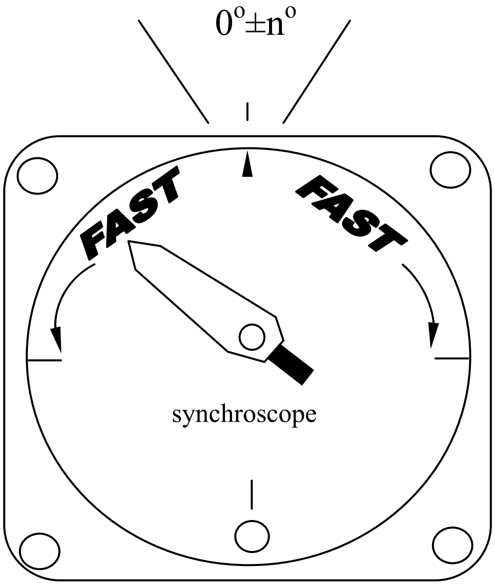


Figure (2.9): Synchroscope

The synchroscope is a multiple parameter information source. It indicates if there is a slip rate (a frequency difference between generator and bus), if the generator frequency exceeds the system frequency, the indicator on the scope will rotate in a clockwise direction. If the generator frequency is below that of the power system, rotation will be in the counterclockwise direction.

As seen in the Figure 2.9, the twelve o'clock position indicates zero degrees of phase angle difference. Any instantaneous position of the pointer indicates the phase angle difference between the bus and generator voltage. Of course, the object of the synchronizing process is to close the generator breaker at a zero degree of phase angle to minimize power flow transients when the breaker is closed.

The speed of rotation is indicative of the frequency difference (slip) between the two systems. The position of the scope also indicates the instantaneous phase displacement between the two voltages. At the 12:00 position, voltages are in phase. At the 2:00 position, the two voltages would be 360*2/12 = 60° apart. It is recommended that the operator initiates breaker closure when the absolute value of the phase shift is less than 10 ° this implies that the synchroscope indicator is between one-third of the distance from 12:00 to 11:00 and one-third the distance from 12:00 to 1:00 (10/360*12=0.33) [22].

In a pure manual synchronizing scheme, the operator initiates an unsupervised close command to the breaker from the breaker control switch. This operator-only design has become nearly extinct.

Now, at a minimum, manual mode closing is supervised by a sync-check relay (Device 25) in series with the control switch as seen in Figure 2.9. The sync-check relay measures the phase angle between the generator and system voltage. The relay will close its contact only when the voltages are within a preset angular limit, which is typically 10^{o} or less either side of the in-phase position for a generator application, and slip is within a preset limit. This design retains the operator's control over closing, and prevents him from making a gross out-of-phase closure.

The supervisory relay sets up an operating tolerance that must be equaled before the circuit breaker can be closed to parallel the alternator. These parameters and some typical ranges are listed below. The supervisory relay does not close its output contacts until all system parameters are satisfied according to the limits summarized below.

Parameters	Range
Slip Frequency	0.1 Hertz
Phase Angle	0° to 10°
Voltage	4 volts

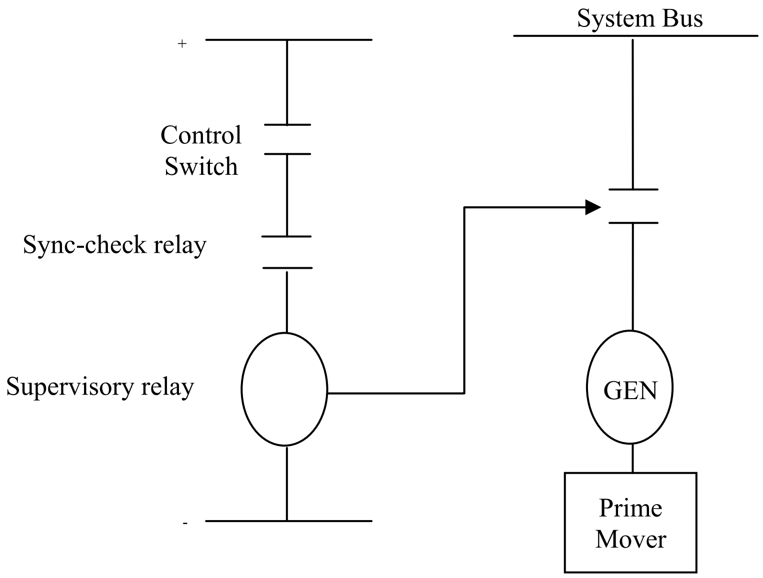


Figure (2.10): Breaker closure with supervisory control

The relay's output contacts are placed in series with the operator's control switch. Closure of the circuit breaker only occurs when

1) The operator manually attempts to close the circuit breaker, and

2) The supervisory relay contacts are closed. This is illustrated in Figure 2.10.

A functional block diagram of the supervisory type relay is shown in Figure 2.11.

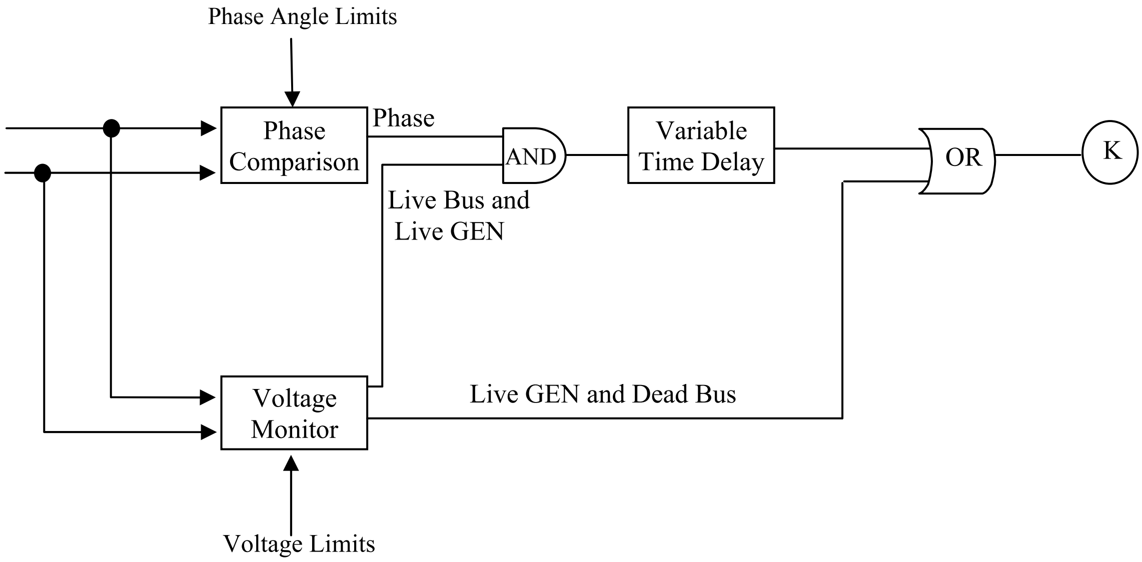


Figure (2.11): Synch-Check block diagram

The manual system uses two types of sync-check relay:

The first one is the electromechanical sync-check relays use the induction disk principle, with two sets of coils acting on the disk. Operating torque proportional to the vector sum of the two input voltages is produced by one set of coils. The other coil set produces restraining torque in proportion to the vector difference of the voltages. The assembly also includes a restraint spring and drag magnet.

Electromechanical sync-check relays should not be applied such that both inputs are continuously energized. This will result in vibration that will over time damage the relay. Instead, one relay input should be connected through the contact of the synchronizing switch (Device 43).

The second type is the solid-state and microprocessor technology. This type allowed the development of algorithms to monitor a host of voltage and frequency conditions applicable to safe synchronization. The most important of these is the direct slip calculation afforded by many microprocessor-based relays.

2.4.2 Automatic Synchronizing

For the first 40-year of the power industry, the synchronizing was entrusted to the skill of a well trained operator. Such responsibility would not be delegated to an automatic scheme that could malfunction and initiate a disastrous out-of-phase closure.

However, as generator size increased and designs became more efficient, both electrical and mechanical systems became less tolerant of the manual synchronization. A less-tolerant design is reflected by the tight limits now placed on closing angle, voltage difference and slip frequency by manufacturers.

Plant complexity also increased significantly, putting more demands on the operating staff and diverting the operator from the act of synchronizing. These changes and the disastrous damage resulting from some operator misjudgments led to the evolution of synchronizing equipment from unrestricted operator-controlled to the fully automated synchronizing schemes that have now become common.

The intent is that the automatic system is preferred and the manual system is used only when the automatic system is unavailable. However, in practice, the method actually implemented is dependent on individual plant philosophy and, in some cases, the level of frustration with the automatic synchronizing equipment.

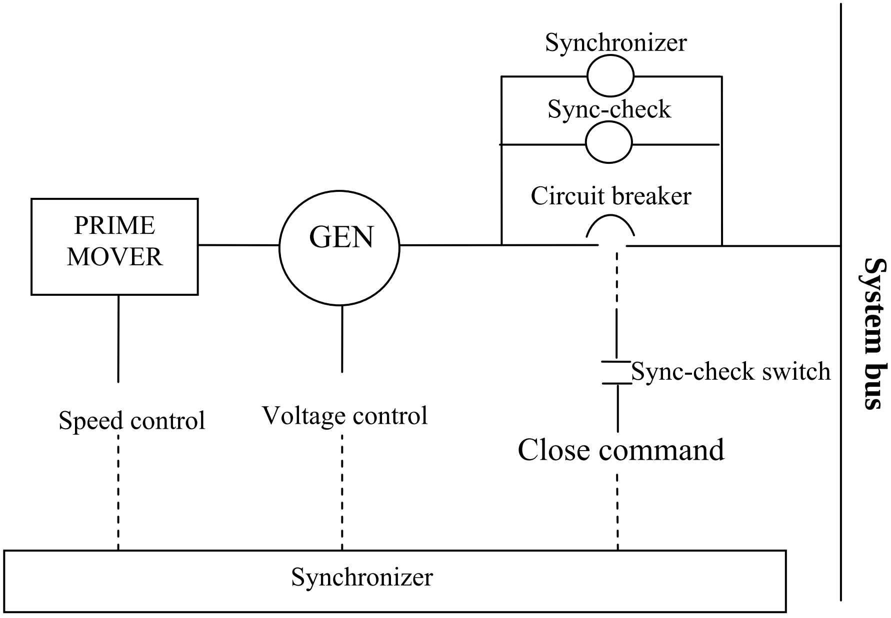


Figure (2.12): Automatic synchronizing

Automatic synchronizers perform all the monitoring and control functions necessary to synchronize the generator and close the breaker without operator involvement, as illustrated in Figure 2.12.

The operator controls the initial startup and early acceleration of the generator. As the generator accelerates, voltage rises. At about of 70% to 80% rated voltage, the automatic synchronizer is capable of measuring generator frequency and takes control of the synchronizing.

The auto synchronizer actuates the governor and voltage regulator to meet slip, voltage magnitude and phase angle limits set within the synchronizer. When operating parameters are within the preset limits, the synchronizer issues a close command to the synchronizing breaker.

Most electronic synchronizers are of the anticipatory type, when all limits are satisfied, the synchronizer will use real-time slip measurements and the breaker closing time to calculate the close initiation angle necessary to produce a closure at the zero degree position.

At the calculated angle, the synchronizer issues the close command. Anticipatory synchronizers require some minimum system slip to operate. State-of-the-art synchronizers can operate with slip as low as 0.0001 Hz. This equates to one synchroscope revolution in 2.8 h.

Speed matching to this accuracy is not normally achieved. Although such a close match is ideal for a smooth synchronization, at this slip breaker closing will be delayed about 5 min for every 10^o the generator voltage must travel to reach the in-phase position. In order to speed up the breaker closing, most preventive synchronizers issue a start pulse to the governor if voltage is within acceptable closure limits but slip is very low [22].

Automatic synchronizers include a variety of settable closing limit parameters to assure safe synchronization. In case of malfunctioned synchronizer, these limits are certainly useless.

To prevent damage from this type of failure, the breaker close command from the automatic synchronizer is normally supervised by a sync-check relay as shown in Figure 2.11. This is often the same sync-check relay that supervises manual synchronizing.

2.5 CAN protocol

The Controller Area Network (CAN) is a serial bus communications protocol. It has been developed by Bosch (an electrical equipment manufacturer in Germany) in the early 1980s. Shortly after that, the CAN was standardized as ISO-11898 and ISO-11519, and it has been adopted as the standard protocol for in-vehicle networking in the auto industry [28].

The CAN protocol plays an important role in many fields in applications that like networked embedded control, including industrial automation, medical applications, building automation, weaving machines, and production machinery. CAN offers an efficient communication protocol between control system components such as sensors, actuators, controllers, and other nodes in real-time applications.

The CAN protocol has many features over the other buses such that RS232, RS485, and parallel bus. It is simple, reliable, high performance, multimaster, flexible, Remote Transmit Request (RTR), can detect an error, and Multiple devices can be connected to the bus at the same time. Besides that, The CAN protocol is based on a serial bus topology, and only two wires are needed for communication over a CAN bus.

Each device on the bus can send or receive data. Therefore, one device can send data at any time while all the others listen. The transmission is prioritized so if two or more devices attempt to send data at the same time, the one with the highest priority is allowed to send its data.

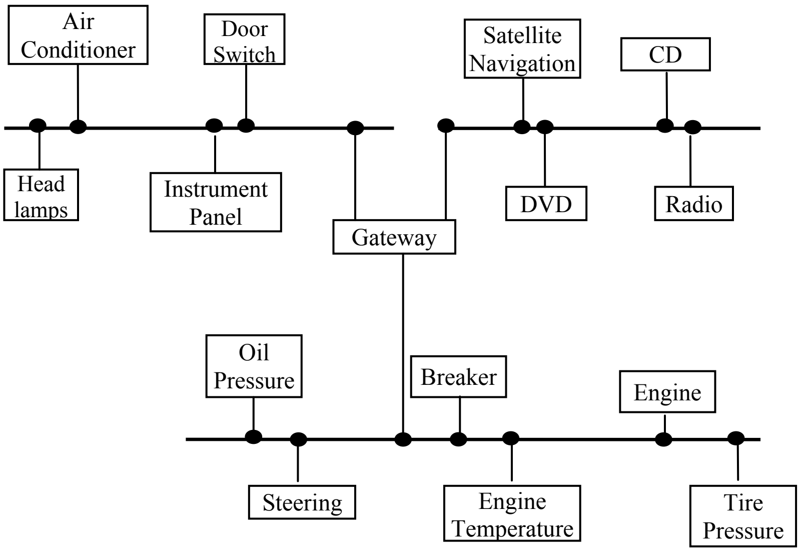


Figure (2.13): Typical CAN bus application in a vehicle

For instance, let us see the CAN protocol in a typical vehicle application in Figure 2.13, in where there is usually more than one CAN bus, operating at different speeds. Slower devices, such as door control, climate control, and driver information modules, can be connected to a slow speed bus.

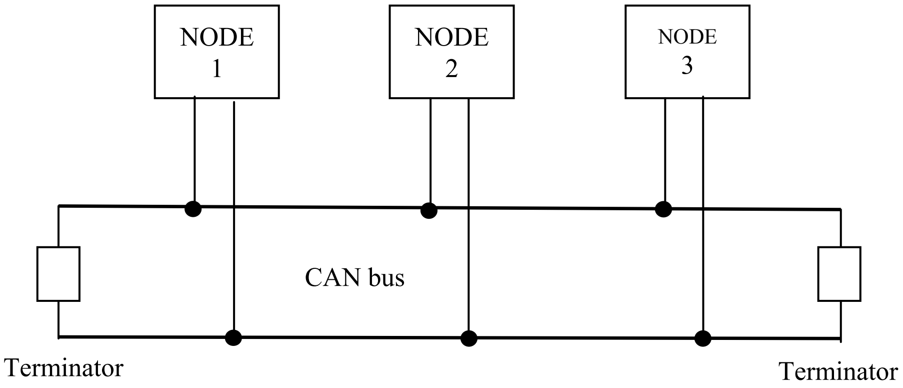


Figure (2.14): Example CAN bus

Devices that require faster response, such as the ABS antilock braking system, the transmission control module, and the electronic throttle module, are connected to a faster CAN bus.

Figure 2.14 shows a CAN bus with three nodes. The CAN protocol is based on CSMA/ CDAMP (Carrier-Sense Multiple Access/Collision Detection with Arbitration on Message Priority) protocol.

CAN protocol solves the collision problem, where only the highest priority node is given the right to send its data [28].

CAN protocol can be classified to two basic types: 2.0A and 2.0B. CAN 2.0A is the earlier standard with 11 bits of identifier, while CAN 2.0B is the new extended standard with 29 bits of identifier.

The ISO-11898 CAN bus specifies that a device on that bus must be able to drive a forty-meter cable at 1Mb/s.

A much longer bus length can usually be achieved by lowering the bus speed. Figure 2.15 shows the variation of bus length with the communication speed. For example, with a bus length of one thousand meters we can have a maximum speed of 40Kb/s [28].

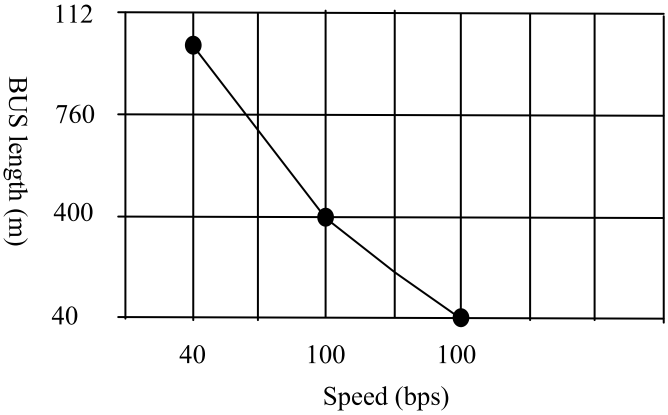


Figure (2.15): CAN bus speed and bus length

2.5.1 CAN bus termination

A CAN bus is terminated to minimize signal reflections on the bus. The ISO-11898 requires that the bus has a characteristic impedance of 120 ohms.

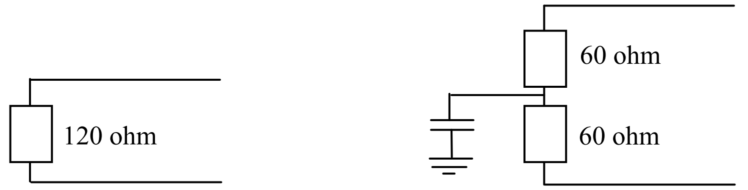
One of the following methods can terminate the bus [28]:

- Standard termination
- Split termination
- Biased split termination

In standard termination, the most common termination method, a 120-ohm resistor is used at each end of the bus, as shown in Figure 2.16(a).

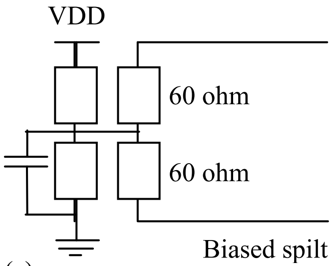
In split termination, the ends of the bus are split and a single 60-ohm resistor is used as shown in Figure 2.16(b). Split termination allows for reduced emission, and this method is gaining popularity.

Biased split termination is similar to split termination except that a voltage divider circuit and a capacitor are used at either end of the bus. This method increases the EMC performance of the bus (Figure 2.16(c)).



(a) Standard termination

(b) Spilt termination



(c) Biased spilt termination

Figure (2.16): Bus termination methods

2.5.2 CAN bus layers

Many network protocols are described using the seven-layer Open Systems Interconnection (OSI) model.

The CAN protocol includes the data link layer, and the physical layer of the OSI reference model (see Figure 2.17). The data link layer (DLL) consists of the Logical Link Control (LLC) and Medium Access Control (MAC).

LLC manages the overload notification, acceptance filtering, and recovery management.

MAC manages the data encapsulation, frame coding, error detection, and serialization/deserialization of the data.

The physical layer consists of the physical signaling layer (PSL), physical medium attachment (PMA), and the Medium Dependent Interface (MDI).

PSL manages the bit encoding/decoding and bit timing. PMA manages the driver/receiver characteristics, and MDI is the connections and wires.

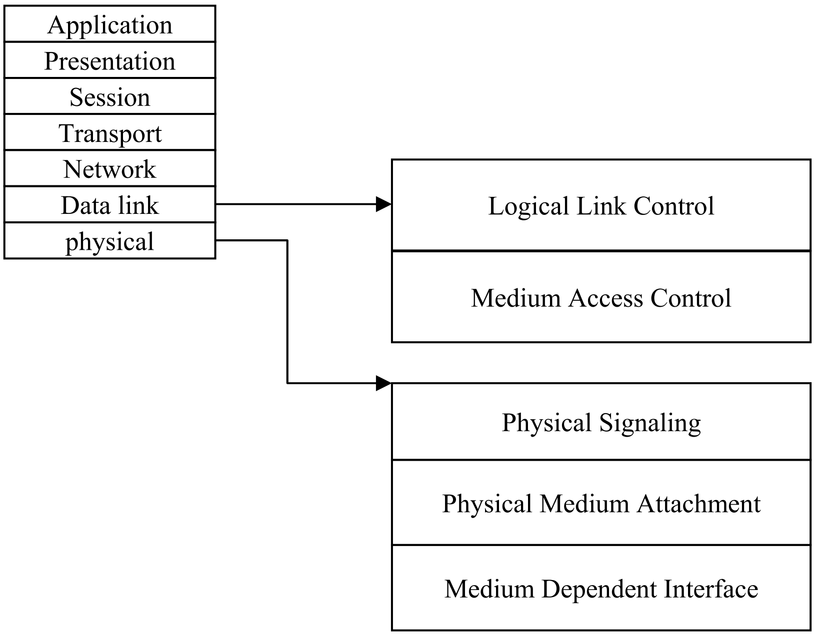


Figure (2.17): CAN and the OSI model

2.5.3 CAN bus frames

There are basically four message frames in CAN: data, remote, error, and overload. The data and remote frames need to be set by the user. The other two are set by the CAN hardware.

2.5.3.1 Data Frame

There are two formats to the data frame: standard (having an 11-bit ID) and extended (having a 29-bit ID).

The transmitting device uses data frame to send data to the receiving device, and the data frame is the most important frame handled by the user. Figure 2.18 shows the data frame's structure.

A standard data frame starts with the start of frame (SOF) bit, which is followed by an 11-bit identifier and the remote transmission request (RTR) bit.

The identifier and the RTR form the 12-bit arbitration field. The control field is 6 bits wide and indicates how many bytes of data are in the data field. The data field can be 0 to 8 bytes.

The data field is followed by the CRC field, which checks whether or not the received bit sequence is corrupted.

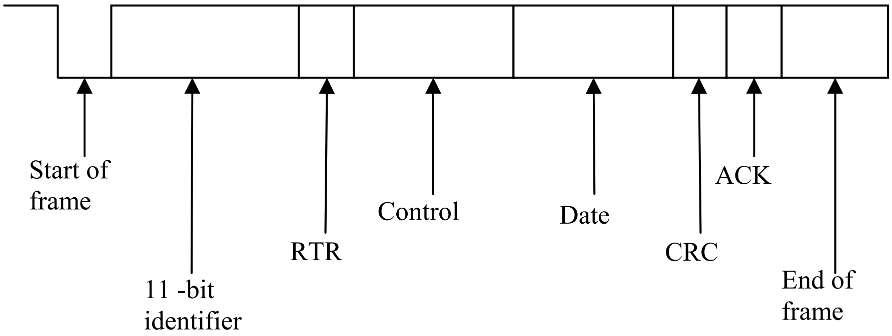


Figure (2.18): Standard data frame

The ACK field is 2 bits and is used by the transmitter to receive acknowledgment of a valid frame from any receiver. The end of the message is indicated by a 7-bit end of frame (EOF) field. In an extended data frame, the arbitration field is 32 bits wide (29-bit identifier 1-bit IDE to define the message as an extended data frame 1-bit SRR which is unused 1-bit RTR) (see Figure 2.19).

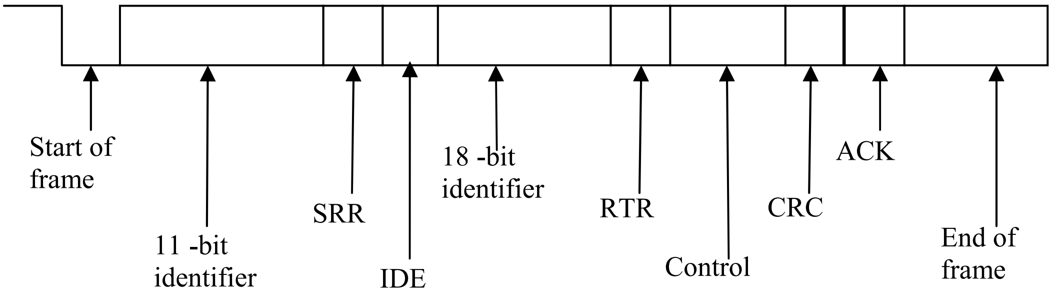


Figure (2.19): Extended data frame

2.5.3.2 Remote Frame

The receiving unit uses the remote frame to request transmission of a message from the transmitting unit.

It consists of six fields (see Figure 2.20): start of frame, arbitration field, control field, CRC field, ACK field, and end of frame field.

A remote frame is the same as a data frame except that it lacks a data field.

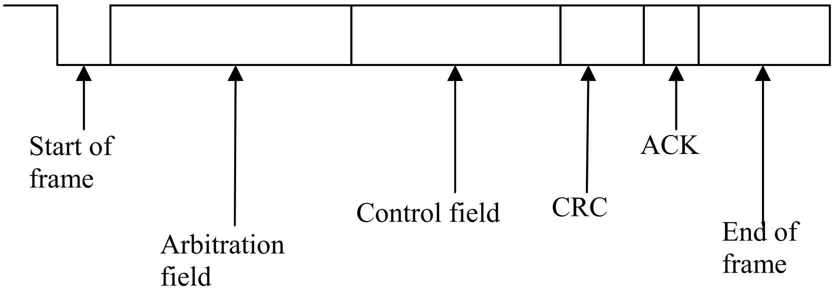


Figure (2.20): Remote frame

2.5.3.3 Error Frame

The CAN hardware generates and transmits error frames and uses it to indicate when an error has occurred during transmission. An error frame consists of an error flag and an error delimiter. There are two types of error flags: active, which consists of 6 dominant bits, and passive, which consists of 6 recessive bits. The error delimiter consists of 8 recessive bits.

2.5.3.4 Overload Frame

The receiving unit uses the overload frame to indicate that it is not yet ready to receive frames. This frame consists of an overload flag and an overload delimiter. The overload flag consists of 6 dominant bits and has the same structure as the active error flag of the error frame. The overload delimiter consists of 8 recessive bits and has the same structure as the error delimiter of the error frame.

2.5.4 PIC Microcontroller CAN Interface

In general, any type of PIC microcontroller can be used in CAN bus–based projects, but some PIC microcontrollers have built-in CAN modules, which can simplify the design of CAN bus–based systems. Microcontrollers with no built-in CAN modules can also be used in CAN bus applications, but additional hardware and software are required, making the design costly and also more complex.

Figure 2.21 shows the block diagram of a PIC microcontroller–based CAN bus application, using a PIC16 or PIC12-type microcontroller with no built-in CAN module. The microcontroller is connected to the CAN bus using an external MCP2515 CAN controller chip and an MCP2551 CAN bus transceiver chip. This configuration is suitable for a quick upgrade to an existing design using any PIC microcontroller.

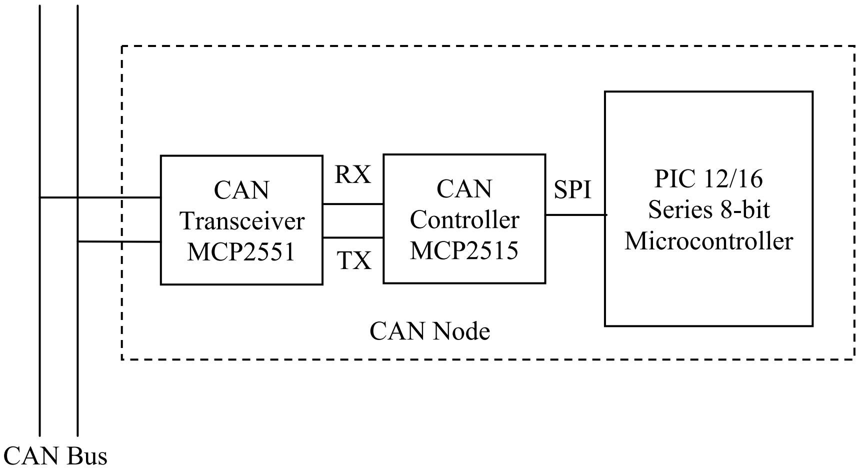


Figure (2.21): CAN node with any PIC microcontroller

For new CAN bus–based designs it is easier to use a PIC microcontroller with a built-in CAN module. As shown in Figure 2.22, such devices include built-in CAN controller hardware on the chip. All that is required to make a CAN node is to add a CAN transceiver chip.

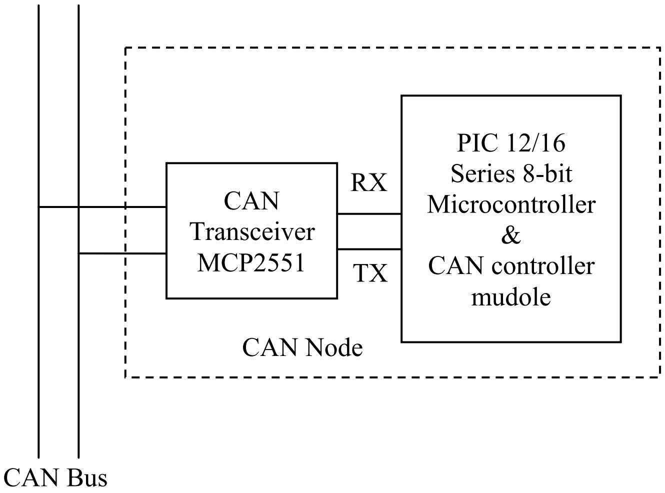


Figure (2.22): CAN node with integrated CAN module

2.5.5 Variable resistance CAN Bus example

The following is a simple two-node CAN bus–based example. The block diagram of the example is shown in Figure 2.23. The examlpe is made up of two CAN nodes. One node (called DISPLAY node) requests the resistance value every second and displays it on an LCD. This process is repeated continuously. The other node (called detector node) reads the resistance value.

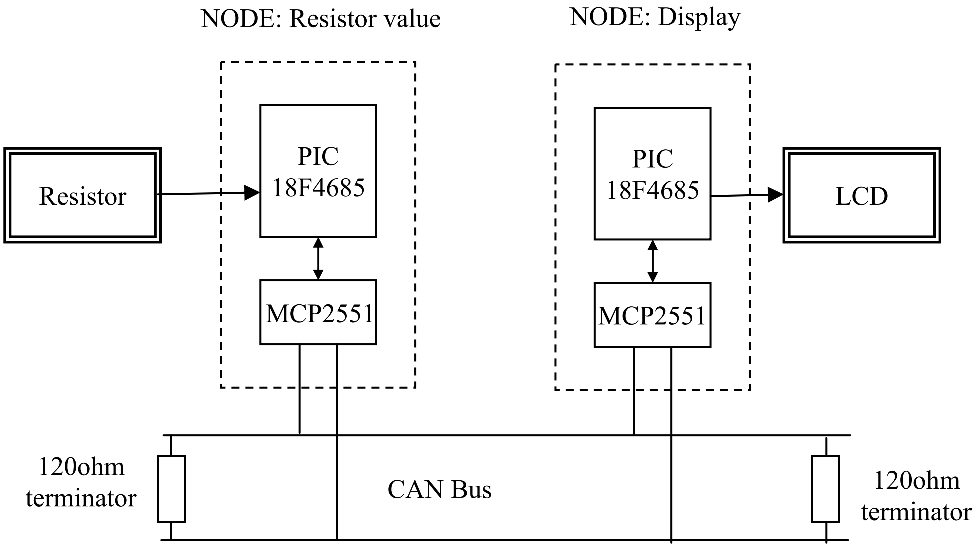


Figure (2.23): CAN block diagram of the example

The example's circuit diagram is given in Figure 2.24. Two CAN nodes are connected together using a two-meter twisted pair cable, terminated with a 120-ohm resistor at each end.

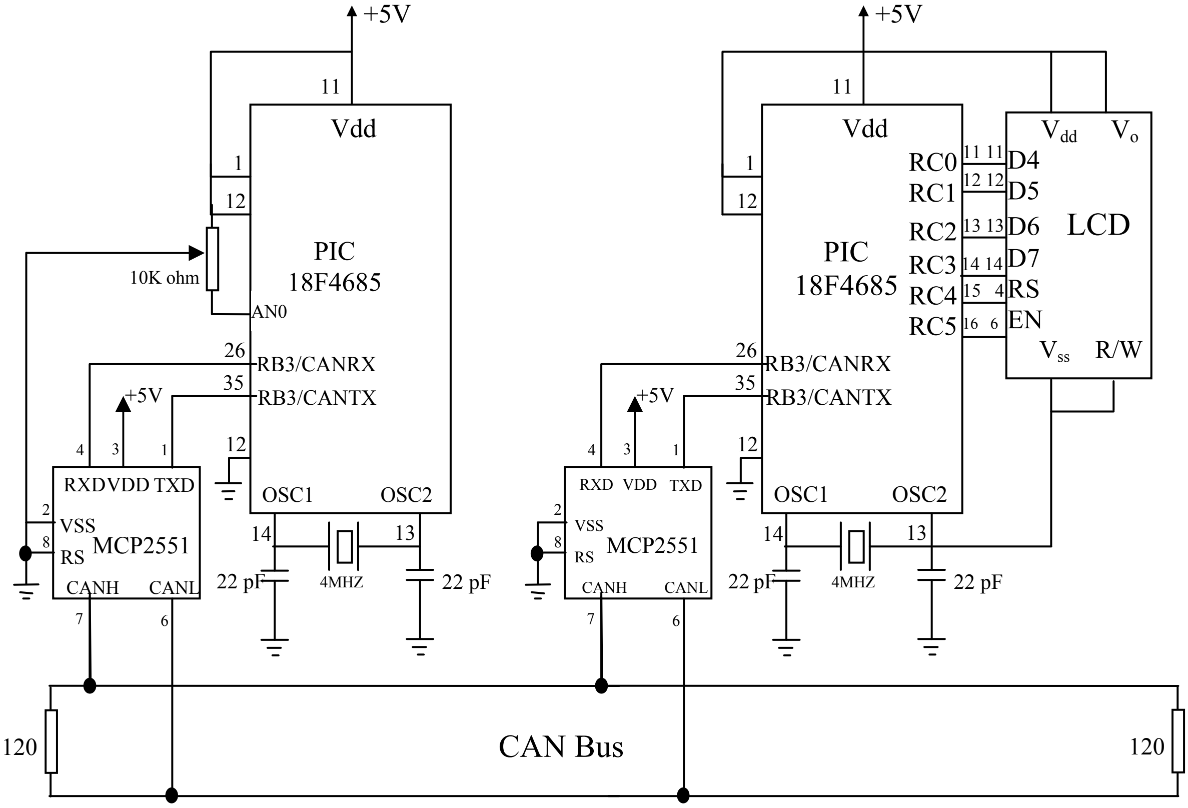


Figure (2.24): Circuit diagram of the example

The COLLECTOR Processor

The COLLECTOR processor consists of a PIC18F4685 microcontroller with a built-in CAN module and an MCP2551 transceiver chip. Analog input AN0 of the microcontroller is connected to a variable resistor 10K ohm.

The resistor can give range of 0 ohm to 10k ohm and generates an analog voltage directly proportional to the output resistor (i.e., the output is 0.5 mV/ohm). For example, at 5k ohm the output voltage is 2.5V.

The CAN outputs (RB2/CANTX and RB3/CANRX) of the microcontroller are connected to the TXD and RXD inputs of an MCP2551-type CAN transceiver chip. The CANH and CANL outputs of this chip are connected directly to a twisted cable terminating at the CAN bus. The MCP2551 is an 8-pin chip that supports data rates up to 1Mb/s. The chip can drive up to 112 nodes.

An external resistor connected to pin 8 of the chip controls the rise and fall times of CANH and CANL so that electromagnetic interference can be reduced. A reference voltage equal to VDD/2 is output from pin 5 of the chip.

The DISPLAY Processor

Like the COLLECTOR processor, the DISPLAY processor consists of a PIC18F4685 microcontroller with a built-in CAN module and an MCP2551 transceiver chip.

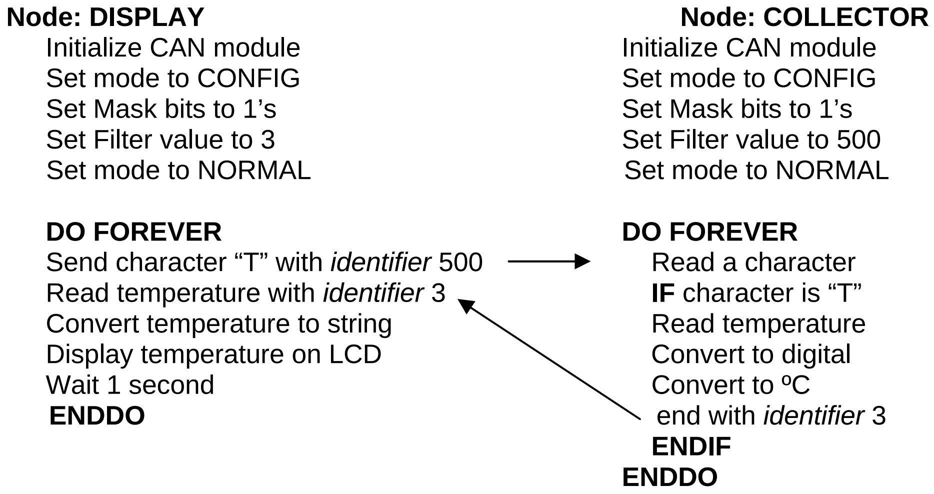
The CAN outputs (RB2/CANTX and RB3/CANRX) of the microcontroller are connected to the TXD and RXD inputs of the MCP2551. Pins CANH and CANL of the transceiver chip are connected to the CAN bus.

An HD44780-type LCD is connected to PORTC of the microcontroller to display the temperature values.

The program listing is in two parts as in Figure (2.25): the DISPLAY program and the COLLECTOR program. The operation of the system is as follows:

- The DISPLAY processor requests the current resistor value from the COLLECTOR processor over the CAN bus
- The COLLECTOR processor reads the voltage on AN0, calculate the resistor value, and sends to the DISPLAY processor over the CAN bus

- The DISPLAY processor reads the resistor value from the CAN bus and then displays it on the LCD
- This process is repeated every second



```
Node: DISPLAY
    Initialize CAN module
    Set mode to CONFIG
    Set Mask bits to 1's
    Set Filter value to 3
    Set mode to NORMAL

    DO FOREVER
    Send character "T" with identifier 500
    Read temperature with identifier 3
    Convert temperature to string
    Display temperature on LCD
    Wait 1 second
    ENDDO
```

```
Node: COLLECTOR
Initialize CAN module
Set mode to CONFIG
Set Mask bits to 1's
Set Filter value to 500
Set mode to NORMAL

DO FOREVER
    Read a character
    IF character is "T"
    Read temperature
    Convert to digital
    Convert to ºC
    end with identifier 3
    ENDIF
ENDDO
```

Figure (2.25): Operation of both nodes

The complete program written in mikroC is given in Appendix A.

2.6 Summary

This chapter described the theoretical background of the synchronizing component and the related topic where section one demonstrate the mechanical construction of the synchronous generator. Section 2 define the synchronizing process and the conditions to do it. The third section explained the damage to the generator and the disturbance to the system in asynchronize case. In section four the methods of the synchronizing are listed and discussed in details.

CHAPTER 3 MEASUREMENT METERS AND CONTROLLER UNITS

In this chapter, we explain the electrical operation of three meters: Volt meter, Frequency meter, and Phase meter. This is referred to as power meter. The volt meter, gives a map 240v to 5v. The Electrical equivalent of speed (RPM) is the frequency (HZ), provided by frequency meter. A phase meter determines the phase shift between the signal generator and a reference signal. In addition of that we explain the driver of the dc motor to control the governor of the generator.

A single 16F877 PIC microcontroller has been used in the volt meter part. Microcontroller has been operated with 4 MHz oscillator. Loop time of each command is one microsecond. Comparator, ADC, Timer-1 modules and A, B, E ports of the microcontroller has been used in the application. Port configuration of the microcontroller used for the study is shown n Figure 3.1.

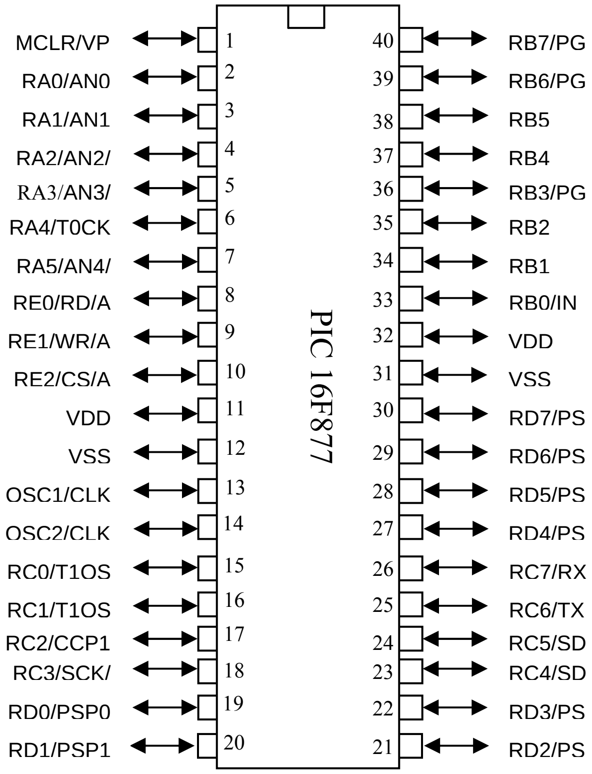


Figure (3.1): Pin diagram of 16F877

A single 16F628A PIC microcontroller has been used in the frequency and phase meters parts instead of 16F877 because it is capable of measuring frequency and phase by Timer module embedded in 16F628A without need of additional module, beside it is cheaper and smaller. Microcontroller has been operated with internal 4 MHz oscillator. Loop time of each command is 1µs, Comparator, Timer-1 modules and A, B ports of the microcontroller has been used in the application. Port configuration of the microcontroller used for the study is shown n Figure 3.2.

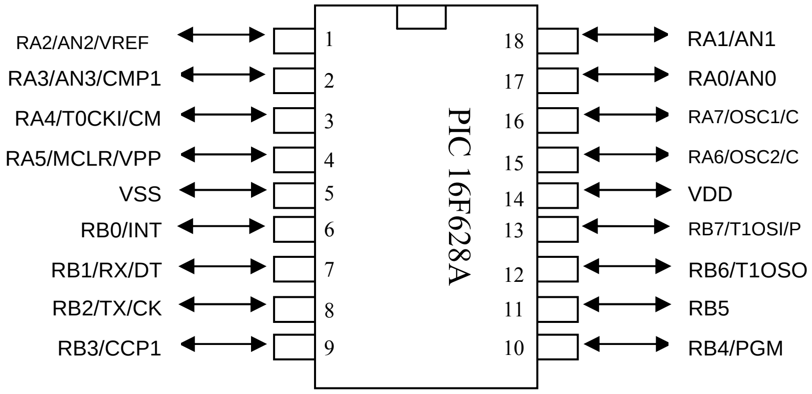


Figure (3.2): Pin diagram of 16F628A

Beside that, a memory 24C02 “SERIAL 2K (256 x 8) EEPROM” is used to facilitate the connection between the data acquisition module and the control generator unit, the chip provide 1 million erase/write cycles with 40 years data retention, working with single supply voltage1.8V to 5.5V, using hardware wire control versions: ST24W02 and ST25W02, connected with two wire serial interface, can write byte and multibyte (up to 4 BYTES), and self timed programming cycle. The pin diagram of the chip is shown in Figure 3.3.

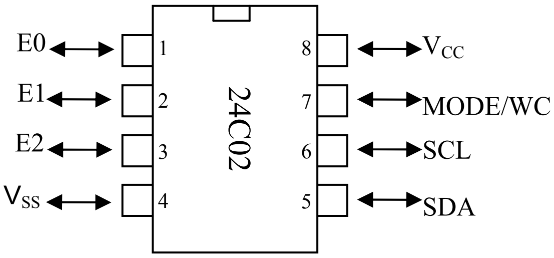


Figure (3.3): Pin diagram of 24C02

3.1 Volt meter design

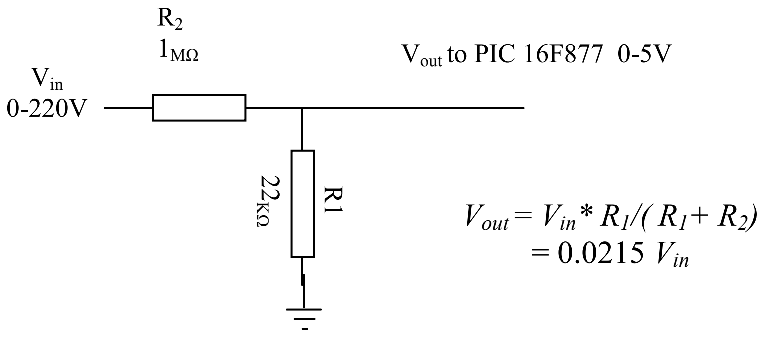
In this section, we describe how to make a simple digital voltmeter using a PIC16F877 microcontroller. The range of this meter is 0-5V.

3.1.1 The principle

The PIC 16F877 microcontroller reads the input voltage through a voltage divider circuit, which maps 240v AC to 5v DC. The PIC recalculates the amplitude of the voltage by using the ADC module and sends it to the memory unit. The voltage difference at the output of the transformer circuit is calibrated to achieve an exact 5 volt from the 220 v at the input. This is done by using a variable resistance connected in series with the transformer input and a 22-KW resistance in parallel with the output. The potentiometer is calibrated with the help of Oscilloscope in order to realize the 5V signal. The variable resistance is found to be 1M-ohm.

3.1.2 Hardware design

A 220V signal cannot be fed directly to a PIC microcontroller's input channel. It is much higher than the operating voltage range of the PIC, so the microcontroller can be damaged. Therefore, we need to design a voltage scalar that will scale down the input voltage to the safe operating voltage range of PIC16F877. A simple resistor divider network as Figure 3.4 can achieve it.



$$V_{out} = V_{in} * R_1/(R_1 + R_2)$$
$$= 0.0215\ V_{in}$$

Figure (3.4): Voltage scalar circuit

Using two resistors, R_1 and R_2, the input voltage ranging from 0-220V AC can be scale converted to 0-5V AC. For the chosen values of R_1 and R_2, you can see that the output (Va) from the resistor divider network is 1/44 of the input voltage. Before passing the voltage to the PIC, it is passed through bridge to rectifier the sinusoidal signal then a low pass filter circuit is used to eliminate the AC component of the signal. So a capacitor is paralleled with R_3 1 Mohm, which value is 10 µF. The voltage Va will go to AN0 (pin 2) channel of the PIC16F877 microcontroller. The whole circuit is shown in Figure 3.5.

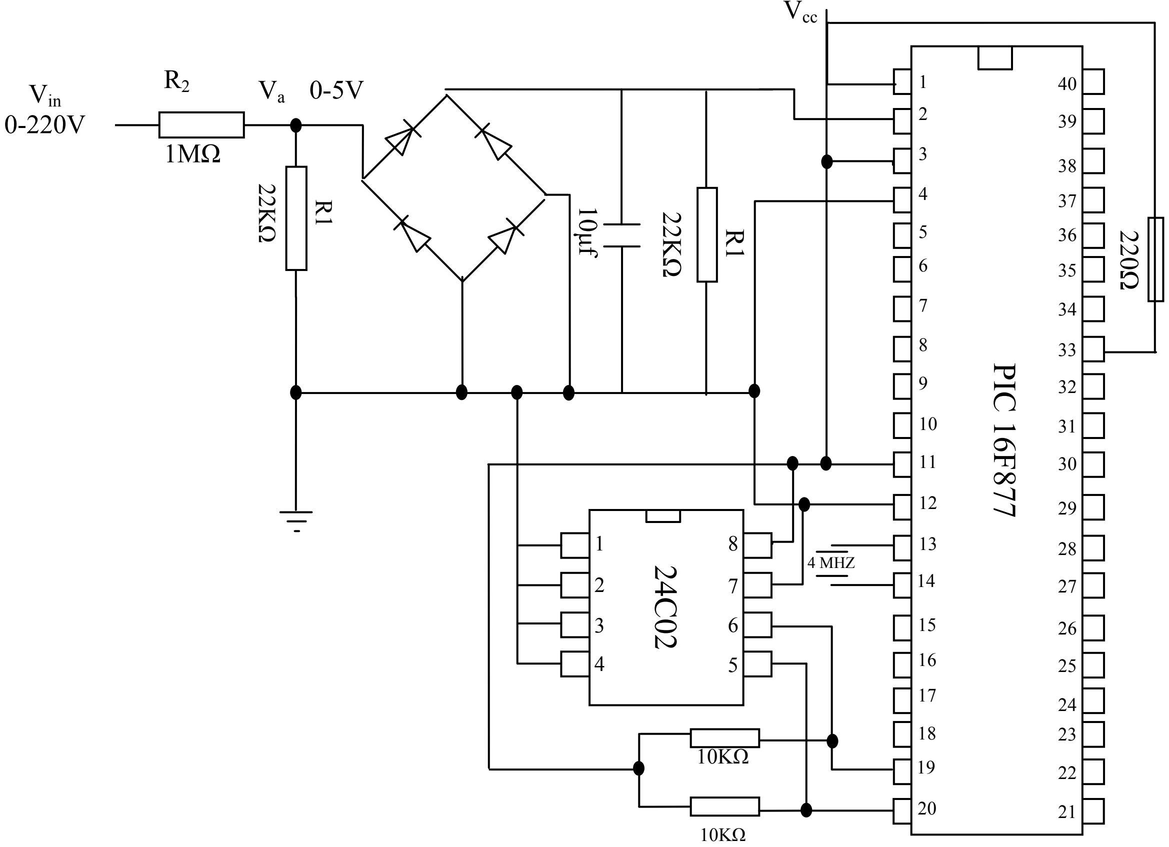


Figure (3.5): Volt meter circuit

3.1.3 Software design

The 5V is regulated as it is used as a fixed reference voltage for mapping 220v to 5v to provide accurate digital count for input signal. If the reference voltage is not stable, the scaling factor output is meaningless. Therefore, the PIC will convert any input voltage between 0-5 V in to a digital count between 0-1023. A major source of error in this module is the accuracy of R_1 and R_2 resistors. In our module, R_1 = 1 MΩ and R_2 = 22 KΩ. Now,

0 – 5 V Analog I/P is mapped to one of the 1024 levels (0-1023 Digital Count)

=> Resolution = 5/1024 = 0.0049 V/Count

$$V = \text{Digital Count} * 0.0049 \qquad (3.1)$$

Also, $Va = 22k*Vin/(22k+1M) = 0.0215*Vin$

$$\Rightarrow \; Vin = 46.4*Va \qquad (3.2)$$

Substitute (1) in (2)

$Vin = 46.4*$ Digital Count * 0.0049

= 0.227 * Digital Count (Approx.)

The above calculations are not precise due to losses in countered in the rectification and filter circuit along with the value of the resistors used in the voltage divider circuit. The firmware is developed in C and compiled with mikroC Pro for PIC compiler from Mikroelektronika [36].

The complete program written in mikroC is given in Appendix A.

3.1.4 Volt meter testing

Here is the volt meter measurement output for various input voltages ranging from 0-5V obtained through a variable AC power supply source.

S.No.	**Volt measured on oscilloscope “volt”**	**Volt measured by volt meter “volt”**	**Volt measured by Ref [28] “volt”**
1	207	209	210
2	207	210	211
3	207	210	211
4	206	209	210
5	207	209	210
6	208	209	210
7	208	209	210
8	207	209	210
9	207	209	210
10	206	208	209

Table (3.1): Volt meter results

The oscilloscope picture of the design is shown in Figure 3.6.

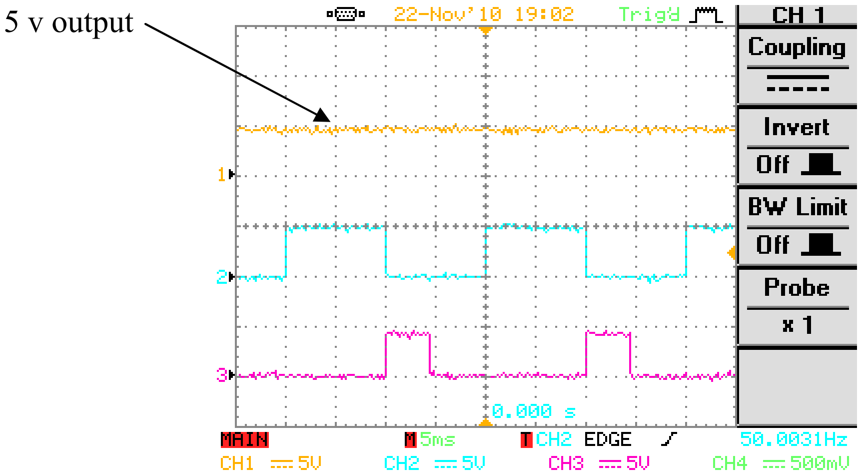


Figure (3.6): Volt meter results by oscilloscope

3.2 Frequency meter

There are various types of frequency counters that might have analogue or digital forms. Some of the instruments commonly used for measuring various frequency ranges are, oscilloscopes, Wein Bridge frequency meters, zero-beat frequency meters, direct reading frequency meters, binary frequency counters, digital frequency counters and university time-counters [29].

The oscilloscope can measure a wide range of frequencies. However, the accuracy of oscilloscope is somewhat limited. The Wein Bridge frequency meter is a device designed only to produce accurately known audio-frequency signals. Zerobeat Frequency Meter is used to measure the radio-frequency range. The University time-counter can be used to measure both frequency and time intervals. However, it can be only used to accurately measure low frequencies instead of frequency counters [29].

Digital frequency counters are the most accurate and flexible instruments available for measuring unknown frequencies. Frequencies from zero to the giga-hertz range can be measured with digital frequency counters.

In this module, a microcontroller is used to build the digital frequency counter. In many ways, the PIC microcontroller has more support available than other electronics device. In this research, the PIC 16F628 is used to construct a high resolution frequency counters which can measure up to 100 kHz.

3.2.1 The principle

The internal frequency counter in the PIC 16F628 is adopted for our module. The PIC 16F628 operated as a frequency counter which can read frequencies from 10 Hz to 100 kHz. It is used the method of measuring the 8-bits (1-byte) counter value from the prescaler, TMR0 (timer 0 module) and some other registers, such as Option Register.

The basic hardware for the measurement circuit is depicted in Figure 3.7. The first step in measuring the frequency is to down-convert the voltage from 220 V -AC to 5 V-AC, so the generator output is fed into voltage scalar (which is explained in the voltage meter section). After that, a power supply circuit is built to convert the AC 220 V to the required 5 V DC.

So, The voltag scaler converts 220 V to 5 V AC. The 5 V AC is passed to through the sinusoidal to square waveform circuit which will be illustrate in the next section. The frequency of the square signal is measured by PIC 16F628 with the help of software.

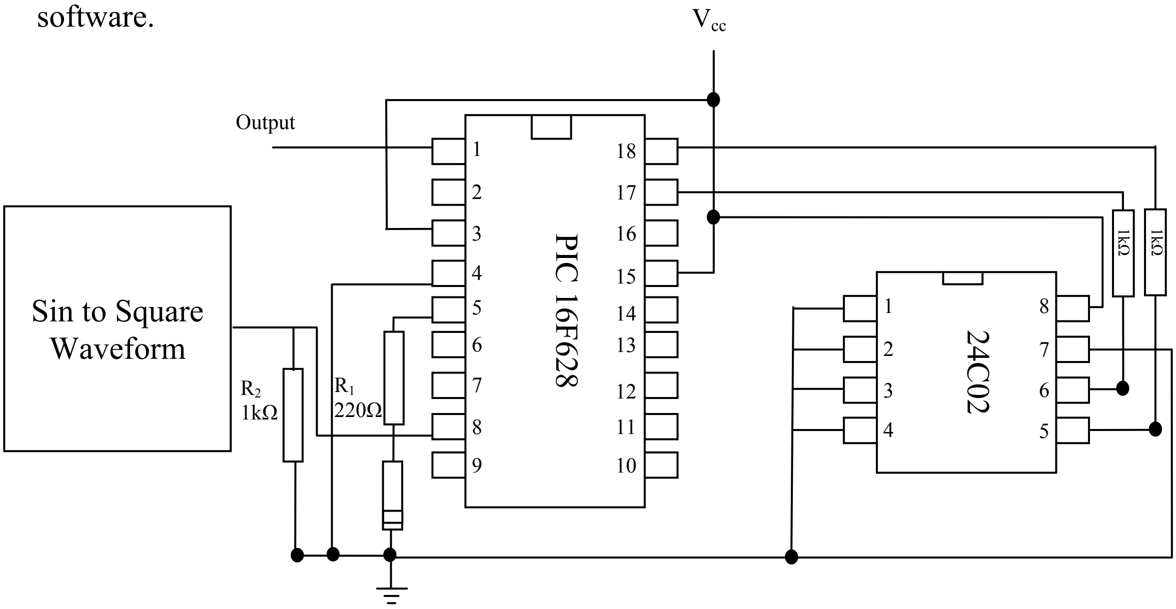


Figure (3.7): Frequency counter circuit

The squarewave signal from the buffer is fed as input frequency to TMR0 (TOCKI or RA4 in a PIC 16F628) . TMR0 is configured to measure the input frequency, at RA4 of the PIC 16F628. The input frequency is “gated” for a precise duration of time.

Before starting this precise “gate”, TMR0 is cleared. The precise “gate” is implemented in software as an accurate delay. A 8-bit value of the input frequency is now saved in TMR0, Registers and 3-bit prescaler. By concatenating the calculated value and the original value from TMR0 (256-N), the 8-bit value for the frequency is determined.

3.2.1 Hardware design

The hardware diagram of the Sinusoidal to square waveform is shown in Figure 3.8. The -5V sinusoidal signal is fed to LM741 operational amplifier to convert the signal into ± 5 square wave signal.

The signal is fed to the input of the IC: 4N25/A (6-pin DIP Optoisolator Transistor Output) which gives +5 V when the input is positive and zero when the input is negative.

The output of this circuit is not completely stable which might affect our phase and frequency measurement. Therefore, the output of 4N25/A is connected to buffer to stablize signal.

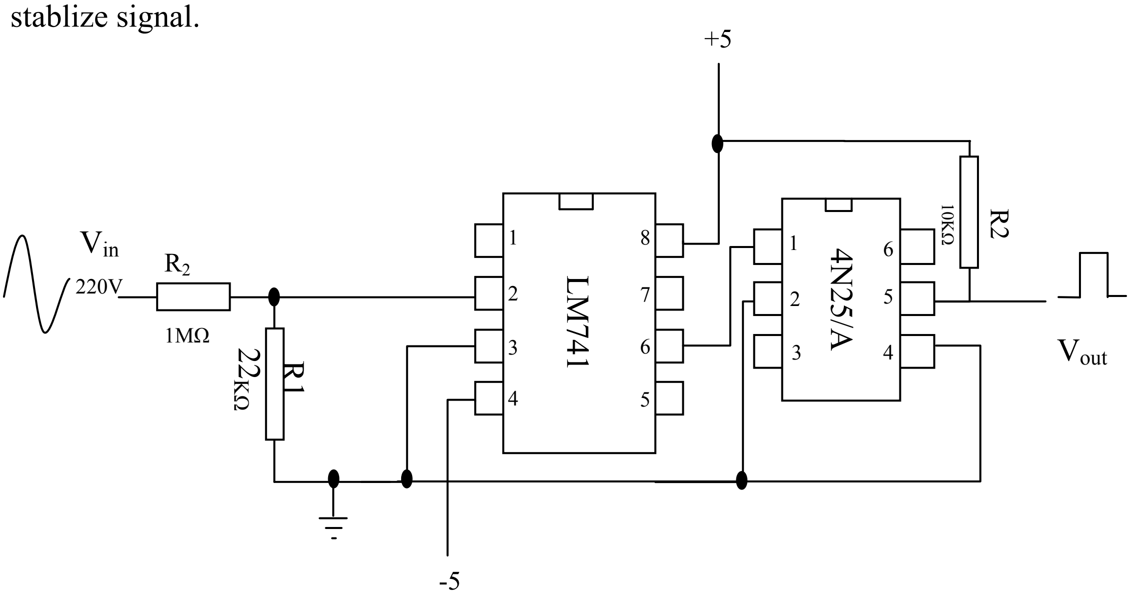


Figure (3.8): Sinusoidal to square waveform converter circuit

The oscilloscope output is captured after and befor the Sinusoidal to square waveform converter circuit as in Figure 3.9

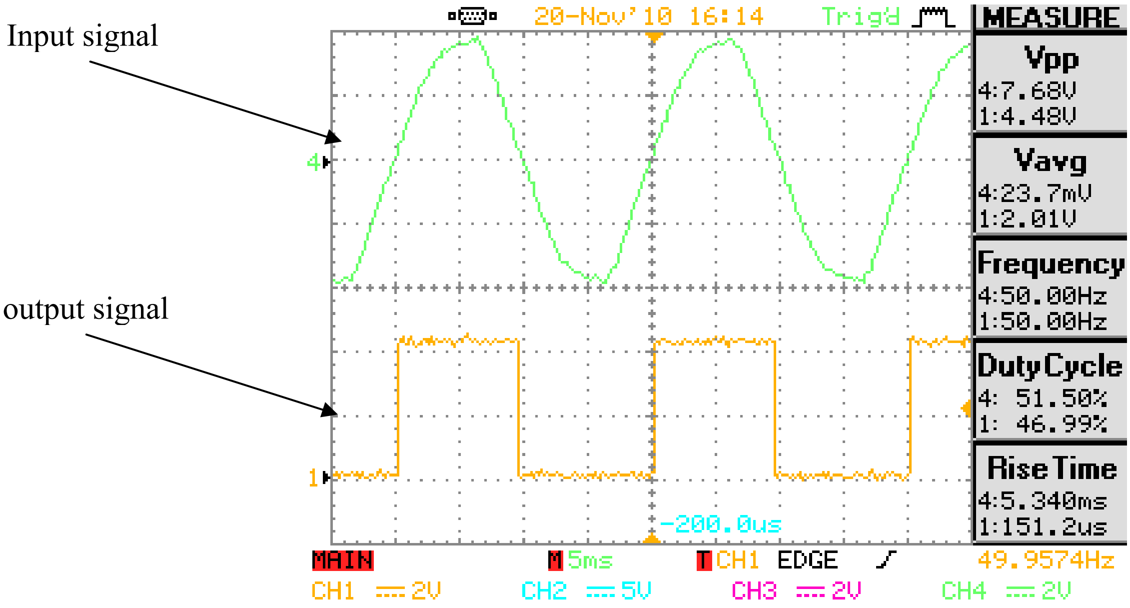


Figure (3.9): The oscilloscope output of the Sinusoidal to square waveform converter circuit

3.2.2 Software design

This is a 3-digit counter counting up to 99.9 Hz. Hardware is very simple. It contains:

1. PIC 16F62
2. LM741 operational amplifier
3. 4N25/A (6-pin DIP Optoisolator Transistor Output)
4. Some resistors, and some capacitors.

The counter uses internal prescaler of PIC. Timing loop values must be from 1 to 255. The overall program steps are:

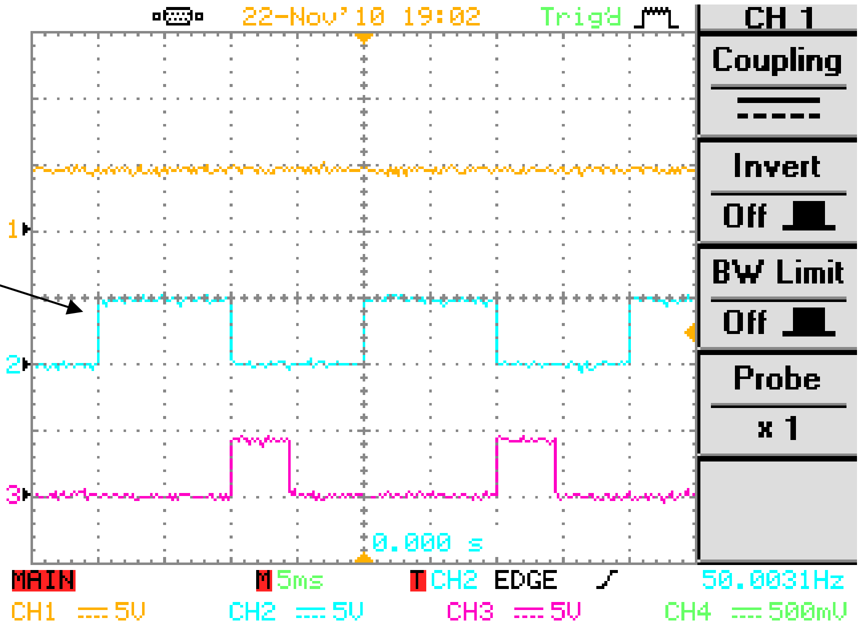
1. Configure the timer to count all rising edges of the square wave,
2. Configure a window of time slot equal 1 s,
3. At the end of one secound the value of the counter will be captured,
4. Send the vale to the memory,
5. Clearing the counter,
6. Goto 1

3.2.2 Frequency meter testing

The frequency counter has been built and tested as a standalone sub-unit. A function generator output has been connected to the counter and the frequency is measured. Table 3.2 shows the difference between the real frequency from the function generator screen and the frequency measured by the counter.

S.No.	Frequency measured on oscilloscope “HZ”	Frequency measured by frequency meter “HZ”	Frequency measured by Ref [29] “HZ”
1	49.8	50	50.1
2	49.7	49.9	50
3	49.9	50	50.1
4	49.8	50	50.1
5	49.9	50	50.1
6	49.9	50	50.1
7	49.9	50	50.1
8	49.9	50	50.1
9	49.8	49.9	50
10	49.9	50	50.1

Table (3.2): Frequency counter results



Frequency signal

Figure (3.10): The oscilloscope output of frequency counter circuit

3.3 Phase meter

The phase meter has an important role in our application as it is vital to have accurate and precise measurement for the phase. Inaccurate measurement might cause phase-lag or lead where both can lead to severe damage to the system as explained in Chapter 2. Several attempts have been made to design and fabricate digital phase meters, they suffer from limitations like compactness, complexity in design, lack of storage, and serial communication facilities etc. [30], which are very important for our application. The PIC16F628 microcontroller based phase meter has been used to get over the above difficulties.

3.4.1 The principle

The main idea behind the measurement of phase-difference between two signals is to relate the time difference between the zero crossing for both signals and the actual phase difference as shown in Figure 3.11.

In Figure 3.11, the phase shift "Φ" and the delay ' τ ' between the zero crossings of the signals is presented by a simple equation below;

$$\Phi = (\tau / T)^{*}\, 360 \qquad (3.1)$$

Where T is the time period of the signal and 'Φ' is the phase difference in degrees.

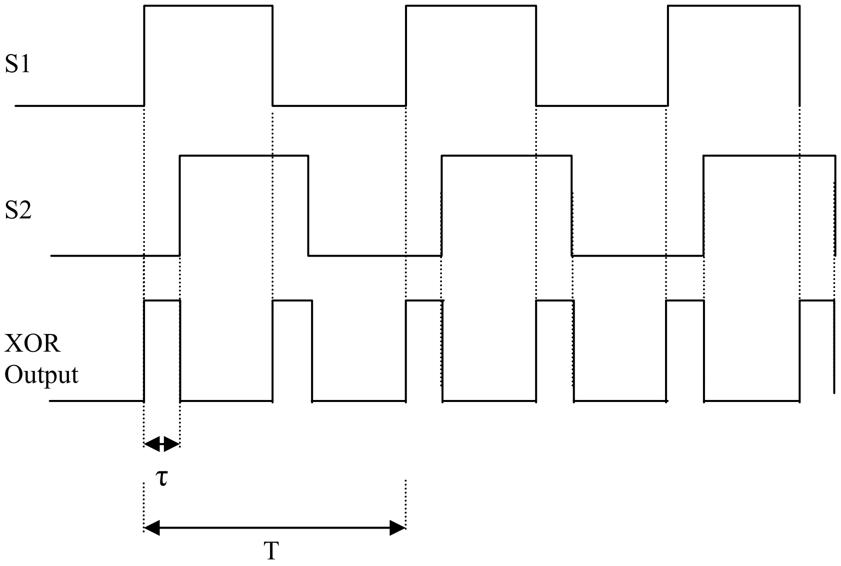


Figure (3.11): Diagram for phase shift

3.4.2 Hardware design

Figure 3.12 shows the schematic diagram of a PIC16F628 microcontroller based phase meter.

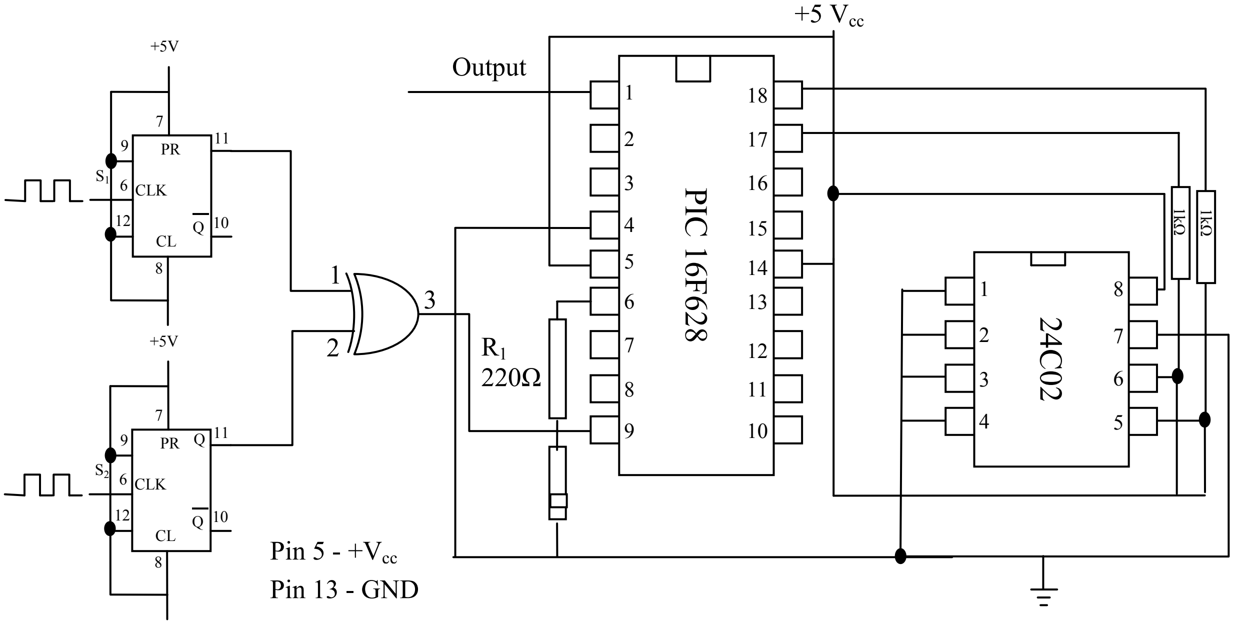


Figure (3.12): Phase meter circuit

To measure the time difference between the zero crossings, we can multiply the two signal modules using XOR gate. The relationship between the on times for the output of the XOR with the phase difference is given in Figure 3.13. However, the relationship is not one-to-one (i.e. two phase difference one below 180 and one above can be related to the same on time).

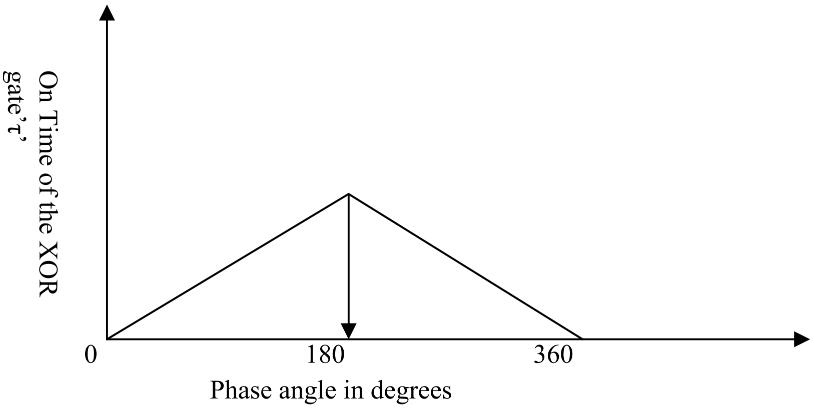


Figure (3.13): Phase angle versus on time of output of XOR gate

To overcome this problem and convert the relationship to be one-to-one, two jk Flip-flops 1 and 2 have been used as shown in Figure 3.9. Each signal is passed to one of these flip-flips to convert it into a square wave signal of a 50% duty cycle. The frequencies of these square waves are half that of the input signals. The outputs of these two flip-flops are applied to an XOR gate which produces a signal that's 'ON' time 'τ' is proportional to the phase difference between these two signals as in Figure 3.14.

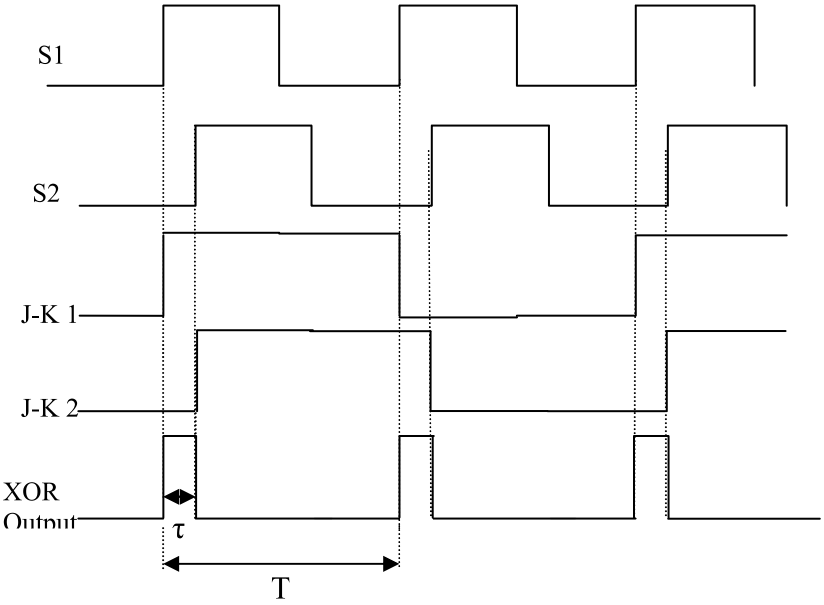


Figure (3.14): Phase angle versus on time of output of XOR gate

The 'ON' time of the XOR gate varies linearly with the phase difference from 0-360°. Figure 3.15 shows the linear relationship between the phase angle and the output of the XOR gate.

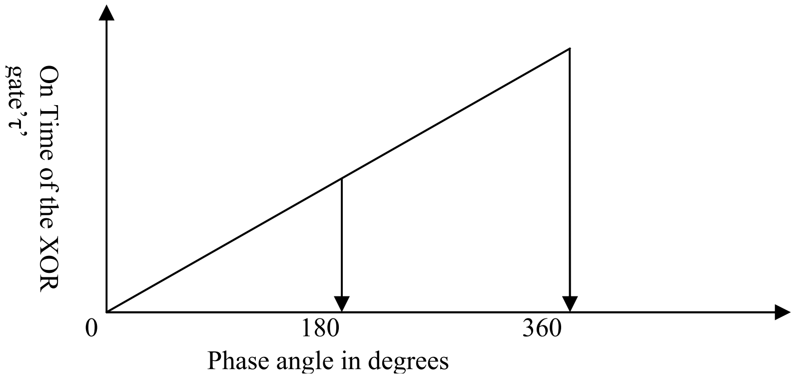


Figure (3.15): Phase angle versus on time of output of XOR gate

The output of the XOR gate is fed to the input INT of the PIC16F628. The PIC16F628 calculates the on time of each pulse of the input signal. Then the PIC calculates the phase difference using equation 3.1. The results is stored in the memory.

3.4.3 Software design

The whole process is illustrated as following:

1. Configure timer as a window with time slot 1 us.
2. Configure RA4 as digital input
3. Clearing the timer
4. At every rising edge, the timer will bigan counting
5. At every falling edge the timer will stop
6. The value of the timer will be captured
7. Send the value the the memory
8. Go to 3

3.4.4 Phase meter testing

The phase meter has been built and tested as standalone sub-unit. Two independent function generators output have been connected to the circuit. The phase difference is measured by means of our device and the oscilloscope. Table 3.3 shows the resultant output. Figure 3.15 is the oscillscope output that show the output of the phase meter.

S.No.	Phase measured on oscilloscope “degree”	Phase measured by phase meter “degree”	Phase measured by Ref [30] “degree”
1	13.3	13.5	13.6
2	13.3	13.5	13.6
3	13.12	13.4	13.5
4	13.2	13.4	13.5
5	13.2	13.4	13.5
6	13.1	13.35	13.4
7	13.03	13.2	13.3
8	12.9	13.2	13.3
9	1301	13.2	13.3
10	13.1	13.2	13.3

Table (3.3): Phase meter results

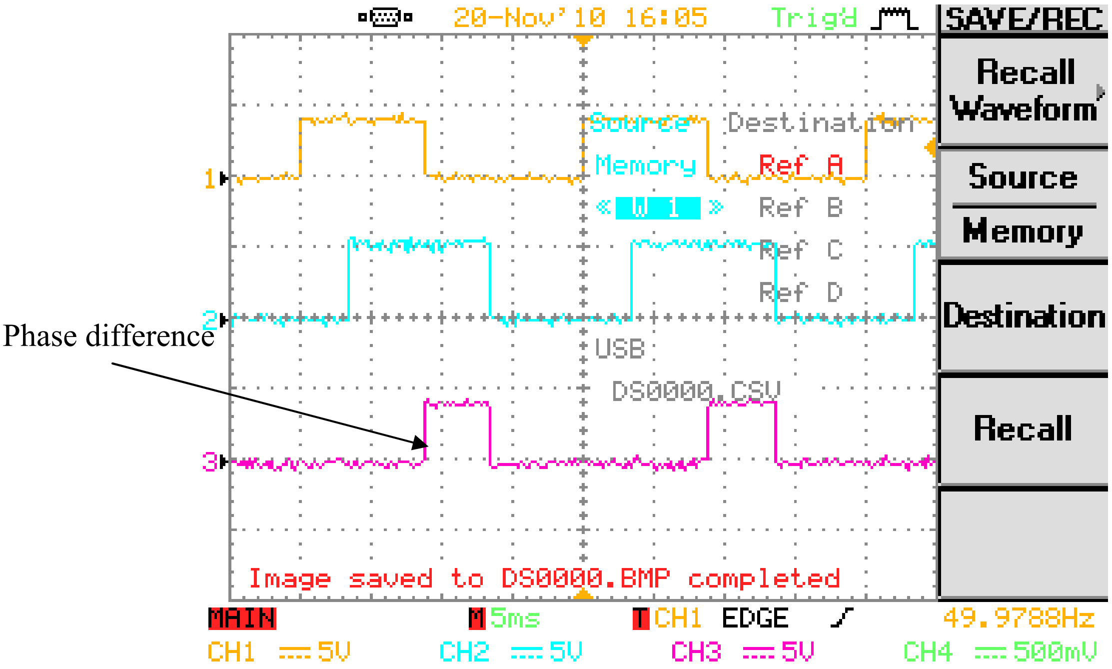


Figure (3.16): The output of the phase meter by oscilloscope

3.4 Motor driver

Automatic Mechanical system is used on the generator “Launtop” to adjust the speed of rotation that keep the output frequency at the desired level (50Hz). A small arm on the governor of the generator doses this adjustment.

If the arm is rotated counterclockwise the fuel pumped to the generator is reduced and therefore the rotation speed is also reduced. On the other hand, if the arm is rotated in the clockwise direction, the fuel pumped to the generator is increased and therefore the rotation speed is also increased.

To control this arm, we have connected it to a small motor by a strong thread. To control the motor, we need an interface driver between the controller and the motor.

So, the Dual full-bridge driver L298 is used to control the speed and the direction of rotation of the motor according to predefined inputs.

3.4.1 The principle

The conventional full bridge method is used to control the motor direction. Where four transistors are used two of them are active as shown in Figure 3.11.

The active transistors in the L298 chip determine which lead of the motor is connected to the ground and which is connected to the power supply and therefore can force it to rotate on both directions.

In order to control the speed, we use the Pulse Width Modulation PWM which is implemented using software. Pulses with different widths are generated so the input dc value is changed and thus it increase/decrease the rotation speed.

3.4.2 Hardware design

Figure 3.17 shows the schematic diagram of connection of Dual full-bridge driver with the motor.

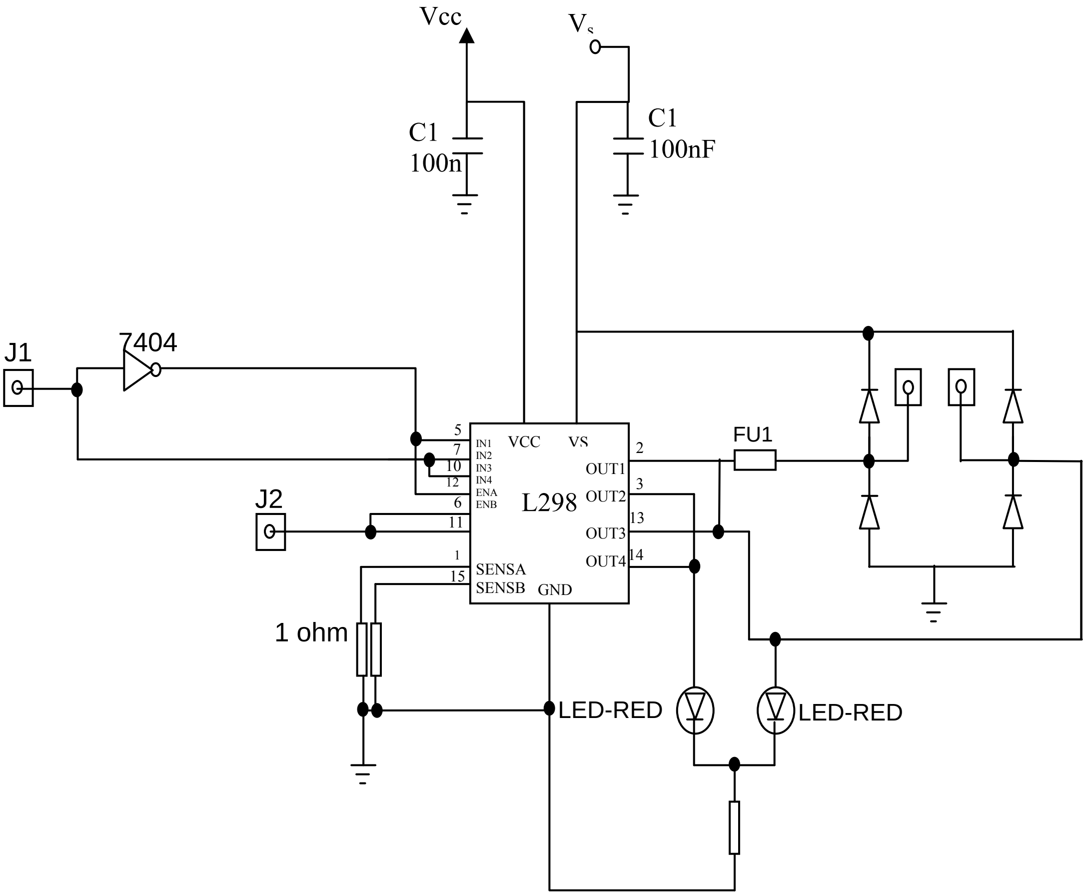


Figure (3.17): Motor driver circuit

The L298 is an integrated monolithic circuit in a 15- lead Multiwatt and Power SO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device in dependently of the input signals.

3.4.3 Software design

This part discussed the algorithm of the control strategy of the motor. As mentioned before the motor is used to regulate the phase of the generator by rotating its calibrating screw. This operation will vary the frequency of the generator by a small fraction referenced to the 50Hz from the board then this small increment in frequency will shift the generated signal then vary the phase. After arriving the desired phase value, the motor will return and stop.

The strategy of the code will be as follow:

1- reading the phase value
2- comparing it with desired value
3- control the motor such that vary the phase to the desired value
4- stop the motor and maintain this state

3.5 Summary

There are many types of metering device used in to measure the characteristics of the generator. This chapter focused on the voltmeter circuits to capture the characteristics value from 220 V AC. section one demonstrated the volt meter circuit and show the results of it. Where section two listed the frequency counter types and demonstrated the frequency counter circuit and show the results of it. In the third section the phase meter circuit is designed tested. Section four illustrated the connection of the DC motor of the generator's governor.

CHAPTER 4 OVERALL SYSTEM

In this chapter, we explain the whole system for synchronizing two generators. This system consists of three main components. The first one is the breaker closing control unit, which is used to take a decision for breaker closing when the two generators are satisfying the synchronization conditions. The generator Control Unit reads the power meter output provided by the data acquisition module for each generator and regulates it to satisfy the reference values. The third component is the data acquisition module and power meter module, which is formed from the meter circuits. Figure (4.1) shows the diagram of the overall system.

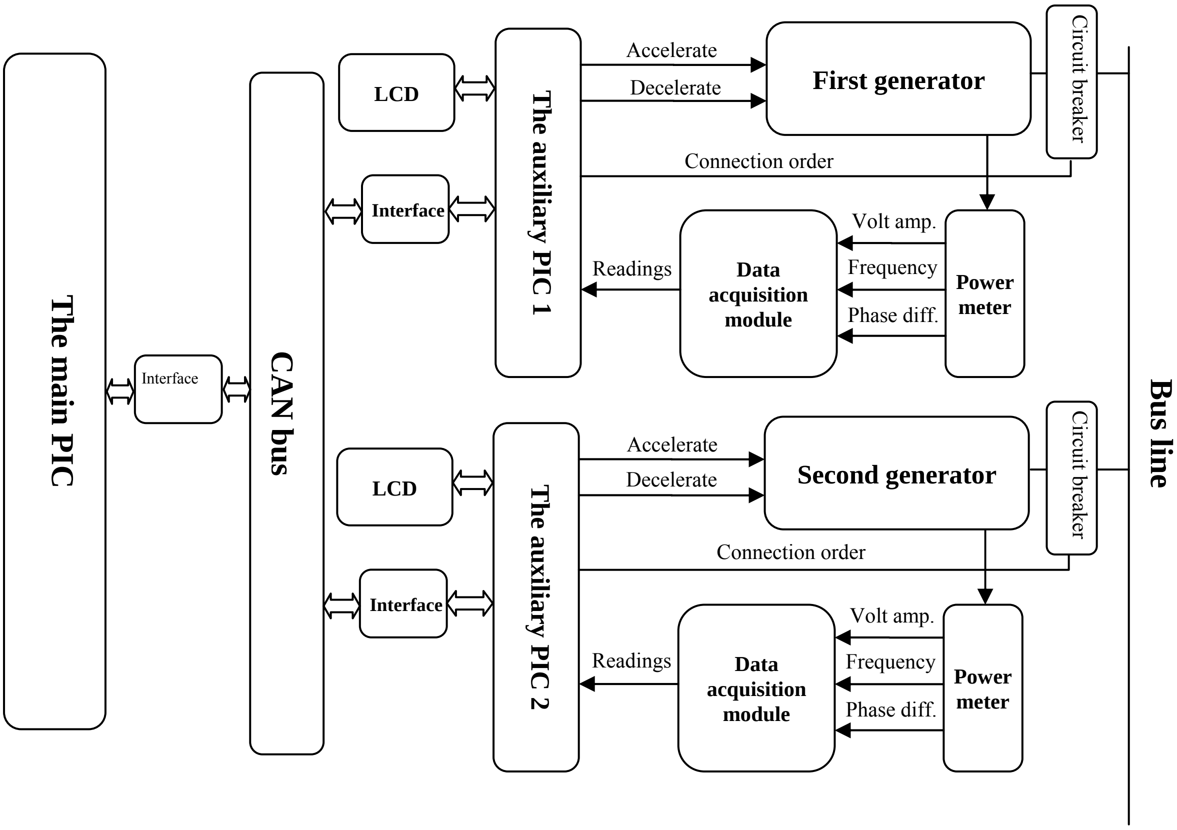


Figure (4.1): The overall system diagram

4.2 The breaker closing control unit

Many devices can synchronize the generator to a power system or to another generator. Such a device must first test if the voltage, frequency and phase between the two generators are within the acceptable limits then takes the decision to connect them. The principles of the synchronizing devices are different, some of them are located at crossing point between the two system needed to be synchronized, in other systems devices are need to be located at the system's sides.

For example SYNCHROTACT 5 shown in Figure 4.2 (a) is the fifth generation of synchronizing equipment produced by ABB. SYNCHROTACT products are used for automatic synchronization of generators with power lines and for paralleling of synchronous lines. They are designed for fully automatic operation by dual channel or single-channel systems [34].

(a) (b)

Figure (4.2): Synchronizing devices

Another example is shown in Figure 4.2 (b) has been developed for parallel operation of AC generators isolated or with the grid. It owes the built-in load sharing function. Connection of one synchronizer to each generator is required [35].

The main component in our design is PIC18F4685. This PIC has 8Kbit program memory, has been operated with 4 MHz oscillator, Loop time of each command is 1µs, Comparator, ADC, Timer-1 modules and A, B, E ports of the microcontroller has been used in the application. Port configuration of the microcontroller used for the study is shown in Figure (4.3), 3328 bytes SRAM and 1024 bytes EEPROM.

The main PIC "18F4685" controls auxiliary PIC "18F4685" for each generator to capture signal voltage, frequency and phase angle from the measuring modules. After that, the auxiliary PIC for each generator sends ok signal, which means that the measuring data are in the acceptable limits.

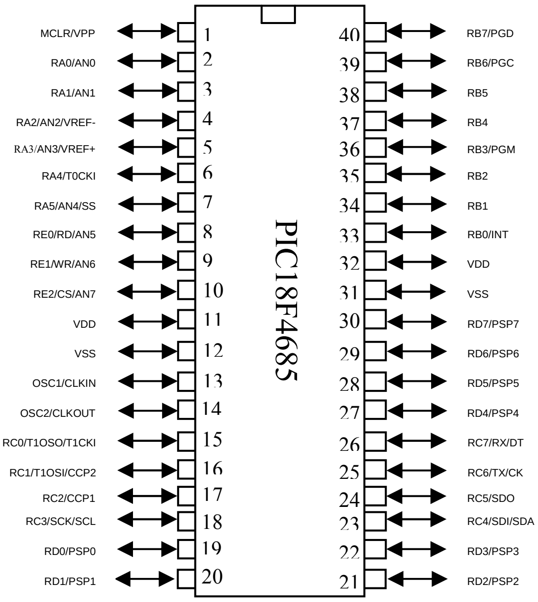


Figure (4.3): Pin diagram of PIC 18F4685

The PIC performs the generator paralleling which is the main function of the synchronizing system when the voltage, frequency, and phase deference are within the acceptable limits. A relay realizing the parallel connection is connected to input number 4 input of the port B. The layout and pin connection of the main PIC is shown in Figure (4.4).

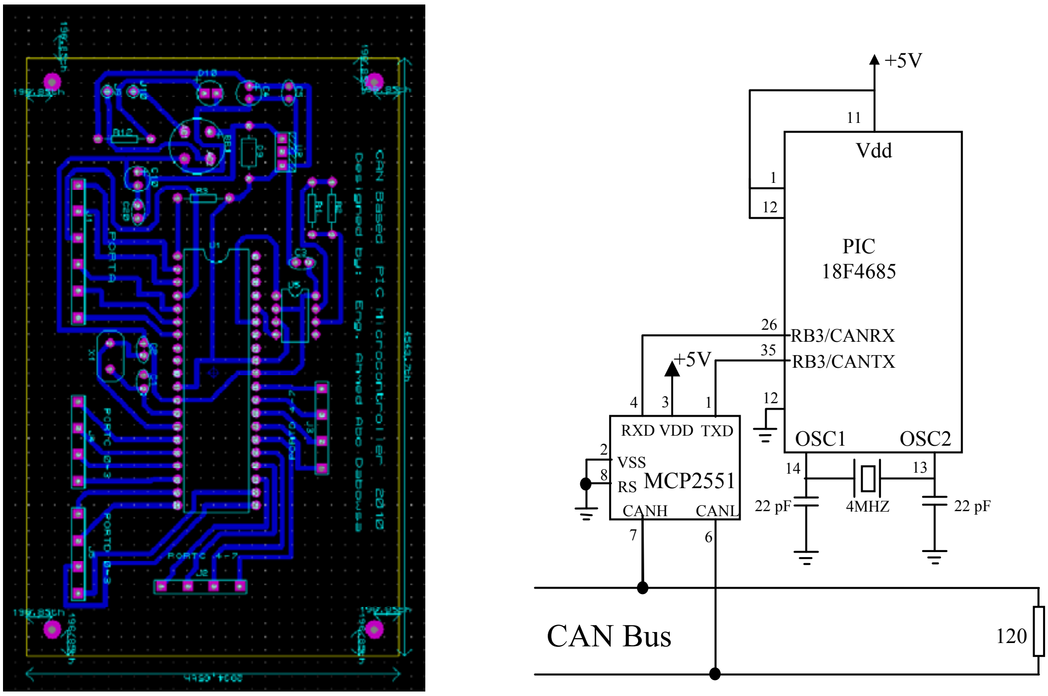


Figure (4.4): Layout of the main PIC 18F4685

Figure 4.5 shows the flowchart of the main software program. The software is uploaded on the main PIC. As shown in the flowchart; the main PIC sends a start signal to the auxiliaries PICs to start controlling the generators to achieve the acceptable limits of the parallel connection conditions. The main PIC waits for both auxiliary PICs to send their ‘OK’ signals and then it issues the closing command.

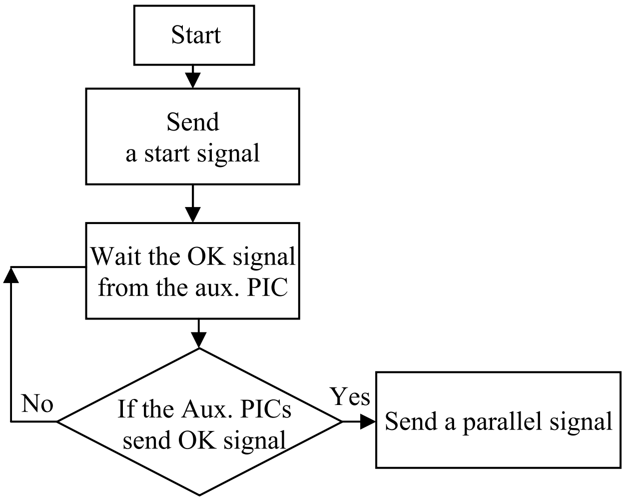


Figure (4.5): Flowchart of the main software program

4.3 The generator control unit

In order to minimize external circuitry and not to lose the time of main chip, an auxiliary PIC18F4685 is used as a timer and supplementary controller for each generator. This gives a room for extra time for the main controller to complete control and communication functions. Therefore, the complete system has higher accuracy, reliability, and celerity.

The output voltmeter, frequency meter and phase meter described in details in Chapter 3 are connected to the inputs of the auxiliary PIC via the data acquisition module which will be described in detail in the section 4.3.

The auxiliary PIC issues an order to the data acquisition module to increase or decrease each of the measured values in order to realize the accepted predefined margins.

When these margins are achieved, the auxiliary PIC sends 'OK' signal to the main PIC. The connection diagram between the Auxiliary PIC and the data acquisition module is shown in Figure 4.6.

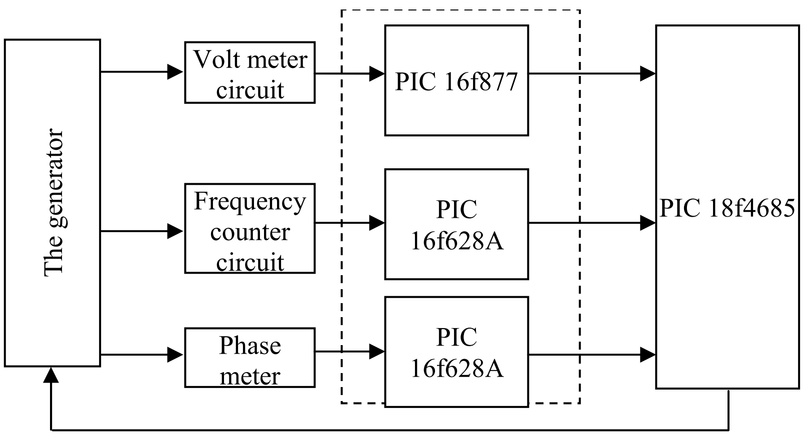


Figure (4.6): Connection circuit of the Auxiliary PIC 18F4685 with the data acquisition module

The layout of the auxiliary PC is the same as that of the main PIC and pin connection of the auxiliary PIC is the same one for the main PIC except that it has a connection to an LCD to show the measured data and also it has a connection with data acquisition module as shown in Figure (4.7) below.

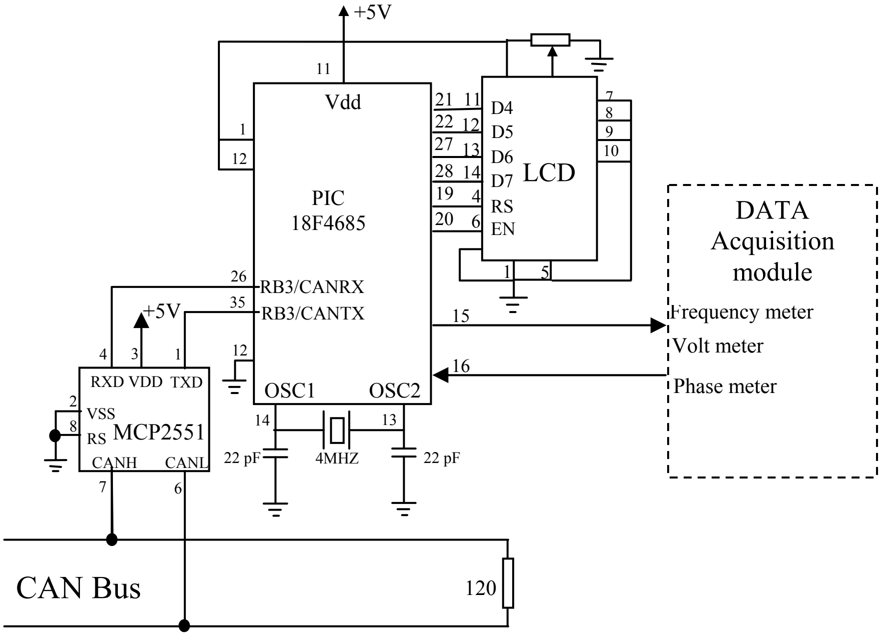


Figure (4.7): The pin connection of the auxiliary PIC with the data acquisition module and LCD

Figure 4.8 shows the flowchart of the auxiliary software program. The software is uploaded on the auxiliary PIC. As shown from the flowchart; frequency, voltage, and phase of the generator are measured, respectively.

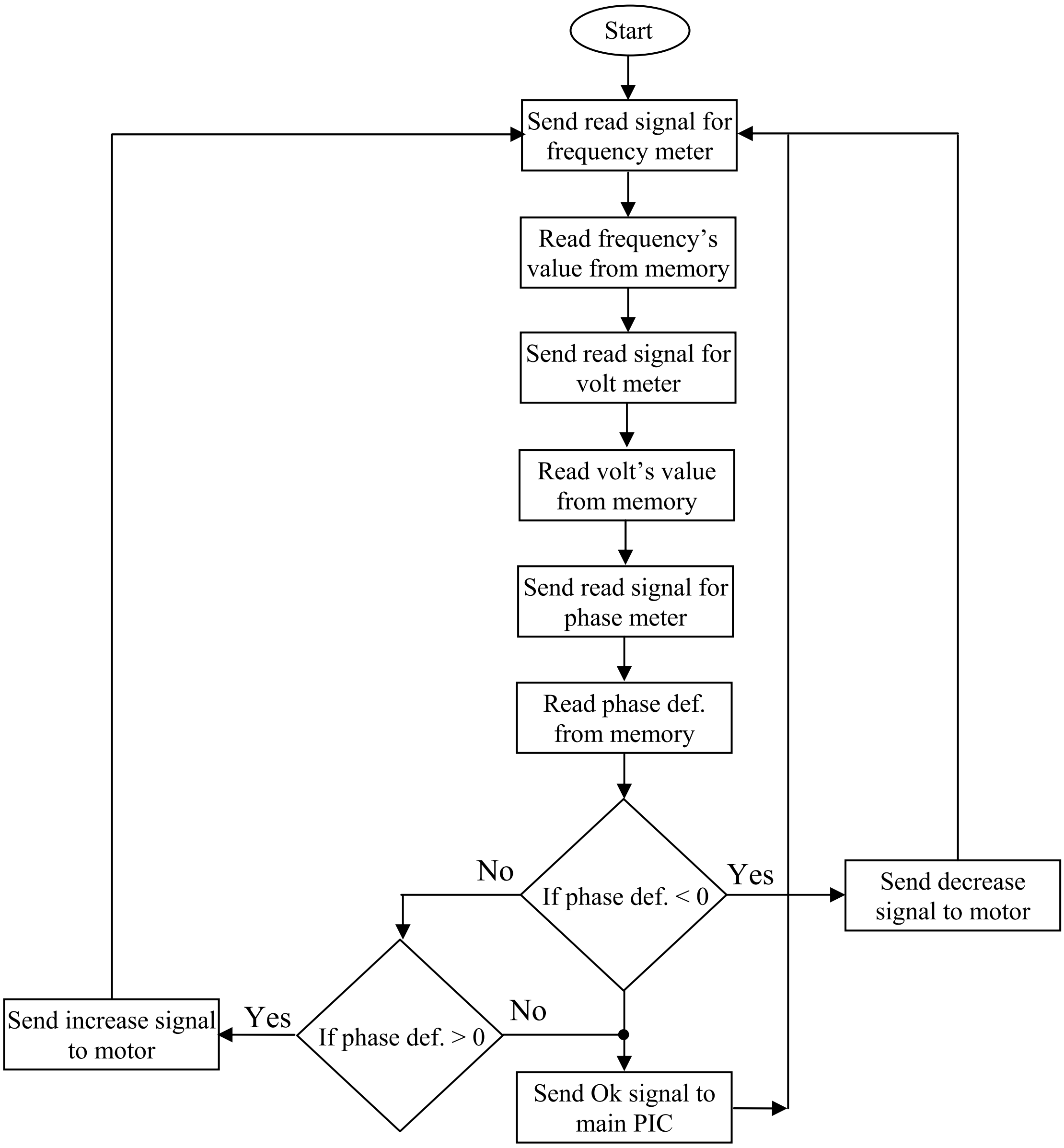


Figure (4.8): The flowchart of the auxiliary software program

The measured phase is compared to the phase of the reference signal as explained in Section 3.3. According to this phase difference, the PIC sends a signal to the speed motor connected with the generator governor. If the phase difference is negative, the "decrease" signal is issued to the motor to rotate in the counterclockwise direction and hence the frequency of the generator is reduced. Otherwise, i.e. the phase difference is positive; the “increase” signal is issued to the motor to rotate in the clockwise direction and hence the frequency of the generator is increased.

In both cases, the phase difference is measured and if it falls within the acceptable limitation, no action is taken. This guarantees that both phase and frequency approximately match those of the reference signal. The chosen generator, which is “5KW LUNTOP generator”, has Automatic Voltage Regulator (AVR). The measured voltage is only for double-checking.

4.4 The power meter

The power meter module consists of two parts. The first part is the collection of the volt, frequency, and phase meter circuits as shown in Figure 4.9. The second part is the collection of the PIC 16F877 and 16F628A as shown in Figure 4.10. The second part process the data from the first part and sends it to the auxiliary PIC.

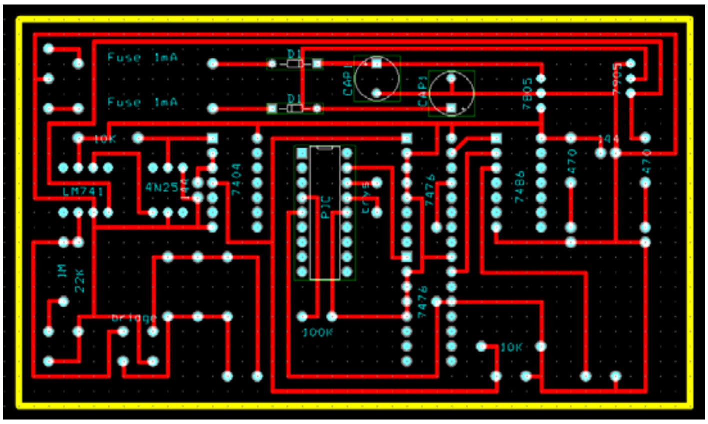


Figure (4.9): Part one of the power meter module

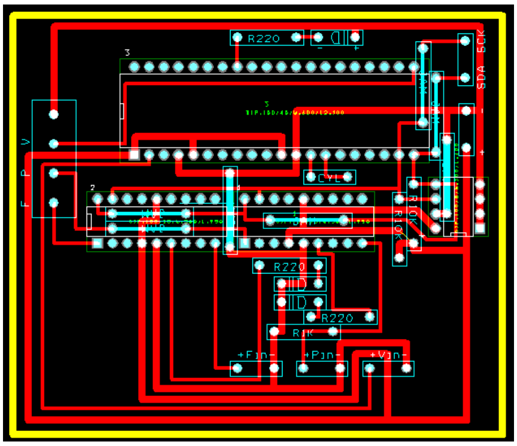


Figure (4.10): Part two of the power meter module

At the end of this chapter the overall system can be describe as follows:

First, the output of each generator connect to the power meter module that read the three components of the voltage (voltage amplitude, frequency, and phase difference with respect to reference signal) by using three circuits, one for each measure.

Second, the outputs of the power meter is fed to the data acquisition module which consists of three PICs. The first PIC is PIC16F877, which reads the analog AC voltage; convert it to digital, and saves it in a memory chip. The second PIC is PIC16f628 that reads the frequency output; counts it and saves the number in the same memory mention before. The third PIC is also PIC16F268 reads the phase difference and recalculate to get a digit and saves it in the same memory.

Third, the auxiliary PIC which is PIC18F6485 captures the three components from the memory and compares with the acceptable limits. If they are out of range it sends a control signal to a motor driver module where the motor driver module will control the governor of the generator to decrease or increase the fuel fed the generator. At the moment that the voltage components satisfying the acceptable limits the auxiliary PIC send an OK signal to the main PIC

Last, the main PIC, which is also PIC18F6485, receives the ok signals from the two auxiliary PIC and once it receives them it send a connection order to auxiliary PIC, where it sends an order to the circuit breaker, which connects the line of the generator to the bus line. The connection diagram between the main PIC and the auxiliary PIC shown in Figure 4.11.

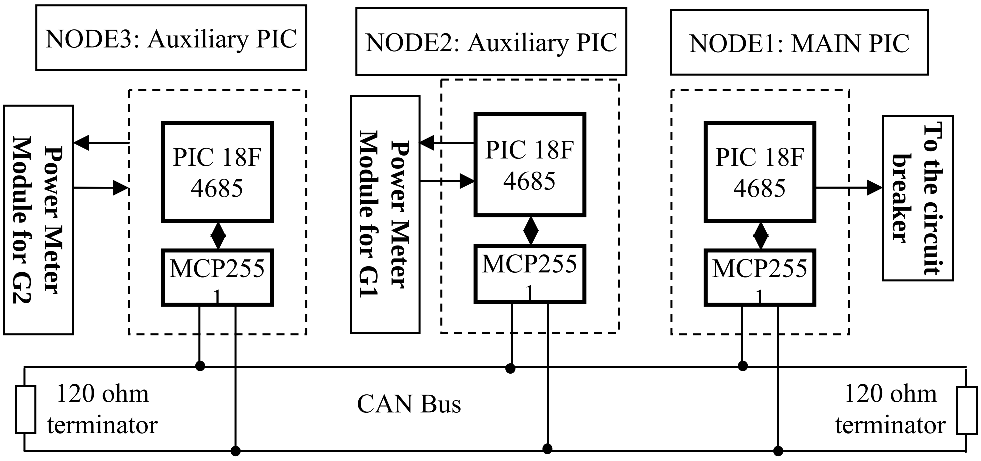


Figure (4.11): Connection diagram of the main PIC 18F4685 with auxiliary PIC

4.5 Summary

Chapter 5 displayed the overall system and the main components of it and the connection between them. Section 1 described the breaker control unite. In section two the generator control unit was declared. Moreover, in section three the power meter component explained.

CHAPTER 5 CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

5.1 Conclusion

There are intensive and chronicle problems in the power distribution in Gaza Strip. To overcome these problems, two or more generators are connected together which is referred as synchronizing generators. Synchronization offers many advantages like reliability, expandability, flexibility. To do that the volt and frequency and phase differences between the generators must be taken into account, where the voltage, frequency and phase differences must be within acceptable parameters.

Faulty synchronizing can damage the electrical and mechanical generating systems, causes disturbances to the power system and causes the unit to trip offline.

There are two main methods to synchronize generators: automatic and manual. The manual one depends on a well trained operator where the automatic depends on a device based on hardware or software technique.

The synchronizing process needs fast real time data communication so after along research the CAN protocol is chosen which it is the main contribution of the thesis.

Synchronizing device proposed in the thesis is an Automatic device based on CAN protocol. It includes three control units to read the characteristics of two generators and adjusting the generators to become identical with each other.

The three control units are built from PIC 18F4680 connected via CAN protocol, two of them are similar to each other, which are called the generator control unit, and they are amounted on the two generators to read the characteristics of the generators, and regulate the phase difference relative to a reference signal, and send OK signal to the third unit via CAN protocol. The third unit is the circuit breaker control unit, which waits for the OK signal from the generator control units to issue an order for closing the circuit breaker.

The system was implemented, tested and had acceptable results.

5.2 The Suggestions for future work

The synchronize device developed in this thesis require to add volt control and frequency control to its controller which give the device ability to adjust the volt and frequency within the acceptable limits of the synchronizing process to get full control to the generator

The most important thing is to generalize the device to synchronize bigger generators. We hope that this work will be completed, and the device will become available for practical Appling.

Another suggestion is to add the SCADA system characteristic for supervisory control and data acquisition data.

REFERENCES

[1] Colak I., Synchronous Machines, 2003, Seckin Press, In Ankara-Turkish.

[2] Generac Power Systems Inc, Generator Paralleling-Technical Perspective, 2006, In USA.

[3] Z. Ying, D. Cun-Lu, J. Chong-Peng, C. Li “Design of Generator Synchronizing Device Based on PIC16F877 and FPGA”. ISA 2009 “Intelligent Systems and Applications 2009”, Wuhan, 978-1-4244-3893-8, IEEE 2009.

[4] R. A. EVANS, “A Manual/Automatic Synchronization Circuit for a 37.5-MVA Steam-Turbine-Driven Generator”. IEEE Transaction on Industry Applications. vol. 26, no 6, November/December 1990

[5] “Power O and M. Bulletin” No. 27, US Dept., of the Interior, Bureau of Reclamation, Denver, Colorado, 1957.

[6] H. Chun, J. Yaqun, J. Yan, C. Li “Novel Automatic Synchronizer Based on Dual Principles and Dual Microprocessors”, 10.1109/ICPST.2006.321439 International Conference on Power System Technology 2006.

[7] L. C. Gross, L. S. Anderson, and R. C. Young, “Avoid generator and system damage due to a slow synchronizing breaker” in Proc. 24th Annu. Western Protective Relay Conf., Oct. 21–23, 1997, pp. 1–20, 1997.

[8] W. G. Hartmann, “Automatic synchronizing for generation and tie lines” in Proc. 18th Annu. Western Protective Relay Conf., Oct. 22–24, 1991, pp. 1–28, Aug 2002.

[9] Richard A. Evans, "A Manual/Automatic Synchronization Circuit for a 37.5-MVA Steam-Turbine-Driven Generator," IEEE Transactions on Industry Applications, Vol. 26 No.6, pp. 1080- 1085, 1990.

[10] W. G. Hartmann, "Automatic Synchronizing for Generation and Tie Lines," 18th Annual Western Protective Relay Conf, pp. 22-24, in Proc. 1991.

[11] Y. H. Yang, G. C. Shang, and Y. J. Fang, "A Fast Following Synchronizer of Generators," IEEE Transactions. On Energy Conversion, vol. 3, No. 4, pp.765-770, 1988.

[12] Metwally H.M.B. “Operation of new variable speed constant voltage and frequency generator connected to the grid”, Energy Conversion and Management ,Volume 41, Issue 7, Pages 701-712, May 2000.

[13] IEEE Std C37.118-2005, IEEE Power Engineering Society, IEEE standard for synchrophasors for power systems, 2005.

[14] Shao Ming, Wang Yanfang, “Development of the-same-term switch Equipment Based on 89C51”, Journal of Zhengzhou Ytechnic Institute,pp.6-8, 2004.

[15] Shi Dejia, Lu Guangde, Zhu Jianlin, “Automatic Accurate Circle Controlling System of Synchronous Dynamo”, Natural Science Journal of Xiangtan University, pp.99-101, 2003.

[16] Shi Dejia, Lu Guangde, Zhu Jianlin, “Automatic Accurate Circle Controlling System of Synchronous Dynamo”, Natural Science Journal of Xiangtan University, pp.99-101, 2003.

[17] Emam, S.E.A., Automatic digital synchronization, International Conference on Electrical, Electronic and Computer Engineering, ICEEC'04,p. 778-784, 2004.

[18] Colak, Bayındır, R., Sefa, Demirba_, Bal, G.,Developing of a control unit for the parallel operation of AC generators, TPE’06 3rd International Conference on Technical and Physical Problems in Power Engineering, p.134-137, 2006.

[19] E. Bekiroglu, A. Bayrak “Automatic Synchronization Unit For The parallel Operation Of Synchronous Generators” EUROCON 2009, EUROCON '09.18-23 On page(s): 766 – 771, May 2009.

[20] Li Yexing, Deng Zhijie, Li Wenhui, “Design and Realization of Automatic Synchronize Device Based on DSP”, Electric Application, pp.27-30, 2006.

[21]. Bhag S. Guru, Hiiseyin R. Hiziroglu, “Electric Machinery and Transformers”, Oxford University Press, New York, In USA, 2001.

[22]. Reimert, Donald., “Protective Relay for Power Generation System”, CRC Press, New York, In USA, 2006.

[23] Michael J. Thompson, “Fundamentals and Advancements in Generator Synchronizing Systems”. Power-GEN International, in Las Vegas, 2011.

[24] IEEE Standard for Salient-Pole 50 Hz and 60 Hz Synchronous Generators and Generator/Motors for Hydraulic Turbine Applications Rated 5 MVA and Above, IEEE Standard C50.12-2005.

[25] IEEE Standard for Cylindrical-Rotor 50 Hz and 60 Hz Synchronous Generators Rated 10 MVA and Above, IEEE Standard C50.13-2005.

[26] IEEE Guide for Operation and Maintenance of Turbine Generators, IEEE Standard 67-2005.

[27] http://www.scribd.com/doc/22987422/Introduction-to-Automatic-Synchronizing

[28] Dogan Ibrahim, “Advanced PIC Microcontroller Projects in C”, Oxford University Press, In USA, 2008.

[29] Wai P. Aung, , “Implementation of PIC based Digital Frequency Counter”, World Academy of Science, Engineering and Technology, 2008.

[30] L. Shrimanth Sudheer, S.S. Patil*, R. Laxmikant, Nagabhushana Katte and P. Bhaskar, “Microcontroller based phase meter”, Vol. 39 No. 1, Jl. of Instrum. Soc. of India, 2009.

[31] STMicroelectronics GROUP OF COMPANIES Inc, L298 Data Sheet, www.st.com, 2000.

[32] http://embedded-lab.com/blog/?p=396

[33] http://www.allaboutcircuits.com/vol_3/chpt_8/3.html

[34] ABB Switzerland Ltd, SYNCHROTACT® 5 Synchronizing and Paralleling Equipment and Systems for Synchronous Machines and Networks, www.abb.com/synchrotact, Inc.2010.

[35] SELCO Ltd, synchronizer E7600.

[36] http://www.mikroe.com/eng/home/index/

APPENDIX A

1. MicroC Software Code Implemented Inside PIC16F4685 for Variable resistance CAN Bus example

Program for LCD

```
void main()
{
unsigned char temperature, data[8];
unsigned short init_flag, send_flag, dt, len, read_flag;
char SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_Seg, txt[4];
long id, mask;

TRiSC = 0; // PORTC are outputs (LCD)
TRiSB = 0x08; // RB2 is output, RB3 is input
//
// CAN BUS Parameters
//
SJW = 1;
BRP = 1;
Phase_Seg1 = 6;
Phase_Seg2 = 7;
Prop_Seg = 6;
init_flag = CAN_CONFiG_SAMPLE_THRiCE &
CAN_CONFiG_PHSEG2_PRG_ON &
CAN_CONFiG_STD_MSG &
CAN_CONFiG_DBL_BUFFER_ON &
CAN_CONFiG_VALiD_XTD_MSG &
CAN_CONFiG_LiNE_FiLTER_OFF;
send_flag = CAN_TX_PRiORiTY_0 &
CAN_TX_XTD_FRAME &
CAN_TX_NO_RTR_FRAME;
read_flag = 0;
//
// lnitialize CAN module
//
CANinitialize(SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_Seg, init_flag);
//
// Set CAN CONFlG mode
//
CANSetOperationMode(CAN_MODE_CONFiG, 0xFF);
mask = -1;
//
// Set all MASK1 bits to 1's
//
CANSetMask(CAN_MASK_B1, mask, CAN_CONFiG_XTD_MSG);
//
// Set all MASK2 bits to 1's
//
CANSetMask(CAN_MASK_B2, mask, CAN_CONFiG_XTD_MSG);
//
// Set id of filter B2_F3 to 3
//
CANSetFilter(CAN_FiLTER_B2_F3,3,CAN_CONFiG_XTD_MSG);
//
// Set CAN module to NORMAL mode
//
CANSetOperationMode(CAN_MODE_NORMAL, 0xFF);

//
// Configure LCD
//
Lcd_Config(&PORTC,4,5,0,3,2,1,0); // LCD is connected to PORTC
Lcd_Cmd(LCD_CLEAR); // Clear LCD
Lcd_Out(1,1,"CAN BUS"); // Display heading on LCD
```

```
Delay_ms(1000); // Wait for 2 seconds
//
// Program loop. Read the temperature from Node:COLLECTOR and display
// on the LCD continuously
//
for(;;) // Endless loop
{
Lcd_Cmd(LCD_CLEAR); // Clear LCD
Lcd_Out(1,1,"Temp = "); // Display "Temp = "
//
// Send a message to Node:COLLECTOR and ask for data
//
data[0] = 'T'; // Data to be sent
id = 500; // ldentifier
CANWrite(id, data, 1, send_flag); // send 'T'
//
// Get temperature from node:COLLECT
//
dt = 0;
while(!dt)dt = CANRead(&id, data, &len, &read_flag);
if(id == 3)
{
temperature = data[0];
ByteToStr(temperature,txt); // Convert to string
Lcd_Out(1,8,txt); // Output to LCD
Delay_ms(1000); // Wait 1 second
}
}
}
```

Program for collector

```
//Transmitter

void main()
{
unsigned char temperature, data[8];
unsigned short init_flag, send_flag, dt, len, read_flag;
char SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_Seg, txt[4];
unsigned int temp;
unsigned long mV;
long id, mask;
TRiSA = 0xFF; // PORTA are inputs
TRiSB = 0x08; // RB2 is output, RB3 is input
//
// Configure A/D converter
//
ADCON1 = 0x80;
//
// CAN BUS Timing Parameters
//
SJW = 1;
BRP = 1;
Phase_Seg1 = 6;
Phase_Seg2 = 7;
BRP = 1;
Prop_Seg = 6;
init_flag = CAN_CONFiG_SAMPLE_THRiCE &
CAN_CONFiG_PHSEG2_PRG_ON &
CAN_CONFiG_STD_MSG &
CAN_CONFiG_DBL_BUFFER_ON &
CAN_CONFiG_VALiD_XTD_MSG &
CAN_CONFiG_LiNE_FiLTER_OFF;
send_flag = CAN_TX_PRiORiTY_0 &
CAN_TX_XTD_FRAME &
CAN_TX_NO_RTR_FRAME;
read_flag = 0;
//
// lnitialise CAN module
//
```

```
CANinitialize(SJW, BRP, Phase_Seg1, Phase_Seg2, Prop_Seg, init_flag);
//
// Set CAN CONFlG mode
//
CANSetOperationMode(CAN_MODE_CONFiG, 0xFF);
mask = -1;
//
// Set all MASK1 bits to 1's
//
CANSetMask(CAN_MASK_B1, mask, CAN_CONFiG_XTD_MSG);
//
// Set all MASK2 bits to 1's
//
CANSetMask(CAN_MASK_B2, mask, CAN_CONFiG_XTD_MSG);
//
// Set id of filter B1_F1 to 3
//
CANSetFilter(CAN_FiLTER_B2_F3,500,CAN_CONFiG_XTD_MSG);
//
// Set CAN module to NORMAL mode
//
CANSetOperationMode(CAN_MODE_NORMAL, 0xFF);
//
// Program loop. Read the temperature from analog temperature
// sensor
//
for(;;) // Endless loop
{
//
// Wait until a request is received
//
dt = 0;
while(!dt) dt = CANRead(&id, data, &len, &read_flag);
if(id == 500 && data[0] == 'T')
{
//
// Now read the temperature
//
temp = Adc_Read(0); // Read temp
mV = (unsigned long)temp*5000/1024; // in mV
temperature = mV/10; // in degrees C
//
// send the temperature to Node:Display
//
data[0] = temperature;
id = 3; // ldentifier
CANWrite(id, data, 1, send_flag); // send temperature
}
}
}
```

2. MicroC Software Code Implemented Inside PIC16f877 to Measure the Voltage value

```
define vref 6

char txt[7],txt1[3],flag,i;
float res,res0,res1;
int res2,res3;

// Software I2C connections
sbit Soft_I2C_Scl           at Rd0_bit;
sbit Soft_I2C_Sda           at Rd1_bit;
sbit Soft_I2C_Scl_Direction at TRISd0_bit;
sbit Soft_I2C_Sda_Direction at TRISd1_bit;
// End Software I2C connections

// Lcd pinout settings
```

```
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D4 at RB4_bit;

// Pin direction
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D7_Direction at TRISB7_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB4_bit;



void write_to_24c02(char EEA,char EED)
 {
  Soft_I2C_Start();              // Issue start signal
  Soft_I2C_Write(0xA0);          // Address PCF8583
  Soft_I2C_Write(EEA);             // Start from word at address 0
(configuration word)
  Soft_I2C_Write(EED);          // Write 0x80 to config. (pause counter...)
  Soft_I2C_Stop();               // Issue stop signal
  delay_ms(10);
 }

void interrupt()
{
/*intcon.t0if=0;
      if (flag==0)
      {
      Lcd_Cmd(_LCD_CLEAR);                 // Clear display
      Lcd_Cmd(_LCD_CURSOR_OFF);            // Cursor off
      Lcd_Out(1,1,"Volt:");                  // Write text in first row
      flag=1;
      }
intToStr(res2,txt);
Lcd_Out(1,11,txt);
   tmr0=0;*/
}

void main() {
adcon1=15;
Soft_I2C_Init();
/*Lcd_Init();
Lcd_Cmd(_LCD_CLEAR);                 // Clear display
      Lcd_Cmd(_LCD_CURSOR_OFF);            // Cursor off
      Lcd_Out(1,1,"Volt:");                  // Write text in first row*/
trisd.f2=1;
option_reg = 0b01000111;
intcon.t0ie=1;
intcon.peie=1;
intcon.gie=0;
tmr0=0;
trisb=0;
portb=0;

while(1)
{portb.f0=~portb.f0;
 res=adc_read(0);
 res1=(res*5*452)/1024;
 res1=res1/10;
 res2=res1;
intToStr(res2,txt);
for (i=3;i<6;i++)
txt1[i-3]=txt[i];
```

```
/*if (isalnum(txt1[0])) { Lcd_chr(1,6,txt1[0]);} else {Lcd_chr(1,6,'0');}
Lcd_chr(1,7,'.');
if (isalnum(txt1[1])) { Lcd_chr(1,8,txt1[1]);} else {Lcd_chr(1,8,'0');}
if (isalnum(txt1[2])) { Lcd_chr(1,9,txt1[2]);} else {Lcd_chr(1,9,'0');}
Lcd_Out(1,10," V");*/

if(portd.f2 == 1)
{
for(i=3;i<6;i++)
{
write_to_24c02(i-3+10,txt[i]);
//delay_ms(10);
}
}
for(i=3;i<6;i++)
{
eeprom_write(i-3,txt[i]);
}
}
}
```

3. MicroC Code Inside PIC16f628 to count the frequency

```
char txt[7],i,txt1[3],flag;
char uart_rd,error=0;
unsigned int cnt,cnt2;
char EEA,EED;
float cnt1,cnt3;

// Software I2C connections
sbit Soft_I2C_Scl           at Ra0_bit;
sbit Soft_I2C_Sda           at Ra1_bit;
sbit Soft_I2C_Scl_Direction at TRISa0_bit;
sbit Soft_I2C_Sda_Direction at TRISa1_bit;
// End Software I2C connections

// Lcd pinout settings
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB0_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D4 at RB4_bit;

// Pin direction
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB0_bit;
sbit LCD_D7_Direction at TRISB7_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB4_bit;

void write_to_24c02(char EEA,char EED)
 {
  Soft_I2C_Start();              // Issue start signal
  Soft_I2C_Write(0xA0);          // Address PCF8583
  Soft_I2C_Write(EEA);             // Start from word at address 0
(configuration word)
  Soft_I2C_Write(EED);          // Write 0x80 to config. (pause counter...)
  Soft_I2C_Stop();               // Issue stop signal
  delay_ms(10);
 }

void interrupt()
{
CCP1IF_bit =0;
cnt=tmr1h;
cnt=cnt<<8;
```

```
cnt=cnt | tmr1l;

tmr1h=0;
tmr1l=0;

}
void main() {

cmcon=7;
pcon.f3=1;
/*Lcd_Init();

Lcd_Cmd(_LCD_CLEAR);                    // Clear display
Lcd_Cmd(_LCD_CURSOR_OFF);               // Cursor off
Lcd_Out(1,1,"Frequency:");                     // Write text in first row*/

Soft_I2C_Init();

trisb=8;
portb=0;
trisa.f2=1;


ccp1con=0b00000101;

CCP1IE_bit=1;
PEIE_bit =1;
GIE_bit =1;

CCP1IF_bit =0;

tmr1l=0;
tmr1h=0;
t1con=0b00000001;
 delay_ms(1000);
while(1)
{
cnt1=10000000/cnt;
cnt2=cnt1;
cnt3=cnt2/10;
intToStr(cnt2,txt);

for(i=3;i<7;i++)
{
eeprom_write(i-3,txt[i]);
delay_ms(10);
}

if(porta.f2 == 1)
{
for(i=3;i<6;i++)
{
write_to_24c02(i-3,txt[i]);
//delay_ms(10);
}
}
txt1[0]=txt[3];  txt1[1]=txt[4];  txt1[2]='.';  txt1[3]=txt[5];
/*Lcd_Out(1,11,txt1);                     // Write text in first row
Lcd_Out(1,16,"Hz");                     // Write text in first row*/
portb.f0=~portb.f0;
}
}
```

4. MicroC Code Inside PIC16f628 to measure the phase difference

```
char txt[7],i;
unsigned int old,ct1,ct2,ct3,ct4,ct5;
unsigned int cnt,cnt1,cnt2,cnt3;
```

```
// Software I2C connections
sbit Soft_I2C_Scl           at Ra0_bit;
sbit Soft_I2C_Sda           at Ra1_bit;
sbit Soft_I2C_Scl_Direction at TRISa0_bit;
sbit Soft_I2C_Sda_Direction at TRISa1_bit;
// End Software I2C connections

// Lcd pinout settings
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB0_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D4 at RB4_bit;

// Pin direction
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB0_bit;
sbit LCD_D7_Direction at TRISB7_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB4_bit;

void write_to_24c02(char EEA,char EED)
 {
  Soft_I2C_Start();               // Issue start signal
  Soft_I2C_Write(0xA0);           // Address PCF8583
  Soft_I2C_Write(EEA);              // Start from word at address 0
(configuration word)
  Soft_I2C_Write(EED);           // Write 0x80 to config. (pause counter...)
  Soft_I2C_Stop();                // Issue stop signal
  delay_ms(10);
 }

void interrupt()
{
t1con=0b00000000;
CCP1IF_bit =0;
GIE_bit =0;
old=ccp1con;
ccp1con=0b00000000;

if (old ==0b00000100)
{
cnt=tmr1h;
cnt=cnt<<8;
cnt=cnt | tmr1l;

ccp1con=0b00000101;
}
else
{
ccp1con=0b00000100;
}

tmr1h=0;
tmr1l=0;

GIE_bit =1;
t1con=0b00000001;
}

void main() {

cmcon=7;
pcon.f3=1;
Soft_I2C_Init();
```

```
/*Lcd_Init();

Lcd_Cmd(_LCD_CLEAR);                // Clear display
Lcd_Cmd(_LCD_CURSOR_OFF);           // Cursor off
Lcd_Out(1,1,"Phase:");                 // Write text in first row*/

trisb=8;
portb=0;
trisa.f2=1;

ccp1con=0b00000101;

CCP1IE_bit=1;
PEIE_bit =1;
GIE_bit =1;

CCP1IF_bit =0;

tmr1l=0;
tmr1h=0;
t1con=0b00000001;

while(1)
{
ct5=ct4;
ct4=ct3;
ct3=ct2;
ct2=ct1;
ct1=cnt;

cnt1=(ct5+ct2)/2;
cnt2=(ct3+ct4)/2;
cnt3=(cnt1+ct1)/2;
cnt=(cnt3+cnt2)/2;

intToStr(cnt,txt);
//Lcd_Out(1,8,txt);

if(porta.f2 == 1)
{
for(i=1;i<6;i++)
{
write_to_24c02(i-1+20,txt[i]);
//delay_ms(10);
}
}
//delay_ms(1000);
for(i=1;i<6;i++)
{
eeprom_write(i-1,txt[i]);
}
portb.f0=~portb.f0;
}
}
```

5. MicroC Code Inside auxiliary PIC18f4685 to measure the phase difference

```
unsigned char Can_Init_Flags, Can_Send_Flags, Can_Rcv_Flags; // can flags
unsigned char Rx_Data_Len;                                   // received data
length in bytes
char RxTx_Data[8];                                        // can rx/tx data
buffer
char Msg_Rcvd;                                               // reception flag
const long ID_1st = 12111, ID_2nd = 3;                       // node IDs
long Rx_ID;

 char uart_rd,cnt,i,ii,flag,start_f=0,start_ff,sw_flag;
 char freq[8],phase[9],volt[8],tmp[5];
```

```
 char EEA,result,time_cc=0;
 int time_c=0;

// Lcd pinout settings
sbit LCD_RS at Rd0_bit;
sbit LCD_EN at Rd1_bit;
sbit LCD_D7 at Rd5_bit;
sbit LCD_D6 at Rd4_bit;
sbit LCD_D5 at Rd3_bit;
sbit LCD_D4 at Rd2_bit;

// Pin direction
sbit LCD_RS_Direction at TRISd0_bit;
sbit LCD_EN_Direction at TRISd1_bit;
sbit LCD_D7_Direction at TRISd5_bit;
sbit LCD_D6_Direction at TRISd4_bit;
sbit LCD_D5_Direction at TRISd3_bit;
sbit LCD_D4_Direction at TRISd2_bit;

///////////
void clr_wdt()
{
 asm clrwdt;
}
///////////////////
char read_from_24c02(char EEA)
 {
  I2C1_Start();              // Issue start signal
  I2C1_Wr(0xA0);          // Address PCF8530
  I2C1_Wr(EEA);             // Start from word at address 0
  I2C1_Repeated_Start();
  I2C1_Wr(0xA1);             // Write 0 to config word (enable counting)
  result = I2C1_Rd(0);
  I2C1_Stop();               // Issue stop signal
  clr_wdt();
  return result;
  }
///////////////////
void interrupt()
{clr_wdt();
 intcon.gie=0;
if (intcon.TMR0IF == 1){
   intcon.TMR0IF =0;

if (portc.f5 == 1)
{
 portc.f0=0;  portc.f1=0;  portc.f2=0;
}
else
{
switch (sw_flag) {
  case 0: portc.f0=1; delay_ms(100); portc.f0=0; sw_flag=1;  break;
  case 1: portc.f1=1; delay_ms(100); portc.f1=0; sw_flag=2; break;
  case 2: portc.f2=1; delay_ms(100); portc.f2=0; sw_flag=3;  break;
  case 3:   break;
}
}
time_c++;  if(time_c>3) {time_c=0; time_cc++;}
if(time_cc>2) time_cc=0;
}
tmr0l=0;
intcon.gie=1;
clr_wdt();
}

void main() {
rcon.ipen=0;
cmcon=7;
```

```
//pie1.RCIE =1;
t0con = 0b10000000;
intcon.TMR0IE =1;
intcon.peie=1;
intcon.gie=1;
clr_wdt();
trisc=0b00100000;
portc=0;

start_f=0;
clr_wdt();
Lcd_Init();
clr_wdt();
I2C1_Init(100000);      // initialize I2C communication
clr_wdt();
//wdtcon.SWDTEN=1;
//CAN BUS
Can_Init_Flags = 0;                                       //
  Can_Send_Flags = 0;                                       // clear flags
  Can_Rcv_Flags  = 0;                                       //

  Can_Send_Flags = _CAN_TX_PRIORITY_0 &                     // form value to
be used
                   _CAN_TX_XTD_FRAME &                      //     with
CANWrite
                   _CAN_TX_NO_RTR_FRAME;

  Can_Init_Flags = _CAN_CONFIG_SAMPLE_THRICE &              // form value to
be used
                   _CAN_CONFIG_PHSEG2_PRG_ON &              // with CANInit
                   _CAN_CONFIG_XTD_MSG &
                   _CAN_CONFIG_DBL_BUFFER_ON &
                   _CAN_CONFIG_VALID_XTD_MSG;

  CANInitialize(1,3,3,3,1,Can_Init_Flags);                  // Initialize CAN
module
  CANSetOperationMode(_CAN_MODE_CONFIG,0xFF);               // set
CONFIGURATION mode
  CANSetMask(_CAN_MASK_B1,-1,_CAN_CONFIG_XTD_MSG);          // set all mask1
bits to ones
  CANSetMask(_CAN_MASK_B2,-1,_CAN_CONFIG_XTD_MSG);          // set all mask2
bits to ones
  CANSetFilter(_CAN_FILTER_B2_F4,ID_2nd,_CAN_CONFIG_XTD_MSG);// set id of
filter B2_F4 to 2nd node ID

  CANSetOperationMode(_CAN_MODE_NORMAL,0xFF);               // set NORMAL mode
  // End CAN



 start_ff=0;
 for (;start_ff <5;)
{
Lcd_Cmd(_LCD_CLEAR);                // Clear display
Lcd_Cmd(_LCD_CURSOR_OFF);           // Cursor off
Lcd_Out(1,1,"Volt:");                 // Write text in first row
Lcd_Out(2,1,"Freq:");                 // Write text in first row
Lcd_Out(3,1,"Phase:");                 // Write text in first row
 start_ff++;
 delay_ms(10);
}

/*for(ii=0;ii<15;ii++)
  { RxTx_Data[ii]=0;}
CANWrite(ID_1st, RxTx_Data, 15, Can_Send_Flags);*/
 RxTx_Data[0]='a';

while(1)
```

```
{

//if (portc.f5 == 1) {flag =1;}
//if (portc.f5 == 0 && flag == 1)
if(sw_flag == 3)
{
sw_flag=0;
intcon.gie=0;
clr_wdt();
delay_ms(100);
freq[0]=read_from_24c02(0);
freq[1]=read_from_24c02(1);
freq[2]='.';
freq[3]=read_from_24c02(2);
freq[4]='H';
freq[5]='z';

if (isalnum(freq[0])) {} else {freq[0]='0';}
if (isalnum(freq[1])) {} else {freq[1]='0';}
if (isalnum(freq[3])) {} else {freq[3]='0';}

phase[0]=read_from_24c02(20);
phase[1]=read_from_24c02(21);
phase[2]='.';
phase[3]=read_from_24c02(22);
phase[4]=read_from_24c02(23);
phase[5]=read_from_24c02(24);
phase[6]='m';
phase[7]='s';
phase[8]=0;
 clr_wdt();
if (isalnum(phase[0])) {} else {phase[0]='0';}
if (isalnum(phase[1])) {} else {phase[1]='0';}
if (isalnum(phase[3])) {} else {phase[3]='0';}
if (isalnum(phase[4])) {} else {phase[4]='0';}
if (isalnum(phase[5])) {} else {phase[5]='0';}

volt[0]=read_from_24c02(10);
//volt[1]='.';
volt[1]=read_from_24c02(11);
volt[2]=read_from_24c02(12);
volt[3]='V';
clr_wdt();
if (isalnum(volt[0])) {} else {volt[0]='0';}
if (isalnum(volt[1])) {} else {volt[1]='0';}
if (isalnum(volt[2])) {} else {volt[2]='0';}

flag=0;


clr_wdt();
Lcd_Out(1,6,volt);                 // Write text in first row
clr_wdt();
Lcd_Out(2,6,freq);                 // Write text in first row
clr_wdt();
Lcd_Out(3,7,phase);                 // Write text in first row
clr_wdt();

clr_wdt();


//Msg_Rcvd = CANRead(&Rx_ID , RxTx_Data , &Rx_Data_Len, &Can_Rcv_Flags); //
receive message
//    if ((Rx_ID == ID_2nd) && Msg_Rcvd ) {
if(time_cc==0){
for (ii=0;ii<6;ii++)
{RxTx_Data[ii] = freq[ii];}
```

```
CANWrite(ID_1st, RxTx_Data, 6, Can_Send_Flags); }         // send initial
message

if(time_cc==1){
for (ii=0;ii<4;ii++)
{RxTx_Data[ii] = volt[ii];}
CANWrite(ID_1st, RxTx_Data, 4, Can_Send_Flags);}          // send initial
message

if(time_cc==2){
for (ii=0;ii<8;ii++)
{RxTx_Data[ii] = phase[ii];}
CANWrite(ID_1st, RxTx_Data, 8, Can_Send_Flags);}          // send initial
message

//         }
//RxTx_Data[0]++;
intcon.gie=1;
clr_wdt();
portc.f7=~portc.f7;
}
}
}
```

6. MicroC Code Inside main PIC18f4685 to measure the phase difference

```
unsigned char Can_Init_Flags, Can_Send_Flags, Can_Rcv_Flags; // can flags
unsigned char Rx_Data_Len;                                   // received data
length in bytes
char RxTx_Data[8];                                           // can rx/tx data
buffer
char Msg_Rcvd;                                               // reception flag
const long ID_1st = 12111, ID_2nd = 3;                       // node IDs
long Rx_ID;

// LCD module connections
sbit LCD_RS at Rc4_bit;
sbit LCD_EN at Rc5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// End LCD module connections


void main() {

  PORTC = 0;                                                // clear PORTC
  TRISC = 0;                                                // set PORTC as
output

  Can_Init_Flags = 0;                                       //
  Can_Send_Flags = 0;                                       // clear flags
  Can_Rcv_Flags  = 0;                                       //

  Can_Send_Flags = _CAN_TX_PRIORITY_0 &                     // form value to
be used
                   _CAN_TX_XTD_FRAME &                      //     with
CANWrite
                   _CAN_TX_NO_RTR_FRAME;
```

```
  Can_Init_Flags = _CAN_CONFIG_SAMPLE_THRICE &              // form value to
be used
                   _CAN_CONFIG_PHSEG2_PRG_ON &              // with CANInit
                   _CAN_CONFIG_XTD_MSG &
                   _CAN_CONFIG_DBL_BUFFER_ON &
                   _CAN_CONFIG_VALID_XTD_MSG &
                   _CAN_CONFIG_LINE_FILTER_OFF;

  CANInitialize(1,3,3,3,1,Can_Init_Flags);                  // initialize
external CAN module
  CANSetOperationMode(_CAN_MODE_CONFIG,0xFF);               // set
CONFIGURATION mode
  CANSetMask(_CAN_MASK_B1,-1,_CAN_CONFIG_XTD_MSG);          // set all mask1
bits to ones
  CANSetMask(_CAN_MASK_B2,-1,_CAN_CONFIG_XTD_MSG);          // set all mask2
bits to ones
  CANSetFilter(_CAN_FILTER_B2_F3,ID_1st,_CAN_CONFIG_XTD_MSG);// set id of
filter B2_F3 to 1st node ID

  CANSetOperationMode(_CAN_MODE_NORMAL,0xFF);               // set NORMAL mode

   Lcd_Init();                        // Initialize LCD

  Lcd_Cmd(_LCD_CLEAR);                // Clear display
  Lcd_Cmd(_LCD_CURSOR_OFF);           // Cursor off


  while (1) {                                                               //
endless loop
    Msg_Rcvd = CANRead(&Rx_ID , RxTx_Data , &Rx_Data_Len, &Can_Rcv_Flags); //
receive message
    if ((Rx_ID == ID_1st) && Msg_Rcvd) {                                   //
if message received check id
      for(ii=0;ii<Rx_Data_Len;ii++)
      {
       Lcd_Chr(1, ii+1, RxTx_Data[ii]);
      }
      CANWrite(ID_2nd, 'T', 1, Can_Send_Flags);                      // send
incremented data back
    }
  }
}
```