Design & Construction of 100+MHz Frequency Counter

Submitted in partial fulfilment of
The requirements for the award of the degree of

Master of Technology
In
Microwave and Optical Communication Engineering

Submitted by
Aishwarya
2K15/MOC/04

Under the guidance of
Dr. Priyanka Jain
Assistant Professor
Electronic and Communication Department

Department of Electronics and Communication
and
Applied Physics Department
Delhi Technological University
Delhi, India 110042

July 2017

DELHI TECHNOLOGICAL UNIVERSITY

Established by Govt. of Delhi vide Act 6 of 2009

(*Formerly Delhi College of Engineering)*

SHAHBAD DAULATPUR, BAWANA ROAD, DELHI-110042

CERTIFICATE

This is to certify that the work which is being presented in the dissertation entitled "**Design & Construction of 100+MHz Frequency Counter**" is the authentic work of **Aishwarya** under my guidance and supervision in the partial fulfilment of requirement towards the degree of Master of Technology in Microwave and Optical Communication Engineering jointly run by Department of Applied Physics and Department of Electronics and Communication in Delhi Technological University during the 2015-17.

As per the candidate declaration this work has not been submitted elsewhere for the award of any other degree.

Dr. Priyanka Jain

Supervisor

Assistant Professor

Delhi Technological University

Declaration

I hereby declare that all the information in this document has been obtained and presented in accordance with academic rules and ethical conduct. This report is my own, unaided work. I have fully cited and referenced all material and results that are not original to this work. It is being submitted for the degree of Master of Technology in Engineering at the Delhi Technological University. It has not been submitted before for any degree or examination in any other university.

Abstract

A frequency counter is an electronic instrument used to measure the frequency of an input signal, it operates on the principle of the input signal into the counter which assembles the number of event occurring within a specific period of time. The accuracy of the measurement is directly related to the internal resolution of the counter and the accuracy of its internal time base

The frequency counter in the project is based on direct counting approach. The power consumption of this counter is very low, it can operate with 12V battery/adapter. The circuit contains only a few components. The brain of this unit is a PIC 16F628A microcontroller which has an instruction clock of up to 4MHz, which is also the input sampling rate. It has a 16x2 LCD i.e. it can display 32 characters. This counter is capable of measuring frequency up to 1.8 GHz in normal condition with a maximum amplitude of15V. This project also includes the construction of wave form generator which is AD 9850 (DDS) chip and able to generate signal of up to 50 MHz

Acknowledgements

I would like to express my sincere gratitude to my project supervisor, Asst. Professor Dr. Priyanka Jain, for her supervision, invaluable guidance, motivation and support throughout the extent of the project. I have benefitted immensely from her wealth of knowledge.

My gratitude is extended to my colleagues and friends who have not been mentioned here personally in making this project a success.

Last but not least, I want to thank my parents, for inculcating good ethos, as a result of which I am able to do my post-graduation from such an esteemed institution. I would thank my friends for believing in my abilities and for always showering their invaluable support and constant encouragement.

Aishwarya
M. Tech, MOCE
2K15/MOC/04

Table of Contents

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

Chapter 1

Introduction

1.1. Introduction

Frequency counters are the electronic components which are widely used for the accurate measurement of radio frequency (RF) signals, or measuring the oscillation or pulses of repetitive electronic signals and measuring the elapsed time between events.

A frequency counter works in two ways: direct counting and reciprocal counting. Most frequency counters work on direct counting approach in which it counts the number of times the input signal crosses a given trigger voltage with a given direction in a given period, which is known as 'gate time.' After the gate time, the value of the counter is displayed, and the counter is set back to zero.

Another approach is 'Reciprocal approach' which is less common and less widely used. In this method, the period for one cycle of the waveform is measured, and then its reciprocal is taken. This method gives a higher resolution and can make very fast readings. It can give 1MHz resolution in 1ms where a direct counter gives 1Hz resolution in 1s. But this method is more expensive so less widely used.

Accuracy and resolution are two important aspects of the frequency counters. The internal oscillation in the frequency counter which provides the time signals is called the timebase. The accuracy of a frequency counter is highly dependent on its timebase. The timebase is very delicate and can be easily changed by interface or age. This can lead to higher or lower readings as compared to the actual ones. For this, timebases are generated by highly accurate circuits, e.g. a quartz crystal oscillator, within a sealed temperature controlled chamber (known as an oven controlled oscillator). For the accurate readings, timebase should always be well calibrated.
Resolution of a single count is proportional to the timebase oscillator frequency and the gate time. The resolution of a frequency counter is its ability to differentiate between two signals that are close to each other. It is a measure of the number of digits in the reading of the signal frequency. Resolution can be improved by techniques like oversampling, averaging, etc.

The frequency counter in the project is based on direct counting approach. The power consumption of this counter is very low, it can operate with 12V battery/adapter. The circuit contains only a few components. The brain of this unit is a PIC 16F628A microcontroller which has an instruction clock of up to 4MHz, which is also the input sampling rate. It has a 16x2 LCD i.e. it can display 32 characters. This counter is capable of measuring frequency up to 1.8 GHz in normal condition with a maximum amplitude of15V. This project also includes the construction of wave form generator which is AD 9850 (DDS) chip and able to generate signal of up to 50 MHz

Chapter 2

Method and Theory

This chapter discusses the project method and theory on the subject of frequency counting and AD9850 waveform generator

2.1 METHOD

As discussed in the previous chapter frequency counters are used for measuring the oscillation or pulses of repetitive electronic signals. Although the frequency counters and interval timers are different from each other, it is possible to use the same basic test instrument work like both. So, frequency counter-timers are widely used. As the input signal is a sinusoidal wave, the signalmust beconverted tothe squarewaveform with thesame frequencyand sameamplitudebeforemeasuring thefrequency of the input signal. It can be converted using optical isolatorcircuitry with *4N25 optocoupler*, a Schmitttrigger or a direct square wave generator AD 9850 module.

Here direct wave generator is used which uses a125MHz crystal oscillator to generate square wave up to 50MHz.

The method used here is direct counting of frequency which is discussed in next section.

2.2 Direct frequency counting:

Frequency is defined as the rate of occurrence of in a given specified time. The "occurrence" in this case is the repetition of a period of a signal. This can be writtenas.

$$\mathbf{f = n/t} \quad \text{..} 2.1$$

Where,

'f' is the frequency

'n'is a number of periods that occurred during the interval 't.' This time interval is also known as gate time. After the gate time, the value of the counter is displayed, and the counter is set back to zero.

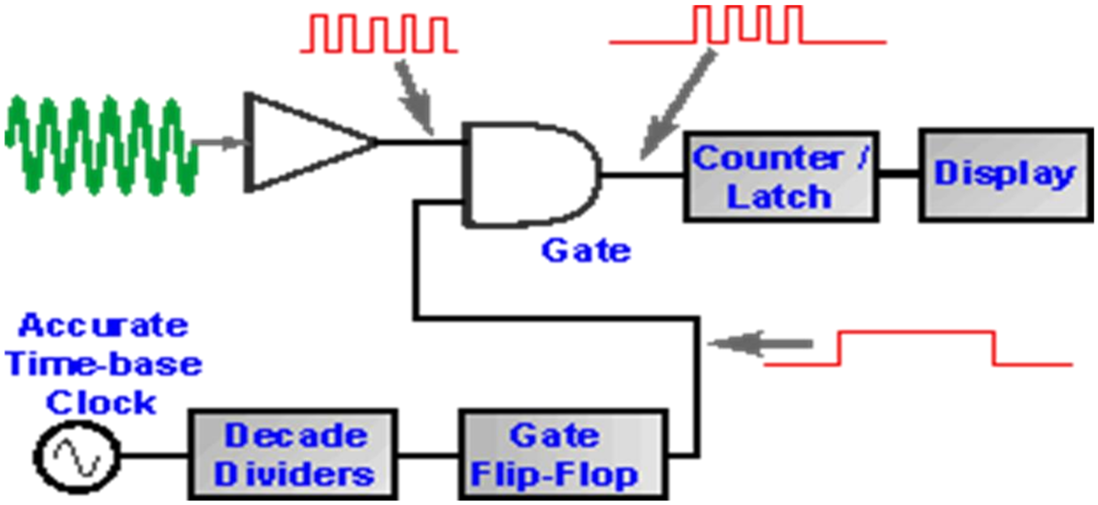


Figure 2.1 : Basic frequency counter block diagram

Figure 2.1 shows the primaryprocess of the counter. The input wave is shaped into the square wave before passing through the gate to counter have internal clock instruction and output will display on LCD.

Input: Inputsignal enters the input amplifier, in which the signal is transformed into a logically square wave for processing. Trigger levels and sensitivity are controlled within this area of the frequency counter.

Accurate ime-base/clock : To generate the various gates/timing signals in the frequency counter, an accuratetimebase or clock is needed.Typically a crystal as an oven controlled crystal oscillator is used.

Decade dividers and flip flop: The clock oscillator is used to provide an accurately timed gate signal that will allow through pulses from the incoming signal. Thisis generated from the clock by dividing the clock signal by decade dividers and then feeding this into a flip flop to give the enabling pulse for the main.

Gate:Thetimebaseenabling signal from the clock is applied to one input of a gate, and the other has a train of pulsesof the incoming signal. The resultant output of the gate is a series of pulses for the timebase. For example, if the incoming signal was at 1 MHz and the gate was opened for 1 second, then 1 million pulses would be allowed.

Counter/ latch:The counter takes the incoming pulses from the gate. It has a set of divide-by-10 stages (a number equal to the number of display digits minus 1). Each stage divides by ten and therefore as they are chained the first stage is the input divided by ten, the next is the input divided by 10 x 10, and so forth. These counter outputs are then used to drive the display.

To hold the output in place while the figures are being displayed, the output is latched. The latch will hold the last result while the counter is counting a new reading. In this way, the display will remain static until a new result can be displayed. The latch will be updated and the new reading presented to the display.

Display: The display takes the output from the latch and displays it in a readable format. LCD or LED displays are the most common.

2.3 Prescaling

While dealing with Radio frequencies, special measures have to be taken. The majority of standard components are not fast enough to keep up with such high frequencies. To manage these frequencies, these should be lowered with the help of pre-scalars. Prescalars also was known as a frequency divider, divides the incoming frequency N times, where N can be a fixed integer or a programmable value. Prescaling is further described later.

2.4 Characteristic Impedance

When designing p rinted circuit board (PCB) that will include board traces used to carry signals within the RF spectrum, special considerations have to be taken. Every PCB trace has a characteristic impedance. Thiscan be described by illustrating a section of a transmission line to realize the components of a typical PCB traces or any transmission line. Four components can describe a small length of a transmission line

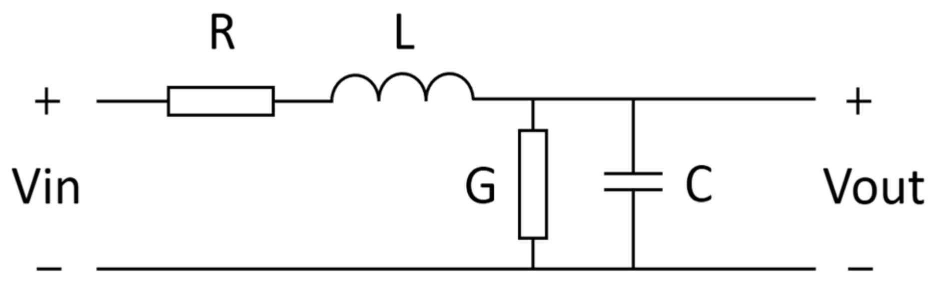


Figure 2.2: Illustration of a transmission line.

Where 'R' can be described as the series resistance due to the non-ideal conductive characteristics of the trace material.'L' represents the current and magnetic energy in the line segment. 'G' represents nonideal insulator properties of the isolation material between the copper layers of the PCB. Where as'C'represents the electric energy stored in the transmission line.The formula of characteristic impedance can be described as equation 2.2,

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \qquad \text{………………..2.2}$$

Characteristic impedance can be expressed as the impedance seen for a signal at every point along the transmission line. There is a possibility for the PCB designer to alter the characteristic impedance by changing the layout of the trace. By changing the material of the insulator, the permittivity can be altered and thus make the impedance change. That can also becontrolled by changing the thickness and width of the copper trace. The reason that the characteristic impedance of a trace is important is that, when a signal hits a discontinuity in the medium in which it propagates, part of the signal is reflected off leading to a wave with less power that continues along the transmission line. The ratio of the voltage of the wave that continues forward and the voltage of the wave that is reflected is called the reflection coefficient and can be written as in equation 2.3.

$$\Gamma_L = \frac{V_L^-}{V_L^+} = \frac{Z_L - Z_0}{Z_L + Z_0} \qquad \text{………………..2.3}$$

Where $V^-_L$is the voltage of the reflected wave, V^+_L is the incident voltage

Z_L is the impedance of the second propagation medium

Chapter 3

Operation and Components

3.1 Frequency Counter Timer and operation on PIC 16F628A

The whole counter design is based on the PIC micro-controller PIC 16F628A. The two most important components in this project are internal counters/timers called TMR0 and TMR1. The counter is used as a pre-scaler to the PIC's counter TMR, it divides the frequency by 256, so the PIC gets frequency more than 100MHz, which is in the range of minimum input clock pulse 0.0189s i.e. timer will run for 2684 cycles = 0.08191 seconds.

Basically, Timer0 (TMR0) is one of the 3timers available on the 16F628A, that is in running state all the times. It is an 8-bit wide register, and its valueiscontrolled byaclock source.The clock source for TMR0 can either be an external strobe signal fed in on pin RA4/T0CK(T0CKI= Timer0 ClocK Input)or an internal instruction cycleclock. We are using internal clock of 4MHz.

When operating with the instruction clock cycle, it works as a timer. The value residing in the TMR0 register starts at 0x00, then it is incremented by a factor of 1 upon every low to high transition of the clock source (can be configured to increment on the high-low or the low-high transition if using an external clock source). Basically, it's just a register that counts pulses from one of the two possible clock pulse sources, and it can count as high as 0xFF (decimal 255).

The instruction clock runs at 1/4th the F_{OSC} frequency. This means that if F_{OSC} = 4MHz, instruction clock runs at 1MHz or one instruction per microsecond. In this scenario, assuming the pre-scalar is not assigned to TMR0, TMR0's value increment on every pulse of the instruction clock. At 1MHz instruction clock frequency, this increments the value in register TMR0 once every microsecond (1/µS).

Because the TMR0 register is only 8-bits wide, the maximum value to which TMR0 can be incremented is 0xFF (decimal 255). Once TMR0 is increment to 0xFF, it will roll back to 0 upon the next clock pulse. Once this rollback occurs, the Timer 0 interrupt flag in the INTCON register gets SET (T0IF). This flag must be cleared in software.

Since TMR0 is currently running at 1 increment perµS, this means that this rollback occurs and the T0IF flag gets SET once every 256µS (0x00 - 0xFF or 0-255 for a total of 256 increments). If we want to know when a period of 256µS has passed, we can clear TMR0, clear the T0IF flag, then poll interrupts flag T0IF until T0IF goes high (rollback occurred), then continue on with our program. A timer would not be used without some way to slow it down to different increment rates. This is the job of the "prescaler." The prescaler prescales the timer so that instead of it being incremented on every clock pulse, we can have it incremented every 2 clock pulses, every 4, every 8, etc. all the way up to every 256 clock pulses. If we were to set the prescaler to0:256 with a 1MHz instruction clock, this means the timer would be incremented once every 256µS.

3.2 PRESCALE

The basic functionality of a Pre-scalar is to divide the frequency of the incoming signal N times where N can be a fixed value or a programmable value. A primitive realization of a pre-scaler can be seen in **figure 3.1**.

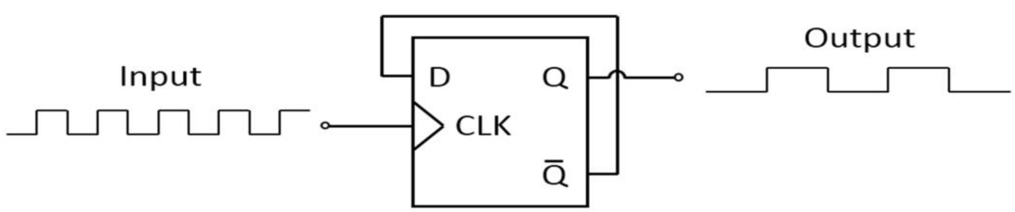
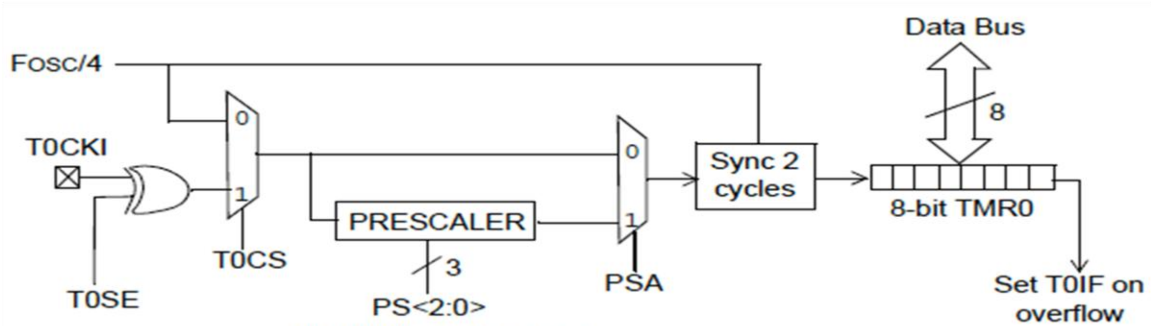


Figure3.1 Illustration of a Frequency Divider

In figure 3.1,a common latch has been used to present the functionality of the circuit. The input signal is used to clock the latch which is connected with feedback from the inverted output to the input. This causes the latch to toggle the output every other pulse, and a divide-by-2 pre-scaler is implemented.

3.3 Timer and Counter Function on Pre-scale :



BIT VALUE	TMR0 RATE
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

Figure 3.2: Timer and Counter Function On Pre-scale

Timer 0 is an 8-bit synchronous counter that stores the value of the counter in specific function driven by instruction clock. It is incremented at a constant rate. F_{OSC} of 4MHz is used to synchronize the value of input signal after pre-scaling. Timer clock along with timer oscillation frequency passes through NOR gate to pre-scaling assembly (PSA), where TOIF i.e. TIMER 0 sets the interrupt by TOIE with GIE bit along with INTCON.

Pre-scalars are selected through PS0, PS1, and PS2 of OPTION register and must be declared with TMR0. If it is SET, no pre-scaler value will be assigned to TMR0.

Counter mode- It is selected by TOCS bit in the OPTION register. TMR0 counts the external clock pulse at RA2/TOC1K1/PIN. If TOSE is SET, the timer will increase on falling edge of the clock pulse, and it can be extended by the use of pre-scale. The maximum clock frequency of TOCK1 after PSA is synchronized with the internal clock by sampling pre-scale O/Pin Q2 & Q4.

TABLE 3.1 **OPTION_REG – OPTION REGISTER (ADDRESS: 81h OR 181h)**

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RAPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RAPU}}$:** PORTA Pull-up Enable bit
1 = PORTA pull-ups are disabled
0 = PORTA pull-ups are enabled by individual port latch values in WPUA register

bit 6 **INTEDG:** Interrupt Edge Select bit
1 = Interrupt on rising edge of RA2/INT pin
0 = Interrupt on falling edge of RA2/INT pin

bit 5 **T0CS:** TMR0 Clock Source Select bit
1 = Transition on RA2/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE:** TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA2/T0CKI pin
0 = Increment on low-to-high transition on RA2/T0CKI pin

bit 3 **PSA:** Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS<2:0>:** Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate[1]
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Note 1: A dedicated 16-bit WDT postscaler is available for the PIC16F688. See **Section 11.7 "Watchdog Timer (WDT)"** for more information.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

3.4 COMPONENTS

This frequency counter constructed using PIC 16F628Auses a 4MHz oscillator for the CPU clock which is also the input sampling rate. An overview of a microcontroller with LCD has shown below.

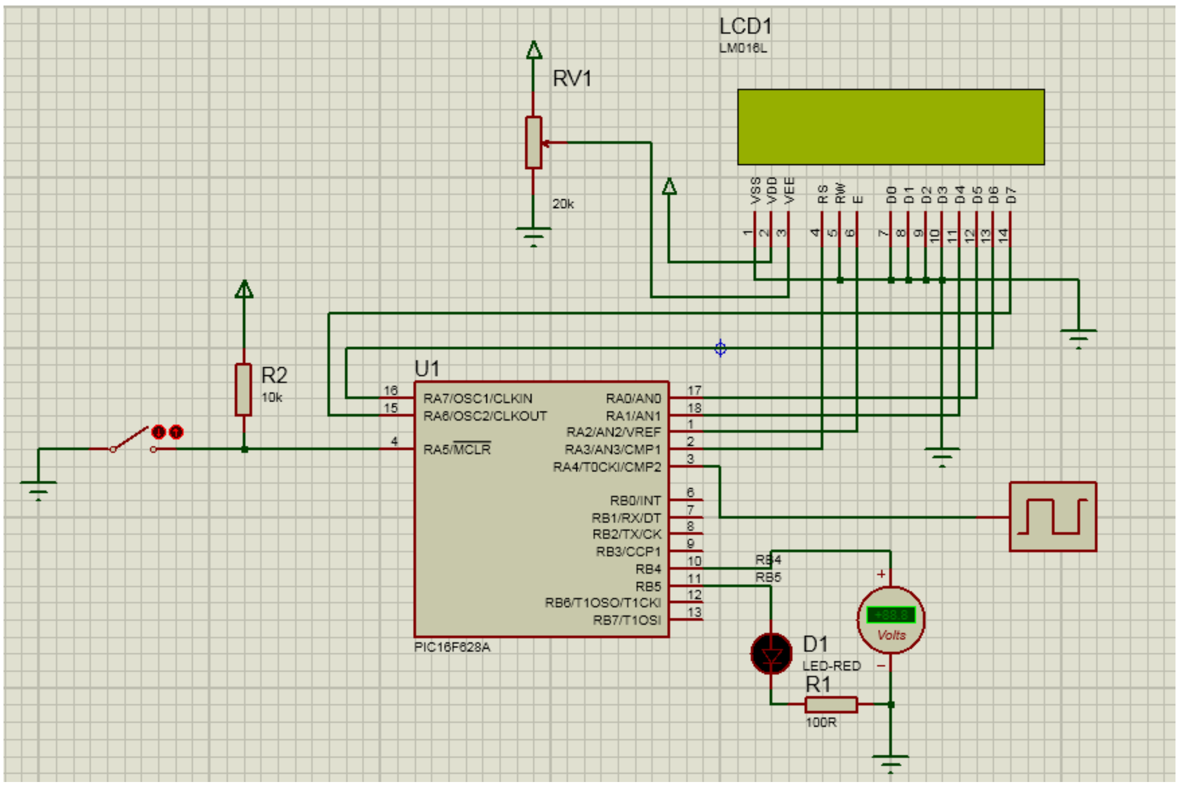


Figure 3.3: Block Diagram of PIC 16F628A Based Frequency Counter

The circuit is capable of countingup andcounting downand alsocapable of programming with an initial count value. The input signal is connected at Pin 3 RA4/TOC1/CMP2, where asRB4is connected with the output voltage of 5V.A LED is connected to RB5 along with a 100R resistor. The required voltage for PIC and LCD is 2V and 5V respectively. Pin no 5,7,8,9 and 10 are grounded, and data bus pins are connected to the microcontroller. The 20k potentiometer is connected to LCD to adjust the brightness and backlight of LCD. The LCD is connected to pin 1 of port A for controlling the process as hardware. We could use external crystal oscillator at RA3 and RA4 for generating time interval of afrequency counter, but here an internal clock oscillator of 4MHz is used.

3.4.1 Microcontroller

The microcontroller used in this projectis the PIC 16F628A CMOS FLASH based 8-bit microcontroller. Summary of its specification is as follows:

- Clock speed 4MHz.
- 128 bytes of EEPROM data memory.
- Flash memory 2028
- Low power feature -- 100 nA @ 2.0V, typical

  Operating current --- 12 μA @ 32 kHz, 2.0V,

  -- 120 μA @ 1 MHz, 2.0V, typical

  Watchdog TimerCurrent : - 11 μA @ .2.0V,typical

  Timer1 Oscillator Current: - 11.2μA@.32 kHz,,2.0V,typical

  Dual-speed Internal Oscillator: --Runtime selectable between 4 MHz and 48 kHz
- Programmable V_{REF}
- Low voltage programming and low clock speed.

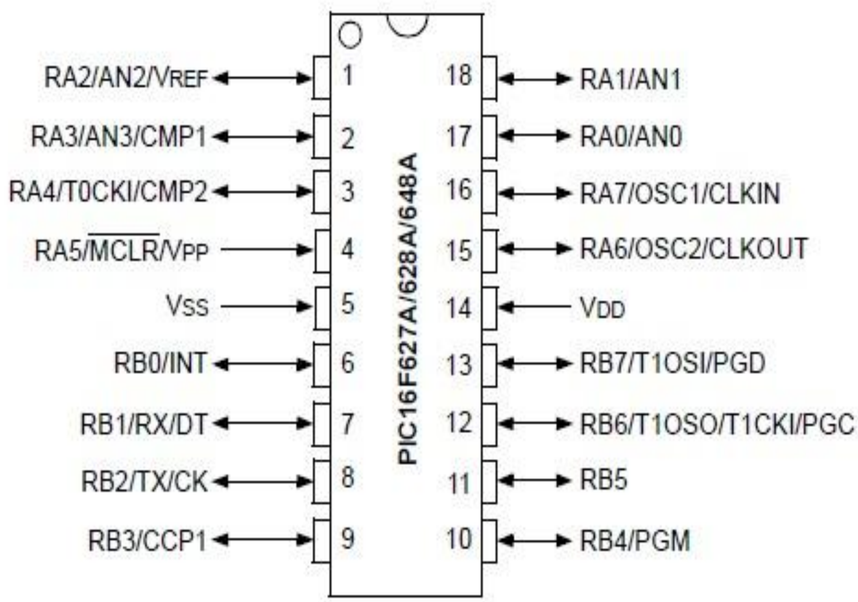


Figure 3.4 Pin Diagram of PIC16F628A

Pin Description

Pin Number	Description
1	RA2/AN2/Vref - Port A
2	RA3/AN3/CMP1 - Port A
3	RA4/TOCK1/CMP2 - Port A
4	RA5/MCLR/THV - Port A
5	Vss - Ground
6	RB0/INT - Port B
7	RB1/RX/DT - Port B
8	RB2/TX/CK - Port B
9	RB3/CCP1 - Port B
10	RB4/PGM - Port B
11	RB5 - Port B
12	RB6/T1OSO/T1CK1 - Port B
13	RB7/T1OS1 - Port B
14	Vdd - Positive Power Supply
15	RA6/OSC2/CLKOUT - Port A
16	RA7/OSC1/CLKIN - Port A
17	RA0/AN0 - Port A
18	RA1/AN1 - Port A

Table 3.2: Pin Description of PIC16F628A

3.4.2

16x 2 Character LCD display

Description

LCD(Liquid Crystal Display) screen is an electronic display module and standard interface application.The16x2 LCD display is very basic and most commonly used in various devices and circuits.

In a 16x2 LCDwe can display 32 character in 2 rows. In this LCD each character is displayed in 5x7 pixel matrix.

The LCD has two registers known as Command register and Data register.

Command register is basically used to insert a particular command into the LCD. Data register is used to insert data into the LCD. Generally, command is a specific set of instructionswhich is used to give the internal command to LCD like clear screen, move to line character, setting up the cursor, etc[22]

Pin Description

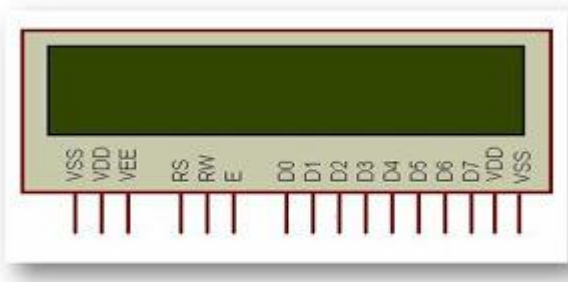


Figure 3.5LCD Diagram

Sr. No	Pin No.	Pin Description
1	Pin 1 (GND)	This is a ground pin to apply a ground to LCD.
2	Pin 2 (VCC)	This is the supply voltage pin to apply voltage to LCD.
3	Pin 3 (VEE)	This is the pin for adjusting a contrast of the LCD display by attaching a veriable resistor in between VCC and GND.
4	Pin 4 (RS)	RS stands for Register Select. This pin is used to select command/data register. If RS=0 then command register is selected. If RS=1 then data register is selected.
5	Pin 5 (R/W)	R/W stands for Read/Write. This pin is used to select the operation Read/Write. If R/W=0 then Write operation is performed. If R/W=1 then Read operation is performed.
6	Pin 6 (EN)	En stand for Enable signal. A positive going pulse on this pin will perform a read/write function to the LCD.
7	Pin 7-14 (DB0-DB7)	This 8 pin is used as a Data pin of LCD.
8	Pin 15 (LED+)	This pin is used with pin 16(LED-) to setting up the illumination of back light of LCD. This pin is connected with VCC.
9	Pin 16 (LEC-)	This pin is used with pin 15(LED+) to setting up the illumination of back light of LCD. This pin is connected with GND.

Table 3.3: Important Commands of LCD

3.4.3

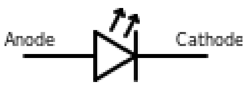
Light-emitting-Diode

A light-emitting diode (LED) is a semiconductor light source. LĔDs are widely used as indicator lamps in most devices and are increasingly used for standard lighting. [23]

Figure 3.6 LED Bulb

LEDs work on the principle of electroluminescence. These are two types one is passive and other is optoelectronics.

Electronic symbol



3.4.4

IC LM 7805

Every project requires a power supply voltage regulator circuit which is often an IC-7805.The counter needs 9-12V highly stabilized power supply. As an input signal is always needed through input jack, a DC input jack is connected to the DC Adapter of 12V. This will be connected to an ON/OFF switch. The primary function of ON/OFF switch is to either ON or OFF the main supply. A diode is placed between the input and the first pin of a voltage regulator which protects the circuit from the reverse polarity.

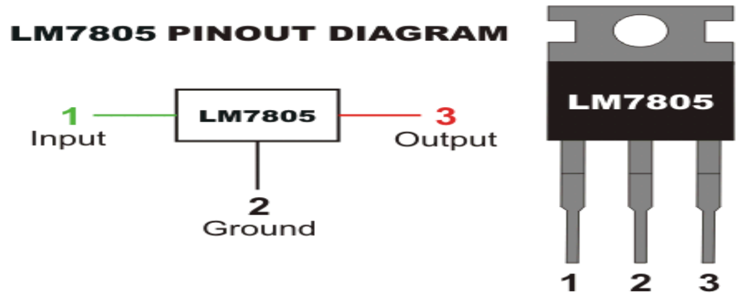


Figure 3.7: Circuit diagram of voltage regulator and Pin Diagram of IC 7805

The primary function of IC7805 is to convert an input voltage to 5V DC. Two capacitors of 100mf and 10mf are also placed which basically filter out any noise component or AC component which may be present in the supply. Usually, the power indicator LED bulb is set along with the resistor of 220Ω to limit the current through LED. The Same circuit has been used in frequency counter as well as in waveform generator [24]

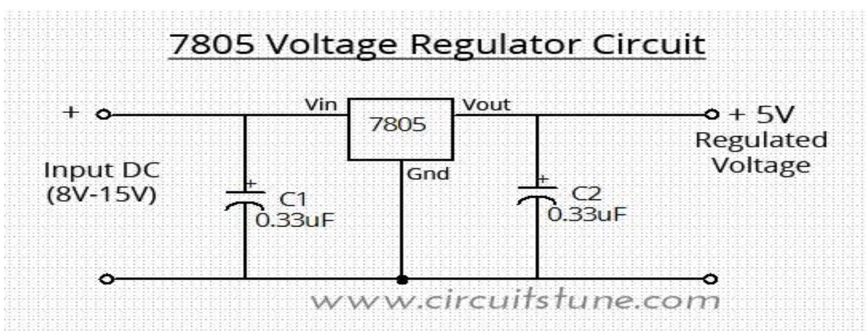


Figure 3.8 Voltage Regulator Circuit

In basic network circuit of IC 7805, the input DC voltage of 8-15V is connected to Vin of IC 7805 with 2 ceramic capacitors of 0.33µf each. The output regulated voltage is 5V, which is necessary for the circuit.

3.4.5ATmega-328

ATmega-328 is very popular microcontroller by Atmel in mega AVR family. It has 32 KB flash memory with read/write capability, 1KB EEPROM memory and 2k of internal

SRAM. It is widely used with the Arduino Uno modules. Arduino Uno comes with either ATmega16 or ATmega-328. It is the most advanced chip used for the generation of the waveform. It has 28 pins out of which 6pins are used with PWM output, 6are analog output pins and 14 pins are used as digital I/O pins [16]

The basic pin configuration of AT-mega 328is shown below;

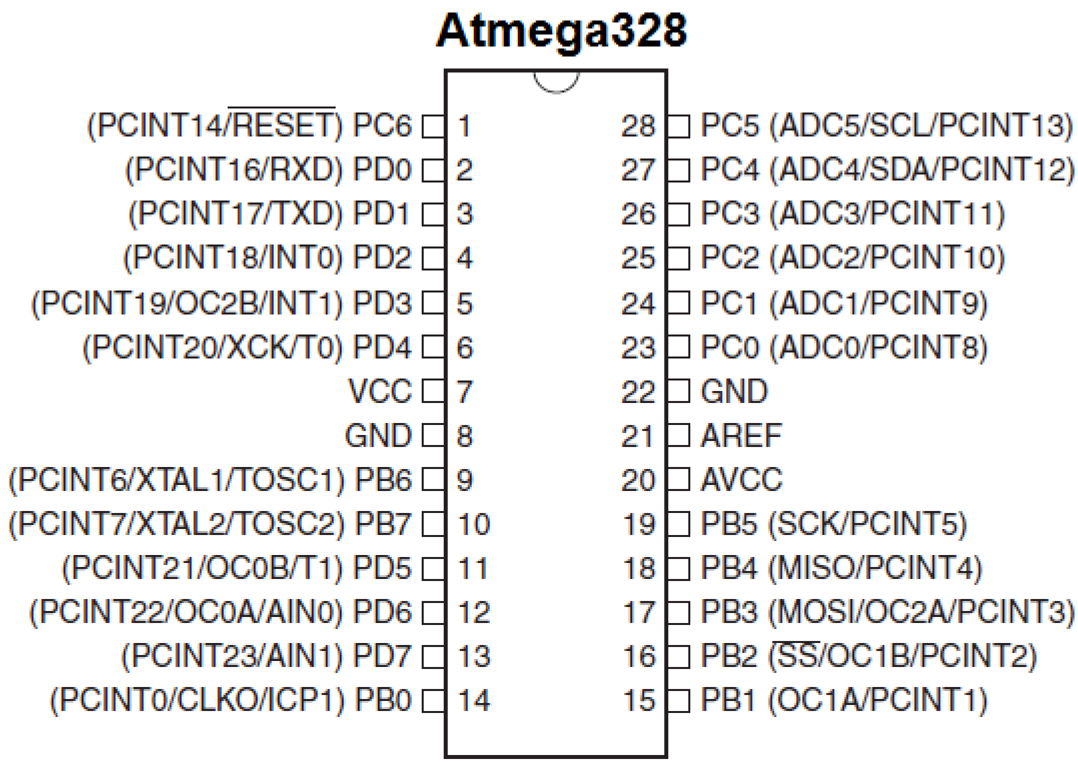


Figure 3.9 Pin diagram of ATmega-328

Remaining two pins are used as crystal oscillator which provides clock pulse for AT mega chip. A clock pulse is needed to synchronize the signal so that communication can occur in the same way. This chip needs power, so *Vcc* and GND are used.

Pin Number	Description	Function
1	PC6	Reset
2	PD0	Digital Pin (RX)
3	PD1	Digital Pin (TX)
4	PD2	Digital Pin
5	PD3	Digital Pin (PWM)
6	PD4	Digital Pin
7	Vcc	Positive Voltage (Power)
8	GND	Ground
9	XTAL 1	Crystal Oscillator
10	XTAL 2	Crystal Oscillator
11	PD5	Digital Pin (PWM)
12	PD6	Digital Pin (PWM)
13	PD7	Digital Pin
14	PB0	Digital Pin
15	PB1	Digital Pin (PWM)
16	PB2	Digital Pin (PWM)
17	PB3	Digital Pin (PWM)
18	PB4	Digital Pin
19	PB5	Digital Pin
20	AVCC	Positive voltage for ADC (power)
21	AREF	Reference Voltage
22	GND	Ground
23	PC0	Analog Input
24	PC1	Analog Input

Table 3.4: pin description of ATmega-328

This chip uses ADC (analog to digital converter) to interpret analog input. It has16 pins for the analog input signal, and three pins are used for supply ground voltage and $V_{REF.}$ The power is always positive for ADC,it converts an analog signal to digital signal. Those voltages above the V_{REF} will be marked as the digital value 1 while those voltages that are below the V_{REF} will be marked as 0. [13]

3.4.6
Rotary Encoder

Rotary Encoder is used to determine the angular position of the shaft. It generates an analog or a digital signal according to the movement of the shaft. It is also known as Quadrature encoder or relative rotator encoders which basically generates output pulses of the square wave. The encoders have 2 out pins A or B and a common pin C. When the ring will rotate, point A and B will touch the common-C and will generate the 2 square wave signal accordingly.

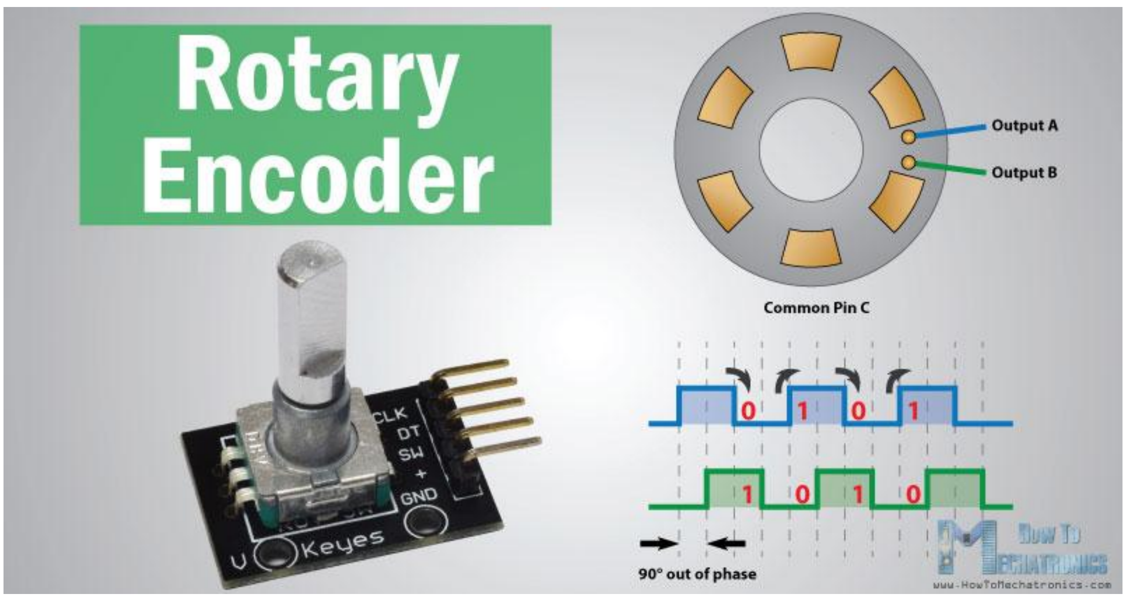


Figure 3.10 Descriptionof Rotary Encoder

The 2 clock pulses are 90^{o} out of phase with each other i.e. when the encoder rotates clockwise the output pulse A will be ahead of pulse B and vice versa. So, the step of each signal results in the two O/P signals which have totally opposite values. In the case of counterclockwise, the o/p signals have equal values. By programming, we can read the encoder position and rotation direction.[27]

Chapter 4

Project Overview

4.1 Specifications

The goals were set high to test the limits of what can be produced in this short time frame using low-cost components and with AD9850 waveform generator. A pure wave can be used as a frequency source [14][15]. AD9850 waveform generator allows us for thegeneration of output frequencies of up to half of the reference clock frequency.

Frequency rating **-** The resulting frequency counter shall be able to detect and calculate frequencies up to 1.8GHz.

Accuracy **-** The calculated frequency shall not differ more than $\pm$ 250 Hz from the input frequency over the whole bandwidth

Power consumption **-** The resulting frequency counter should not draw currents exceeding 150 MW at 15 V. Whole circuit can be connected to the computer with a high power USB cable.

Cost **-** The cost of the frequency counter along with Chip AD9850 (DDS) waveform generator should be profitably lower that is about 9000INR.

4.2 AD9850 (DDS) WAVEFORM GENERATOR

A signal generator is an electronic device that generates electronic signals and waveforms. These electronic signals are either repeating or non-repeating as per the requirements and field of application. A signal generator can generate various kinds of waveforms. Most common are the sine waves, square waves, sawtooth waves, and triangular waves.

Here a 1-50MHz signal generator is used. It uses AD9850 (DDS) synthesizers which can generate waves of higher stability. The high-speed core technology of AD9850 (DDS) provides a 32-bit frequency which results in an output resolution of 0.0191Hz for 125MHz. It can be programmed using Arduino by connecting the RESET pin. Generated waves of higher amplitude are aliased that occur at multiples of input clock frequencies and are not very smooth. 70MHz low pass filter is placed between the o/p of DAC and input of the comparator to suppress the effect of aliased wave further.

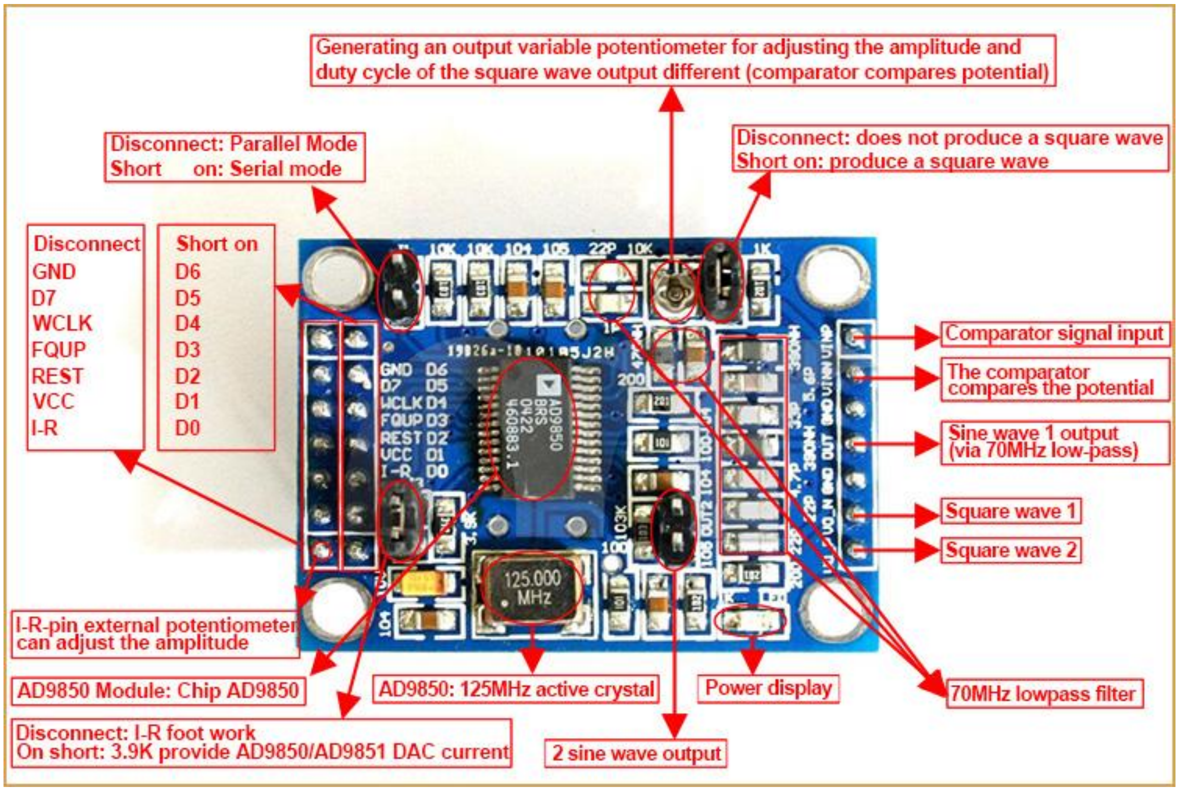


Figure 4.1 AD9850 chip Description

Setting the output frequency

$$f_{OUT} = \frac{\Delta Phase \times REFCLOCK}{2^{32}}$$

F_{out} is the frequency of output signal, CLKIN is the input reference clock in MHz

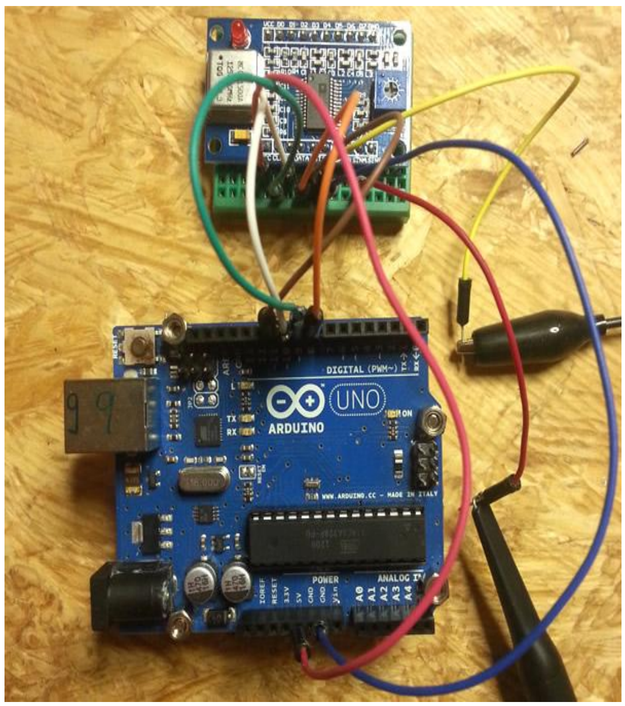
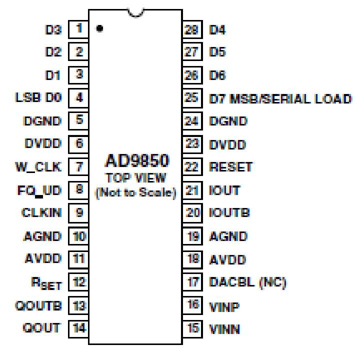


Figure 4.2 AD 9850 (DDS) Chip With Arduino[17]

This signal generator is composed of ATmega-328 microcontroller, which controls the whole circuit. It is programmed using Arduino UNO. Pins 8,9,10 and 11 are connected to the DDS CLOCK, DATA, LOAD AND RESET lines respectively. Signal output can be taken from pin 21 of the DDS chip.

AD9850

PIN CONFIGURATION



NC = NO CONNECT

Table I. PIN FUNCTION DESCRIPTIONS

Pin No.	Mnemonic	Function
4 to 1, 28 to 25	D0 to D7	8-Bit Data Input. This is the 8-bit data port for iteratively loading the 32-bit frequency and the 8-bit phase/control word. D7 = MSB; D0 = LSB. D7 (Pin 25) also serves as the input pin for the 40-bit serial data-word.
5, 24	DGND	Digital Ground. These are the ground return leads for the digital circuitry.
6, 23	DVDD	Supply Voltage Leads for Digital Circuitry.
7	W_CLK	Word Load Clock. This clock is used to load the parallel or serial frequency/phase/control words.
8	FQ_UD	Frequency Update. On the rising edge of this clock, the DDS updates to the frequency (or phase) loaded in the data input register; it then resets the pointer to Word 0.
9	CLKIN	Reference Clock Input. This may be a continuous CMOS-level pulse train or sine input biased at 1/2 V supply. The rising edge of this clock initiates operation.
10, 19	AGND	Analog Ground. These leads are the ground return for the analog circuitry (DAC and comparator).
11, 18	AVDD	Supply Voltage for the Analog Circuitry (DAC and Comparator).
12	R_{SET}	DAC's External R_{SET} Connection. This resistor value sets the DAC full-scale output current. For normal applications (F_S I_{OUT} = 10 mA), the value for R_{SET} is 3.9 kΩ connected to ground. The R_{SET}/I_{OUT} relationship is $I_{OUT} = 32\ (1.248\ V/R_{SET})$.
13	QOUTB	Output Complement. This is the comparator's complement output.
14	QOUT	Output True. This is the comparator's true output.
15	VINN	Inverting Voltage Input. This is the comparator's negative input.
16	VINP	Noninverting Voltage Input. This is the comparator's positive input.
17	DACBL (NC)	DAC Baseline. This is the DAC baseline voltage reference; this lead is internally bypassed and should normally be considered a no connect for optimum performance.
20	IOUTB	Complementary Analog Output of the DAC.
21	IOUT	Analog Current Output of the DAC.
22	RESET	Reset. This is the master reset function; when set high, it clears all registers (except the input register), and the DAC output goes to cosine 0 after additional clock cycles—see Figure 7.

Table 4.1 Pin Configuration of AD9850

Chapter 5

Simulation and Results

This section will present the design and all of its features. The different sections throughout this chapter contain the differentparts of the PCB design along with what was considered valuable input about that section. Since the implementation and design of the hardware has been one of the biggest parts of this thesis work, it was seen appropriate to dedicate a chapter to present

5.1 Circuit Description

While designing the circuit, proper care should be taken in selecting, handling and soldering the components. It is betterto assemble the circuit on a breadboard before assembling on a printed circuit board. First, the IC base and connectors are assembled followed byall resistors andcapacitor. The 20k potentiometer is used along with220k resistor to preventwarmth dissipation.The pins of PIC16F628A microcontroller, the power supply pin, and input-output port are accessed by male headers. It can beeasily plugged into a breadboard and useful for fast prototyping. The PIC16F628A allow us to study/write 8-bit facts at once through PORTB, which is 18-bitwide.

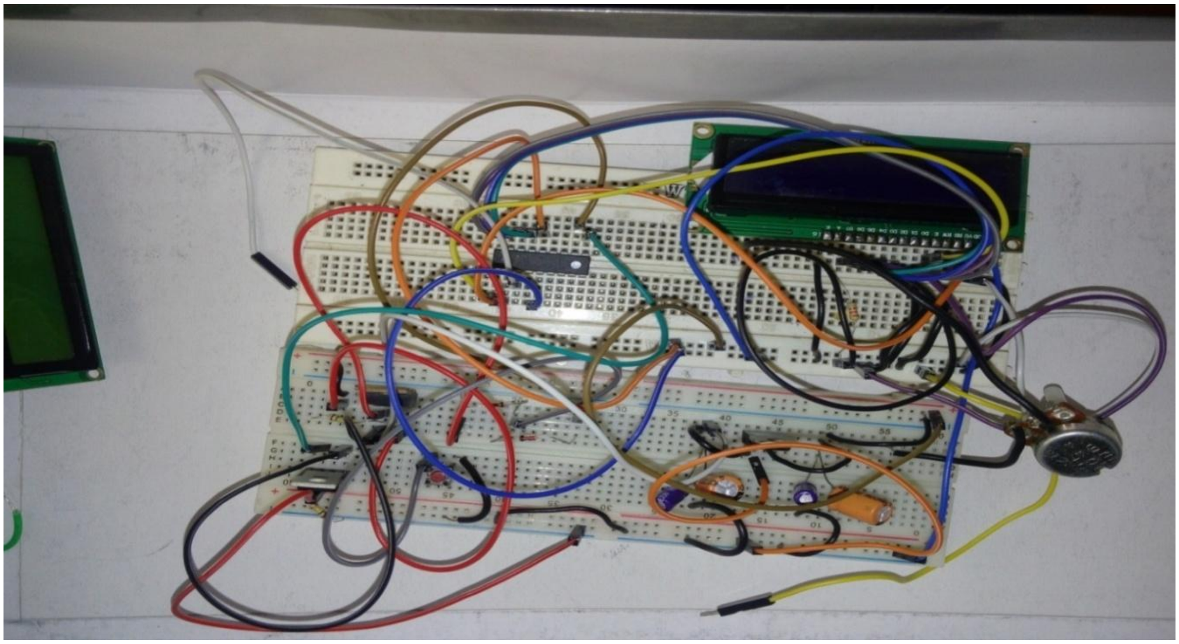


Figure 5.1: Physical Model of Low-Cost Frequency Counter on Breadboard

Figure 5.1represents the connection on a breadboard, which is readily available and easy to use.[3]

5.2 Simulation on Software:

Here we will discuss the simulation on software.

First, PROTEUS CAD program is started which displays a splash screen, then, workspace with interface buttons for circuit designing will appear. The whole circuit should be designed under the blue rectangular box on the screen. Next step is selecting the components of our required circuit. The required components are:

- An LCD display,
- A PIC 16F628Amicrocontroller,
- SWITCH button,
- DIODE,
- 10k and 20k resistor and
- Square wave input.

After the selection of all the components from the library, they are placed in the workspace. The angle of the device can be changed using rotation button. All devices are placed in the workspace and connections are drawn with pen symbol.All the components are connected according to the circuit. Now Hex file is loaded into microcontroller which was built in Micro C IDE.

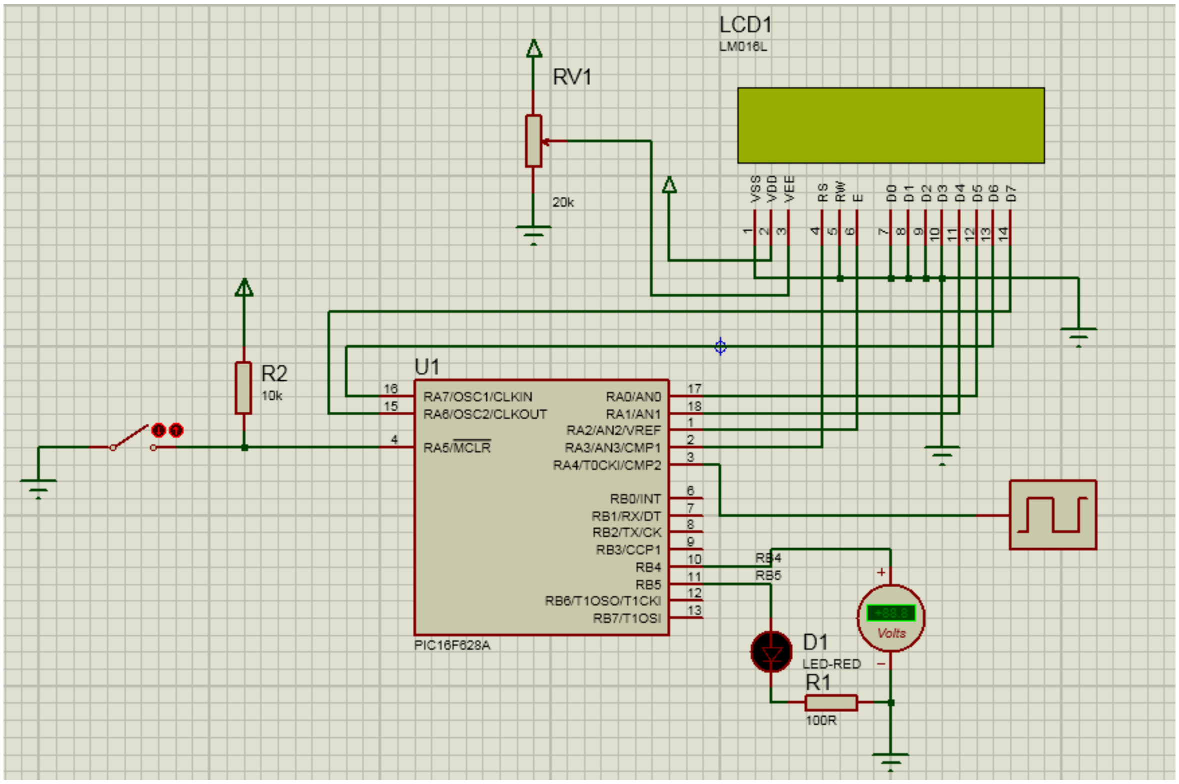


Figure 5.2: Circuit Modeling On PROTEUS

The input square wave is directly given with clock frequency varying from100Hz to 1.8 GHz to pin 3 to PIC since we are using timer/pre-scaler, the corresponding output signal varies from Hz to GHz on LCD display. The **results are shown in below figures.**

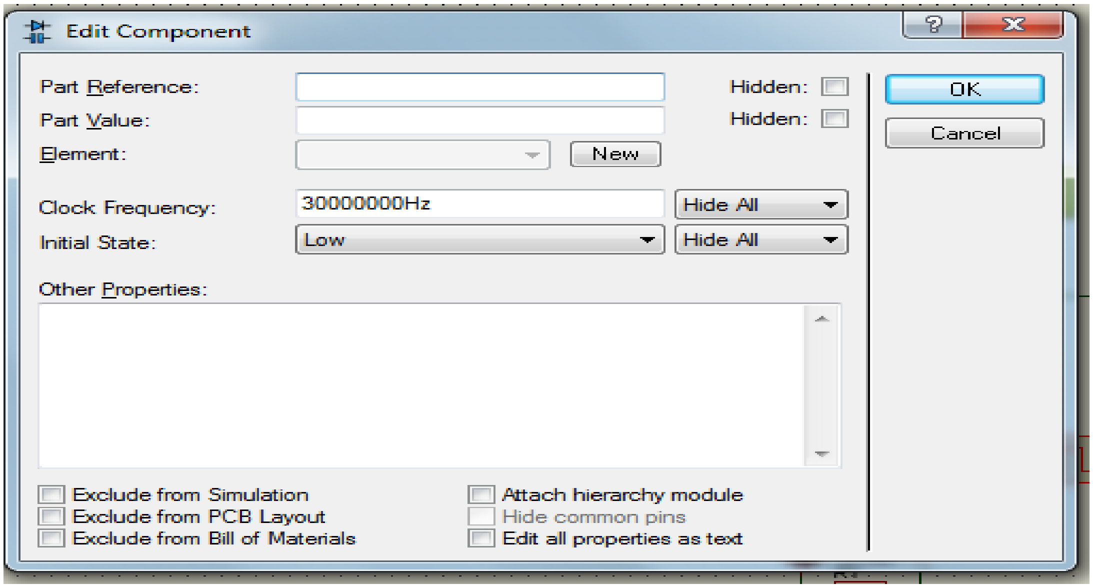


Figure 5.3: Change Clock Frequency of Input Square Wave

We connected the RESET switch along with 10kΩ resistor to pin 4 of PIC which returns the whole hardware if it gets hanged. Pin 10- RB4 or Pin 11- RB5 is connected to the 5V power indicator with LED bulb where as pin 7-14 in LCD are data bus line, along with pin 3 there is a 10kΩ potentiometer to adjust the contrast or backlight of LCD display. Below are the pictures of simulations in Professional PROTEUS software where we are getting frequency up to 1.8GHz on LCD display by varying the clock frequency of input square wave.

5.3 Simulation Results on Proteus

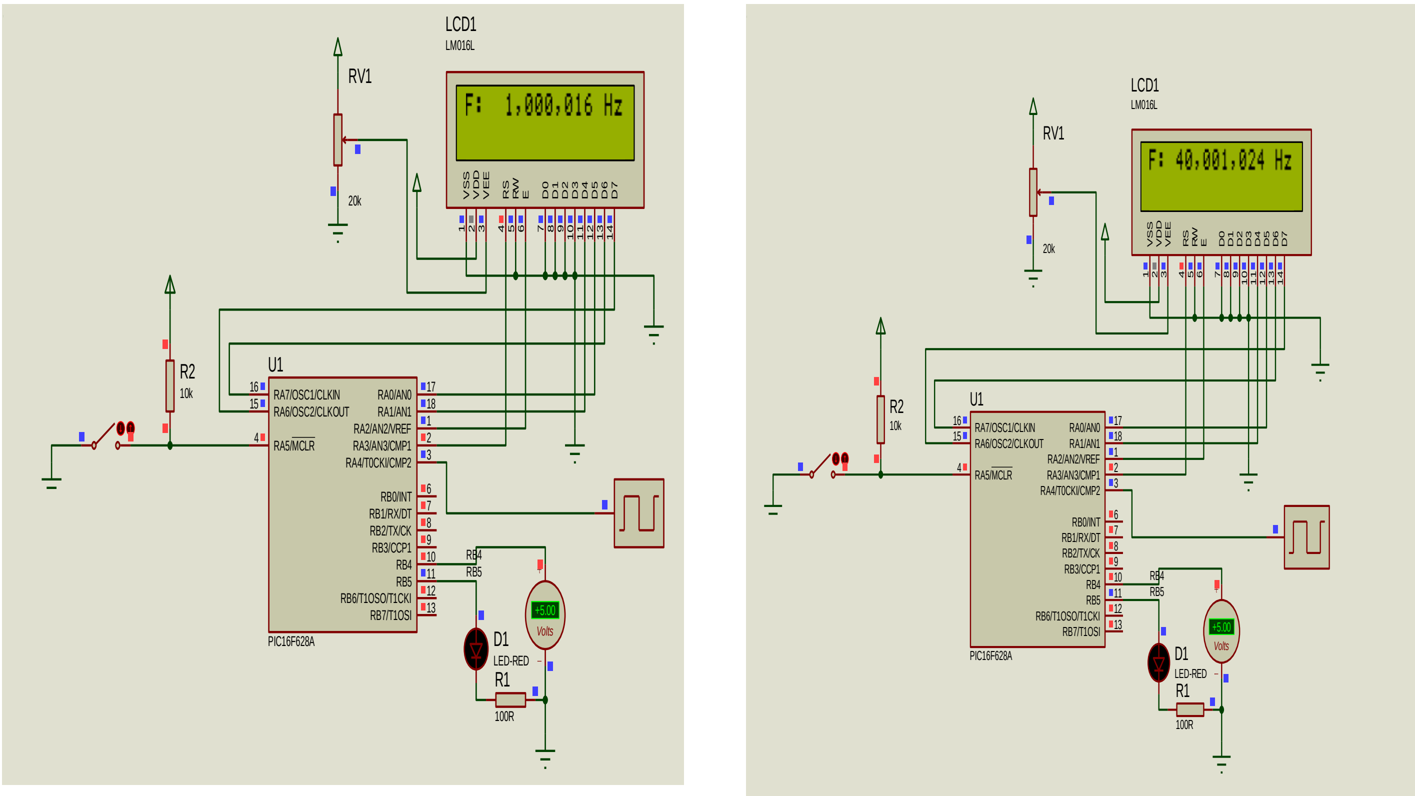


Figure 5.4: 100MHz and 400MHz Signal as Output on LCD Display

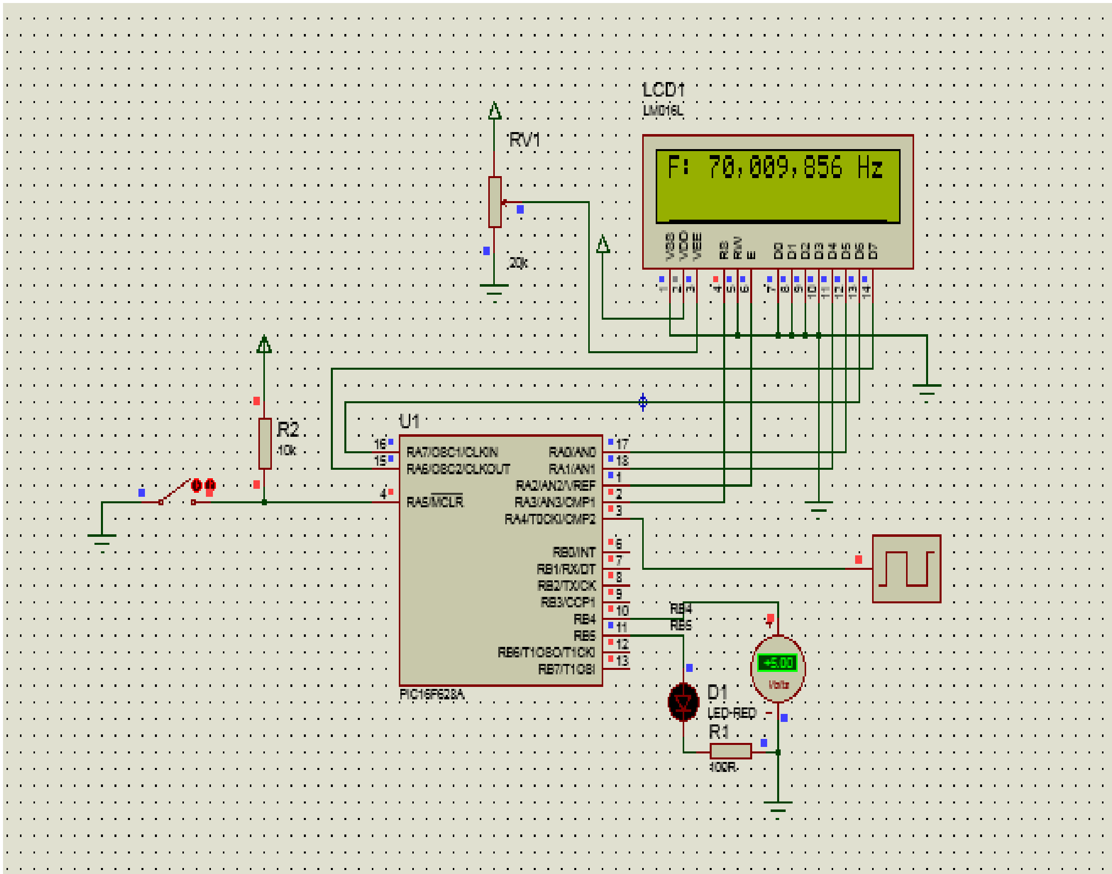


Figure 5.5 Output Result of 700MHz

The only limitation of this circuit is the input square wave if the period of the incoming signal pulse exceeds one second, then the counter may not count any rising edges, during the execution of the program.

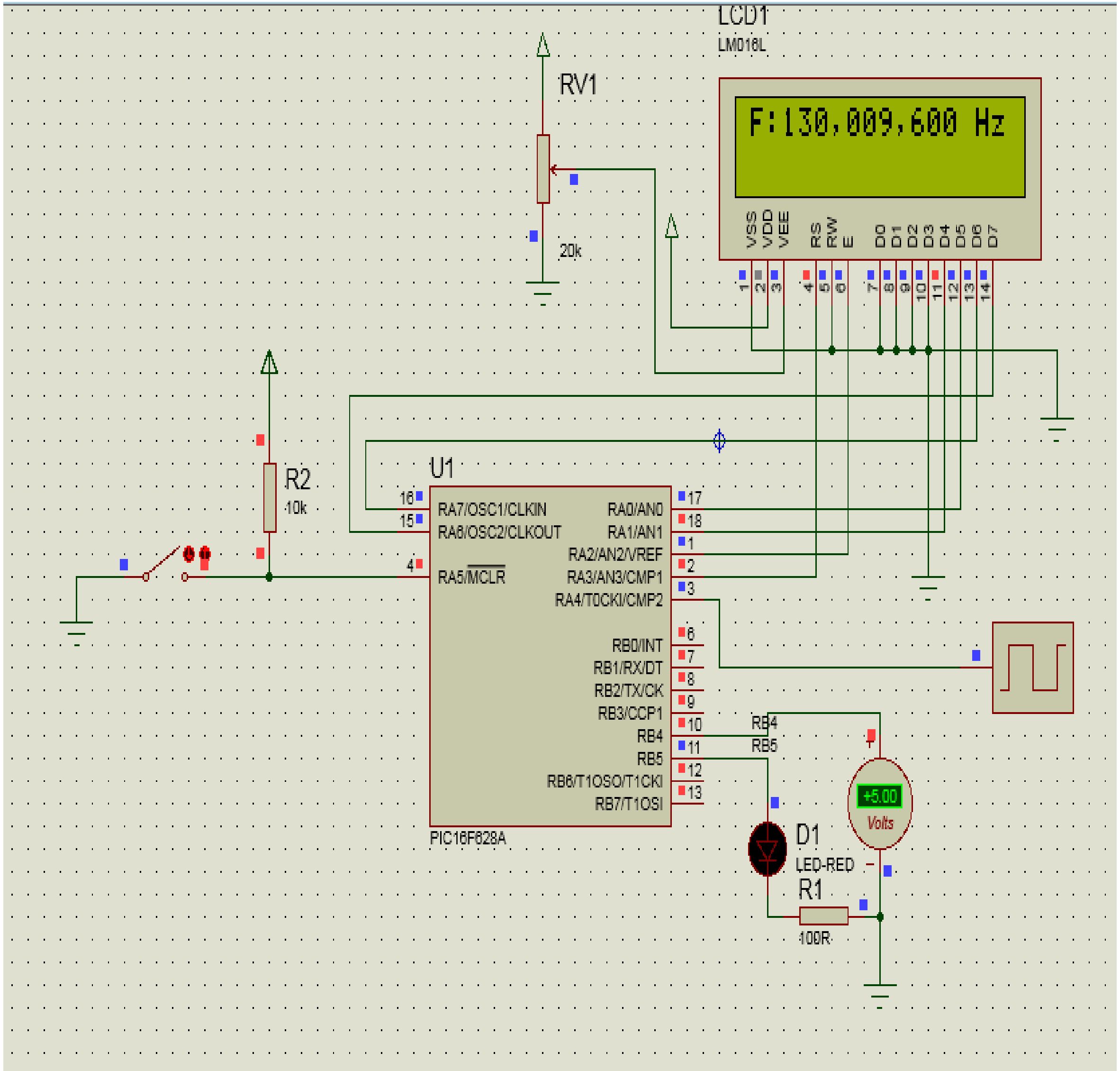


Figure 5.6 1.3 GHz Frequency Output

If the amount of the signal is just too short, then the frequency obtained will be lower than the actual input frequency. Figure 5.6 represents the output signal 1.3 GHz.

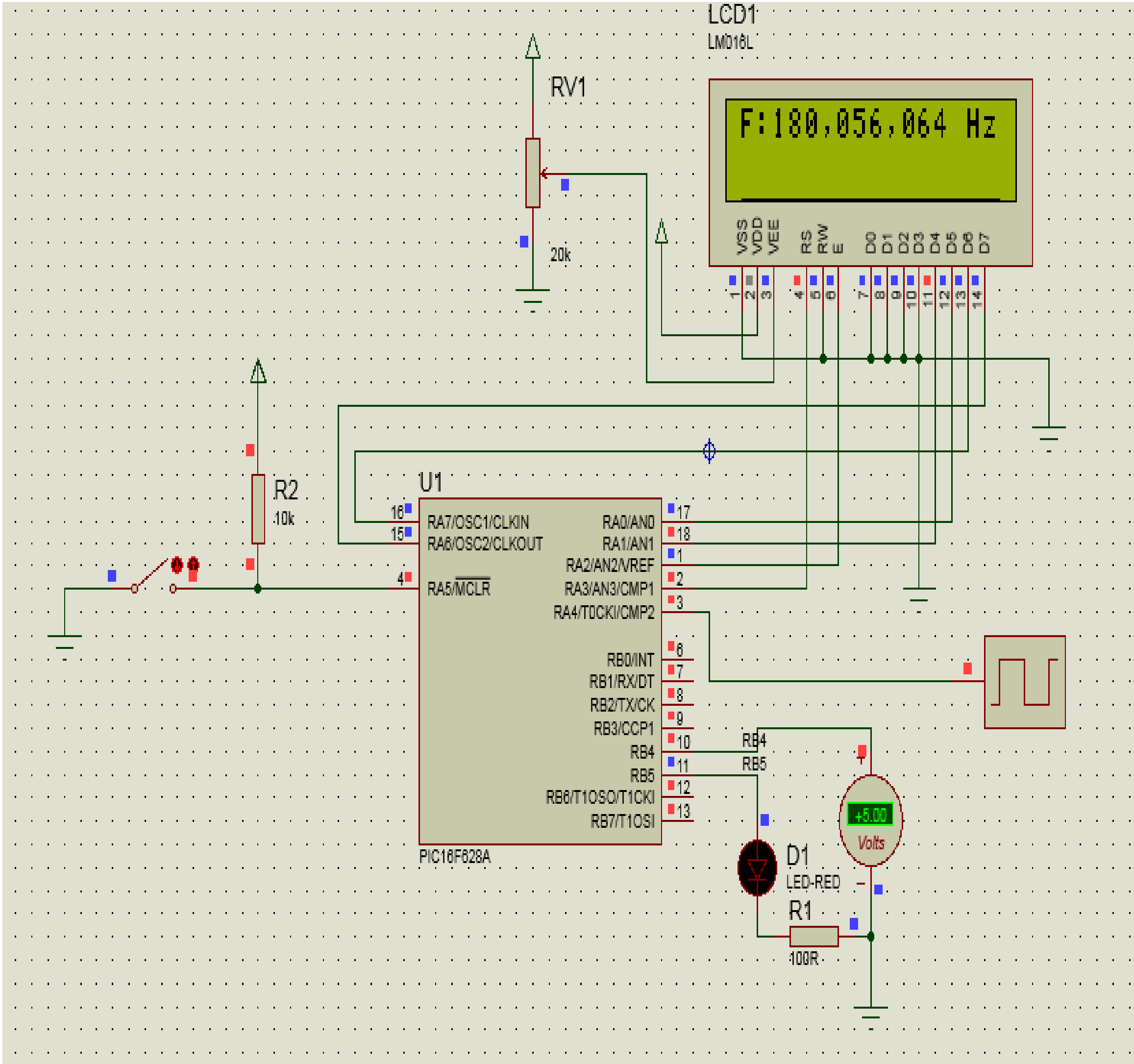


Figure 5.7: 1.8 GHz Frequency Output

Figure 5.7 shows the maximum output of frequency counter which is 1.8 GHz.

5.4 PCB DESIGN

Frequency Counter

In all micro-controller based circuit, we need three main things:

First is the power supply that is 5V ground, in this case, we have generated it using IC 7805.

The second thing is clock circuit which consists of crystal oscillator of 4 MHz frequency along with two ceramic capacitors of 22µf used to filter out any unwanted frequencies that may be generated by the crystal.

The third thing is reset circuit. It consists of reset switch and 10k ohm resistor. The system can be reset whenever hardware gets hanged. Before picking components, we read data sheets of all the components relevant to our circuit, and we took proper care while soldering the components. [28] In the picture shown below, the components are marked with their name on PCB.

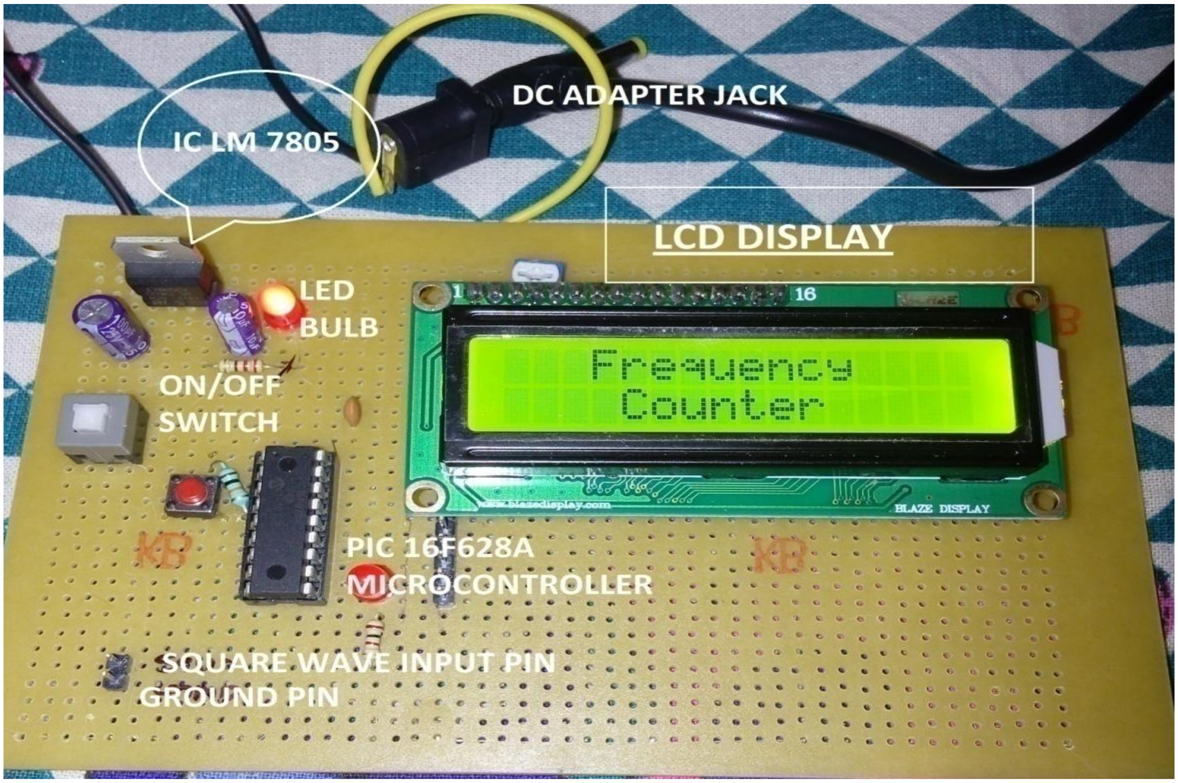


Figure 5.8 Front Side of PCB

Picture below represents the back side of PCB circuit

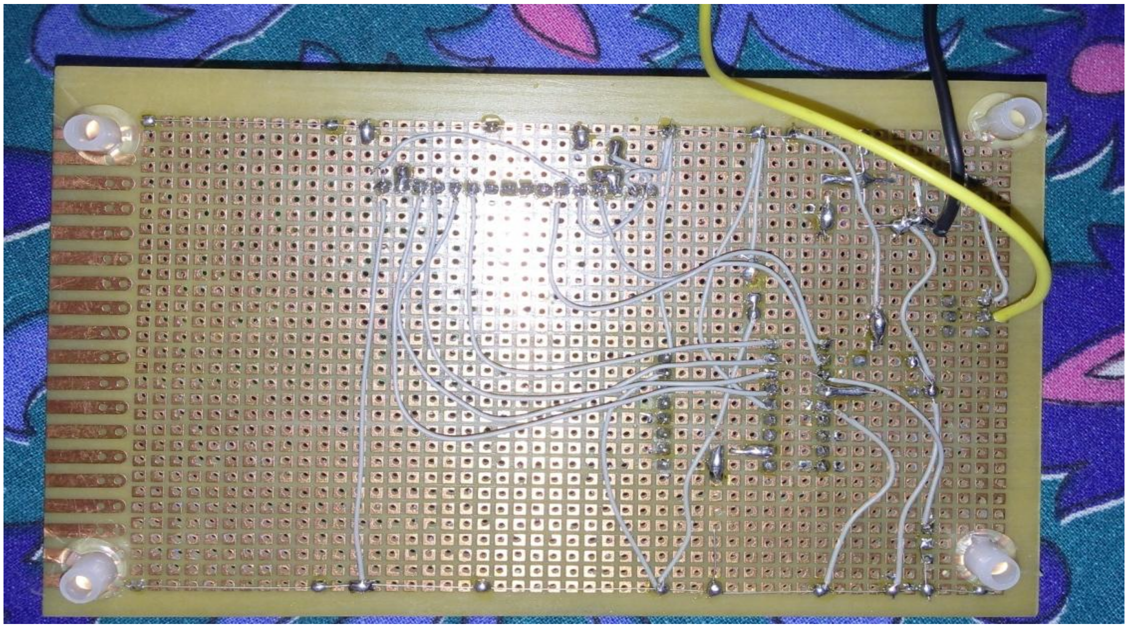


Figure 5.9 Back side of PCB

Since circuitsimulation is working properly and able to count the signal up to 1.8 GHz. Our hardware design will also work the same till we have a signal source to generate signal up to the range.

5.5 AD 9850 PCB DESIGN

After the complete circuit of a frequency counter, next step is to design a signal generator. For generator we need following components:

- Arduino Uno,
- A 16x2 line LCD display,
- AD9850 module and
- A rotary encoder
- External crystal
- Resistors and Capacitors
- Switch Button
- Jumper wires

After designing the whole circuit on board according to the schematic diagram, this will generate the signal with high stability. For designing the PCB, we need to understand the power supply section, which we have studied in above section next, we will move to the waveform generator circuit AD9850 (DDS)using ATMEGA-328 micro-controller generate analog waveforms with digital input we program it using ARDUINO module through ICSP lines have MOSI, MISO, SCK, RESET, VCC AND ground pins which control the DDS chip we use Uno, or we can make our own circuit on PCB using ATMEGA-328 that is what we did.

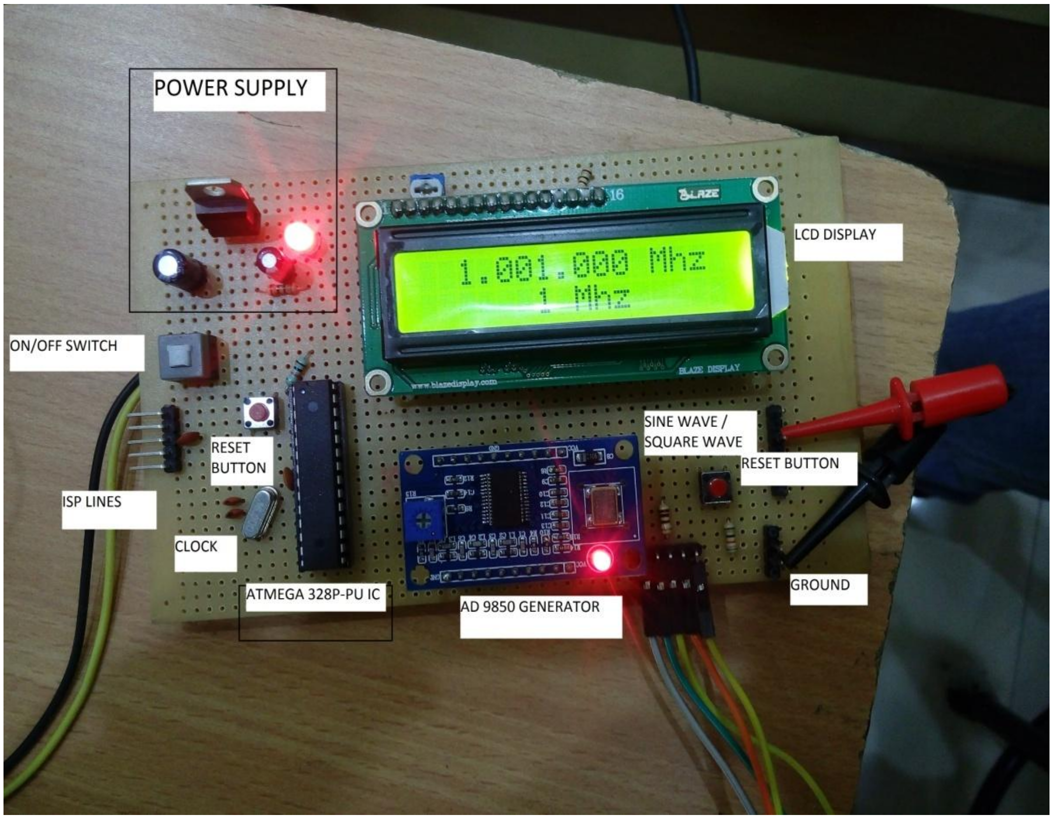
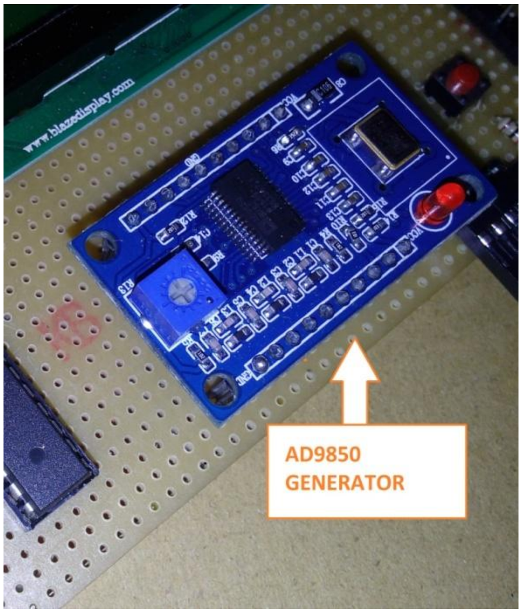


Figure 5.10 Components Marking On PCB

This is the basic circuit which is required to make AD 9850 signal generator.

If reset clock and power supply are working ok, then the circuit must work properly. First, thecodewas run after designing the circuit which is used to blink the LED bulb connected to pin 19 of PIC or pin 13to cross check the functionality of the circuit. After the successful test, the whole circuit was reprogrammed using ICSP lines.[29]



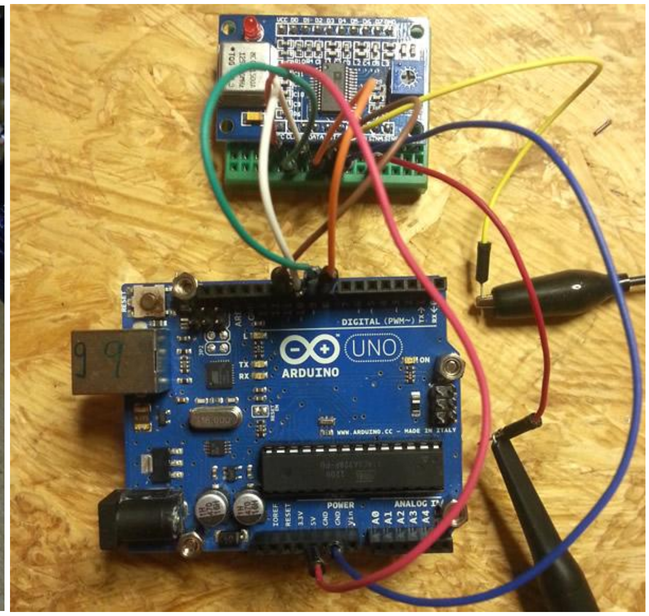


Figure 5.11AD 9850 (DDS) CHIP **Figure5.12**AD 9850 Connections With Arduino

AD 9850 waveform generator

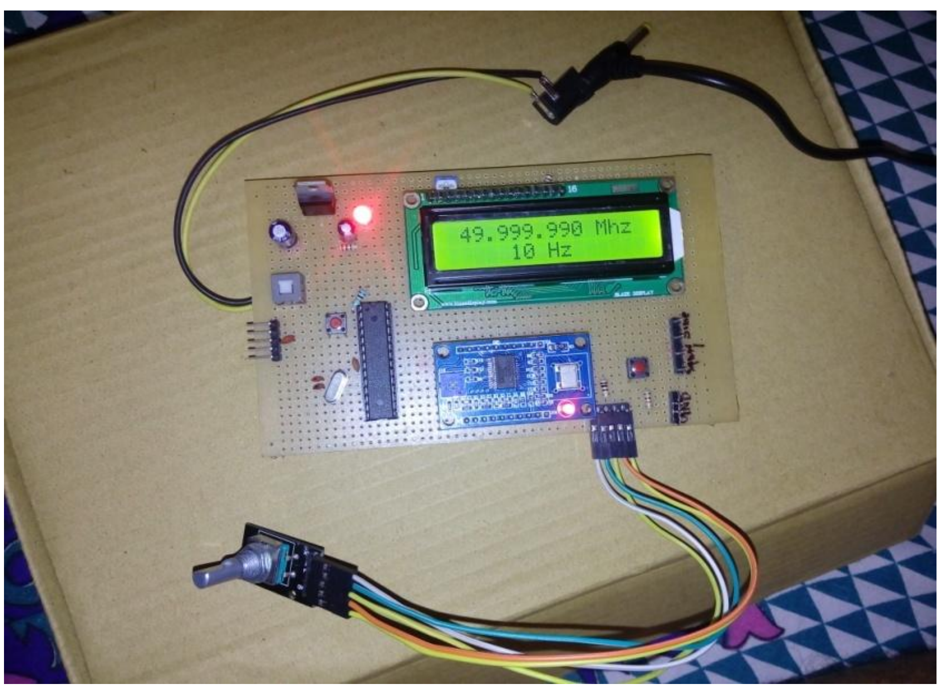


Figure 5.13: 50 MHz output of generator

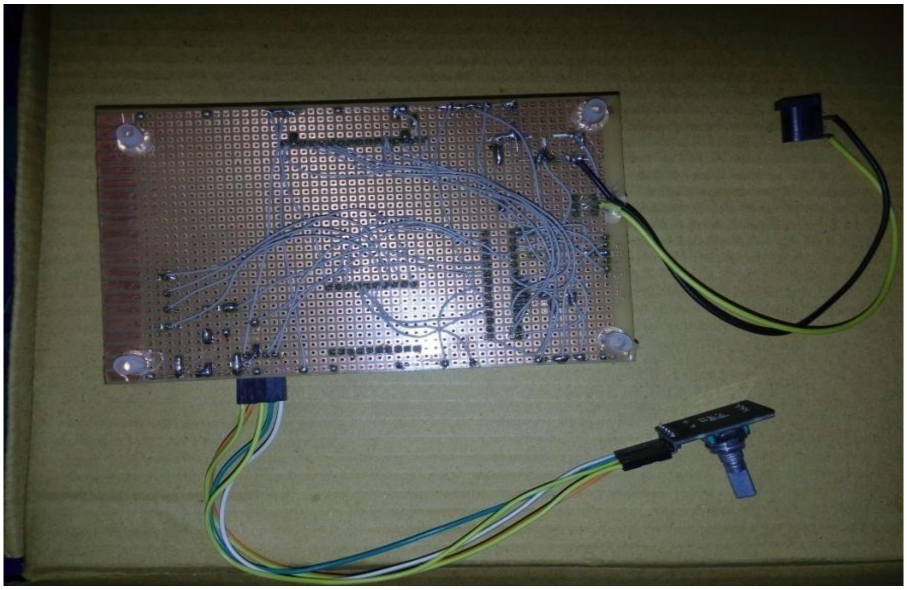


Figure5.14 Back side of PCB

The above figures show the completed hardware design of frequency counter and a waveform generator.

The signal generator device does not have its own display to show generated a waveform, it can just represent the decimal value of generated signal, so, to represent the waveform we need to connect it with a CRO. For connection with CRO, the main point to keep in mind is that ground pin should be connected properly else it will not work. In the figure below, the red color wire is connected to the signal generator pin while the black color wire represents ground.

We have 6 pins out of which upper 2 pins are used for sine wave generation, lower 2 pins are used for square wave signal, and the remaining 2 pins are used as ground pins.

When the power of the AC adapter turns on, and if the led bulb on PCB glows this means the supply function is working properly. Next, the phase step size is adjusted using a rotary encoder and significant output wave will be generated will be displayed on display.

Below figures show the waveform generated wave, by fixing the step size at 1MHz, we are getting 40 MHz frequency on display.

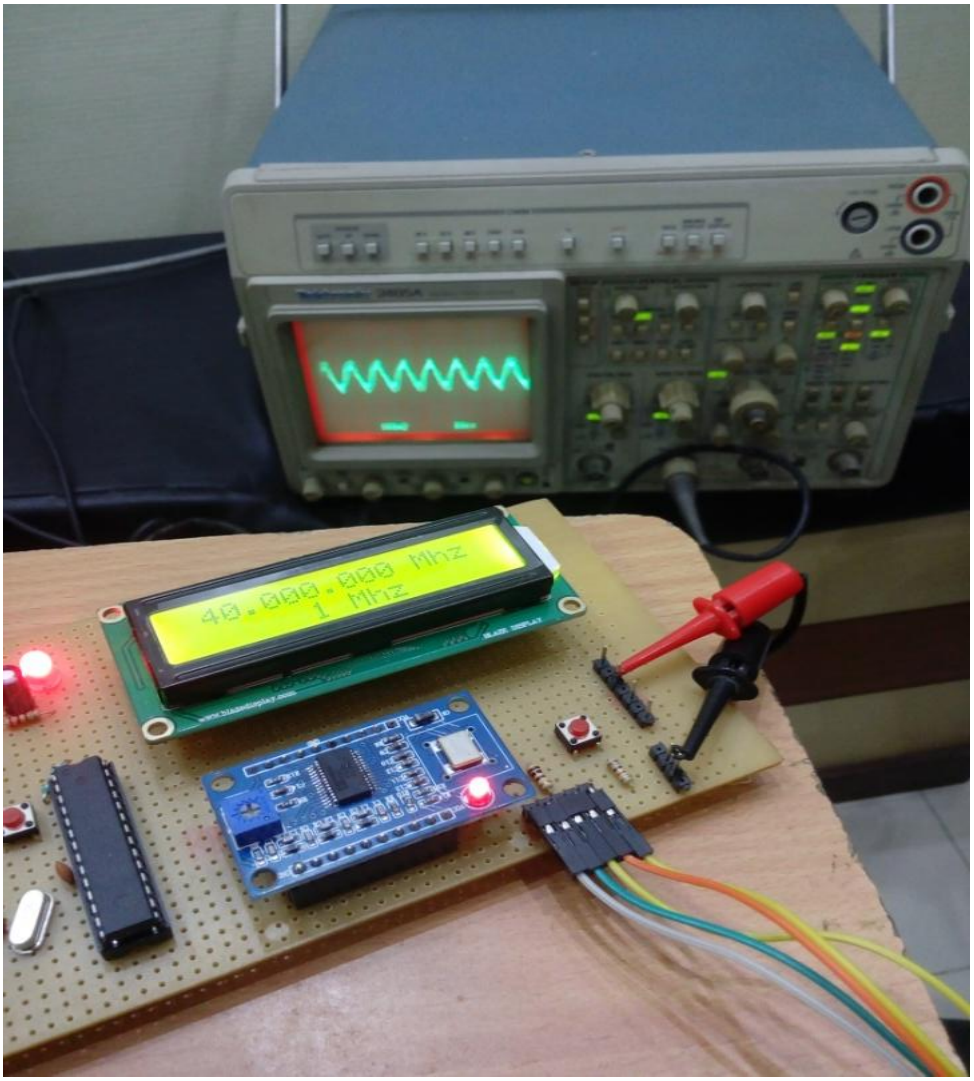


Figure 5.15 Test Result on CRO

Similarly, we can set step size maximum at 1MHz, and the50MHz signal will be generated.

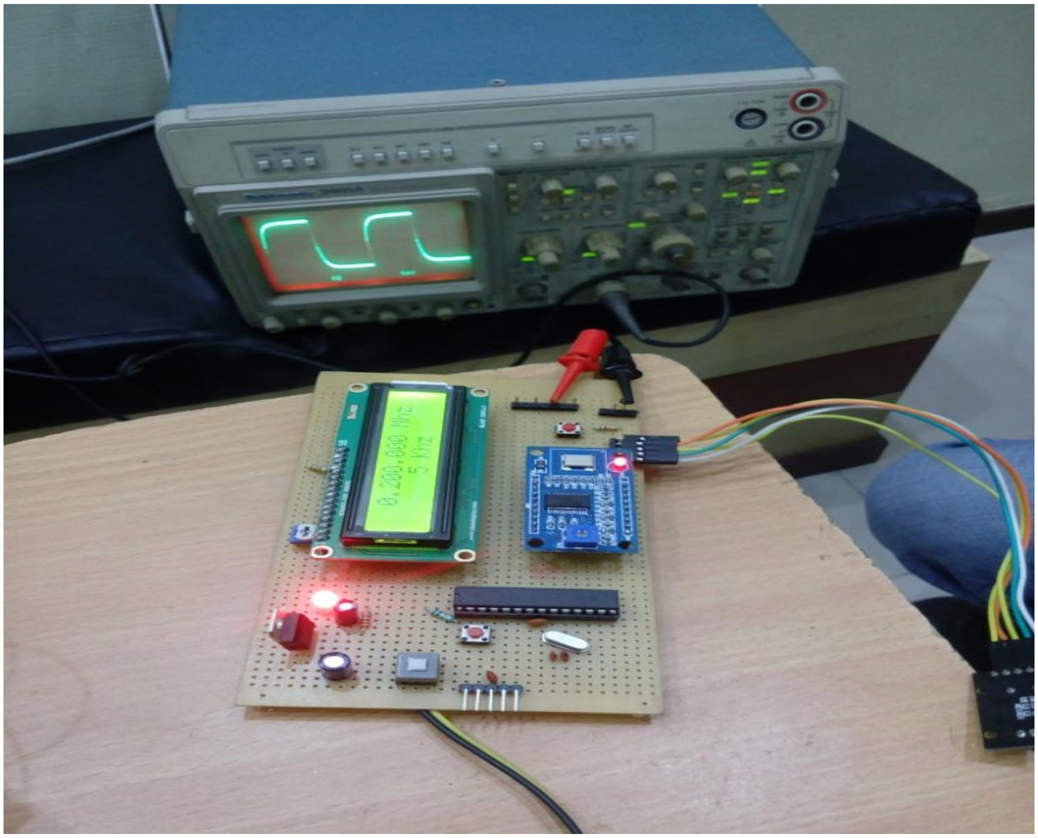


Figure 5.16 Generation of The Square Wave at 200 Hz

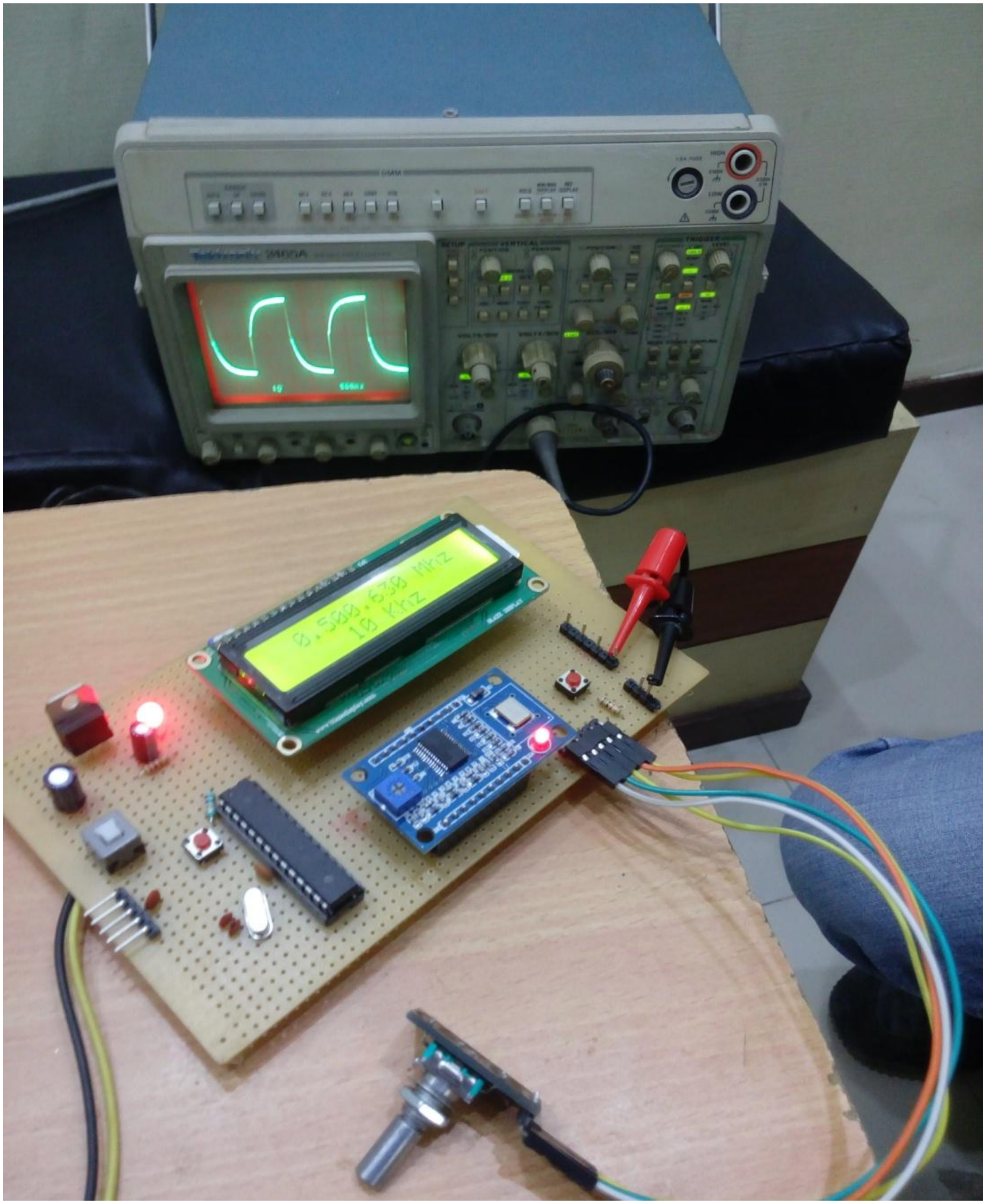


Figure 5.17 Test result of Square Wave Generation

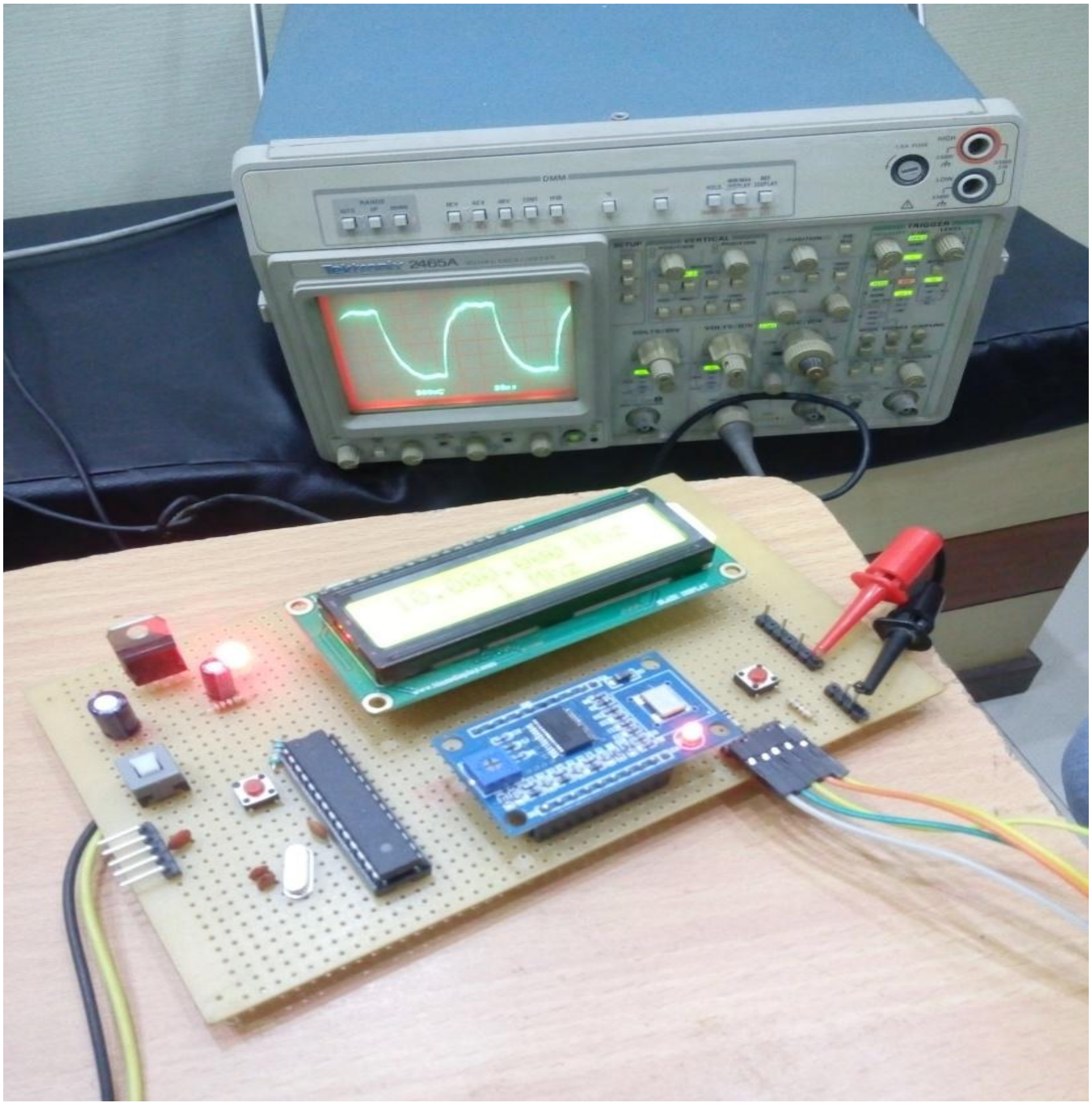


Figure 5.18 Laboratory Test Result for Square Wave at 10MHz

Figure 5.19 represents the connection of frequency counter with AD 9850 waveform generator. The complete system (including the hardware components and software routines) is working as per the initial specifications and requirements of our project.

Certain aspects of the system can be modified as operational experience is gained with it,and as the user works with the system, new ideas may emerge for the development and enhancement of the project.

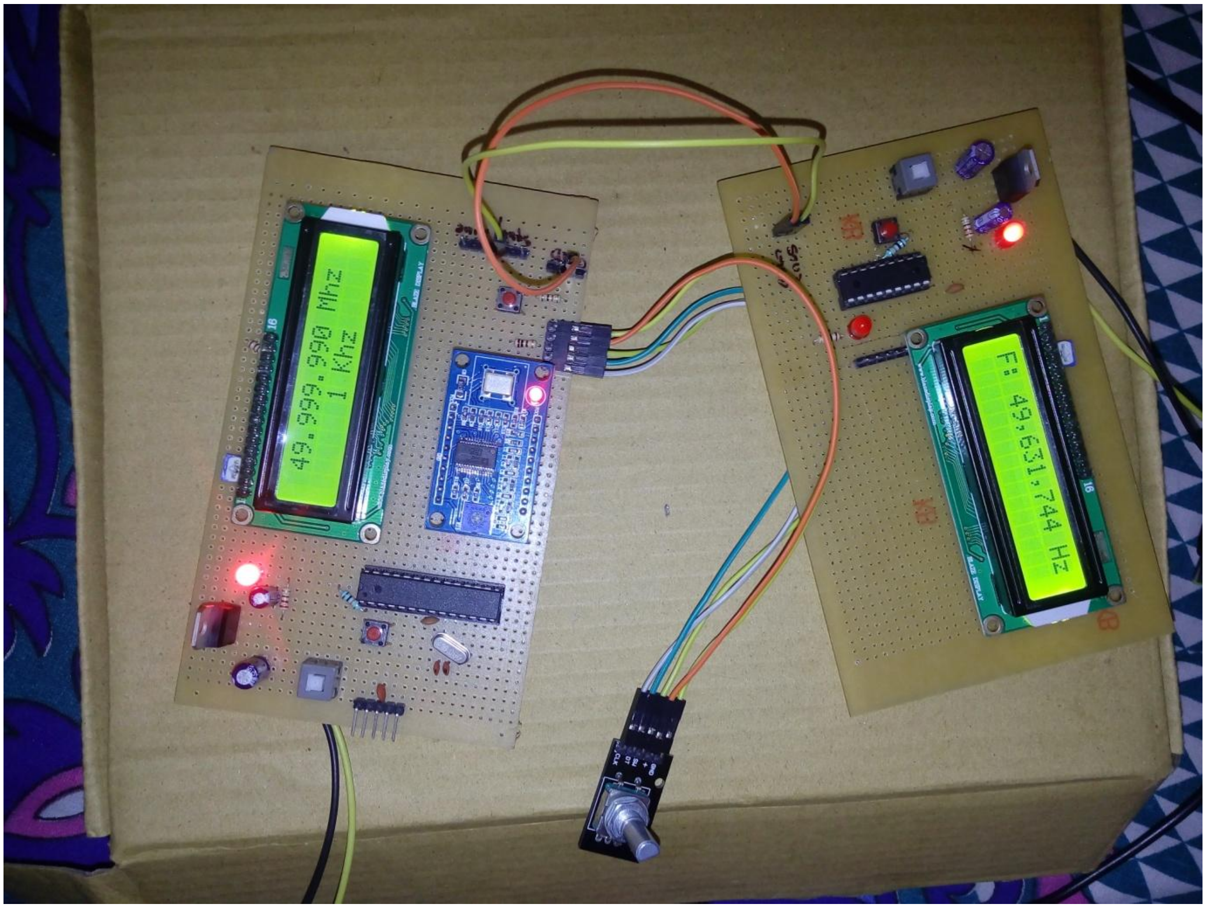


Figure 5.19 Connection of Frequency Counter with AD9850 Waveform Generator

Chapter 6

Conclusion & Future Scope

The scope of this project was to present a possible design of a frequency counter and verify its functionality. Our complete system (including hardware and software simulation) is working as per the initial specifications and requirements of our project. Schematic simulation is done in PROTEUS software while coding is done using Micro C IDE which provides a lot of predetermined function.

We designed the hardware of frequency counter using PIC 16F628A which is capable of measuring frequency up to 1.8 GHz in normal condition with a maximum amplitude of 15V. For cross checking of the system, we designed AD9850 (DDS) CHIP waveform generator using ATmega-328 which generates sine and square wave up to 50 MHz We connected both hardware carefully, and the desired result was at the display with stability. So, we can conclude that the frequency measured by Arduino matches with the actual frequency of the input. The frequency counter built here is both power and cost efficient as per the initial specifications and can be used in frequency counter application.

6.2 Future Work

Since Digital frequency counter are becoming more and more popular, RF frequencies are needed to be testing. The designed circuit is both power and cost efficient as per initial specifications, still few factors need to be considered to improve its accuracy primarily by keeping the oscillator warmth continuously will avoid errors, select base time, Noise of instrument input time base and counter resolution, by examine these, the properly designed frequency counter will count precise frequency and can be used in radar based application, temperature and mostly in microwave frequency measurement. Since it is used by a vast majority of testing equipment in the modern day and after accuracy, it will be used to communicate digitally with the devices directly. The range of this frequency counter can be increase to measure high frequency.

References

1. http://lea.hamradio.si/~s53mv/counter/history.html
2. https://en.wikipedia.org/wiki/Frequency_counter
3. http://www.electronics-lab.com/project/100mhz-frequency-counter-with-pic16f628a-2/
4. http://www.radio-electronics.com/info/t_and_m/frequency_counter/counter_basics.php
5. http://www.best-microcontroller-projects.com/article-frequency-counter.html
6. http://www.radio-electronics.com/info/t_and_m/frequency_counter/frequency-counter-timer-interval.php
7. http://www.radio-electronics.com/info/t_and_m/frequency_counter/frequency-counter-specifications.php
8. Analog Devices Inc. (2016) PLL Frequency Synthesizer ADF4108 Data Sheet, Rev. E, Norwood Massachusetts USA Analog Devices Inc. (2013) RF PLL Frequency Synthesizers Data Sheet ADF4110/ADF4111/ADF4112/ADF4113, Rev. F, Norwood Massachusetts the USA

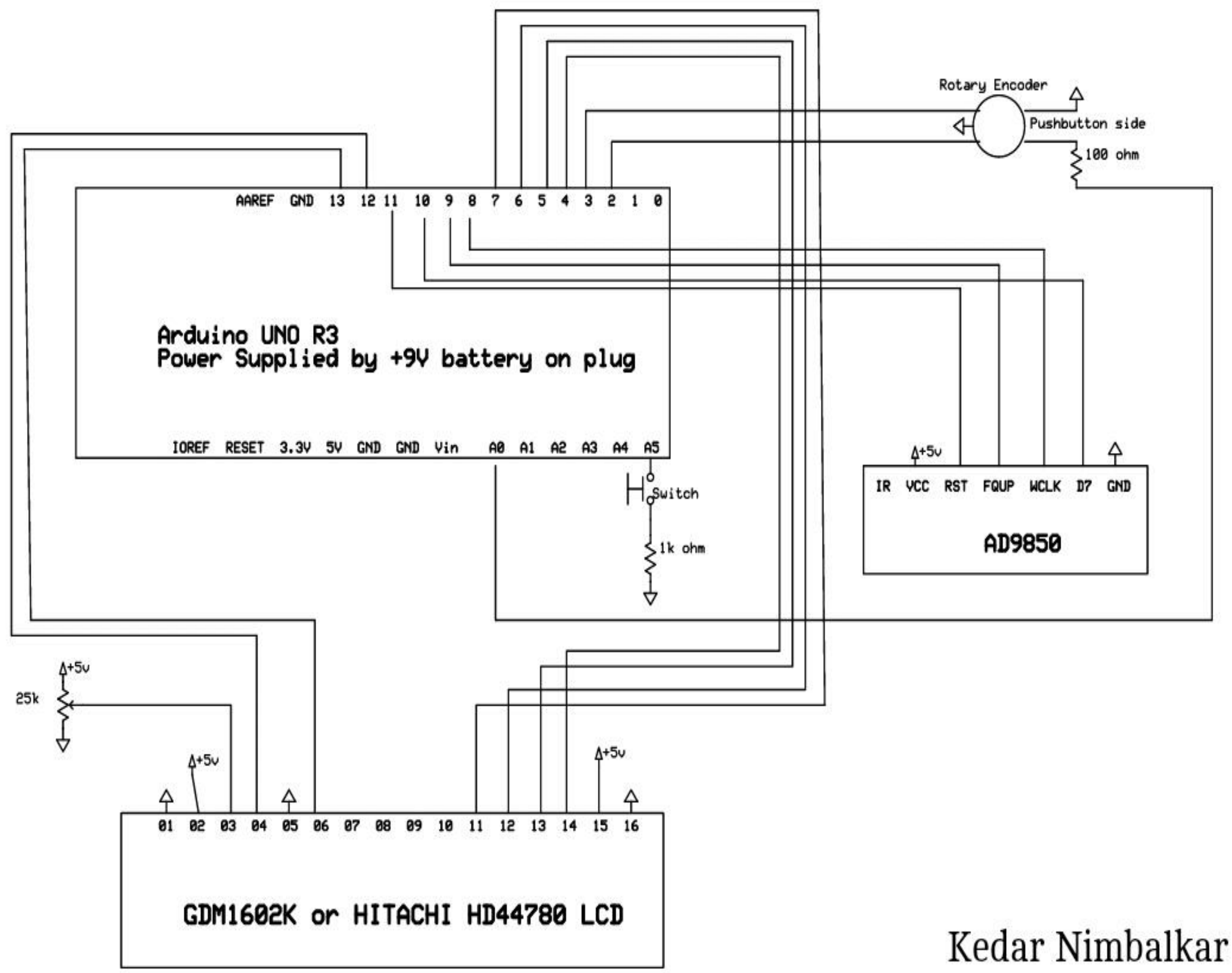
9 http://www.electronics-lab.com/project/100mhz-frequency-counter-with-pic16f628a/

10 http://www.electronics-lab.com/project/100mhz-frequency-counter-with-pic16f628a-2/

11. Analog Devices Inc. (2016) HMC983LP5E DC - 7 GHz FRACTIONAL-N DI-VIDER AND FREQUENCY SWEEPER DATASHEET, v02.0112, Norwood Massachusetts the USA
12. http://www.ohio.edu/people/uijtdeha/ee3954_fall13_09_timers.pdf
13. C.Z Nan, X.P Yu, B.Y Hu, Z.H Lu, W.M Lim, Y Liu, K.S Yeo and C Hu, A 6-GHz dual-modulus prescaler using 180nm SiGe technology, 978-1-61284- 865-5, ISIC, Singapore, December 2011.
14. Design and Implementation of a High-frequency Signal Generator using the (DDS) Mixing Principle Jufang Hu
15. http://electronics.pl7.de/home/(DDS)-sine-signal-generator-for-am-modulation-from-1-Hz-to-40-MHz-with-an-ad9850-or-ad9851-an-ATmega-328-and-an-Arduino-firmware/

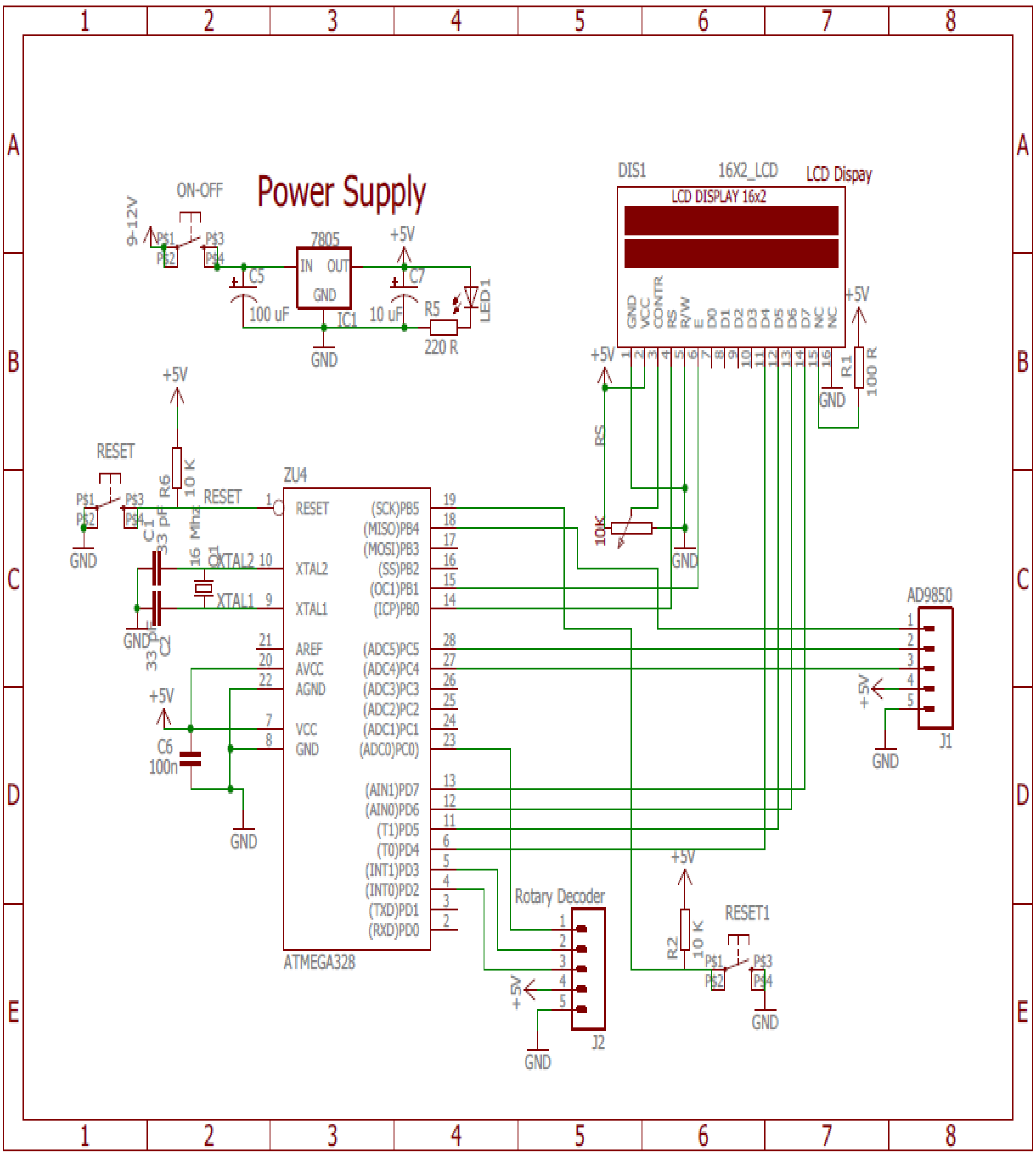
16. Advance in control engineering information science 'Design of an arbitrary waveform signal generator' Xiao Chena,*, and Jianxiang Chena
17. Hewlett-Packard Company (1997) Fundamentals of the Electronic Counters
18. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.192.9577&rep=rep1&type=pdf
19. http://www.foxdelta.com/products/freqcounter/fc1/1011/fc1-1011.pdf
20. Bioenergy frequency measurement of stressed and non-stressed individuals using Resonant Field Imaging (RFI) frequency counter
21. A frequency counter based analog-to-digital converter for a RFID telemetry system
22. http://electronicsforu.com/resources/learn-electronics/16x2-lcd-pinout-diagram
23. https://en.wikipedia.org/wiki/Light-emitting_diode
24. Analysis of RFI effects in voltage regulator ICs: measurement techniques, picking capability prediction and protection methods
25. http://electronics.pl7.de/home/(DDS)-sine-signal-generator-for-am-modulation-from-1-Hz-to-40-MHz-with-an-ad9850-or-ad9851-an-ATmega-328-and-an-Arduino-firmware/
26. Microchip Device INC 2016 ATmega-328/p 8-bit AVR Microcontroller Atmel-42735-8-bit-AVR-Microcontroller-ATmega-328-328P_Datasheet.
27. ST Microelectronics IC LM 7805 INC. 2016, Positive voltage regulator ICs Research technical documents Datasheets.
28. http://Arduinotehniq.blogspot.in/2015/03/signal-generator-with-ad98509-and.html
29. https://codebender.cc/sketch:266550#Arduino%20%2B%20AD9850%2030MHZ%20(DDS)%20Signal%20Generator.ino
30. http://www.electro-tech-online.com/blog-entries/pic16f628a-all-about-timer-0.178/?page=1

Appendix1: Schematic of AD985



Circuit Diagram of Cheap (DDS) Signal Generator Using AD9850 and Arduino Uno

Schematic of AD9850 on Eagle software



Appendix 2: Frequency counter source code implemented in Micro C.

```
#define _XTAL_FREQ       4000000

// Lcd pinout settings

sbit LCD_RS at RA3_bit;

sbit LCD_EN at RA2_bit;

sbit LCD_D4 at RA1_bit;

sbit LCD_D5 at RA0_bit;

sbit LCD_D6 at RA7_bit;

sbit LCD_D7 at RA6_bit;

// Pin direction

sbitLCD_RS_Direction at TRISA3_bit;

sbitLCD_EN_Direction at TRISA2_bit;

sbit LCD_D4_Direction at TRISA1_bit;

sbit LCD_D5_Direction at TRISA0_bit;

sbit LCD_D6_Direction at TRISA7_bit;

sbit LCD_D7_Direction at TRISA6_bit;

// MAX_TIMECOUNT sets the number of loops after which the power is switched off

// one loop is approximately 1.15 seconds long and the time period will be T = MAX_TIMECOUNT x 1.15

#define MAX_TIMECOUNT    208

int f1;

char PowerOFF_flag;

// interrupt procedure

void interrupt() {

   if (T0IF_bit) {
```

```
      f1++;
      T0IF_bit = 0;
    }
    else if (RBIF_bit) {                    // if there is change on PORTB
    if (!PORTB.F5)                     // check if Power button is pressed
PowerOFF_flag = 1;               // sets the PowerOFF_flag
RBIF_bit = 0;                      // clear the interrupt flag
    }
}
// displays the frequency with commas after each 3 digits
void display_freq(long f) {
  char i, n, k, digit[9];
  n = 0;
  // separating digits
  do {
    digit[n] = '0' + f % 10;
    f /= 10;
    n++;
  } while (f > 0 && n < 9);
  k = 11;                     // number of symbols to be displayed: 9 digits + 2 commas
for(i = 0; i< 9; i++)
  {
    if (k == 4 || k == 8)
     {
      if (i< n)
```

```
Lcd_Chr(1,k+2,',');    // displays commas at 4-th and 8-th positions
      else
Lcd_Chr(1,k+2,' ');    // displays ' ' if number is shorter
      k--;
    }
    if (i< n || i == 0) {
Lcd_Chr(1,k+2,digit[i]); // displays digit
    }
    else {
Lcd_Chr(1,k+2,' ');      // displays ' ' if number is shorter
    }
    k--;
  }
}
void main() {
   long freq;
int f2, prescaler, timecount;
   char i, prescaler_bits;
   CMCON = 0b00000111;                          // comparator off
   T1CON = 0b00010000;                          // TMR1 prescale 1:2, osc = off
   TRISA = 0b00110000;
   TRISB = 0b11001111;
   PORTB.F4 = 1;                               // sets the power on
   PORTB.F5 = 1;
delay_ms(1000);
```

```
  PORTB.F5 = 0;
Lcd_Init();
Lcd_Cmd(_LCD_CLEAR);
Lcd_Cmd(_LCD_CURSOR_OFF);
Lcd_Out(1, 1, "   Frequency    ");
Lcd_Out(2, 1, "    Counter     ");
delay_ms(5000);
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(1, 1, "F:");
Lcd_Out(1, 15, "Hz");
PowerOFF_flag = 0;
timecount = 0;
  while (1) {                          // main loop
    OPTION_REG = 0b00100111;           // set TMR0 prescale = 256
    // Setting the initial value of TMR1 to 2458     1
    // Timer1 will run for 2684 cycles = 0.08191 seconds
    TMR1L = 0b00000100;
    TMR1H = 0b01100000;
    TMR1IF_bit = 0;
    T0IF_bit = 0;
    f1 = 0;
    INTCON = 0b10101000;               // Enable interrupt at Timer0
    TMR0 = 0;                          // start counting input signal
    TMR1ON_bit = 1;                    // start Timer1
    while (!TMR1IF_bit) {}             // loop for 0.08191 seconds
```

```
INTCON = 0;                              // Disable interupts
TMR1ON_bit = 0;                          // stop Timer1
prescaler = 256;                         // sets the initial value of prescaler
prescaler_bits = 0b00000111;             // sets the initial value of prescaler bits
// calculating prescaler
// f1 holds the input frequency divided by 256*128 and multiplied by time period 0.08191s
// if the frequency is above 102.4MHz ->prescaler = 256
// 102.4MHz .. 51.2MHz ->prescaler = 128
// 51.2MHz .. 25.6MHz ->prescaler = 64
// 25.6MHz .. 12.8MHz ->prescaler = 32
// 12.8Mhz .. 6.4MHz ->prescaler = 16
// 6.4MHz .. 3.2MHz ->prescaler = 8
// 3.2MHz .. 1.6Mhz ->prescaler = 4
// 1.6MHz .. 800kHz ->prescaler = 2
// 800khz .. 0 ->prescaler = 1
f1 <<= 1;
while (f1 <prescaler) {
prescaler>>= 1;                          // divide prescaler by 2
prescaler_bits--;
}
if (prescaler< 2) {
prescaler = 1;
OPTION_REG = 0b00101000;                 // TMR0 prescale 1:1
prescaler_bits = 0;
}
```

```
else {
  OPTION_REG = 0b00100000 | prescaler_bits; // TMR0 prescale
prescaler_bits++;
}
// Seting the initial value of TMR1 to 3035
// Timer1 will run for 3035 cycles = 0.5 second
T1CON |= 0b00110000;
TMR1L = 0b11011011;
TMR1H = 0b00001011;
TMR1IF_bit = 0;
T0IF_bit = 0;
f1 = 0;
INTCON = 0b10100000;        // Enable interrupt for Timer0
TMR0 = 0;                   // start counting input signal
TMR1ON_bit = 1;             // start Timer1
while (!TMR1IF_bit) {}      // loop for 0.5 second
INTCON = 0;                 // Disable interrupts
TMR1ON_bit = 0;             // stop Timer1
f2 = TMR0;
// calculating frequency
freq = (long)f1;
freq<<= 8;                  // multiply by 256
freq += (long)f2;
freq<<= prescaler_bits+1;   // multiply by prescaler
display_freq(freq);
```

```
timecount++;
    // Shut down sequence
    /*if (PowerOFF_flag || timecount> MAX_TIMECOUNT) {
      INTCON = 0;                    // disable all interrupts
Lcd_Cmd(_LCD_CLEAR);
Lcd_Out(1, 2, "Power OFF...");
delay_ms(1000);
      PORTB.F4 = 0;                  // switch off the power
      while (1) {}                   // endless loop (until power is off)
    }   */
  }
}
```