Hieu Xuan Thuy Tran

REFERENCE RADIATOR DESIGN AND IMPLEMENTATION BY USING PROGRAMMABLE CLOCK GENERATOR

Technology and Communication
2022

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Hieu Xuan Thuy Tran
Title	Reference Radiator Design and Implementation by Using Programmable Clock Generator
Year	2022
Language	English
Pages	62 + 6 Appendices
Name of Supervisor	Jani Ahvonen

The reference radiator has been an indispensable tool for Electromagnetic Compatibility (EMC) laboratories where it is used as a well-calibrated and well-known noise source to ensure the quality of measurement results. The objective of this thesis was to design and implement a reference radiator by using a programmable clock generator Si5351A-B-GT on Microcontroller systems.

The Si5351A clock generator breakout board was used for testing and analysis to build a reference radiator. When being connected to a monopole antenna, the radiator can generate radiated emissions from 12MHz up to 50MHz.

The final device was completed with more features such as measuring temperature based on a sensor and adjusting a frequency being selected by the user through push buttons. The output frequency results are also expected to be more improved. With the criteria of compact and low-cost design, the device is housed into a small metal box. It is very handy for characterizing anechoic chambers or many other things.

Keywords Reference radiator, EMC, and clock generator

CONTENTS

LIST OF FIGURES AND TABLES

TERM AND ABBREVIATIONS

DUT	Device under test
EMC	Electromagnetic compatibility
EUT	Equipment under test
FAC	Fully anechoic chambers
FAR	Fully anechoic rooms
FFT	Fast Fourier transforms
FMD	Feedback Multisynth Divider
I2C	Inter – integrated circuit
IDE	Integrated development environment
LCD	Liquid-crystal display
MSOP	Mini small outline package
NVM	Non-volatile memory
OATS	Open area test sites
OMD	Output Multisynth Divider
OTP	One-time programmable memory
PC	Personal computer
PCB	Printed circuit board
PLL	Phase-locked loop
QFN	Quad-flat no-leads

RAM	Random access memory
RF	Radio frequency
SAC	Semi-anechoic chambers
SCL	Serial clock
SDA	Serial data
SMA	SubMiniature version A
SMD	Surface-mount device
USB	Universal serial bus
VCO	Voltage-controlled oscillator
XTAL	Crystal = quartz

1 INTRODUCTION

The reference radiator is known as a radiated reference signal source or for validation and verification of radiated measurement systems, such as:

- Open Area Test Sites (OATS)
- Fully Anechoic Rooms (FAR)
- Semi-Anechoic Chambers (SAC)

Daily site checks for ensuring the accuracy of measurement data can be done quickly, efficiently, and cost-effectively. Besides, the device is also used for test site comparisons.

The aim of this thesis is to design a reference radiator that can produce two output frequencies within the 12 MHz to 50 MHz range with a step size of 1 MHz. In addition, this device is integrated with one temperature sensor and three push buttons, one button to select the frequency output and other buttons to increase or decrease the frequencies to the desired value. All settings and adjustments are made based on a programmable clock generator Si5351A-B-GT with the Arduino system. The user can observe the output frequencies through the Arduino IDE's Serial Monitor.

The reference radiator is a compact device and can operate with a small power supply from a 9V battery or a 5V USB port. All device components and circuit boards are housed in an aluminum box to prevent radiation emissions and reduce unwanted reflections caused by external cabling. For radiated operation, an antenna can be attached to the output connector of the device.

Regarding the structure of this thesis, Chapter 2 lists the requirement specifications of the device including hardware and software. Then, the digital electronic radiation is discussed in Chapter 3, which models radiation emissions and describes the parameters on which radiation depends. Chapter 4 represents the basic Fourier Analysis of a square wave. Chapter 5 briefly introduces the structure and operating mechanism of the Si5351A-B-GT programmable clock generator. Before starting with the design of an embedded system for the reference radiator according to the goal mentioned above, a Si5351A clock generator breakout board from Adafruit was used for testing and analyzing the measurement results. Therefore, Chapter 6 introduces, wiring of the module to a microcontroller board, assembling to the aluminum box as well as controlling Si5351A by software. This chapter also covers the implementation and testing with this test version. Next, Chapter 7 details the design that has been extended and some features improved compared to the test version. Finally, Chapter 8 indicates the conclusion and evaluation of the product.

2 REQUIREMENTS SPECIFICATION

A reference radiator producing periodic square waves has been an important device for EMC laboratories. It helps to ensure the quality of measurement results, namely, for quick checks of antennas and test set-ups. This project aims to design a compact and low-cost reference radiator.

2.1 Hardware

The device uses Si5351A-B-GT with a crystal frequency is 25/27 MHz. The number of output frequency connectors is two (CLK0, CLK1) with 50 Ω impedance. The radiated frequency range of output connectors is 12MHz to 50MHz. The frequency step size is 1MHz.

An antenna is connected to the output connectors. Monopole antennas with a length of 1 meter are used. The frequency stability is <+/- 15ppm from -30°C to +85°C. The power supply is 9V battery and 5V USB port, ensure no power and cables affecting measurements. The temperature sensor range should be at least from -30°C to +85°C in 0.5°C increments. The ambient temperature is measured and informed constantly in Serial Monitor. The radiator dimensions are 12 cm x 8 cm.

The controls used are 1 switch to control the power battery, 1 push button (CLK) to select the frequency output (CLK0 or CLK1), and 2 push buttons (UP & DOWN) to increase and decrease the frequency of the selected output.

2.2 Software

The reference radiator is implemented on the Arduino system. The I2C protocol is used as a serial communication between a microcontroller with a temperature sensor and a clock generator.

The Arduino's Serial Monitor displays:

- The value of the ambient temperature.

- Two output frequencies are generated from frequency outputs (CLK0 & CLK1).

The device should be so easy to use that no special skills are needed from the user.

3 DIGITAL ELECTRONIC RADIATION

In today's digital world, electrical and electronic devices are being constantly developed and used more and more widely. During product design, consideration should be given to the potential of interference that digital devices may cause to each other. Therefore, electromagnetic compatibility (EMC) testing plays a very important role in the product design project. This testing helps to minimize the possibility of radiated or conducted emissions generated by the device affecting other electronic devices around it. Besides, EMC testing can also ensure that the device can function as expected around other sources of electromagnetic emissions.

Radiofrequency (RF) energy emitted from electronic devices can be classified into two main types: radiated emission and conducted emission. This chapter models the radiated emission, which is unintentional energy escaping from a device in the form of an electric, magnetic, or electromagnetic field, and describes the parameters associated with it.

Radiation from circuits can be either differential mode or common-mode radiation [1].

3.1 Differential-mode Radiation

Differential mode radiation is the result of the normal functioning of the circuit and the current circulating around the loops formed by the circuit conductors. It can be modeled as a small loop antenna carrying the interference current. A small loop antenna is made up of a ring of wire and any current in the wire produces the radiation surrounding the antenna [1].

In Figure 1, a current flows in a closed-loop, therefore this potential forms a small loop antenna and basically emission occurs.

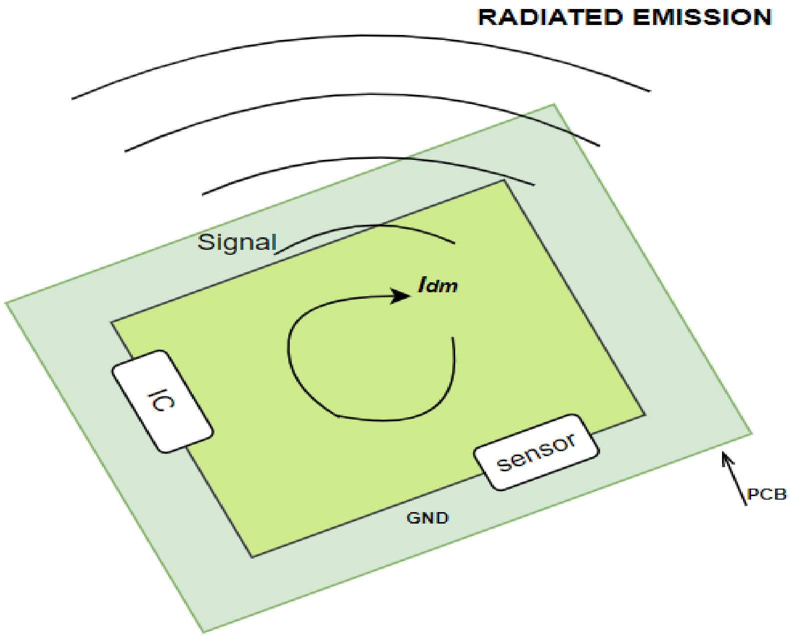

Figure 1. Differential mode radiation from PCB.

Under far-field, A (m^2) is the small loop area and I_{dm} (A) is the carrying current, the magnitude of electric field E (V/m) measured in free space with no reflecting surfaces at distance r (m) can be expressed in this equation [2]

$$E = 131.6 \times 10^{-16}(f^2.A.I_{dm})\left(\frac{1}{r}\right) sin\theta \quad (1)$$

Where f in Hz and θ is the angle between the viewing point and a perpendicular to the loop plane.

Based on equation (1), when the frequency f increased, the magnitude of electric field E is also increased. Therefore, the fundamental frequency must be used as low as possible to prevent radiation.

Furthermore, a small loop is one whose circumference is smaller than a quarter wavelength ($\lambda/4$) and the current in the loop is in phase everywhere. For large loops, the current is not entirely in phase and therefore part of the current can be subtracted from the global emission rather than added [1]. Hence, the large loops may minimize the emission. But, according to equation (1), when increasing the loop, the area increasea, and the emission also increases. The loop should be made as small as possible to reduce the radiated emission.

In fact, EMC measurements of radiation from electronic devices are performed in an open area, over a ground plane instead of free space.

For differential mode radiation occurring over a ground plane, the emission is double because of reflection from the ground plane. Where $\theta = 90$ degrees.

$$E = 263 \times 10^{-16}(f^2.A.I_{dm})\ (\frac{1}{r}) \quad (2)$$

In summary, differential mode radiation can be reduced by [1]

- Limiting the magnitude and frequency, or harmonic of the current.
- The loop area should be made as small as possible.

3.2 Common-mode Radiation

Common mode radiation is the result of undesired voltage drop and the result of parasitic in the circuit, causing some part of the circuit to be at a common mode potential above the true ground plane. It can be modeled as a monopole antenna with less than a quarter wavelength (λ/4) and is driven by a voltage. A monopole antenna is known as half of a dipole antenna, which is basically an open circuit, it can be the cable or PCB trace [4].

In Figure 2, two ground points are shown in a patch of the ground plane that has a potential difference, it creates a ground voltage V_N. The cable or PCB trace is linked from the ground plane creating the radiated emission.

Compared to the differential mode radiation, the common-mode radiation is more difficult to control and normally figures out the overall radiated emission execution of the product [3]. Because of the potential difference between the two ground points, it is much harder to cancel away the ground voltage.

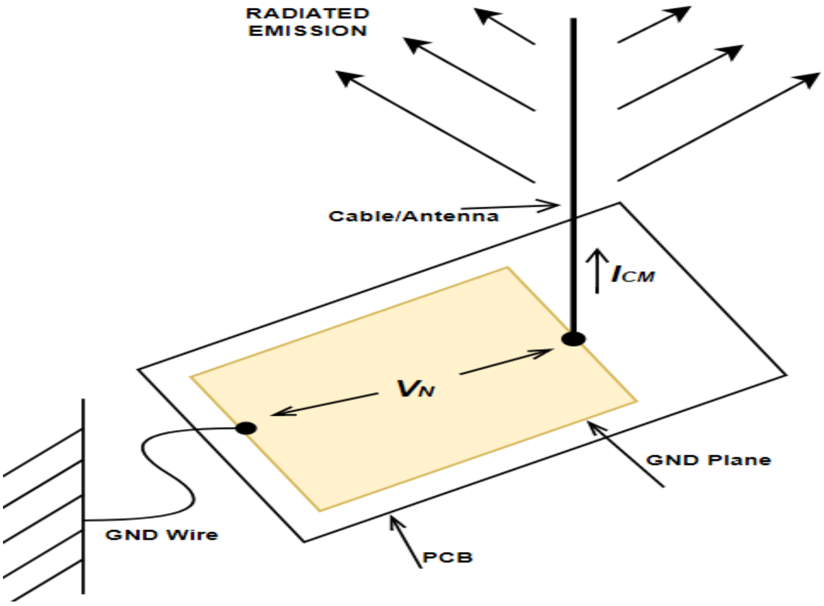

Figure 2. Common mode radiation from cables.

In the far-field, *l* (m) is the length of a short antenna linked to a ground plane, and the electric field strength *E* (V/m) at the distance of observant *r* (m) can be measured in the following equation [5]

$$E = \frac{4\pi \times 10^{-7} (f.l.I_{CM}) \sin\theta}{r} \quad (3)$$

I_{CM} (A) is the common mode current on the antenna. According to equation (3), the radiated emission is proportional to frequency, length of the antenna, and current I_{CM}.

Common mode radiation can be reduced by [3]

- Reducing the source voltage that drives the antenna.
- Providing a high impedance in series with the cable or PCB track.
- Preventing the current to the ground.
- Shielding.
- Preventing the cable links the ground plane.

4 FOURIER ANALYSIS OF SQUARE WAVE

Jean-Baptiste Joseph Fourier (1768-1830) was a French mathematician and Egyptologist who made enormous contributions to mathematics, physics, and astronautics. He not only developed the Fourier series (Trigonometric series) and its applications to problems of heat transfer and vibrations but also discovered the greenhouse effect [6].

There are two types of Fourier expansions: Fourier Series and Fourier Transform [7]. Fourier Series is a mathematical technique to generate any kind of periodic waveform with a combination of sines and cosines. Fourier Transform is used to transform periodic and non-periodic signals from the time domain to the frequency domain. It can also convert Fourier series in the frequency domain.

4.1 Properties of Sine and Cosine Functions

Considering the periodic function with period 2π, the sine function is an odd function that is symmetrical with the origin. Every function $sin(nx)$ has three properties: $sin(x + 2\pi) = sin(x)$, $sin(-x) = -sin(x)$, $sin(0) = sin(\pi) = 0$.

An infinite combination of sine waves is presented by the Fourier sine series [8]

$$S(x) = \sum_{n=1}^{\infty} b_n \, sin(nx) \tag{4}$$

Where, $S(x)$ is the sum of sines, b_n is the Fourier coefficients.

The cosine function is an even periodic function that is symmetrical with the y-axis. With $cos(-x) = cos(x)$, Cosine series [8]

$$C(x) = a_0 + \sum_{n=1}^{\infty} a_n \, cos(nx) \tag{5}$$

Where, $C(x)$ is the sum of cosines, a_0, a_n are the Fourier coefficients.

A periodic waveform is a kind of function that basically repeats itself after regular intervals of time.

$$f(x + T) = f(x) \tag{6}$$

Where x is any given point in time and T is the period of the waveform which is the interval between two repetitions. Its frequency is $f = 1/T$ and its angular frequency is $\omega = 2\pi/T$.

4.2 Fourier Series of Square Wave

The Fourier series is an infinite series that can be represented by a periodic function with period T. There are two common forms, "Trigonometric" and "Exponential" [9]. Assume $0 \leq x \leq T$, the function $f(x)$ can be written as exponential form:

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n e^{in\omega x} \tag{7}$$

where c_n are the Fourier coefficients of $f(x)$,

$$c_n = \frac{1}{T}\int_0^T f(x) e^{-in\omega x} dx \tag{8}$$

Trigonometric Form is as follows:

$$f(x) = a_0 + \sum_{n=1}^{\infty} a_n \, cos\left(\frac{2\pi nx}{T}\right) + \sum_{n=1}^{\infty} b_n \, sin\left(\frac{2\pi nx}{T}\right) \tag{9}$$

The function $f(x)$ has a DC term, cosine terms, and sine terms. The coefficients of the cosine terms are called the a terms and the coefficients of the sine terms are called the b terms where n is a number that goes from one to infinity.

To find the coefficients of the terms, that the following equations are used:

$$a_0 = \frac{1}{T}\int_0^T f(x) dx = average \tag{10}$$

$$a_n = \frac{2}{T}\int_0^T f(x) \cos\left(\frac{2\pi nx}{T}\right) dx, n \neq 0 \tag{11}$$

$$b_n = \frac{2}{T}\int_0^T f(x) \sin\left(\frac{2\pi nx}{T}\right) dx \tag{12}$$

Assumably, the periodic function for the case when T = 2π as in Figure 3. Based on the symmetric of the x-axis, the average value is zero so $a_0 = 0$. As mentioned in Chapter 4.1, if the signal was even, then $f(-x) = f(x)$, and if the signal is odd then $f(-x) = -f(x)$. Figure 3 has double symmetry showing that $f(-x) = -f(x)$, so it is clearly an odd signal. The conclusion is that if the signal is odd, it is not going to have even components, so all cosine terms are equal to zero.

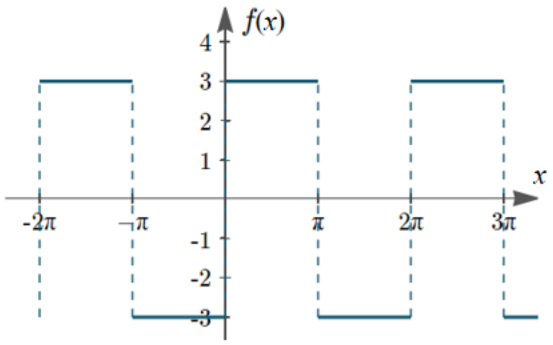

Figure 3. Graph of *f(x).*

Observing Figure 3, $f(x)$ is three when x is between zero and π seconds and it is minus three when x is between π and 2π seconds. Therefore, to find the coefficients of sine terms the equation (12) can be rewritten as:

$$b_n = \frac{2}{2\pi}\int_0^{\pi} 3\sin\left(\frac{2\pi nx}{2\pi}\right)dx + \frac{2}{2\pi}\int_{\pi}^{2\pi}(-3)\sin\left(\frac{2\pi nx}{2\pi}\right)dx$$

$$= \frac{3}{\pi}\int_0^{\pi}\sin(nx)dx - \frac{3}{\pi}\int_{\pi}^{2\pi}\sin(nx)dx$$

$$= \frac{3}{n\pi}\{[-\cos(n\pi) + \cos(0)] + [\cos(n2\pi) - \cos(n\pi)]\}$$

$$= \frac{3}{n\pi}\{[-\cos(n\pi) + 1] + [1 - \cos(n\pi)]\}\ (*)$$

Obviously, $\cos(n\pi)$ can be either 1 or -1. Therefore,

$$\begin{cases}\cos(n\pi) = -1, & n\ is\ odd\\ \cos(n\pi) = +1, & n\ is\ even\end{cases}$$

$$(*) \Leftrightarrow b_n = \begin{cases} \dfrac{12}{n\pi}, & n \; is \; odd \\ 0, & n \; is \; even \end{cases}$$

$$= \frac{12}{n\pi}\left(\frac{1 + (-1)^{n+1}}{2}\right)$$

Finally, the Fourier series of the square wave can be expressed as:

$$(9) \Leftrightarrow f(x) = \frac{12}{n\pi}\left(\frac{1 + (-1)^{n+1}}{2}\right) sin(nx)$$

$$b_1 = \frac{12}{n\pi} \quad , \quad b_4 = 0$$

$$b_2 = 0 \quad , \quad b_5 = \frac{12}{5\pi}$$

$$b_3 = \frac{12}{3\pi} = \frac{4}{\pi} \quad , \quad b_6 = 0$$

Because $a_0 = a_n = 0$, the amplitude spectrum $c_n = \sqrt{{a_n}^2 + {b_n}^2} = b_n$. From the values of $c_1 \; to \; c_6$, we got a spectral sketch shown in Figure 4. The coefficients show that a square wave is composed of only its odd harmonics.

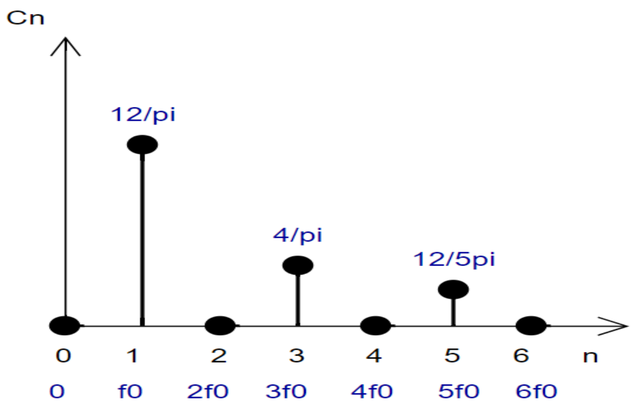

Figure 4. Spectral sketch of the signal.

4.3 Fourier Transform of Square Wave

Fourier Transform is a mathematical technique for transforming a time function $f(x)$ to a frequency function $F(\omega_n)$. It is firmly related to the Fourier Series.

Hence, the function of time $f(x)$ can be represented by a Fourier series as Equation (7).

Similarly, c_n is as equation (8). It can be rewritten as follows:

$$c_n T = \int_0^T f(x) e^{-in\omega x} dx$$

Considering $\omega_n = n\omega$ with $-\infty \leq n \leq +\infty$ and $F(\omega_n) = c_n T$ so, the Forward Fourier Transform is expressed as [10]:

$$F(\omega_n) = \int_{-\infty}^{+\infty} f(x) e^{-i\omega x} dx \quad (13)$$

Likewise, the Inverse Fourier Transform can be derived as follows:

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n e^{in\omega x} = \sum_{n=-\infty}^{+\infty} T c_n e^{in\omega x} \frac{1}{T}$$

$$f(x) = \sum_{n=-\infty}^{+\infty} F(\omega_n) e^{i\omega x} \frac{d\omega}{2\pi} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega_n) e^{i\omega x} d\omega \quad (14)$$

The impulses in every harmonic contained in the Fourier series expansion make up the Fourier Transform of a periodic square wave. Therefore, the Fourier Transform is actually impulses [11].

Considering the Fourier domain function as an impulse at $\omega_n = \omega$ and $F(\omega_n) = \delta(\omega_n - \omega)$,

$$(14) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \delta(\omega_n - \omega) e^{i\omega x} d\omega = \frac{1}{2\pi} e^{i\omega x}$$

From this result, the complex exponential forms a Fourier Transform pair is

$$e^{i\omega x} \overset{F}{\leftrightarrow} 2\pi\delta(\omega_n - \omega) \quad (15)$$

It can be used to find the Fourier Transform of any periodic function that is represented by the Fourier Series. This is the relationship between the Fourier Series and the Fourier Transform for a periodic function.

If a periodic function $f(x)$ and the Fourier Series of it is:

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n e^{in\omega x}$$

Then the Fourier Transform of this function will be:

$$F(\omega_n) = 2\pi \sum_{n=-\infty}^{+\infty} c_n \delta(\omega_n - \mathrm{n}\omega) \quad (16)$$

Considering the periodic function, $f(x)$ with period T, the coefficients of the Fourier Series, c_n relating to the Fourier Transform, $F(\omega_n)$:

$$c_n = \frac{1}{T} F(\omega_n) \quad (17)$$

Summarily, both the Fourier Series and the Fourier Transform have many applications. As mentioned above, the Fourier Series is a branch of the Fourier analysis, it decomposes a periodic function as a linear combination of sines and cosines. Initially, the Fourier Series was developed to solve heat equations but then it is used to solve numerous problems in many fields such as electrical engineering, optics, and vibration analysis. It is very useful in harmonic analysis with periodic functions and provides motivation for the Fourier Transform [12].

The Fourier Transform represents a signal in the frequency domain and time domain. It can be applied to both aperiodic and periodic signals. The Fourier Transform is used to solve differential equations and has application in nuclear magnetic resonance [12]. The Fast Fourier Transform (FFT) is known as a mathematical method for doing the process of Fourier Transform.

5 I²C - PROGRAMMABLE CLOCK GENERATOR

I²C programmable clock generators are also known as I²C programmable timing devices that can be set to provide timing pulses at a specified frequency through the I²C bus (SDA and SCL). Si5351 is a versatile I²C configurable clock generator from the company Skyworks. There are three versions of the Si5351 (A, B, and C) [13]. Version A (Si5351A) is considered to be used to build the reference radiator.

5.1 Si5351A

Si5351A is well-suited for replacing crystals and crystal oscillators by using an internal oscillator based on a PLL and high-resolution MultiSynth fractional divider architecture, it can generate up to 8 free-running clocks [13].

Si5351A has three package types:

10-MSOP: 3 outputs.

16-QFN (3x3 mm): 4 outputs.

20-QFN (4x4 mm): 8 outputs.

5.1.1 Functional Description

Si5351A consists of an input stage, two stages of synthesis, and an output stage. In the input stage, an external crystal (XTAL) as a reference is connected to an internal oscillator through pin XA and pin XB, as shown in Figure 5. In the first stage of the synthesis, the free-running reference from the output of the oscillator will be provided to one or both PLLs to multiply the references to a high-frequency intermediate clock. In the second stage, the MultiSynth dividers are used to generate the desired output frequencies [13].

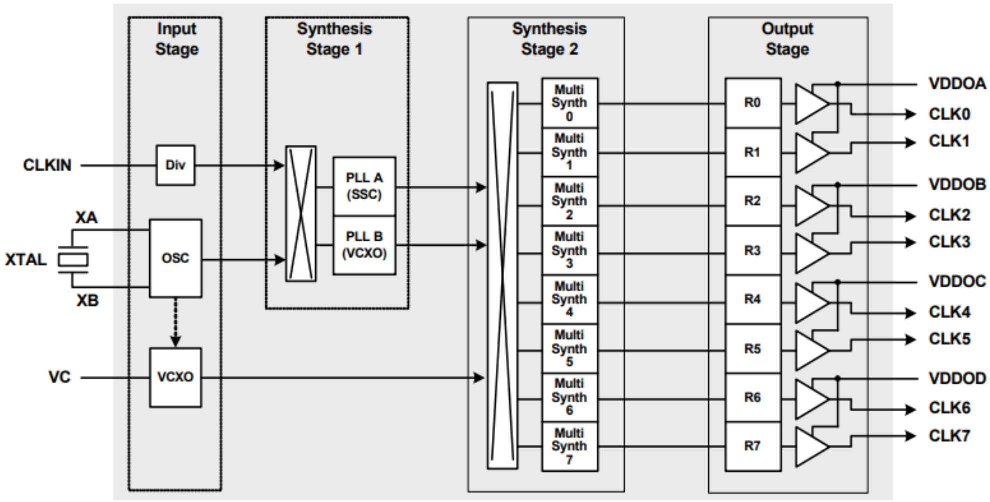

Figure 5. Si5351 Block Diagram [14].

5.1.2 Configuring the Si5351A through I²C Interface

The I²C interface can operate in three modes:

- Slave mode with 7-bit addressing: contains 6 bits fixed address and 1 LSB bit (A0 pin), however, the Si5351A does not have this pin, so A0 has been defaulted to 0 and the corresponding address would be set 0x60, as shown in Figure 6.
- Standard mode 100 kbps
- Fast mode 400 kbps

In addition, the I²C interface can support continuous data transmission with automatic address increments. A bidirectional serial data line (SDA) and a serial clock input (SCL) are the two lines used in the I2C protocol, they are also called the I2C bus. Through an external pull-up, they are connected to the VDD supply.

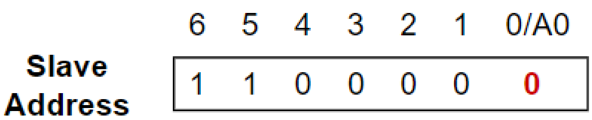

Figure 6. Si5351A I2C Slave address.

The Si5351A is configured via the I2C bus by reading and writing to registers in the RAM space. According to the I^2C specification [15], the data transfers follow the first byte or 8 bits format. A write command contains 8 bits, including 7 bits slave address and 1 write bit. This is shown in Figure 7.

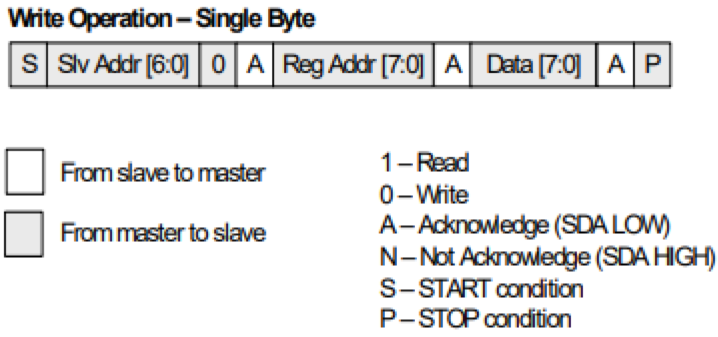

Figure 7. Write Operation [13].

With the read operation, it is performed in two stages as shown in Figure 8, and they are:

- Set the registered address by using a data write.
- Retrieve the data from the set address by performing a data read.

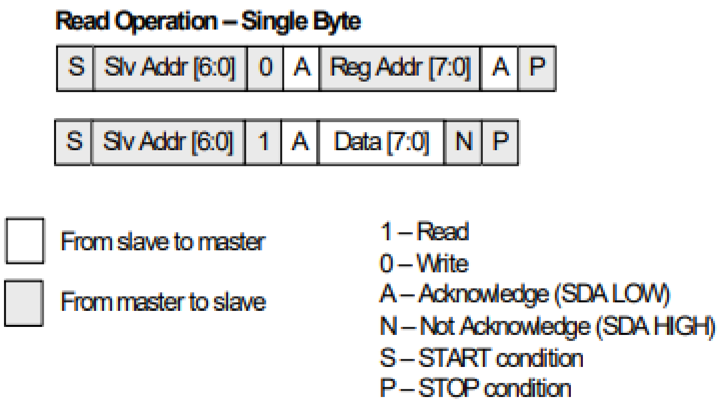

Figure 8. Read Operation [13].

The default configuration of Si5351A is stored in a one-time programmable memory (OTP) as known as non-volatile memory (NVM) and the contents of NVM would be copied into RAM during a power cycle as shown in Figure 9 [13].

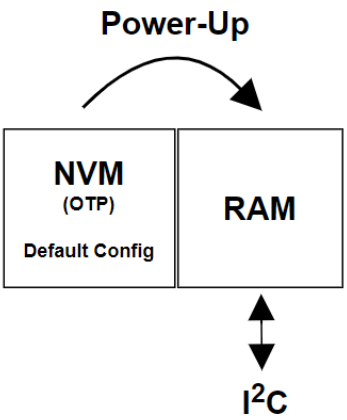

Figure 9. Si5351A Memory Configuration

The Si5351A can be programmed through an I2C bus by following the procedure shown in Figure 10.

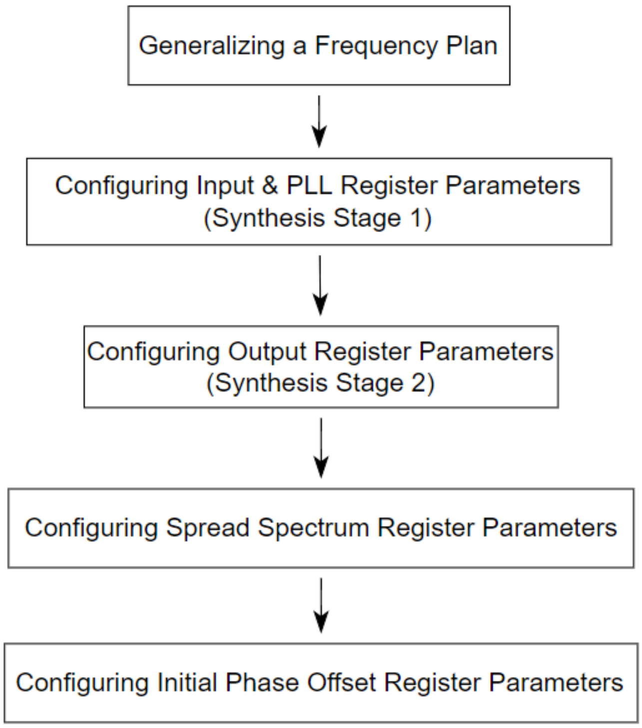

Figure 10. Programming Steps.

The Si5351A contains two PLLs (PLLA and PLLB), each one generates a VCO frequency which can be from 600 to 900 MHz by using the Feedback MultiSynth Divider (FMD). Then, these frequencies are divided down by Output MultiSynth Dividers (OMD) to generate frequencies from 500kHz to 200MHz. Additionally, output R dividers can divide the generated frequency as low as 2.5kHz, R is only relevant if the output frequency is smaller than 500kHz [14]. The VCO and output frequencies can be expressed with the following equations.

$$f_{out} = \frac{f_{VCO}}{OMD \times R} \tag{18}$$

$$f_{VCO} = f_{XTAL} \times FMD \tag{19}$$

where f_{XTAL} (Hz) is the reference frequency of external crystal and f_{out} (Hz) is the desired output frequency. Each *R* can be set to 1, 2, 4, 8, …128.

FMD and OMD are represented as a fractional number, but the variables of FMD are not the same as those for the OMD.

$$\left(a + \frac{b}{c}\right)$$

This fractional ratio of FMD is between 15 + 0/1,048,575 and 90. It is represented in the Si5351A register space using the equations.

$$MSNx\_P1[17:0] = 128 \times a + Floor\left(128 \times \frac{b}{c}\right) - 512 \tag{20}$$

$$MSNx\_P2[19:0] = 128 \times b - c + Floor\left(128 \times \frac{b}{c}\right) \tag{21}$$

$$MSNx\_P3[19:0] = c \tag{22}$$

where *x* could be A or B from PLLA and PLLB. For example, MSNA and MSNB.

The fractional number of OMD can be from 8 + 1/1,048,575 to 2048. It is represented in the Si5351A register space using the equations when the output frequency is smaller or equal to 150 MHz.

$$MSx\_P1[17:0] = 128 \times a + Floor\left(128 \times \frac{b}{c}\right) - 512 \tag{23}$$

$$MSx\_P2[19:0] = 128 \times b - c + Floor\left(128 \times \frac{b}{c}\right) \tag{24}$$

$$MSx\_P3[19:0] = c \tag{25}$$

To avoid violating math principles, *c* is never zero.

5.2 Si5351A-B-GT

The Si5351A-B-GT means Si5351 with Crystal In, product revision B, 10-MSOP package, and it consists of three output dividers.

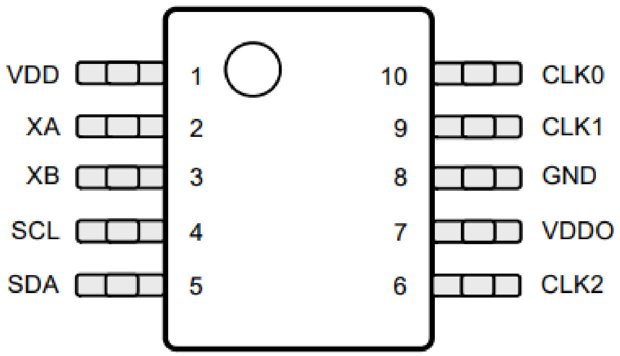

Figure 11. Si5351A 10-MSOP [13].

Table 1. Si5351A 10-MSOP Pin Descriptions.

Pin Name	Pin Number **10-MSOP**	**Function**
XA	2	Input pin for an external crystal.
XB	3	Input pin for an external crystal.
CLK0	10	Output clock 0.
CLK1	9	Output clock 1.
CLK2	6	Output clock 2.
SCL	4	Serial clock input for the I2C bus.
SDA	5	Serial data input for the I2C bus.
VDD	1	Core voltage supply pin. (2.5 or 3.3V)
VDDO	7	Output voltage supply pin for output clocks. (1.8, 2.5, or 3.3V)
GND	8	Ground.

6 TEST VERSION WITH SI5351A BREAKOUT BOARD

6.1 Introduction about Si5351A Breakout Board from Adafruit

The breakout board from Adafruit Industries has a 25MHz crystal reference, internal PLL, and MultiSynth dividers. It can generate the frequency from < 8KHz up to 150+ MHz. This module has three outputs, each one can have another frequency. It also includes a male header and SMA connector. There are a 3.3V low-dropout regulator and a level shifting circuitry on I2C lines [16].

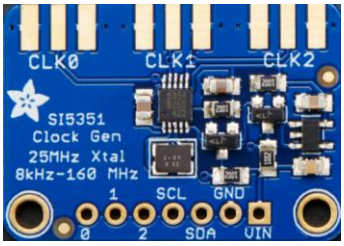

Figure 12. Adafruit Si5351A Breakout Board [15].

The dimensions of the breakout board are 31mm x 22mm x 2mm / 1.22" x .87" x .08" and it weights are 2.5 grams.

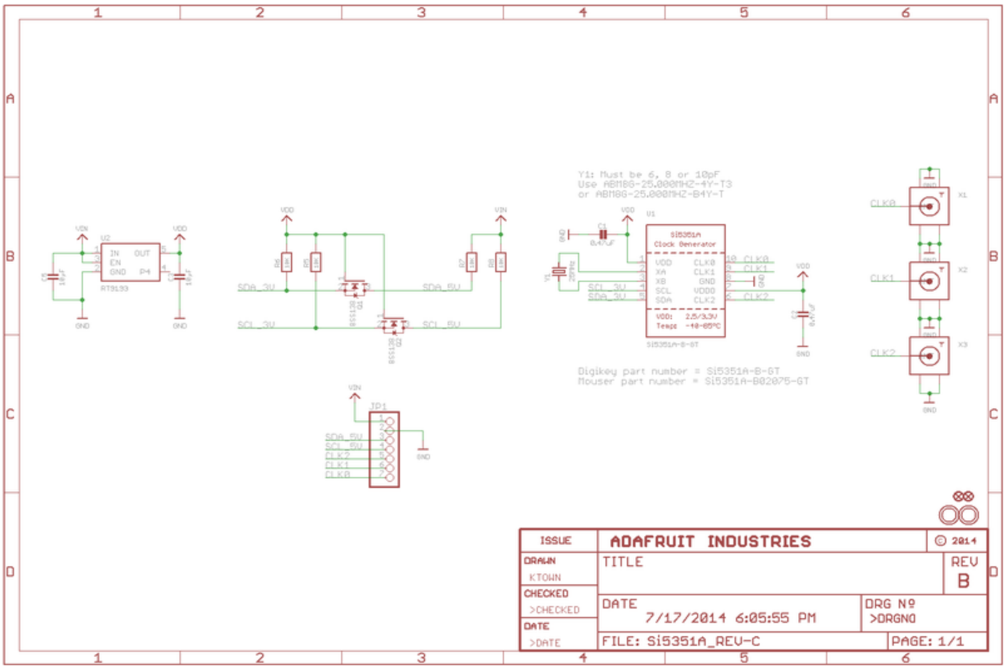

Figure 13. Schematic of Adafruit Si5351A Breakout Board [16].

6.2 Test Version Design

The test version is designed based on the block diagram shown in Figure 14. A microcontroller would be connected to a clock generator, namely, an Adafruit Si5351A Breakout board.

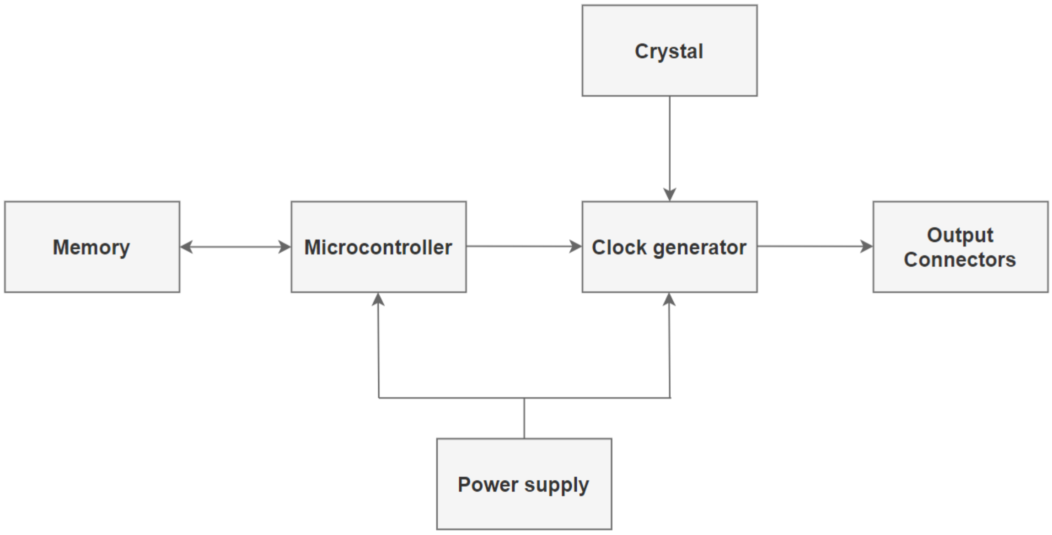

Figure 14. Block diagram of Test Version.

6.2.1 Component List

The electronic and mechanical components were used for the test version, listed in Table 2. All these components were kindly provided by VAMK University of Applied Sciences.

Table 2. Component list of test version.

Name	Description
Clock Generator	1x Adafruit Si5351A Breakout Board
Battery	1x VARTA 550 mAh, 9V
Microcontroller board	1x Arduino Uno Rev3
Switch	1x Power switch
Output Connector (51K507-802N5)	2x Female BNC RF bulkhead jack
Aluminum box	1x (8 cm x 12 cm)
Soldering tools and accessories	

The microcontroller board based on the ATmega328P called Arduino UNO R3 [17] was used to implement and control the reference radiator. Its dimensions are 53.4 x 68.6 mm and its weighs only 25grams. It features a USB-B connector, input voltage pin for power from 7 to 12V, and barrel plug connector works well with a 9V battery.

The 9V battery features a rectangular prism shape and has two types: non-rechargeable (primary) and rechargeable (secondary). The chemistries of primary can be Alkaline, Carbon-Zinc, Lithium. The chemistries of the secondary can be Nickel-cadmium, Nickel-metal hydride, and Lithium-ion [18]. These chemistries give different capacities, so their performance and application vary greatly. Based on the current of the device and the capacity of the battery, we can estimate how long a 9V battery will last. In this case, the reference radiator just used a 9V battery with 550mAh of capacity, so the device is just used for several hours. Therefore, the battery should be replaced with another better battery. However, the device can also use a combination of AA cells to add up to 9V. Some reports show that the AA cells provide a longer operating life when compared to a 9V battery.

Two output connectors are 50Ω BNC RF bulkhead jack coaxial connectors from Rosenberger [19]. The material of their body is brass with a white bronze over silver plating. They are suitable for use with RF coaxial cables.

6.2.2 Assembling the Test Version

The first step of this process was to define the position of components on and into the box. Then, using the drilling machine (Figure 15) to drill the holes for output connectors and power switch.

Figure 15. Drilling machine.

The second step was to connect the Adafruit clock generator Si5351A with the microcontroller board as follows:

Adafruit Si5351A		Arduino Uno R3
SDA	⇔	A4
SCL		A5
Vin		3.3V
GND		GND

The next step was to wire the battery, the power switch, and the microcontroller board together (Figure 16).

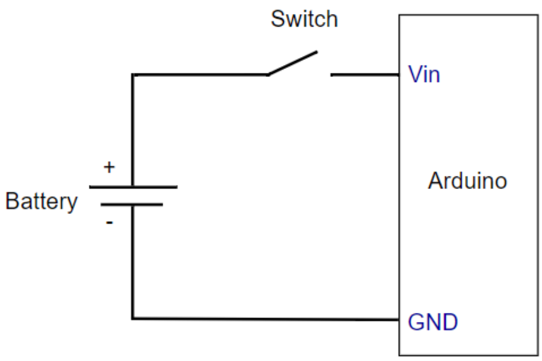

Figure 16. Circuit diagram of battery, switch, and Arduino.

Then, the output of the Adafruit board was wired with RF connectors. The electrical wires should have good quality as shown in Figure 17.

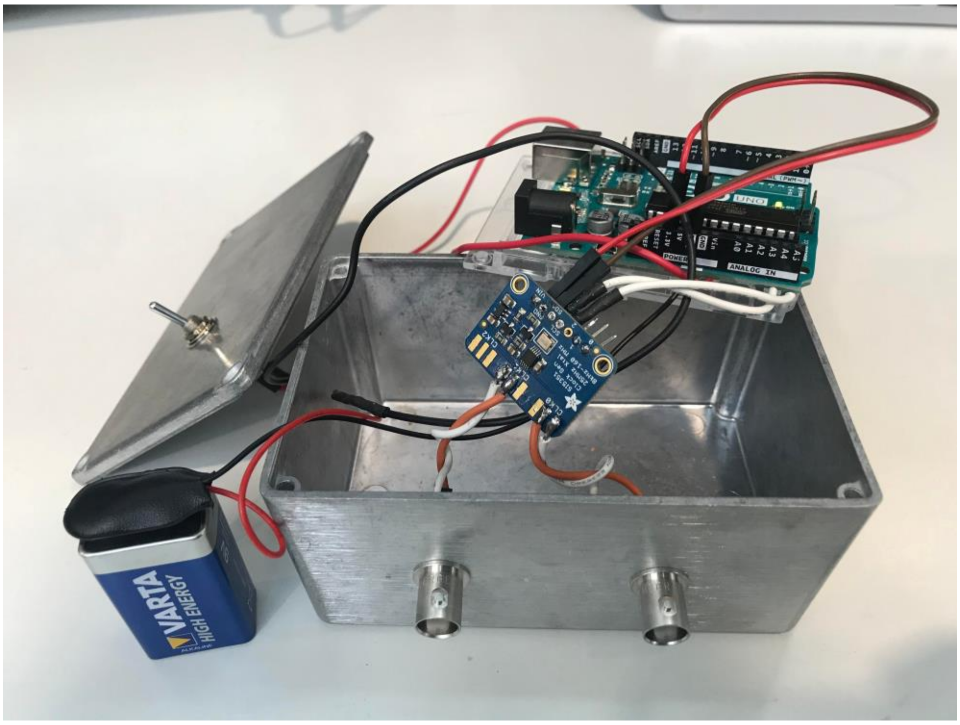

Figure 17. Test version with Si5351A breakout board.

6.3 Configuring and Controlling Si5351A

As mentioned, the Arduino module was used to configure and control the clock generator, it is considerably cheaper and easier to use than other hardware platforms. Its programming environment is Arduino IDE. Arduino supports a language called Arduino language or Arduino programming language which is based upon the Wiring/Processing platform. Basically, it is a framework built on top of C++. This platform is open-source, so users do not need to purchase a license. Furthermore, it is easy to use for non-engineers uploading it to Arduino board as well as using the available libraries. Arduino boards from 8 bits running at 16 MHz up to 32 bits ARM processors at 84MHz for many applications. In this section, the Si5351A is fully programmed via the I2C bus, and the software does not use Arduino libraries to generate the desired frequency. Instead, it is initialized and set through the following steps:

Based on the register map summary for 10-MSOP, all CLK output drivers are disabled and the internal capacitor set to 8pF. At synthesis stage 1, the value of the crystal is set to 25MHz and PLLA to 900MHz for both two outputs.

From the equation (19), we get the value of *FMD*:

$$FMD = \frac{f_{VCO}}{f_{XTAL}} = \frac{900MHz}{25MHz} = 36MHz = \left(a + \frac{b}{c}\right)$$

$$a = \frac{f_{VCO}}{f_{XTAL}} = \frac{900MHz}{25MHz} = 36\text{MHz} \;\; => \text{Integer part}$$

$$b = f_{VCO} \,\%\, f_{XTAL} = 0 \;\; => \;\text{Fractional part}$$

$$c = f_{XTAL} = 25MHz \;\; => \;\; c \neq 0$$

After defining the value of *a, b, and c*, MSNA_P1 is calculated based on the equation (20). Because MSNA_P1 can generate alone 900MHz, MSNA_P2 and MSNA_P3 are not needed.

The next step is triggered at synthesis stage 2. It uses high-resolution MultiSynth fractional dividers to get the desired output frequencies. Similar to the first stage, the selected PLL source is used to calculate the value of *OMD* based on the equation (18). Assumably, the required output frequencies are 12MHz and R dividers are set to 1, *OMD* is:

$$OMD = \frac{f_{VCO}}{f_{out}} = \frac{900MHz}{12MHz} = 75MHz = \left(a + \frac{b}{c}\right)$$

$$a = \frac{f_{VCO}}{f_{out}} = \frac{900MHz}{12MHz} = 75\text{MHz} \;\; => \text{Integer part}$$

$$b = f_{VCO} \,\%\, f_{out} = 0 \;\; => \;\text{Fractional part}$$

$$c = f_{out} = 12MHz \;\; => \;\; c \neq 0$$

From that, the equations (23), (24), and (25) are used to represent *a, b,* and *c* in the Si5351A register space.

Finally, the desired frequencies are sent to the CLK0 and CLK1 of Si5351A by using I^2C bus. Then, they are represented on Serial monitor as shown in Figure 18.

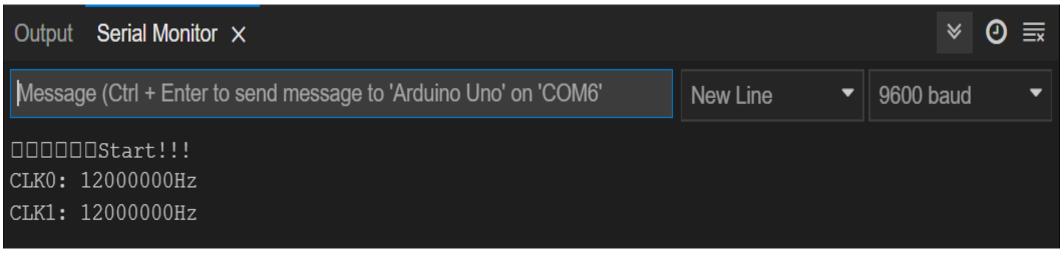

Figure 18. Test version results on Serial monitor.

In summary, writing a configuration to RAM of Si5351A need some essential functions, such as a function to initialize Si5351A, a function to set the desired frequency, and two functions to read and write I^2C register.

6.4 Measurements and Results

Analyzing the frequency content of the device is a critical step in ensuring that the device work properly. The measurements of the output frequency are done in two methods: a direct measurement with an oscilloscope and radiated emission measurement with a 1-meter monopole antenna in an anechoic chamber.

6.4.1 Measurements by Using an Oscilloscope

Normally, the design often appears with noise frequencies which can end up causing the device to malfunction or affecting its accuracy. To detect the noise frequencies, an oscilloscope with an FFT function can be considered to be used.

The Si5351A is a device for the digital domain so the output CLKs will be periodic square waves which are represented by a series of basic functions, as seen in Figure 19. It shows some amount of ringing; however still exhibits a smooth profile.

As mentioned in Chapter 4.3, FFT is basically a math operation that splits up the signal in the time domain, translates that information into the frequency domain, and displays it on the oscilloscope in a graph showing amplitude versus frequency.

Figure 19. Signal waveform of CLK0 at 12MHz and CLK1 at 18MHz.

For example, starting with CLK0 at 12MHz and CLK1 at 18MHz then directly connecting to an oscilloscope, measurement results are depicted in Figure 20 and Figure 21. Observably, there has the appearance both of odd and even harmonics as well as other ones.

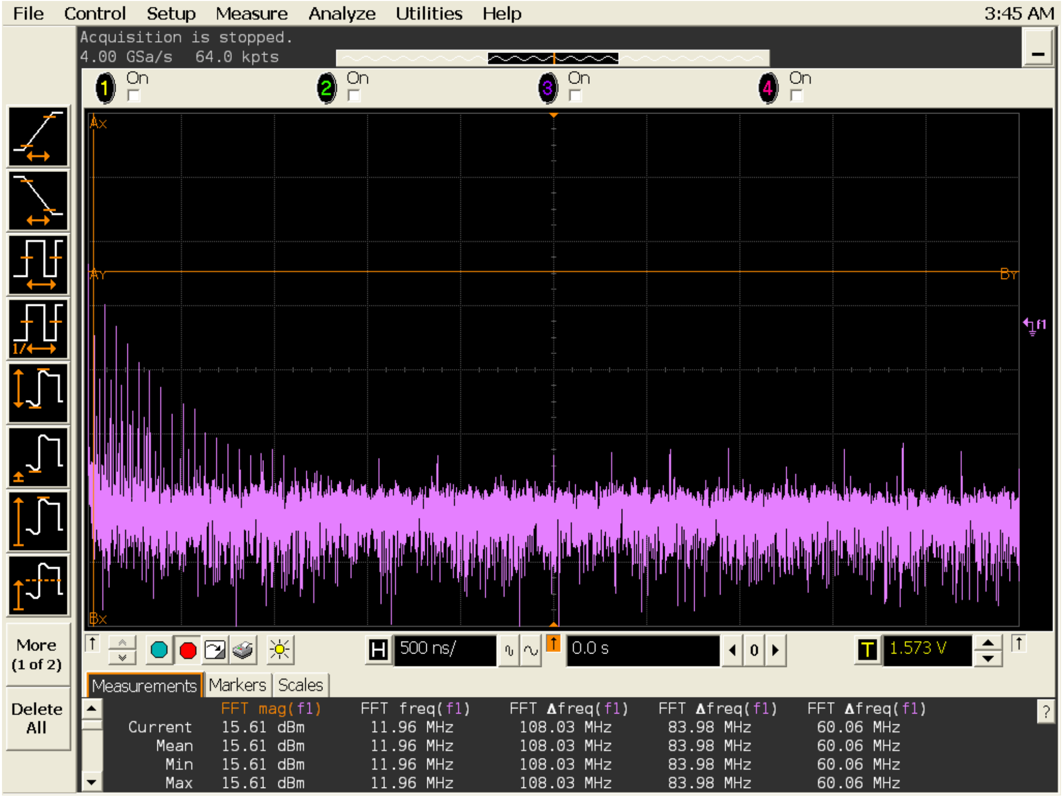

Figure 20. FFT implementation for CLK0 at 12MHz.

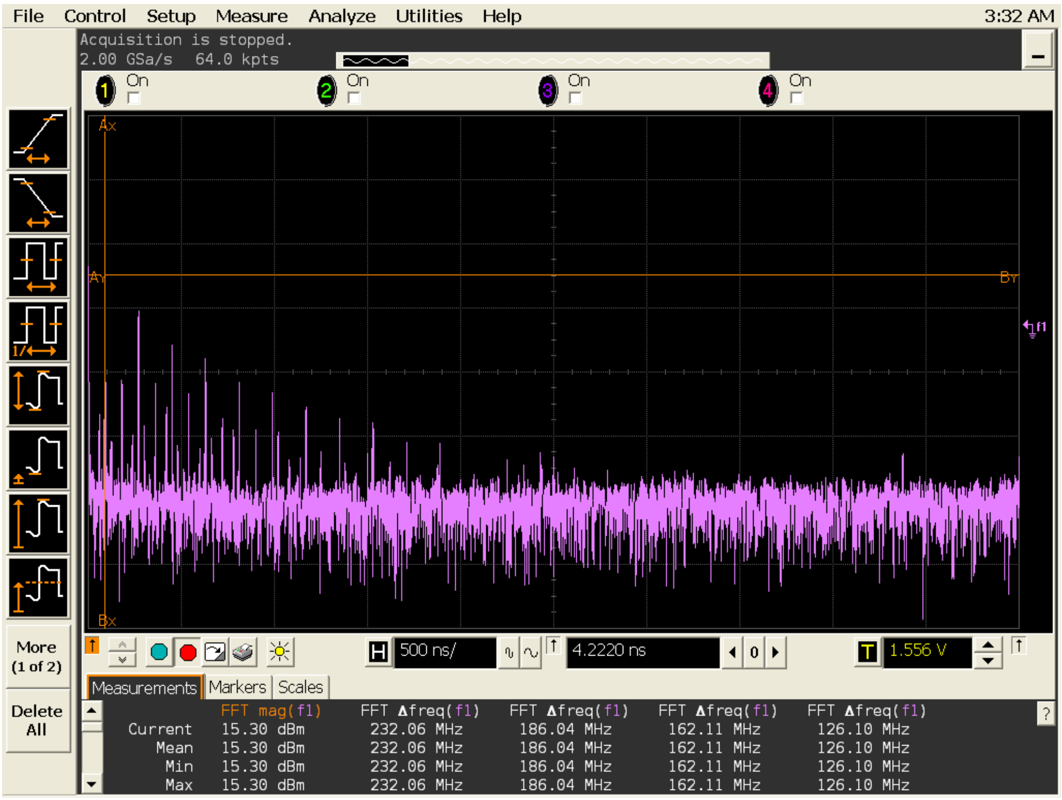

Figure 21. FFT implementation for CLK1 at 18MHz.

6.4.2 Measurements in a Fully Anechoic Chamber

A Fully Anechoic Chamber (FAC) is known as an echo-free room, absorbers are placed on six sides of the chamber to help to absorb reflections of either sound or electromagnetic waves completely [20].

To make sure this test version is operating accurately, a radiated test was performed to ensure emissions emanating from the Equipment Under Test (EUT) or Device Under Test (DUT) are suitable for the expected results [21].

In setting up the radiated testing, the test version was rested on a non-conductive table whose diameter is approximately 0.8 meters and can rotate 360 degrees. As mentioned in Chapter 3, the common-mode radiation can be modeled using a monopole antenna, so the two output connectors of the test version were connected to the monopole antennas at a 1-meter length. The distance from the turntable to the Bilog antenna was 3 meters. This antenna was supported by an antenna mast and could be hoisted up and down from 1 to 4 meters, as shown in Figure 22.

Figure 22. The set-up for Radiated testing in FAC.

After that, the antenna was connected to an EMI receiver. When the device released the electromagnetic wave, the receiver measured the field strength (dBuV/m) captured from the antenna. After capturing the data, this data was routed to a PC. The PC did some calculations and analytics.

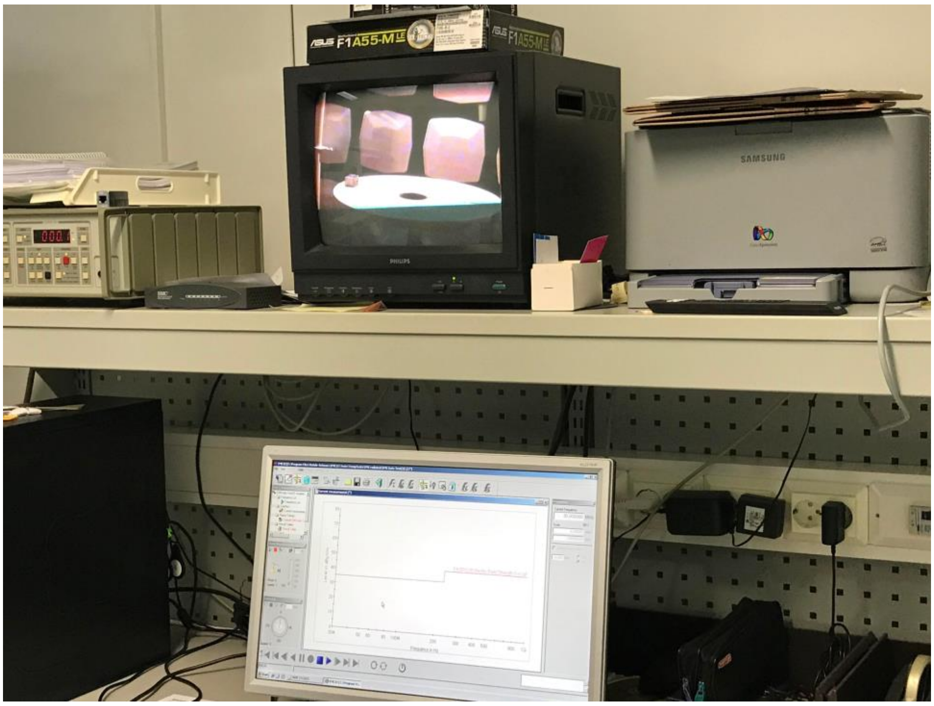

Figure 23. PC and monitor outside FAC.

With a normal device, the emissions released from DUT should comply with the applicable limits, but the test version is a reference radiator, which acts as an EMI source. Hence, its radiated emission could be over the limit.

In the first measurement, 12MHz output connector CLK0 was connected to a 1-meter wire which acted as a monopole antenna role. As shown Figure 10, the radiated electric field strength was captured in FAC using a Bilog antenna with horizontal polarization and a frequency range from 30MHz to 1GHz. This figure shows the harmonics of a 12MHz clock signal, the parts that project upwards as a needle are the harmonics. The odd and even harmonics have a different trend. Besides, the bottom section around 25dB or lower indicates the background noise.

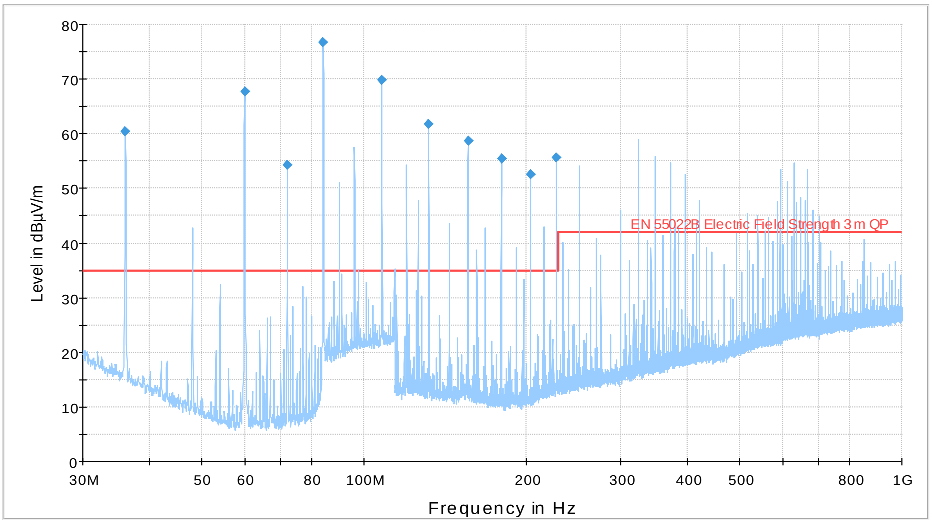

Figure 24. Radiated E-field strength in FAC of CLK0 at 12MHz.

Final Measurement Detector 1

Table 3. Final measurement detector of CLK0.

Frequency (MHz)	QuasiPeak (dBµV/m)	Meas. Time (ms)	Bandwidth (kHz)	Antenna height (cm)	Polarity	Turntable position (deg)	Corr. (dB)	Margin (dB)	Limit (dBµV/m)
36.000000	60.4	1000.000	120.000	155.0	H	0.0	14.4	-25.4	35.0
60.000000	67.6	1000.000	120.000	155.0	H	0.0	5.6	-32.6	35.0
72.000000	54.4	1000.000	120.000	155.0	H	0.0	6.0	-19.4	35.0
84.000000	76.7	1000.000	120.000	155.0	H	0.0	7.9	-41.7	35.0
108.000000	69.9	1000.000	120.000	155.0	H	0.0	11.4	-34.9	35.0
132.000000	61.9	1000.000	120.000	155.0	H	0.0	11.6	-26.9	35.0
156.000000	58.7	1000.000	120.000	155.0	H	0.0	10.5	-23.7	35.0
180.000000	55.4	1000.000	120.000	155.0	H	0.0	9.1	-20.4	35.0
204.000000	52.6	1000.000	120.000	155.0	H	0.0	9.9	-17.6	35.0
228.000000	55.6	1000.000	120.000	155.0	H	0.0	11.5	-20.6	35.0

A square wave can be expressed as a sum of fundamental sine waves which is also known as harmonics. The harmonic order is multiple for the fundamental wave (12MHz). As mentioned above, the frequency range is only from 30 to 1000MHz, so Figure 24 does not show the fundamental wave or the first harmonic order (12MHz). In Table 3, the first frequency is 36MHz, number 36 is a multiple of number 12 with 3 as a factor, so 36MHz is the value of the third harmonic order. In addition, the value of the second harmonic is 24MHz. 36 minus 24 is 12, so the frequency interval is exactly the same as that of the fundamental wave.

A square wave is an infinite series of odd harmonics, and it is an ideal wave when having no even harmonics. Based on the data in Table 3, the odd harmonics are summarized in Table 4.

Table 4. Odd harmonics present when CLK0 tested.

Odd Harmonic	Even Harmonic	Frequency at 12MHz
3		36.000000
5		60.000000
	6	72.000000
7		84.000000
9		108.000000
11		132.000000
13		156.000000
15		180.000000
17		204.000000
19		228.000000

The result of the table shows CLK0 has nine odd harmonics and only one even harmonic per ten measured harmonics and no frequencies other than a multiple of 12MHz.

Similarly, the radiated electric field strength of CLK1 at 18MHz is shown in Figure 25. The first harmonic order (18MHz) is not observed, but the first frequency in Table 5 is 54MHz. Number 54 is a multiple of number 18 with 3 as a factor, so it is accurately the third harmonic order. The frequency interval is 18MHz, the same as the fundamental wave.

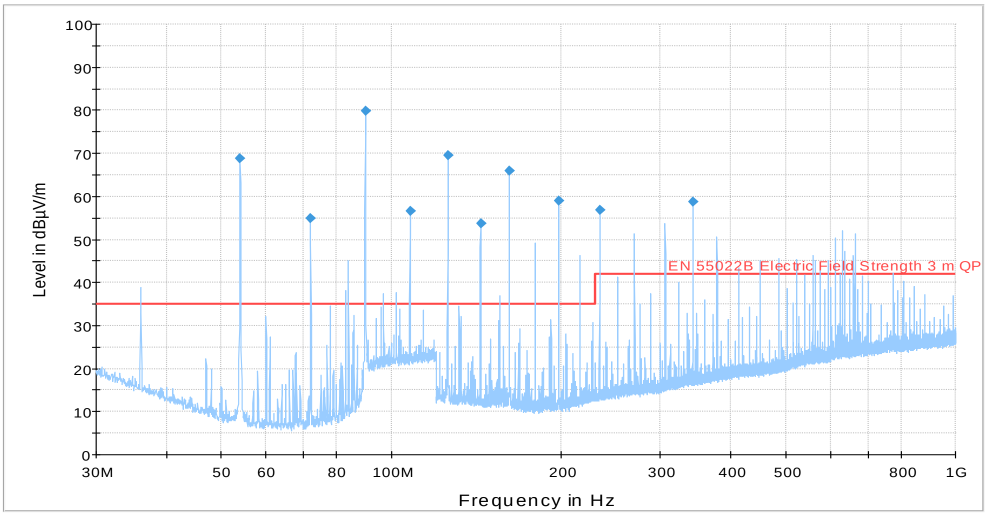

Figure 25. Radiated E-field strength in FAC of CLK1 at 18MHz.

Final Measurement Detector 2

Table 5. Final measurement detector of CLK1.

Frequency (MHz)	QuasiPeak (dBµV/m)	Meas. Time (ms)	Bandwidth (kHz)	Antenna height (cm)	Polarity	Turntable position (deg)	Corr. (dB)	Margin (dB)	Limit (dBµV/m)
54.000000	68.7	1000.000	120.000	155.0	H	0.0	6.8	-33.7	35.0
72.000000	55.0	1000.000	120.000	155.0	H	0.0	6.0	-20.0	35.0
90.000000	79.8	1000.000	120.000	155.0	H	0.0	9.1	-44.8	35.0
108.000000	56.6	1000.000	120.000	155.0	H	0.0	11.4	-21.6	35.0
126.000000	69.5	1000.000	120.000	155.0	H	0.0	11.8	-34.5	35.0
144.000000	53.6	1000.000	120.000	155.0	H	0.0	11.0	-18.6	35.0
162.000000	66.0	1000.000	120.000	155.0	H	0.0	10.2	-31.0	35.0
198.000000	59.0	1000.000	120.000	155.0	H	0.0	9.5	-24.0	35.0
234.000000	57.0	1000.000	120.000	155.0	H	0.0	11.9	-15.0	42.0
342.000000	58.7	1000.000	120.000	155.0	H	0.0	15.3	-16.7	42.0

Based on the data in Table 5, the odd harmonics are summarized in the following Table 6.

Table 6. Odd harmonics present when CLK1 tested.

Odd Harmonic	Even Harmonic	Frequency at 18MHz
3		54.000000
	4	72.000000
5		90.000000
	6	108.000000
7		126.000000
	8	144.000000
9		162.000000
11		198.000000
13		234.000000
15		
17		
19		342.000000

The interference source is generated when using only 18MHz CLK1, which is more unstable than CKL0. The result of the table shows CLK1 has seven odd harmonics and three even harmonics per ten measured harmonics and has no frequencies other than a multiple of 18MHz.

The radiated electric field strength in FAC for both CLK0 & CLK1 is shown in Figure 26 and Table 7 listed the results for both CLK0 and CLK1.

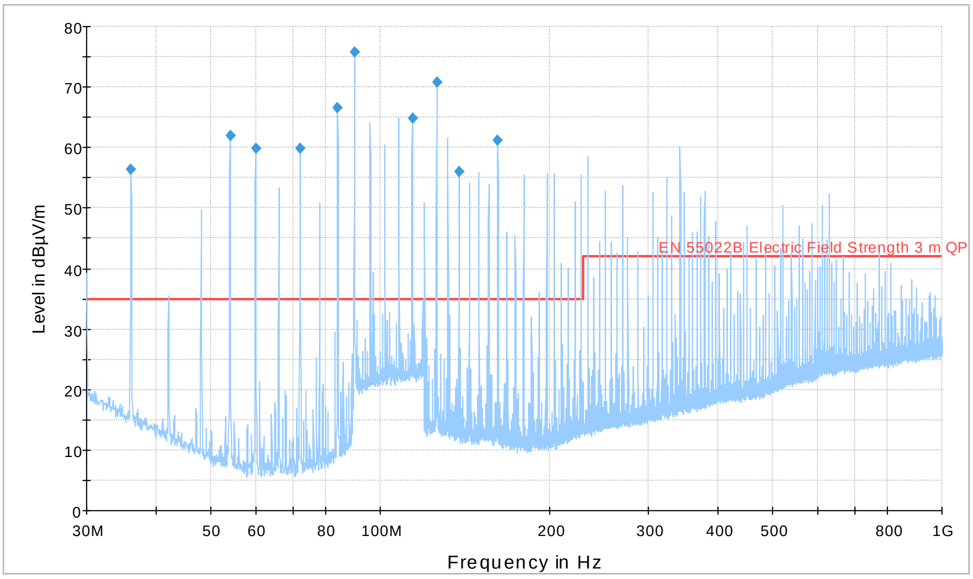

Figure 26. Radiated E-field strength in FAC for both CLK0 &CLK1.

Final Measurement Detector 3

Table 7. Final measurement detector for both CLK0 and CLK1.

Frequency (MHz)	QuasiPeak (dBµV/m)	Meas. Time (ms)	Bandwidth (kHz)	Antenna height (cm)	Polarity	Turntable position (deg)	Corr. (dB)	Margin (dB)	Limit (dBµV/m)
36.000000	56.5	1000.000	120.000	155.0	H	0.0	14.4	-21.5	35.0
54.000000	62.0	1000.000	120.000	155.0	H	0.0	6.8	-27.0	35.0
60.000000	59.8	1000.000	120.000	155.0	H	0.0	5.6	-24.8	35.0
72.000000	59.9	1000.000	120.000	155.0	H	0.0	6.0	-24.9	35.0
84.000000	66.5	1000.000	120.000	155.0	H	0.0	7.9	-31.5	35.0
90.000000	75.7	1000.000	120.000	155.0	H	0.0	9.1	-40.7	35.0
114.000000	64.8	1000.000	120.000	155.0	H	0.0	11.7	-29.8	35.0
126.000000	70.8	1000.000	120.000	155.0	H	0.0	11.8	-35.8	35.0
138.000000	55.9	1000.000	120.000	155.0	H	0.0	11.3	-20.9	35.0
162.000000	61.3	1000.000	120.000	155.0	H	0.0	10.2	-26.3	35.0

Based on the data in Table 7, the odd harmonics are summarized in the following Table 8.

Table 8. Odd harmonics present when CLK0 & CLK1 tested.

Odd Harmonic	Even Harmonic	Frequency at 12MHz	Frequency at 18MHz
3		36.000000	54.000000
	4		72.000000
5		60.000000	90.000000
	6	72.000000	
7		84.000000	126.000000
9			162.000000
		114.000000	
		138.000000	

The interference source is generated when testing both 12MHz CLK0 and 18MHz CLK1 at the same time. This is not really good, especially if using it as a reference source. The result of the table shows they have three odd harmonics for CLK0 and four odd harmonics for CLK; and one even harmonic per ten measured harmonics and have two frequencies other than multiple of 12MHz & 18MHz.

7 EXTENDED VERSION

7.1 Extended Version Design

The extended version was designed with some differences to the test version. In this version, the Clock generator used was not an Adafruit Si5351A breakout board which already has a 25MHz crystal available. It would include Skyworks's Si5351A Chip, an external AT-cut Crystal, and a high-speed inverter that would be used and integrated with other additional components such as temperature sensors, and buttons on a printed circuit board (PCB).

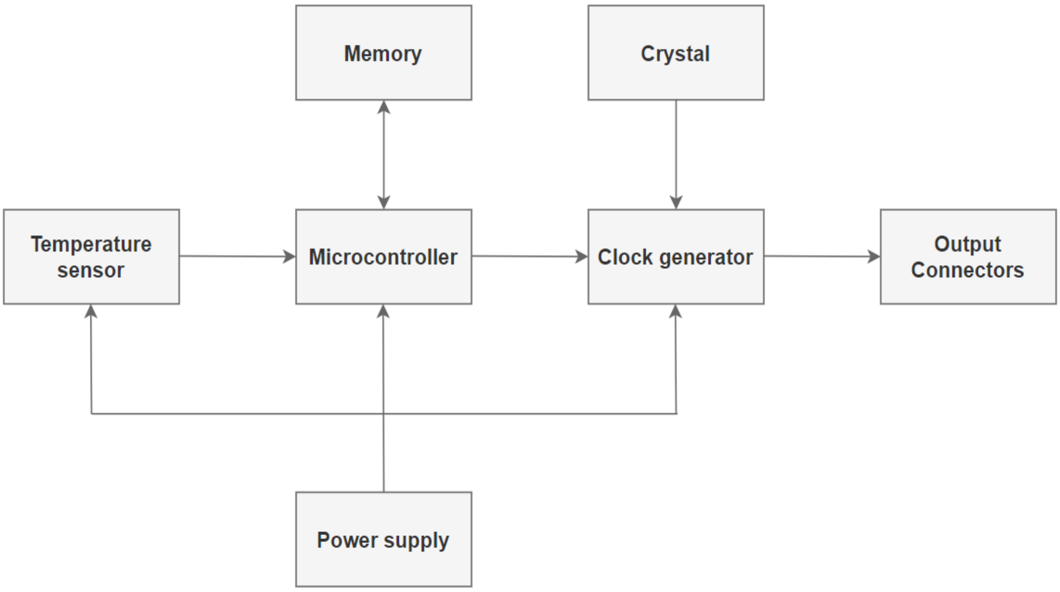

Figure 27. Block diagram of Extended Version.

7.1.1 Component List

Comparing the component list of two versions, the most significant difference between the extended version and the test version is the integration of a temperature sensor, a high-speed inverter, and push buttons to adjust the output frequency.

In Table 9, most of the components are considered to be in SMD form to limit the device size and fit the PCB.

Table 9. Component list of extended versions.

Name	Description
Crystal	25/27 MHz, 12MHz-50MHz
Battery	1x VARTA 550 mAh, 9V
Microcontroller board	1x Arduino Uno Rev3
Switch	1x Power switch
Output Connector (51K507-802N5)	2x Female BNC RF bulkhead jack
Aluminum Box	1x (8x12) cm
Temperature Sensor	1x DS1621 Digital thermometer
Push Button	3x B3W-1050
LCD	1x I2C 16x2 Arduino LCD display
High speed inverter	1x 74HC04 (available) 1x 74ABT04/ 74LVT04 (ordering)
Resistor	8x 2K2,10K
Capacitor	7x 0.1uF, 0.47uF, 100nF
USB cable	1x USB A to B
Breadboard	2x (5.5 x 8.5) cm
LED	1x Blue light
Wires	
Soldering tools and accessories	

A normal oscillator is also temperature-dependent, so it is good to have a sensor to observe their changes. DS1621 sensor is known as a digital thermometer which provides 9 bits temperature readings. It can simply control this system and convert temperature to digital word in less than 1 second. Its temperature range is from $-55°\text{C}$ to $+125°\text{C}$ in $0.5°\text{C}$ Increments. The pin sequence of the component is shown in Figure 28.

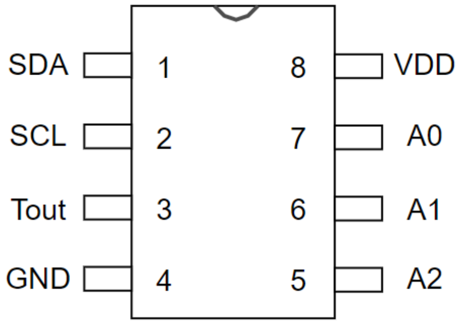

Figure 28. DS1621 sensor pin.

Table 10. Pin Description of DS1621 sensor.

Pin Name	Function
SDA	Serial Data Input/Output
SCL	Serial Clock
GND	Ground pin
Tout	Thermostat Output Signal
A0	Chip Address Input
A1	Chip Address Input
A2	Chip Address Input
VDD	Power Supply Voltage (2.7 to 5.5V)

In addition, this version uses a high-speed inverter at the outputs of the Si5351A. Because the 74HC04 [22] was available at the lab, it was assembled in the prototype of the extended version in Figure 29. Although it belongs to the CMOS logic family, it can be done with a power supply from 2V to 6V, but its operating speed (21ns) does not respond well to the radiator. Therefore, the device needs a higher-speed converter. After the research process, several inverters were found suitable for the power supply and performance speed of this device. Based on the statistical table of digital logic families corresponding to their necessary parameters, such as power supply, and speed, (Table 11), it was possible to choose the most suitable type of inverter for the device.

Table 11. Digital logic family sorted by Speed [23].

Logic Family	Voltage Nodes, V_{CC}						Speed	Output Current		Current Into V_{CC}
	1.2V	1.5V	1.8V	2.5V	3.3V	5.0V	t_{pd} max (ns)	I_{OL} (mA)	I_{OH} (mA)	I_{CC} (mA)
GTLP					●		TBD	100	-24	50
CBT						●	0.25	-	-	0.003
CBTLV				■	●		0.25	-	-	0.010
AVC	■	■	■	●	■		1.9	8	-8	0.040
ALB					●		2	25	-25	0.800
ALVT				■	●		2.5	64	-32	5
ALVC			■	■	●		3	24	-24	0.040
ABT						●	3.5	64	-32	0.250
LVT				■	●		3.5	64	-32	0.190
LVC			■	■	●		4	24	-24	0.010
TVC	■	■	■	■	■	■	4	-	-	-
SSTL				■	●		4.1	20	-20	90
ABTE						●	5.2	90	-60	48
HSTL					●		5.2	24	-24	100
74F						●	6	64	-15	120
GTL					●	■	6.3	100	-32	80
AC					■	●	6.5	24	-24	0.040
BCT						●	6.6	64	-15	90
FB						●	7	100	-3	120
AHC					■	●	7.5	8	-8	0.040
AS						●	7.5	64	-15	143
FCT						●	7.5	64	-15	0.500
AHCT						●	7.7	8	-8	0.040
ACT						●	8	24	-24	0.040

Logic Family	Voltage Nodes, V_{CC}						Speed	Output Current		Current Into V_{CC}
	1.2V	1.5V	1.8V	2.5V	3.3V	5.0V	t_{pd} max (ns)	I_{OL} (mA)	I_{OH} (mA)	I_{CC} (mA)
S						●	9	64	-15	180
ALS						●	10	24	-15	58
LS						●	12	24	-15	95
LV				■	●	■	14	8	-8	0.020
HC				■	■	●	21	6	-6	0.080
TTL						●	22	16	-0.4	22
HCT						●	30	6	-6	0.080
CD4000						●	250	-	10	0.00025

7.1.2 Hardware Setting for Extended Version

Before starting to design a complete PCB, a prototype is built as shown in Figure 29. Pull-up resistors are added at the I2C bus as well as at the push buttons to ensure a known state for the signal as the example in Figure 30.

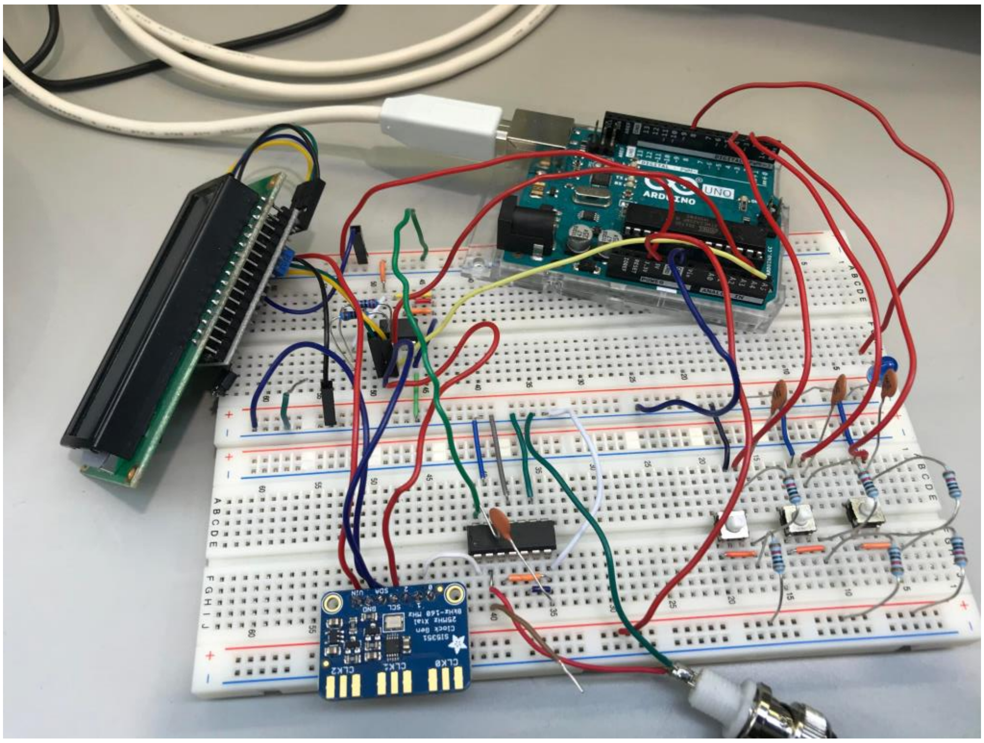

Figure 29. Prototype of extended version.

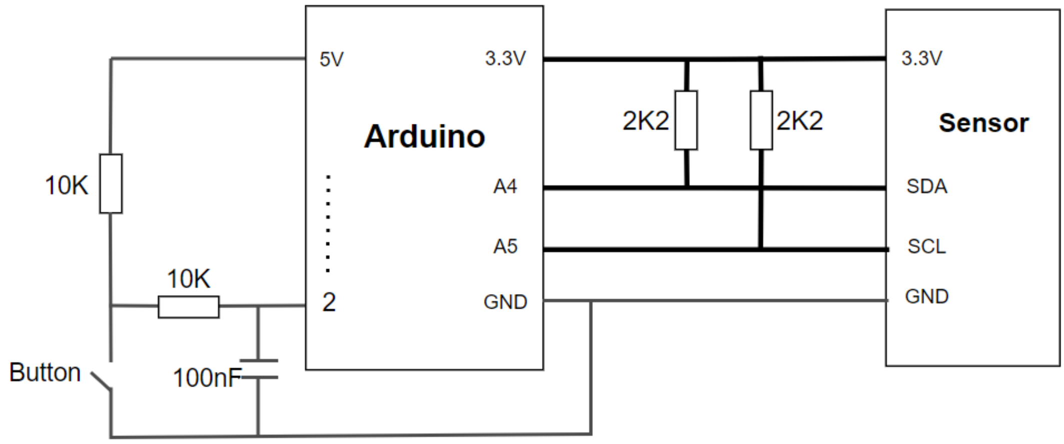

Figure 30. Pull-up resister with Arduino.

Not only each sensor communicates with the MCU via I2C protocol, but also the Si5351A module and LCD. Their SDA and SCL pins are also connected to the Arduino's A4 and A5 pins, respectively. About the inverter, as explained in Section 7.1.1, the 74HC04 inverter is available at the laboratory, so it should be assembled. It would be replaced by a faster inverter.

A 16x2 LCD screen is also used in this prototype to easily observe sensor values as well as output frequency when adjusting the buttons. However, to ensure that the reference radiator has a compact area and no emissions other than those from the two frequency outputs (CLK0 and CLK1), the LCD is considered unnecessary for the final product.

7.2 Controlling the completed version

To implement a completed version, three main parts are needed:

- Configuring and controlling SI5351A. This section will be the same as in chapter 6.3
- Measuring temperature from DS1621 sensor.
- Setting up a control system based on three push buttons.

Before implementing codes of every single part, observe the flowchart as demonstrated in Figure 31. It helps to clarify the workflow of the system.

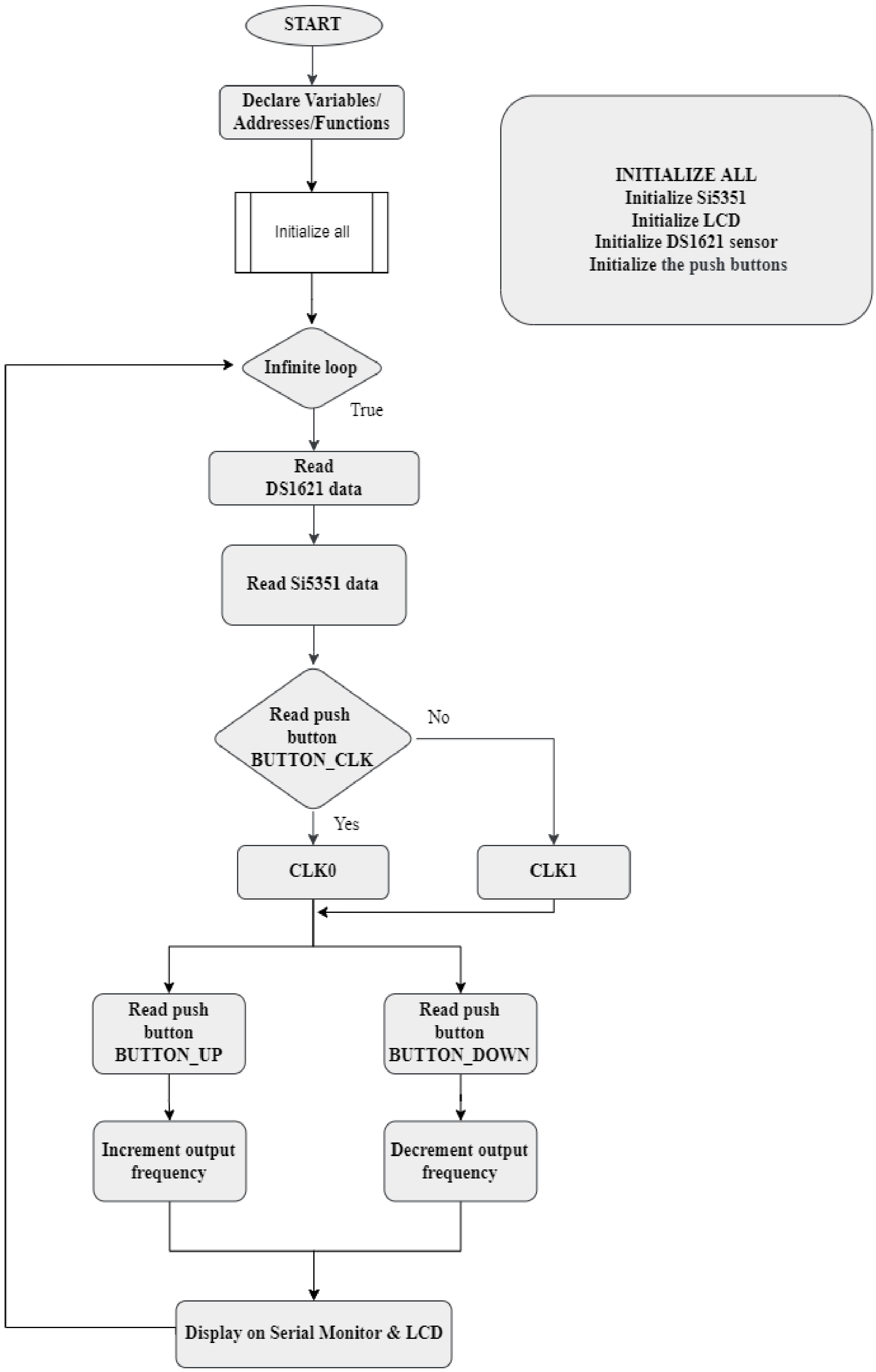

Figure 31. Algorithm of system implementation.

In Arduino development environment, all Arduino programs always follow the following main structure:

```
void setup()
{
        ...
}
void loop()
{
        ...
}
```

Therefore, after defining and declaring variables, functions, and libraries, … the codes that need to run once will be put into setup() function. These codes are often used for initializations. When this function is finished, loop() function would start running and repeat infinity. It only stops when the power is off.

7.2.1 Measuring Temperature

The slave address of DS1621 consists of an 8-bit control byte or it is also known as the first byte. It contains 4-bit MSB that is set default as 1001 binary for reading and writing operations. The next 3-bit can be adjusted by selecting the chip address bits (A0, A1, A2) by connecting the pins to the ground or supply voltage. The last bit of the first byte ($R/\overline{W}$) is a data-oriented bit-defining operation to be performed. In which, set "1" for reading operation and "0" for writing operation [24].

In this case, pins (A0, A1, A2) are grounded, so the slave address of this sensor is $1001 + A2 + A1 + A0 \ = \ 1001\ 000$ binary equivalent to $0x48$ hex.

After defining and initializing DS1621, it starts reading the raw temperature value of the sensor and converts it to tenths degree Celsius. This value is then decomposed into two parts, an integer part and a decimal part based on the following formula:

$$\frac{Temp_{raw}}{10} => Integer\ part$$

$$Temp_{raw}\%\ 10\ => \ Decimal\ part$$

The temperature should be measured continuously, so the code of it would be put in the infinite loop.

Figure 32. Temperature shows on LCD.

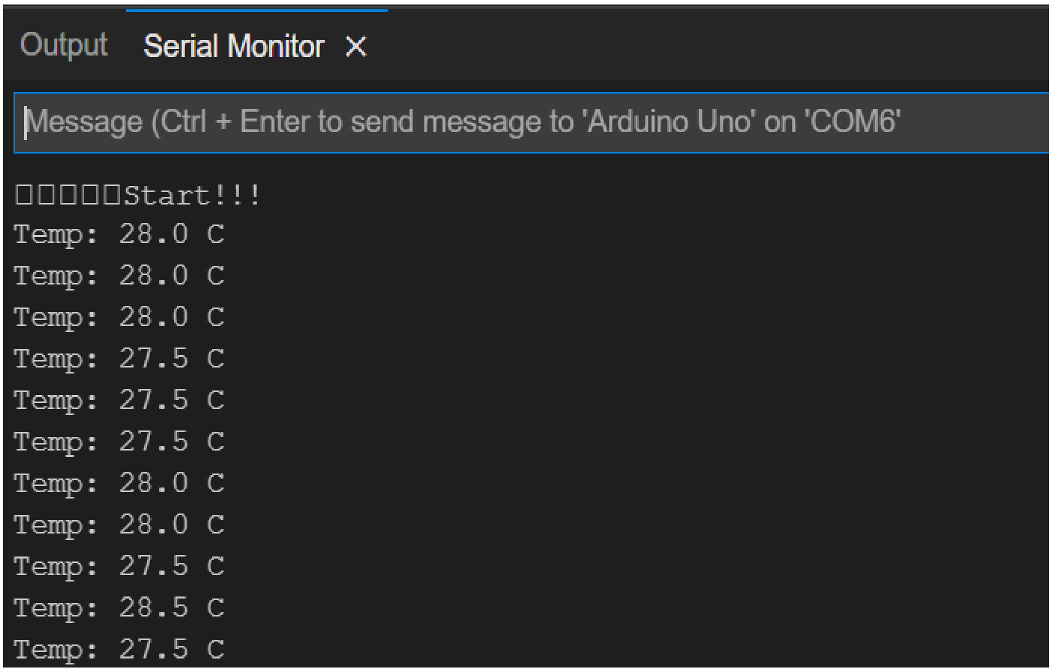

Figure 33. Temperature shows on Serial Monitor.

7.2.2 Adjusting Frequency based on Push Buttons

From Figure 31 it can be seen that the device has 3 buttons: BUTTON_CLK, BUTTON_UP, and BUTTON_DOWN.

The operation of the push button is simply based on pressing action, namely, for the BUTTON_CLK, when this button is pressed, CLK0 will be selected and when pressed again, CLK1 will be selected. In general, the CLK is changed after each press. The default value of the two CLKs is 12MHz.

BUTTON_UP and BUTTON_DOWN have the same operating mechanism. After each press, the selected CLK will increase by 1MHz if BUTTON_UP is used and decrease by 1MHz if BUTTON_DOWN is used. As mentioned in chapter 1, the frequency will only fluctuate between 12MHz and 50MHz.

The desired result is that the user can easily adjust the output frequency at the CLKs and the frequencies of these 2 CLKs can also be set differently or the same, depending on the needs of the user as shown in Figure 33.

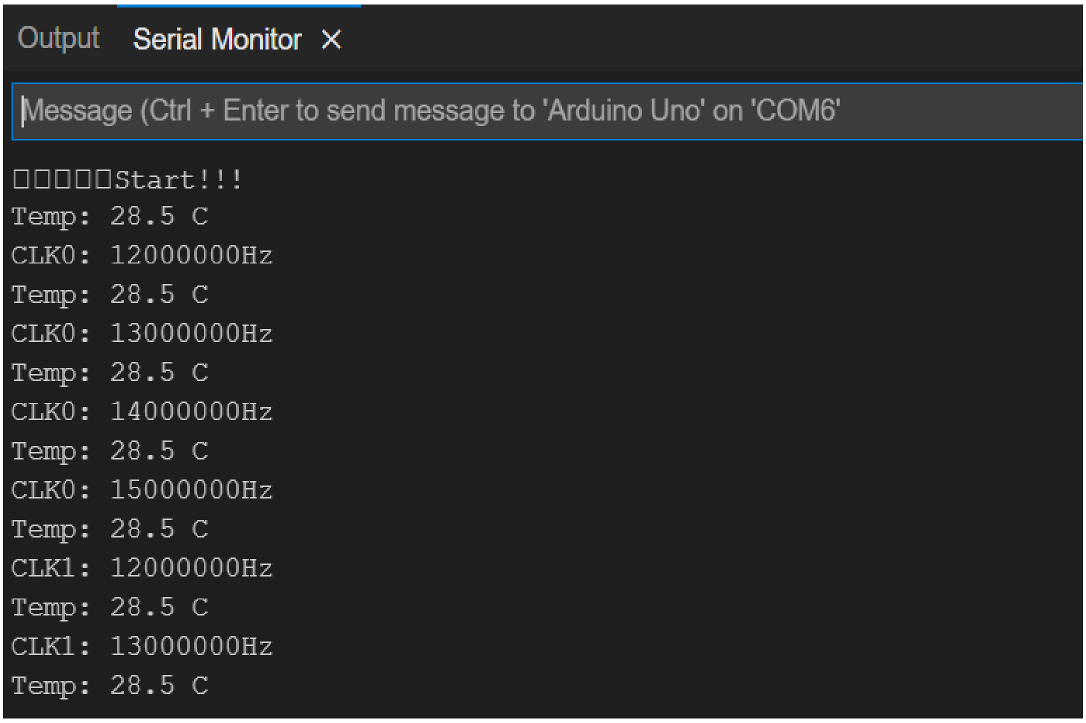

Figure 34. Extended version results on Serial monitor.

7.3 Testing and Results

Figure 34 is the result of the assembly using the 74HC04 inverter measured from the oscilloscope. It is expected to be improved by replacing the Adafruit Si5351A with a Si5351A chip and a higher speed inverter as shown in Table 11.

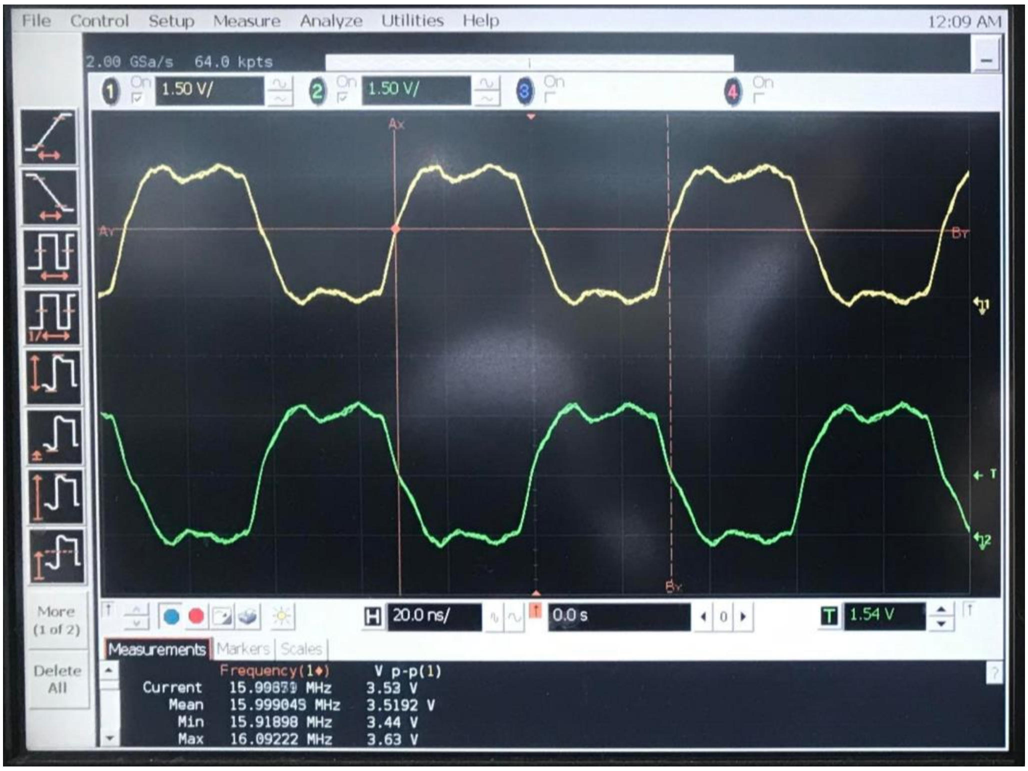

Figure 35. Assembly with 74HC04 inverter.

7.4 Printed Circuit Board Design

The product is expected to be compact with a small area of only 8cmx12cm. So, this reference radiator will be designed based on the idea of RF Solutions Ltd, it is a popular UK Electronics manufacturer. It is a combination of Arduino Uno shield and PCB. The PCB only needs to be plugged directly into the top of Arduino Uno without any wire as shown in Figure 35. This method is not only easy to implement but also cost-effective and gives better results by reducing using of wires.

First of all, a software tool should be used for creating diagrams and PCB, it could be PADS or any similar tool. In this project, KiCad is used to design PCB because it is an open-source tool and has a diverse component library. It is included two main tasks: drawing the electronic schematic by using Eeschema and laying out the printed circuit board by using Pcbnew. In KiCad, the user can export, import, and modify components in the library as well as make the footprints for those components [26]. In addition, the user also admires their board in 3D as seen in Figure 40.

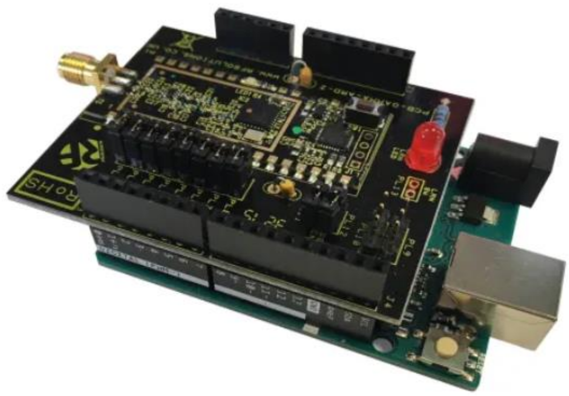

Figure 36. RF Solutions for Arduino Uno [25].

The schematic of the reference radiator consists of 5 parts: Shield, Clock generator, Temperature sensor, Pushbuttons, and Connectors annotated as shown in Figure 36.

After completing the schematic, a blank PCB layout was created to import components. And because RF solution is used, all components must be placed on top of the Arduino Uno's shield.

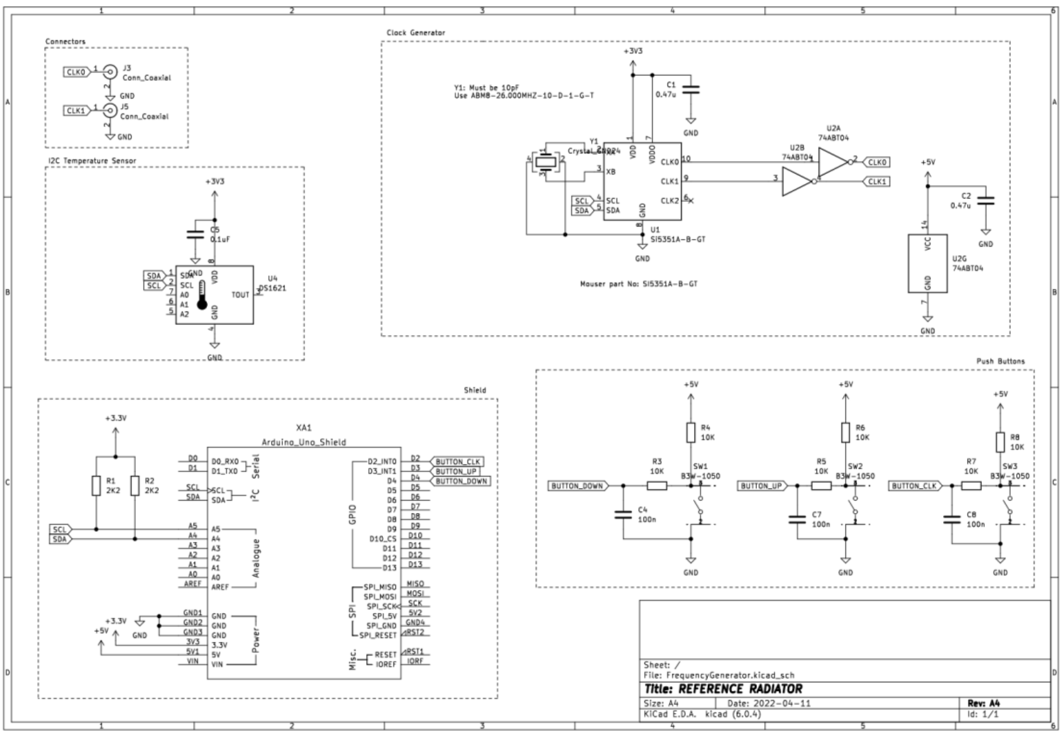

Figure 37. Schematic of Extended version.

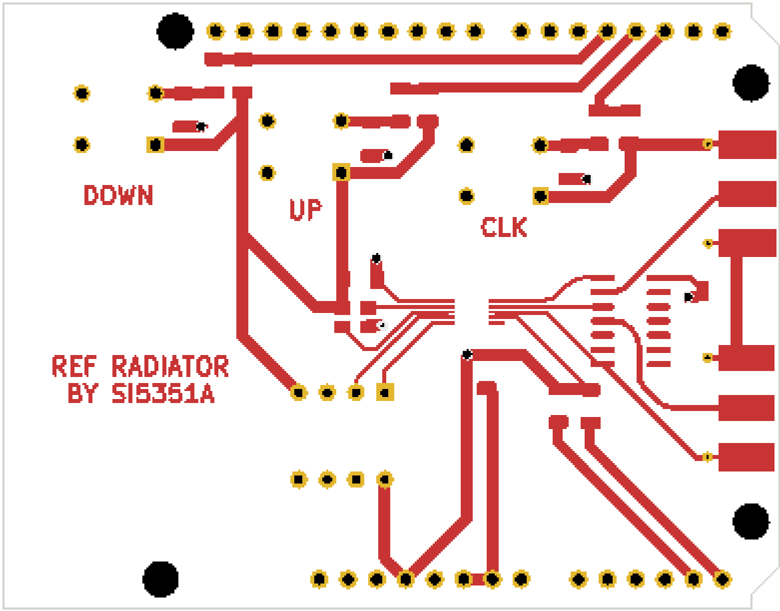

Figure 38. Front view of PCB.

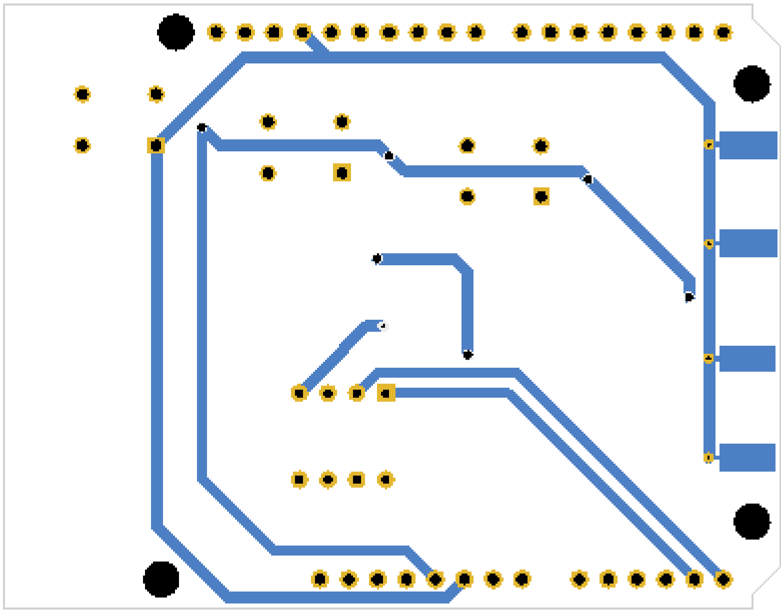

Figure 39. Bottom view of PCB.

To reduce high-frequency noise in the PCB, grounding vias are added as much as possible to cover the board area, in particular, around the high-frequency output traces.

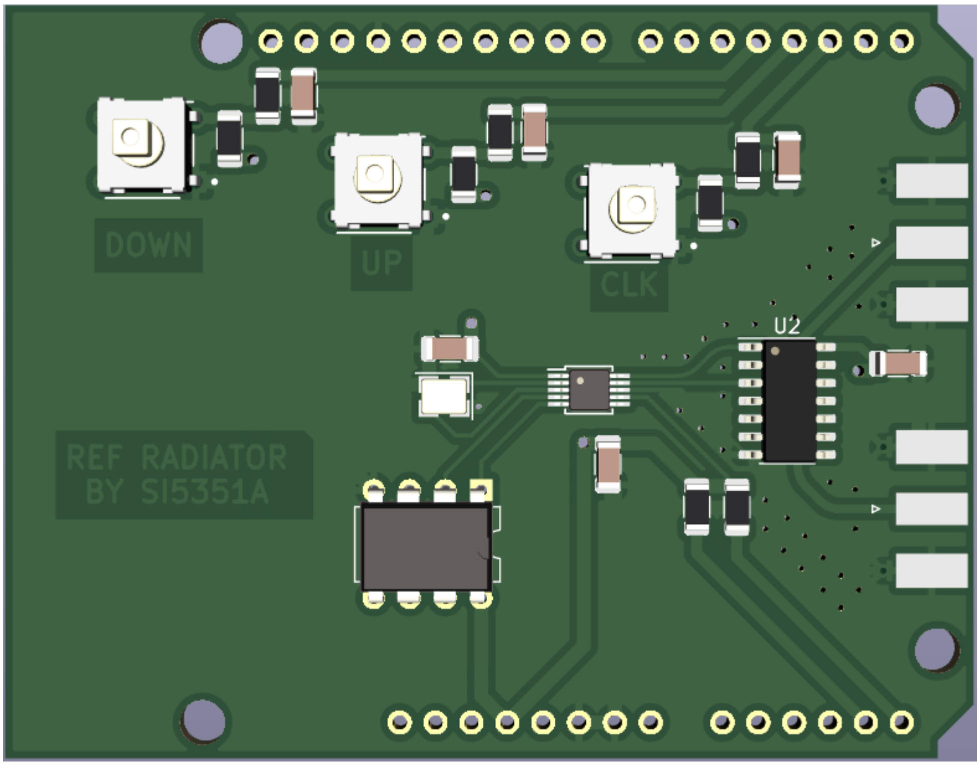

Figure 40. 3D model of PCB.

8 CONCLUSIONS

Overall, the thesis fulfills most of the objectives outlined in this project. The test version was completed and can be used in the EMC laboratory. However, its operating mechanism includes only one function by default: emitting frequencies. The control algorithm of the extended version was completed with full functionality as the stated goal. It can collect components' data and display it on the Serial Monitor or LCD. However, the hardware of this version is still incomplete; its PCB is still in 3D modeling.

It is important to note the accuracy and performance of the reference radiator. However, the extended version implemented in this project mainly covers only the necessary and basic elements, stages, and components to create this device. In addition, it also offers better alternatives rather than creating a finished product. Therefore, if designing a product for long-term use, the power battery, inverter, and crystal oscillator, should be replaced with suitable ones. Because the functions, as well as the quality of the components, should depend on the needs of the user.

The chosen topic allowed studying many aspects related to emission radiation, the basic Fourier calculation for a square wave, circuit design, EMC testing, and algorithms through tools, such as Arduino IDE, and KiCad, as well as devices such as oscilloscope, drilling machine, and soldering tools.

In short, the development of this thesis has roughly covered the entire process of developing a particular product in the field of technology.

REFERENCES

[1] Ott, H., W,(2009) Electromagnetic Compatibility Engineering, p.464-476

[2] Kraus, J.D. and Marhefka, R.J., 2002. Antenna temperature, remote sensing and radar cross section. Antennas for All Applications 3rd Edition, McGraw Hill, NY: New York, p. 199.

[3] Ott, H., W. 2009 Electromagnetic Compatibility Engineering, p. 477-489

[4] Singh, Nisha & Sharma, Madhuri & Verma, Anupam. 2015. Antenna and Its Application. 6, p. 95-97.

[5] Balanis, C.A., 1982. Antenna Theory: Analysis and Design. Harper & Row. Publishers, New York.

[6] Langton, C. and Levin, V., 2017. The Intuitive Guide to Fourier Analysis and Spectral Estimation. Ist edition. USA. Mountcastle Company.

[7] Morin, David. 2009. Waves (Draft, version 1).

[8] Strang, G. 2007. Computational science and engineering (Vol. 791). Wellesley. Wellesley-Cambridge Press.

[9] Cheever, Erik .2005. Derivation of Fourier Series. Accessed 12.5.2022. https://lpsa.swarthmore.edu/Fourier/Series/DerFS.html.

[10] Cheever, Erik. 2005. Introduction to the Fourier Transform. Accessed 12.5.2022. https://lpsa.swarthmore.edu/Fourier/Xforms/FXformIntro.html.

[11] Cheever, Erik. C2005. Fourier Xform of Periodic Functions. Accessed 12.5.2022. https://lpsa.swarthmore.edu/Fourier/Xforms/FXPeriodic.html.

[12] Roshan Ragel. 2011. Difference Between Fourier Series and Fourier Transform. Accessed 16.5.2022. https://www.differencebetween.com/difference-between-fourier-series-and-vs-fourier-transform/

[13] Skyworks, 2021. I2C-Programmable Any-frequency CMOS Clock Generator + VCXO Rev. 1.3. Accessed 18.5.2022. https://www.skyworksinc.com/-/media/SkyWorks/SL/documents/public/data-sheets/Si5351-B.pdf

[14] Skyworks. 2021. Manually Generating an Si5351 Register Map for 10-MSOP and 20-QFN Devices Rev. 0.8. Accessed 20.5.2022. https://www.skyworksinc.com/-/media/Skyworks/SL/documents/public/application-notes/AN619.pdf

[15] NXP B.V. 2021. I^{2}C-bus specification and user manual. UM10204 Rev. 7.0. Accessed 22.5.2022. https://www.nxp.com/docs/en/user-guide/UM10204.pdf

[16] Lady Ada. 2014. Adafruit Si5351 Clock Generator Breakout. Accessed 22.5.2022. https://learn.adafruit.com/adafruit-si5351-clock-generator-breakout/downloads.

[17] Arduino UNO R3. 2021. Product Reference Manual, SKU: A000066, Rev 1.0. Accessed 1.6.2022. https://docs.arduino.cc/static/3a2798446a540d4c8284aba4144c0cec/A000066-datasheet.pdf

[18] MicroBattery. 2021. Battery Bios: Everything you need to know about the 9v battery. Accessed 1.6.2022. https://www.microbattery.com/blog/post/battery-bios:-everything-you-need-to-know-about-the-9v-battery/

[19] Rosenberger Hochfrequenztechnik GmbH & Co. 2020. Bulkhead Jack, BNC 50Ohm, Technical Datasheet, 51K507-802N5. Accessed 2.6.2022. https://docs.rs-online.com/6643/A700000008448618.pdf.

[20] Ustuner, Fatih & Araz, Isa & Yalcin, Nuh & Baran, Ersan. 1999. Fully Anechoic Chamber Construction And Evaluation.

[21] EMC FastPass. 2019. EMC Testing: The Beginner’s Guide. Accessed 15.5.2022. https://emcfastpass.com/emc-testing-beginners-guide/emissions/

[22] Diodes Incorporated. 2013. 74HC04 Hex Inverters. DS35320 Rev. 3-2. Accessed 8.6.2022. https://www.diodes.com/assets/Datasheets/74HC04.pdf

[23] Texas Instruments. Digital Logic Family Selection Matrix (Sorted by Speed). Accessed 10.6.2022. https://digsys.upc.edu/csd/units/Elect/TI_Digital_Logic_Families.pdf

[24] Maxim Integrated Products. 2015. DS1621 Digital Thermometer, DS1621 datasheet, 19-7540, Rev 4. Accessed 15.6.2022. https://datasheets.maximintegrated.com/en/ds/DS1621.pdf

[25] RFSolutions-TF. 2016. The RF Solution for Arduino. Accessed 20.6.2022. https://www.rs-online.com/designspark/the-rf-solution-for-arduino

[26] KiCad. 2021. Getting Started in KiCad. Accessed 25.6.2022. https://docs.kicad.org/4.0/en/getting_started_in_kicad/getting_started_in_kicad.pdf

APPENDICES

APPENDIX 1

Si5351A code

```
#include <arduino.h>
#include <Wire.h>

#define MSNA_0(x) ((uint8_t)x)           // Bust int32 into Bytes
#define MSNA_8(x) ((uint8_t)(x>>8))
#define MSNA_16(x) ((uint8_t)(x>>16))

#define SI5351_ADDR 0x60           // I2C address of Si5351
#define SI5351_XTALPF 2           // 1:6pf  2:8pf  3:10pf

#define SI5351_XTAL 25000000        // Crystal freq in Hz (Ref freq)
#define SI5351_MSA  36             // VCOA is at 25mhz*36 = 900mhz _ (a+b/c)

// #define SI5351_CLKIN  PAGE 3

uint32_t si5351_vcoa = (SI5351_XTAL*SI5351_MSA);  // 25mhzXtal calibrate
uint8_t  si5351_rdiv = 0;           // 0-7, CLK pin sees fout/(2**rdiv)
uint8_t  si5351_drive[3] = {0, 0, 0}; // 0=2ma 1=4ma 2=6ma 3=8ma for CLK 0,1,2
uint8_t  si5351_clken = 0xFF;        // Private, all CLK output drivers off

uint32_t currentCLK = 0;
uint32_t currentFreq = 12000000;

void i2cWrite(uint8_t reg, uint8_t val) {   // write reg via i2c
  Wire.beginTransmission(SI5351_ADDR);
  Wire.write(reg);
  Wire.write(val);
  Wire.endTransmission();
}

void i2cWriten(uint8_t reg, uint8_t *vals, uint8_t vcnt) {  // write array
  Wire.beginTransmission(SI5351_ADDR);
  Wire.write(reg);
  while (vcnt--) Wire.write(*vals++);
  Wire.endTransmission();
}


void si5351_init() {               // Call once at power-up, start PLLA
```

```
  uint8_t reg;  uint32_t msxp1;
  Wire.begin();
  i2cWrite(149, 0);                    // SpreadSpectrum off
  i2cWrite(3, si5351_clken);          // Disable all CLK output drivers
  i2cWrite(183, SI5351_XTALPF << 6);  // Set 25mhz crystal load capacitance
  msxp1 = 128 * SI5351_MSA - 512;     // and msxp2=0, msxp3=1, not fractional
  uint8_t  vals[8] = {0, 1, MSNA_16(msxp1), MSNA_8(msxp1), MSNA_0(msxp1), 0, 0, 0};
  i2cWriten(26, vals, 8);              // Write to 8 PLLA msynth regs
  i2cWrite(177, 0x20);                 // Reset PLLA (0x80 resets PLLB)
}

void si5351_setfreq(uint8_t clknum, uint32_t fout) {  // Set a CLK to fout Hz
  uint32_t  msa, msb, msc, msxp1, msxp2, msxp3p2top;
  if ((fout < 10000000) || (fout > 50000000)) // If clock freq out of range
    si5351_clken |= 1 << clknum;      //  shut down the clock
  else {
    msa = si5351_vcoa / fout;     // Integer part of vco/fout
    msb = si5351_vcoa % fout;     // Fractional part of vco/fout
    msc = fout;             // Divide by 2 till fits in reg

    while (msc & 0xfff00000) {
      msb = msb >> 1;
      msc = msc >> 1;
    }
    msxp1 = (128 * msa + 128 * msb / msc - 512) | (((uint32_t)si5351_rdiv) << 20);
    msxp2 = 128 * msb - 128 * msb / msc * msc; // msxp3 == msc;
    msxp3p2top = (((msc & 0x0F0000) << 4) | msxp2);     // 2 top nibbles
    uint8_t vals[8] = { MSNA_8(msc), MSNA_0(msc), MSNA_16(msxp1), MSNA_8(msxp1),
                      MSNA_0(msxp1), MSNA_16(msxp3p2top), MSNA_8(msxp2), MSNA_0(msxp2)
                     };
    i2cWriten(42 + (clknum * 8), vals, 8); // Write to 8 msynth regs
    i2cWrite(16 + clknum, 0x0C | si5351_drive[clknum]); // use local msynth
    si5351_clken &= ~(1 << clknum);   // Clear bit to enable clock
  }
  i2cWrite(3, si5351_clken);        // Enable/disable clock
}


void initOscillators(){
  //initialize the SI5351
  si5351_init();
  si5351_setfreq(currentCLK, currentFreq);
}
```

APPENDIX 2

Code for reading Oscillator, Sensor, Pushbuttons, LCD, Serial monitor

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD 16x2 address to 0x27
#define DS1621_ADDRESS  0x48

extern void initOscillators(void);
extern void si5351_setfreq(uint8_t clknum, uint32_t fout);

extern uint32_t currentCLK;
extern uint32_t currentCLK0 = 0;
extern uint32_t currentCLK1 = 1;

extern uint32_t currentFreq;
extern uint32_t currentFreq0 = 12000000;
extern uint32_t currentFreq1 = 12000000;

void initialize_all(void);
void showStatus();

uint8_t LED = 8;
uint8_t BUTTON_CLK = 2;
uint32_t BUTTON_UP = 3;   // Button up command
uint32_t BUTTON_DOWN = 4;   // Button down command
uint32_t STEP = 1000000; //1MHz
char sendBuff[16];
char buffer[16];
int k = 0;

void setup() {
  // put your setup code here, to run once:
  initialize_all();
}

void loop()
{
  //DS1621
  // get temp in tenths °C
  int16_t temp = Temperature();

  if(temp < 0) {   // if temp < 0 °C
    temp = abs(temp);  // absolute value
    sprintf(buffer, "-%02u.%1u%c C", temp / 10, temp % 10);
```

```
 }
 else {
  if (temp >= 1000)    // if temp >= 100.0 °C
    sprintf(buffer, "Temp: %03u.%1u C", temp / 10, temp % 10);
  else
    sprintf(buffer, "Temp: %02u.%1u C", temp / 10, temp % 10);
 }
 lcd.setCursor(0, 0);  // move cursor to position (0, 0)
 lcd.print(buffer);  // temp in °C
 delay(100);

//PUSHBUTTON

 if (digitalRead(BUTTON_CLK) == LOW && k == 0) {
  delay(10);
  digitalWrite(LED, LOW);
  currentCLK = currentCLK0;
  currentFreq = currentFreq0;
  si5351_setfreq(currentCLK, currentFreq);

  k = 1;
  showStatus();
 } else if (digitalRead(BUTTON_CLK) == LOW && k == 1) {
  delay(10);
  digitalWrite(LED, HIGH);
     delay(100);
  digitalWrite(LED, LOW);
  currentCLK = currentCLK1;
  currentFreq = currentFreq1;
  si5351_setfreq(currentCLK, currentFreq);

  k = 0;
  showStatus();
 }

//PUSHBUTTON UP & DOWN
  if (digitalRead(BUTTON_UP) == LOW) {
  delay(10);
  currentFreq = currentFreq + STEP;
  if (currentCLK == currentCLK0)
  {
    currentFreq0 = currentFreq;
  } else  currentFreq1 = currentFreq;
  si5351_setfreq(currentCLK, currentFreq);
    if (currentFreq > 50000000){
     currentFreq = 50000000;
```

```
      si5351_setfreq(currentCLK, currentFreq);
    }

  showStatus();
  } else if (digitalRead(BUTTON_DOWN) == LOW)  {
  delay(10);
  currentFreq = currentFreq - STEP;
  if (currentCLK == currentCLK0)
  {
    currentFreq0 = currentFreq;
  } else  currentFreq1 = currentFreq;
  si5351_setfreq(currentCLK, currentFreq);
    if (currentFreq < 12000000){
      currentFreq = 12000000;
      si5351_setfreq(currentCLK, currentFreq);
    }

  showStatus();
  }
}

int16_t Temperature() {
  Wire.beginTransmission(DS1621_ADDRESS); // link to DS1621 (send DS1621 address)
  Wire.write(0xAA);                       // read temp command
  Wire.endTransmission(false);            // send repeated start condition
  Wire.requestFrom(DS1621_ADDRESS, 2);    // request 2 bytes from DS1621 & re-
lease I2C bus at end of reading
  uint8_t Temp_MSB = Wire.read();          // read temp MSB register
  uint8_t Temp_LSB = Wire.read();          // read temp LSB register

  // calculate full temp (raw value)
  int16_t raw_value = (int8_t)Temp_MSB << 1 | Temp_LSB >> 7;
  // convert raw value to tenths °C
  raw_value = raw_value * 10/2;
  return raw_value;

}

void initialize_all(void)
{
  //Init Serial Monitor
  Serial.begin(9600);
  Serial.write("Start!!!\r\n");

  //Init PushButtons
  pinMode(BUTTON_CLK, INPUT);
```

```
  pinMode(BUTTON_UP, INPUT);
  pinMode(BUTTON_DOWN, INPUT);
  pinMode(LED, OUTPUT);

  //Init Si5351
  initOscillators();
  si5351_setfreq(currentCLK0, currentFreq0);
  si5351_setfreq(currentCLK1, currentFreq1);
  delay(1000);

  //Init LCD
  lcd.init();
  lcd.begin(16, 2);     // set up the LCD's number of columns and rows
  lcd.backlight();   // Turn on the backligt
  lcd.setCursor(0, 0);  // move cursor to position (0, 0)
  Wire.begin();          // join i2c bus

  //Init Temp Sensor
  Wire.beginTransmission(DS1621_ADDRESS); // connect to DS1621 (send DS1621 address)
  Wire.write(0xAC);                    // send configuration register address (Access Config)
  Wire.write(0);                       // perform continuous conversion
  Wire.beginTransmission(DS1621_ADDRESS); // send repeated start condition
  Wire.write(0xEE);                    // send start temperature conversion command
  Wire.endTransmission();              // stop transmission and release the I2C bus
}

void showStatus()
{
  Serial.println(buffer);

  Serial.print("CLK");
  Serial.print(currentCLK);
  Serial.print(": ");
  Serial.print(currentFreq);
  Serial.println("Hz");

  lcd.clear();
  sprintf(sendBuff, "CLK%lu: %luHz\n", currentCLK, currentFreq);

  lcd.setCursor(0, 1);  // move cursor to position (0, 1)
  lcd.print(sendBuff);  // print sendBuff (Freq)
}
```