**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

**(An Autonomous Institute, Affiliated to VTU, Belgaum, Approved by AICTE & State Govt. of Karnataka), Yelahanka, Bangalore-560064**

**Department Of Electronics And Communication Engineering**

****

**CERTIFICATE**

Certified that the Internship entitled **“PSOC 5 MC BASED FREQUENCY TUNE WORD GENERATION FOR TRANSRECEIVER MODULE ”** conducted in **BEL** is carried out by **SUMANTH G.V (1NT15EC159) and TUSHAR C DESAI (1NT15EC168) ,**bonafide student of Nitte Meenakshi Institute of Technology in partial fulfillment for the award of **Bachelor of Engineering** in Electronics and Communication of the Visvesvaraya Technological University, Belgaum during the academic year 2018-2019.The internship report has been approved as it satisfies the academic requirement in respect of internship work for completion of autonomous scheme of Nitte Meenakshi Institute of Technology for the above said degree.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Signature of the Guide** |  | **Signature of the HOD** |
|  |  | **(Dr. S. Sandya)** |

|  |
| --- |
|  |
|  |

**External Viva**

**Name of the Examiners Signature with Date**

……………………………… ………………………………

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

**(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belgaum, Approved by AICTE & Govt. of Karnataka)**

**ACADEMIC YEAR 2014-15**

****

Internship Report ON

**“SUMMER TRAINING”**

**CONDUCTED BY BHARAT ELECTRONICS LIMITED**

Submitted in partial fulfillment of the requirement for the award of the degree of

**BACHELOR OF ENGINEERING**

Submitted by

**SUMANTH G.V. (1NT15EC159)**

**TUSHAR C DESAI (1NT15EC168)**

Under the guidance of

**"MS. DIVYA G"**

(Associate Professor)



**Department of Electronics and Communication Engineering**

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY

Yelahanka, Bangalore-560064

**ACKNOWLEDGEMENT**

Behind every achievement there are people who inspire us to do it. To them we take the opportunity to lay the words of gratitude imprinted not just on the paper but deep in our heart.

I express my deepest thanks to our principal **Dr H. C. Nagaraj** and **Dr N. R Shetty**,Director of **NitteMeenakshi Institute of Technology**, Bangalore for following me to carry out the mini project and supporting us throughout.

I am grateful to **Dr. S Sandya**, HOD, Department of Electronics and Communication Engineering, NMIT, Bangalore for granting me the permission and guiding me throughout.

I wish to convey my deep sense of gratitude to **Ms. DIVYA G**, Assistant Professor, ECEDepartment, NMIT for her valuable guidance through the conduction of this Mini project.

Finally we thank all other unnamed who helped us in various ways to gain knowledge and have a good training.

PSOC: Programmable System On Chip

General Description:

With its unique array of configurable blocks, PSoC 5 is a true system-level solution providing microcontroller unit (MCU), memory ,analog, and digital peripheral functions in a single chip. The CY8C55 family offers a modern method of signal acquisition, signal processing, and control with high accuracy, high bandwidth, and high flexibility. Analog capability spans the range from thermocouples (near DC voltages) to ultrasonic signals. The CY8C55 family can handle dozens of data acquisition channels and analog inputs on every GPIO pin. In addition to communication interfaces, the CY8C55 family has an easy to configure logic array, flexible routing to all I/O pins, and a high-performance 32-bit ARM Cortex-M3 microprocessor core. Designers can easily create system-level designs using a rich library of prebuilt components and boolean primitives using PSoC Creator, a hierarchical schematic design entry tool.

Features:

* 32-bit ARM Cortex-M3 CPU core
* DC to 67 MHz operation
* Flash program memory, up to 256 KB
* Up to 64 KB SRAM memory
* 128 bytes of cache memory
* 24-channel direct memory access (DMA)
* Low voltage, ultra low power
* Operating voltage range:2.7 V to 5.5 V
* Versatile I/O system
* 46 to 70 I/Os (60 GPIOs, 8 SIOs, 2 USBIOs)
* Any GPIO to any digital or analog peripheral routability
* LCD direct drive from any GPIO, up to 46×16 segments
* Four 16-bit configurable timers, counters, and PWM blocks
* 66-dB signal to noise and distortion ratio (SINAD)
* Two SAR ADCs
* –40 °C to +85 °C industrial temperature

CREATING A NEW WORKSPACE

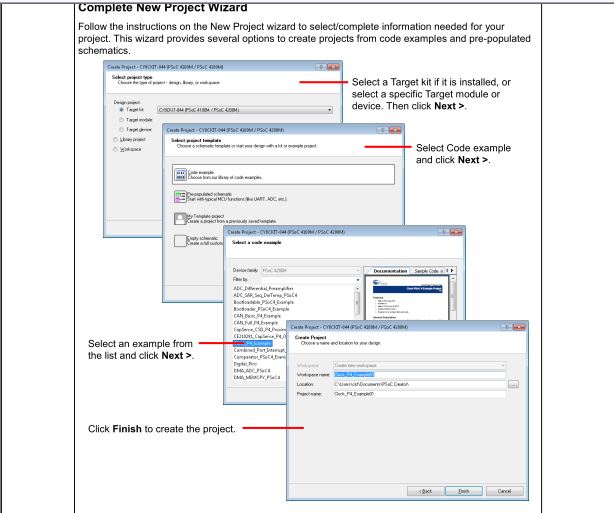


FIG 1. CREATING A WORKSPACE

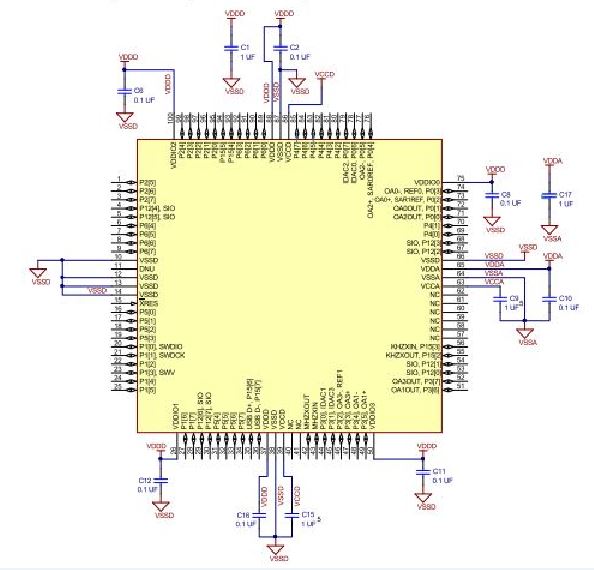


FIG 2: PIN DETAILS OF 100 PIN TQFP

SPI MASTER

Features

* 3- to 16-bit data width
* Four SPI operating modes
* Bit rate up to 18 Mbps

CODE FOR TRASFERING DATA FROM SPI MASTER

#include "project.h"

int main(void)

{

CyGlobalIntEnable;

for(;;)

{

lcd\_Start();

spi\_Start():

lcd\_Cleardisplay();

lcd\_Position(0,0);

lcd\_PrintString("WELCOME TO");

CyDelay(1000);

lcd\_Cleardisplay();

lcd\_Position(1,7);

lcd\_PrintString("BEL");

CyDelay(1000);

spi\_WriteTxData(0b01010101);

}

}

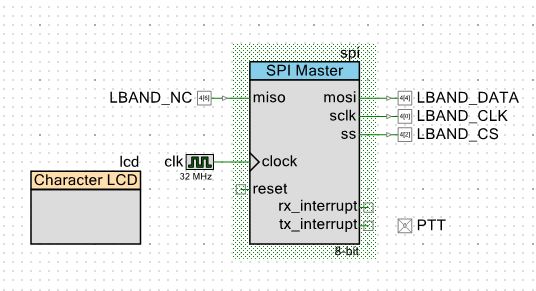


FIG 2 : SPI MASTER AND LCD

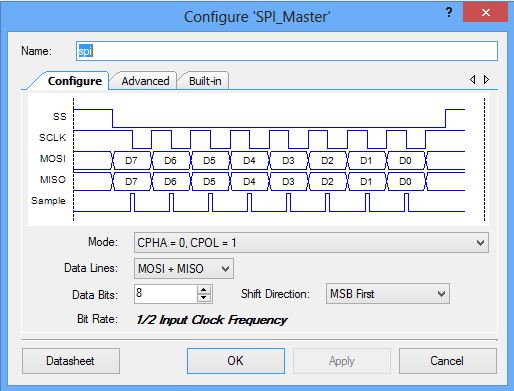


FIG 2.0 : MODE DECLARATION

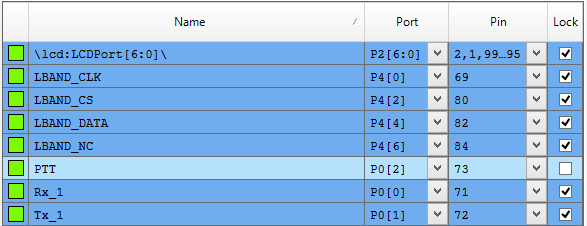


FIG 3: PORT LIST

* SPI master was given one digital input (no operation) and three digital outputs. Also a 4 Mhz clock frequency. The digital outputs were assigned to ports as following :
* MOSI – LBAND\_DATA - port P4[0]
* SCLK – LBAND\_CLK – port P4[4]
* SS – LBAND\_CS – port P4[2]
* The data is dumped after the execution of the program which can be verified on the CRO.

SPI SLAVE WITH INTERRUPT

Features

* 3- to 16-bit data width
* Four SPI modes
* Bit rate up to 5 Mbps

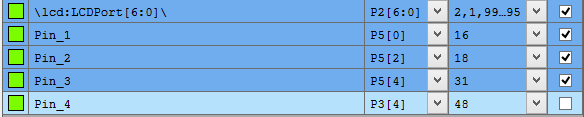


FIG 4: PORT LIST

CODE FOR SPI SLAVE

#include "project.h"

CY\_ISR\_PROTO(GPIOIsrHandler);

int main(void)

{

isr\_GPIO\_StartEx(GPIOIsrHandler);

CyGlobalIntEnable;

SPIS\_Start();

lcd\_Start();

for(;;)

{

lcd\_ClearDisplay();

lcd\_Position(0,0);

lcd\_PrintString("waiting");

CyDelay(1000);

}

}

CY\_ISR(GPIOIsrHandler)

{

isr\_GPIO\_ClearPending();

uint8 j;

while(!(SPIS\_GetRxBufferSize()));

j=SPIS\_ReadRxData();

lcd\_ClearDisplay();

lcd\_Position(0,0);

lcd\_PrintHexUint8(j);

}

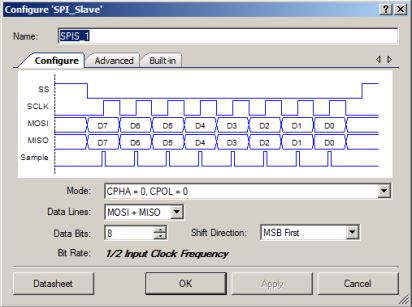
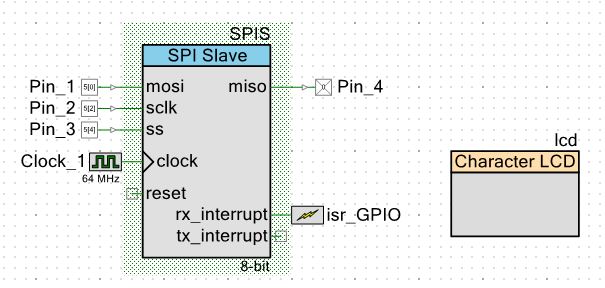


FIG 5: MODE DECLARATION

  
 FIG 6 :SPI SLAVE AND LCD

* SPI slave was given three digital input and one digital outputs. Also a 8 Mhz clock frequency. The digital inputs were assigned to ports as following :
* MOSI – LBAND\_DATA - port P\_in[0]
* SCLK – LBAND\_CLK – port P\_in[2]
* SS – LBAND\_CS – port P\_in[4]
* The mode for both master and slave must be the same. Here it is 01. CPHA=0, CPOL=1.

RESULTS

* On the LCD display “waiting” was displayed until the data 11001100 was transmitted. Once data is transmitted we observe hexadecimal format of the received data i.e CC.
* On the CRO we observed three kinds of signals, the data, the clock and slave select. The data transmitted and received on both master and slave was observed to be the same.

DATA : 11001100

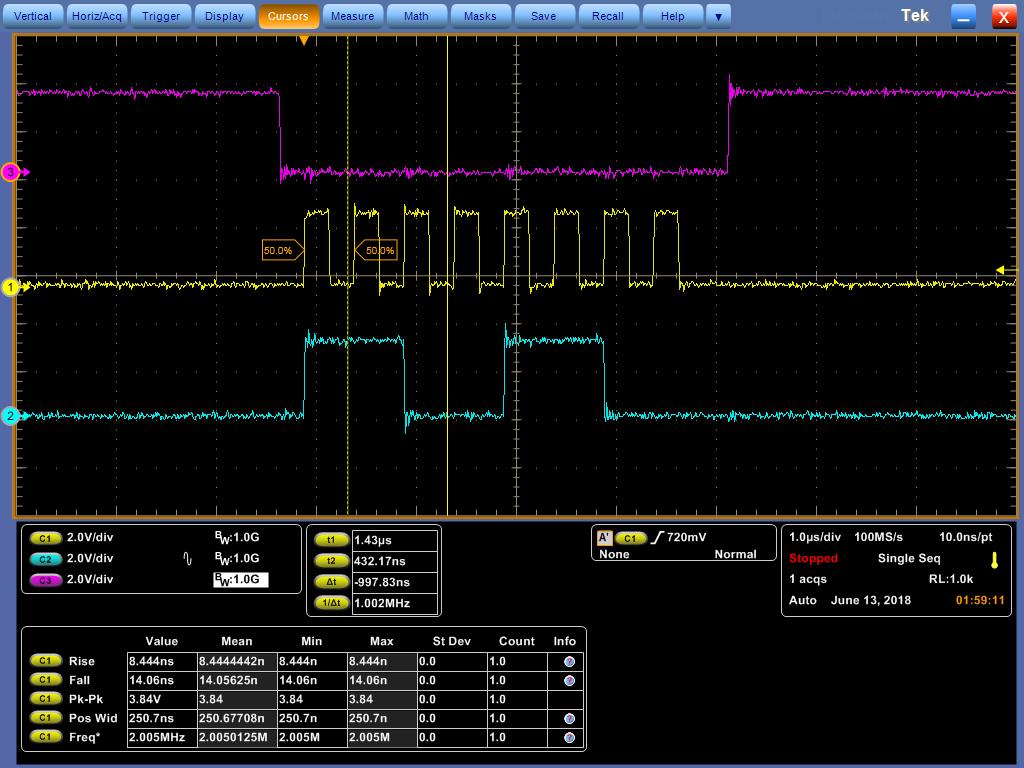
MODE : 00



MODE : 01



MODE : 10



MODE : 00 for data 10101010



LCD DISPLAY

Features of character LCD

* Implements the industry-standard Hitachi HD44780 LCD display driver chip protocol
* Requires only seven I/O pins on one I/O port
* Contains built-in character editor to create user-defined custom characters
* Supports horizontal and vertical bar graphs

CODE FOR LCD DISPLAY

#include "project.h"

int main(void)

{

CyGlobalIntEnable;

for(;;)

{

lcd\_Start();

lcd\_Position(0,0);

lcd\_PrintString("LCD");

lcd\_Position(1,2);

lcd\_PutChar(0x62);

CyDelay(1000);

lcd\_ClearDisplay();

lcd\_Position(1,1);

lcd\_PutChar(0x61);

lcd\_Position(0,5);

lcd\_PrintString("string");

lcd\_Position(1,4);

lcd\_WriteData(0x01);

CyDelay(1000);

lcd\_ClearDisplay();

}

}

The above program was written to display LCD.

RESULTS

* At position (0,0) “LCD” was displayed. At position (1,2) “b” was displayed for a stipulated period of time.
* Next “a” was displayed at position (1,1), “string” was displayed at position (0,5) and hyphen was displayed at position (1,4) for a stipulated period of time.

UART

CODE FOR UART

#include "project.h"

int main(void)

{

uint8 a;

UART\_Start();

lcd\_Start();

CyGlobalIntEnable;

UART\_LoadRxConfig();

for(;;)

{

while(!(UART\_GetRxBufferSize()));

a=uart\_GetChar();

lcd\_Position(0,0);

lcd\_PutChar(a);

}

}

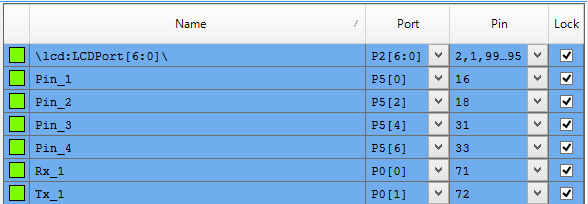


FIG 7: PORT LIST

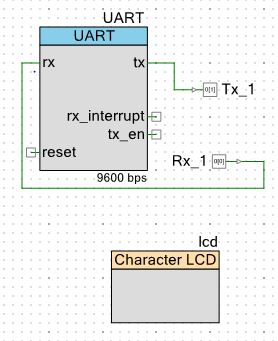
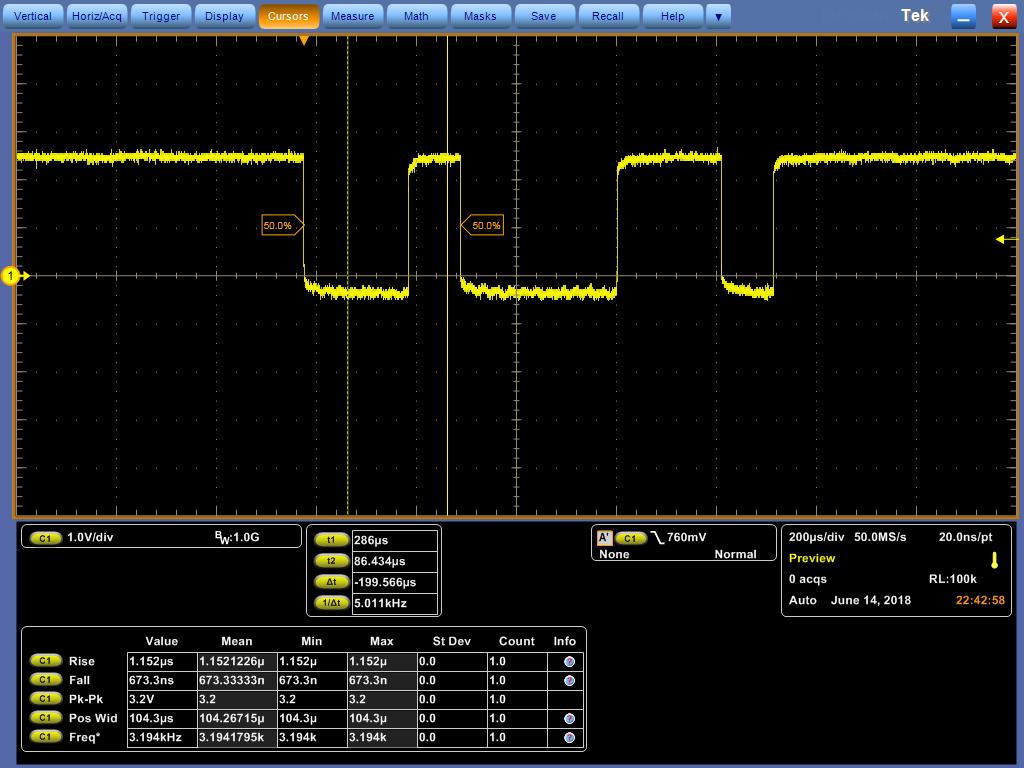


FIG 8: UART WITH LCD

Transferring 8 bits data using UART. And the same was displayed on CRO

( FIG 9 AND 10 )





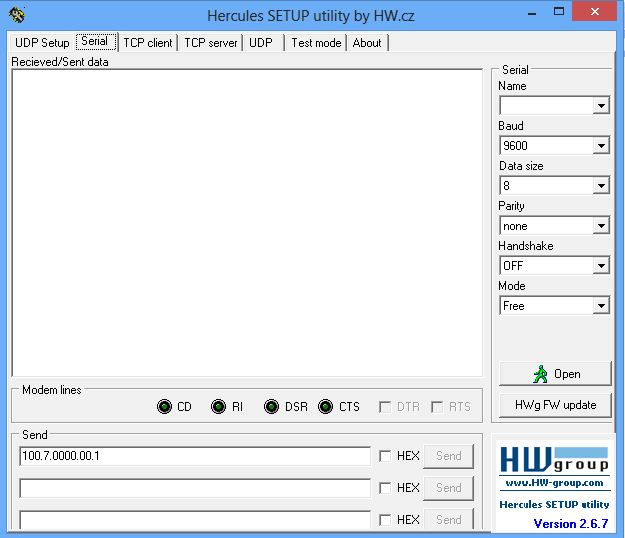


FIG 11: HERCULUS SETUP

The above image shows the herculus setup through which data was sent.

ADC

CODE FOR ADC

#include <project.h>

int main()

{

uint16 output;

LCD\_Start();

CyGlobalIntEnable;

ADC\_Start();

for(;;)

{

output = ADC\_CountsTo\_mVolts( ADC\_Read32() ) ;

LCD\_ClearDisplay();

LCD\_Position(0,0);

LCD\_PrintString("V(mv) = ");

LCD\_Position(0,9);

LCD\_PrintNumber(output);

CyDelay(1000);

}

}

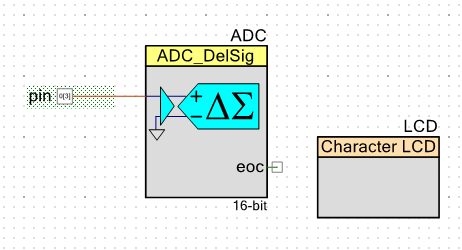


FIG 12: ADC AND LCD

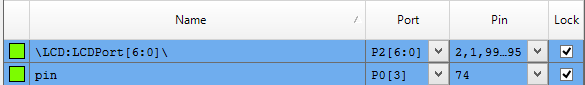


FIG 13: PORT LIST

We have studied delta sigma analog to digital convertor. The varying voltage was displayed through LCD and was verified using a multimeter.

UART AND MASTER SPI CODE

#include "project.h"

int main(void)

{

CyGlobalIntEnable;

//for(;;)

{

lcd\_Start();

spi\_Start();

}

uint8 a,b,c,z1,z,e1,e,f,g,h,i1,i,j,cc,cc1,cc2,cc3,cc4,P1,P;

uint32 fout;

uint16 flsb, fmsb,att,att1,cb,cb1,k;

UART\_Start();

lcd\_Start();

UART\_LoadRxConfig();

for(;;)

{

spi\_Start();

while(!(UART\_GetRxBufferSize()));

{

a=UART\_GetChar();

lcd\_Position(0,0);

lcd\_PrintString("a=");

lcd\_Position(0,2);

lcd\_PutChar(a);

}

while(!(UART\_GetRxBufferSize()));

{

b=UART\_GetChar();

lcd\_Position(1,0);

lcd\_PrintString("b=");

lcd\_Position(1,2);

lcd\_PutChar(b);

}

while(!(UART\_GetRxBufferSize()));

{

c=UART\_GetChar();

lcd\_Position(1,10);

lcd\_PrintString("c=");

lcd\_Position(1,12);

lcd\_PutChar(c);

}

fout = (a-0x30)\*100 + (b-0x30)\*10 + (c-0x30);

fout = fout\*1000000;

flsb = fout;

fmsb = fout>>16;

CyDelay(1000);

lcd\_ClearDisplay();

lcd\_Position(0,0);

lcd\_PrintString("flsb=");

lcd\_Position(0,7);

lcd\_PrintHexUint16(flsb);

CyDelay(1000);

lcd\_ClearDisplay();

lcd\_Position(0,0);

lcd\_PrintString("fmsb=");

lcd\_Position(0,7);

lcd\_PrintHexUint16(fmsb);

CyDelay(1000);

lcd\_ClearDisplay();

lcd\_Position(0,0);

lcd\_PrintString("fout=");

CyDelay(100);

lcd\_Position(0,7);

lcd\_PrintHexUint16(fmsb);

CyDelay(100);

lcd\_Position(0,11);

lcd\_PrintHexUint16(flsb);

CyDelay(1000);

lcd\_ClearDisplay();

z1=UART\_GetChar();

z=UART\_GetChar();

cc = (z-0x30);

{

if(cc==1)

{

lcd\_Position(0,0);

lcd\_PrintString("ReadRun");

}

if(cc==4)

{

lcd\_Position(0,0);

lcd\_PrintString("ChangeFreq");

}

if(cc==0)

{

lcd\_Position(0,0);

lcd\_PrintString("ReadRSSI");

}

if(cc==7)

{

lcd\_Position(0,0);

lcd\_PrintString("ChangeAtten");

}

if(cc==3)

{

lcd\_Position(0,0);

lcd\_PrintString("ChangeControl");

}

e1=UART\_GetChar();

CyDelay(1);

}

CyDelay(1000);

lcd\_ClearDisplay();

e=UART\_GetChar();

cc1 = (e-0x30);

if(cc1==0)

{

lcd\_Position(0,0);

lcd\_PrintString("R/F");

}

if(cc1==1)

{

lcd\_Position(0,0);

lcd\_PrintString("BITE");

}

f=UART\_GetChar();

cc2 = (f-0x30);

if(cc2==0)

{

lcd\_Position(0,8);

lcd\_PrintString("LNON");

}

if(cc2==1)

{

lcd\_Position(0,8);

lcd\_PrintString("LNOFF");

}

g=UART\_GetChar();

cc3 = (g-0x30);

if(cc3==0)

{

lcd\_Position(1,0);

lcd\_PrintString("NB");

}

if(cc3==1)

{

lcd\_Position(1,0);

lcd\_PrintString("WB");

}

h=UART\_GetChar();

cc4 = (h-0x30);

if(cc4==0)

{

lcd\_Position(1,8);

lcd\_PrintString("AGCFAST");

}

if(cc4==1)

{

lcd\_Position(1,8);

lcd\_PrintString("AGCSLOW");

}

CyDelay(2000);

lcd\_ClearDisplay();

i1=UART\_GetChar();

i=UART\_GetChar();

j=UART\_GetChar();

P1=UART\_GetChar();

P=UART\_GetChar();

if (P==0x30)

PTT\_Write(0);

else if (P==0x31)

PTT\_Write(1);

att = (i-0x30)\*10 + (j-0x30);

att1=att\*2;

lcd\_Position(0,0);

lcd\_PrintHexUint16(att1);

CyDelay(1000);

lcd\_ClearDisplay();

cb=(cc<<13)+(cc1<<11)+(cc3<<9)+(cc4<<7)+(att1);

lcd\_PrintHexUint16(cb);

lcd\_ClearDisplay();

lcd\_Position(0,0);

lcd\_PrintHexUint16(cb);

lcd\_Position(0,4);

lcd\_PrintHexUint16(fmsb);

lcd\_Position(0,8);

lcd\_PrintHexUint16(flsb);

spi\_WriteTxData(cb);

CyDelayUs(4);

spi\_WriteTxData(fmsb);

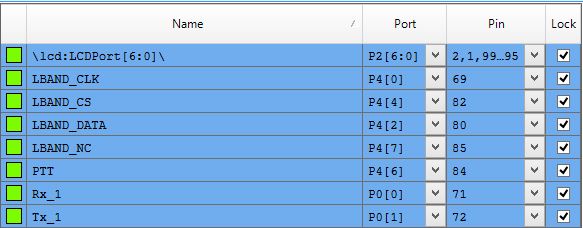
CyDelayUs(4);

spi\_WriteTxData(flsb);

CyDelayUs(4);

}

}



SLAVE SPI

CODE FOR SPI SLAVE

#include "project.h"

CY\_ISR\_PROTO(GPIOIsrHandler);

int main(void)

{

isr\_GPIO\_StartEx(GPIOIsrHandler);

CyGlobalIntEnable;

SPIS\_Start();

lcd\_Start();

for(;;)

{

lcd\_ClearDisplay();

lcd\_Position(0,0);

lcd\_PrintString("waiting");

CyDelay(1000);

}

}

CY\_ISR(GPIOIsrHandler)

{

isr\_GPIO\_ClearPending();

uint16 j,k,l;

while(!(SPIS\_GetRxBufferSize()));

j=SPIS\_ReadRxData();

lcd\_ClearDisplay();

lcd\_Position(0,0);

lcd\_PrintHexUint16(j);

while(!(SPIS\_GetRxBufferSize()));

k=SPIS\_ReadRxData();

lcd\_Position(0,8);

lcd\_PrintHexUint16(k);

while(!(SPIS\_GetRxBufferSize()));

l=SPIS\_ReadRxData();

lcd\_Position(1,0);

lcd\_PrintHexUint16(l);

CyDelay(5000);

SPIS\_ClearRxBuffer();

}

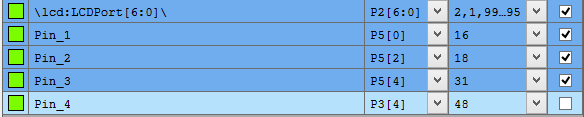
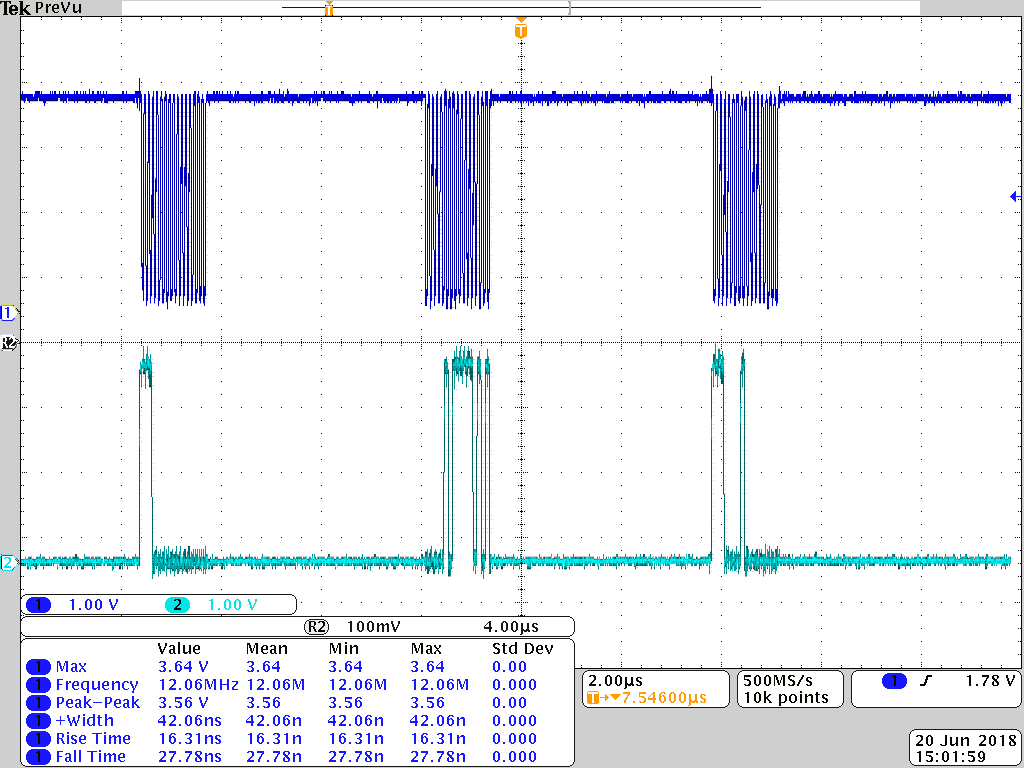


FIG 15: PORT LIST

IMAGE SHOWING THE CLOCK AND DATA TRANSMITTED THROUGH SPI (FIG 16)



The first channel (dark blue) indicates the clock signal. The second channel indicates the data transmitted. The baseband is 48 bits wide and divided into a control word of 16 bits and a frequency component of 32 bits. The detailed representation of the above image is shown below.

**Data Format for sending frequency & Control word:**

Three words, each containing 16 bits, are to be sent from baseband as shown below.

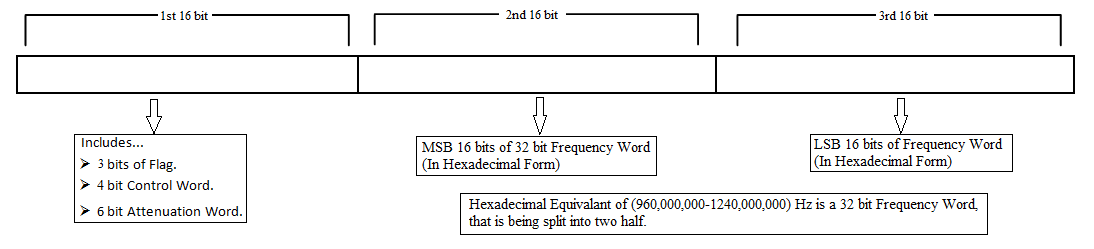


Figure 1: WORD Format

**1st 16 bits:**

Should be send as per the format shown below.

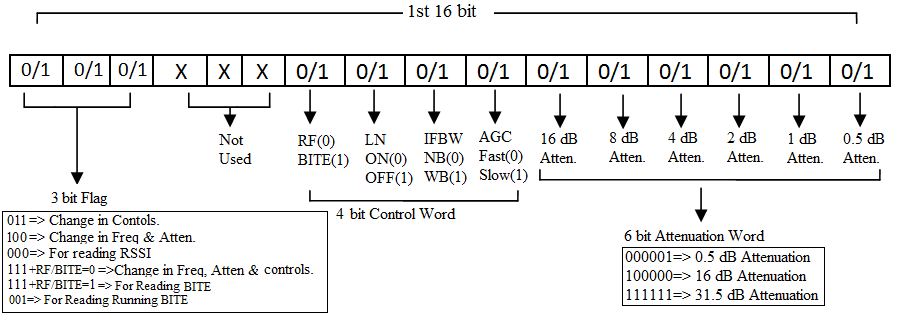


Figure 2: 1st 16 bit word

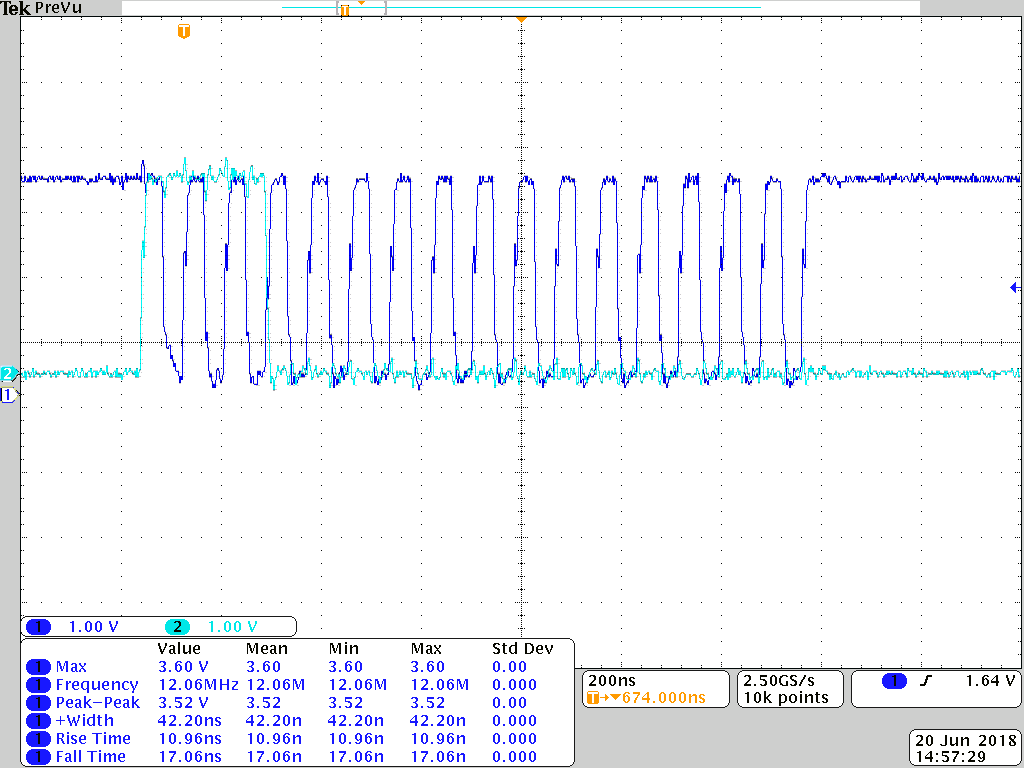


FIG 17: CONTROL WORD

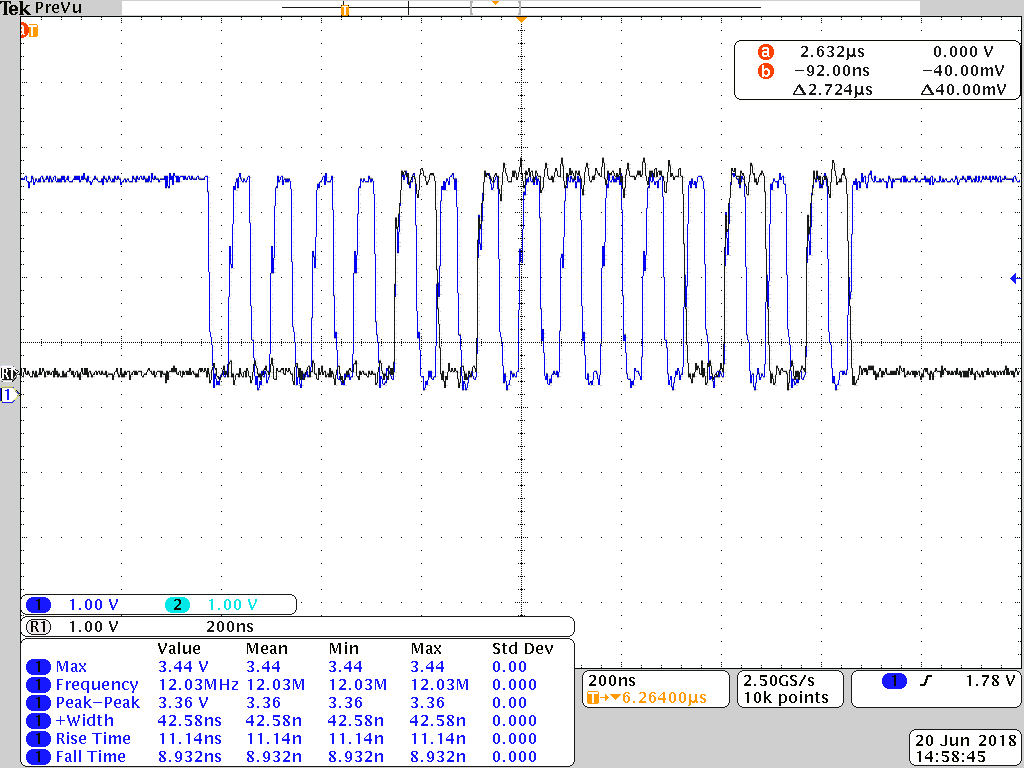


FIG 18:FREQUENCY(MSB)

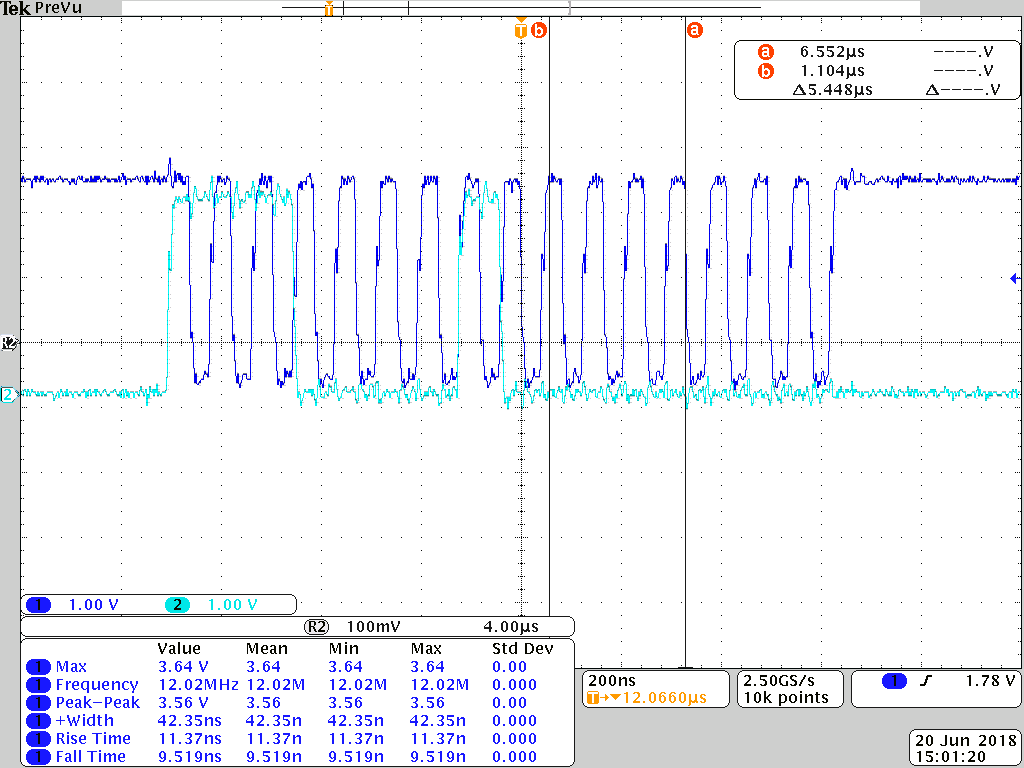


FIG 19: FREQUENCY (LSB)

The above three images indicates the baseband sent through SPI. The first image is that of a control word assuming we have sent the data as 7.0000.00.0.

7.0000.00.0

7.0000.00.0 : 7 indicates 111 which represents change in attenuation state.

7.0000.00.0 : 0 indicates RF level.

7.0000.00.0 : 0 indicates LN on.

7.0000.00.0 : 0 indicates NB.

7.0000.00.0 : 0 indicates AGC fast.

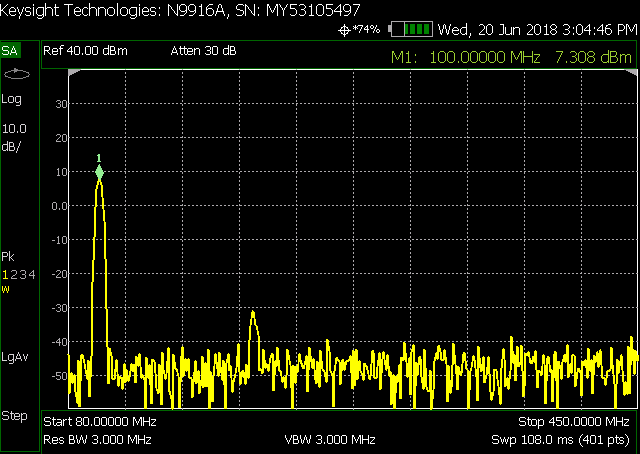
7.0000.00.0 : 00 indicates attenuation.

7.0000.00.0 : 0 indicates that it acts as a transmitter.

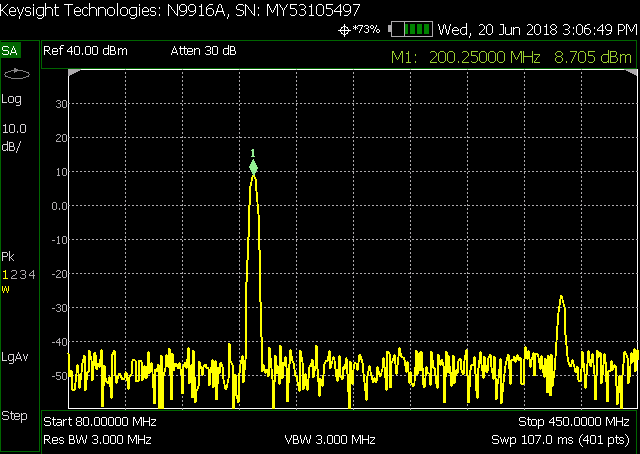
The next two images is that of frequency, we have given a frequency of 100 MHZ, which is represented as 05F5E100.

IMAGES OBTAINED FROM SPECTRUM ANALYSER WHEN VARIOUS FREQUENCIES WERE SENT TO THE TRANSRECEIVER ( TRANSMITTER ) (FIG 20 TO 23)

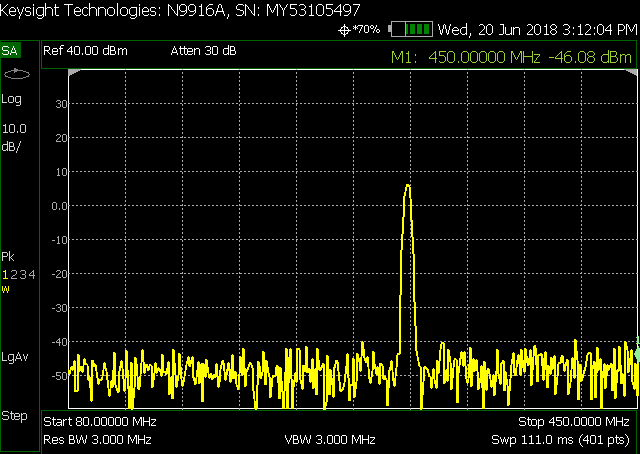
f= 100MHZ



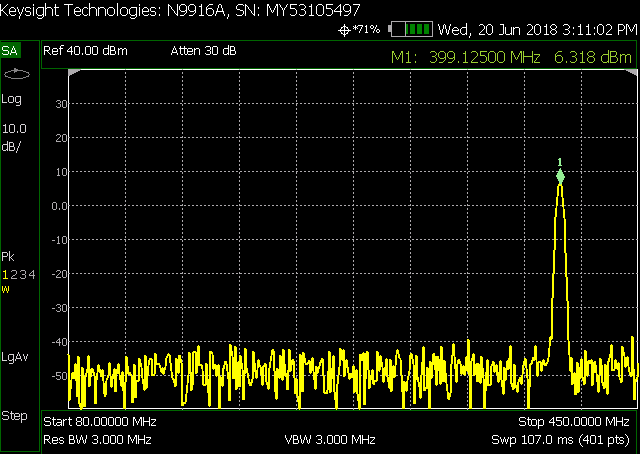
f=200MHZ



f=300MHZ

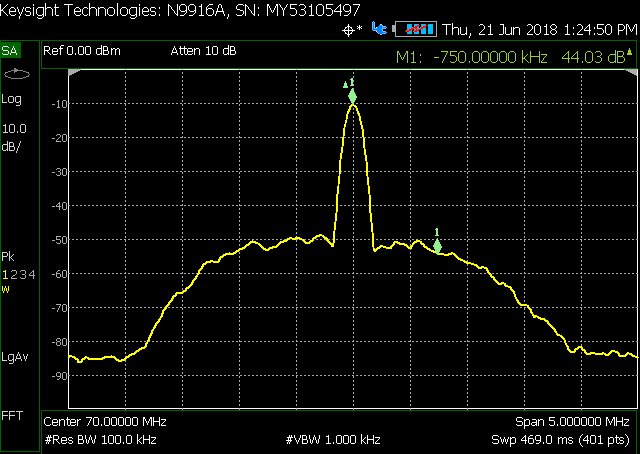


f=399MHZ

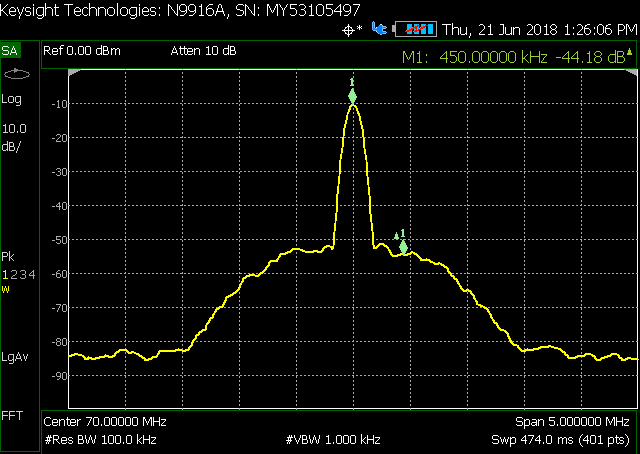


IMAGES OBTAINED FROM SPECTRUM ANALYSER WHEN FREQUENCY OF 100MHZ WAS SENT TO THE TRANSRECEIVER ( RECEIVER ) (FIG 24 TO 25)

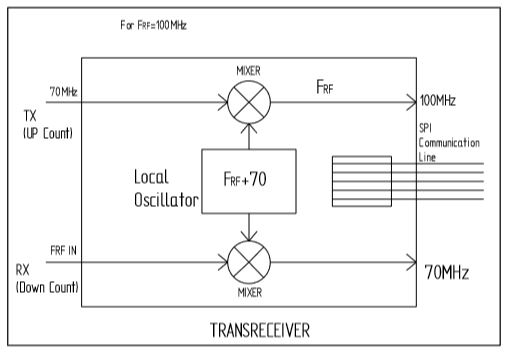
WIDEBAND

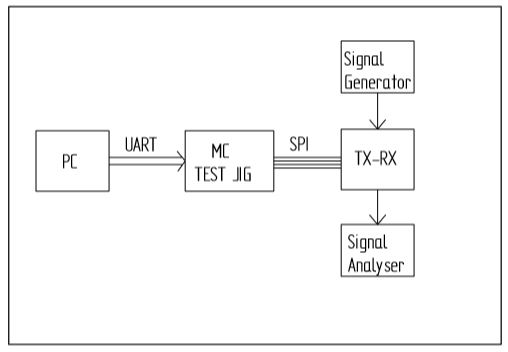


NARROWBAND



Block Diagram:





**CADDS:** Computer Aided Design And Drafting Software

Computer aided design and drafting (CADD) is a subfield of engineering which deals with the design and drafting of objects and materials through the use of specialized software that visualizes designs as modular 3D computer models.

Computer aided design (CAD) software is used for creating the 3D models complete with detailed documentation such as dimensions, materials used and even details the design process.

Even in PCB designing CADDS is used.

The output of the CADDS in PCB designing is GERBER data and drill data.

The standards followed are IPC(Associate connecting Electronic Industries)

The parameters that has to be maintained while designing are

1. Power

2. Audio

3. RF

4. Video

5. Digital

* Power:

For high voltage more spacing must be given

* Audio:

Simple connections are enough, the grounding (return path) connection should be proper

* RF:

In RF only power is considered, voltage is not considered.

In RF components are closely packed. (less spacing)

* Video
* Digital

**NAVAL SYSTEMS:**

**LYNX UX-GFCS**



LONG RANGE GFCS

* GFCS: Gun Fire Control System
* The range is upto 20km
* The main sensors used are
* **RADAR:** Radio Detection and Ranging

Survelliance RADARs has range more than 200kms

Tracking RADARs has range less than 45kms

* **XBAND**: In RADAR engineering, the frequency range is specified by the IEEE

at 8.0 to 12.0 GHz. The X band is used for RADAR, Satellite communication and wireless computer networks.

* TV camera
* Thermal Vision
* SRGM(Super Rapid Gun Mount) upto 120 shells
* AK-36: Pumps upto 4500 shells
* Short range guns have range upto 4kms
* Medium range guns have range upto 13 kms
* The Standards followed are MIL grade
* HAT: Harbour Acceptance Test
* SAT: Sea Acceptance Test

**PASSIVE VACCUM DEVICES:**

**VACCUM INTERRUPTERS**

****

**VACCUM INTERRUPTERS**

* Switching mechanism needs vaccum interrupters
* From 12 KV to 240Vs
* Vacuum interrupters are used in utility power transmission systems and in power-distribution systems for railway and industrial plants.
* In electrical engineering, a **vacuum interrupter** is a switch which uses electrical contacts in a vacuum. It is the core component of medium-voltage switches.
* Separation of the electrical contacts results in a vacuum arc, which is quickly extinguished.
* Since the arc is contained within the interrupter, switchgear using vacuum interrupters can be very compact compared with switchgear using air or oil as the arc-suppression medium.
* Vacuum interrupters can be used for circuit-breakers and load switches. Circuit-breaker vacuum interrupters are primarily used in the power sector in substation and power-grid facilities, and load-switching vacuum interrupters are used for power-grid end users.
* The materials used to make vaccum interrupters are copper, Chromiun copper, Brazzing foils, Covar.

**MICROWAVE SUPER COMPONENTS:**

**RADAR**

**C BAND ANTENNA:**

****

* The CBAND QTRM is used in CBAND antenna.
* The 7 Mtr C-Band Earth Station Antenna (ESA) is designed for satellite communication operation in C-Band worldwide.
* The **C-band** is a designation by the Institute of Electrical and Electronics Engineers (IEEE) for a portion of the electromagnetic spectrum in the microwave range of frequencies ranging from 4.0 to 8.0 gigahertz (GHz)
* Nearly all C-band communication satellites use the band of frequencies from 3.7 to 4.2 GHz for their downlinks, and the band of frequencies from 5.925 to 6.425 GHz for their uplinks.

**X-QTRM:**

* QTRM: Quad Trans Receiver Mode
* Automatic shut-down if internal temperature reaches a critical limit where damage could occur. Hysteresis applies
* Can operate over a wider bandwidth at reduced performance. Limited by circulator bandwidth.
* Up to 3dB reduction in useable attenuation range due to Calibration.

**MMF: MASS MANUFACTURING FACILITIES**

**SURFACE MOUNT TECHNOLOGY:**

* The highest quality automatic pick and place machine with the most advanced technology
* Placement rates up to 20000CPH
* Up to 88 tape feeders, 132 - 8mm tape feeders with bank feeders
* The best GUI (graphical user interface) in its class using Windows-based software allows programming in minutes