线路故障自动检测系统（J 题）

摘要

本装置以 STC32G 单片机为核心控制器件，结合 LM311 振荡电路等电路组成。LC 振荡电路作为装置的核心，通过监测 LM311 输出频率的变化来检测电感和电容元件的故障，并通过检测二极管两边 IO 口信号的变化来判断二极管的故障问题。这种方案具有结构简单、调节便捷的特点。为了实现只允许按一次启动键的功能，我们采用了单片机主控模块发送控制信号，并通过继电器控制电路中各个部分的开关，从而实现对元件故障位置和类型的准确定位和识别。同时，通过精心设计的程序，单片机可将检测结果传输到液晶屏上显示，使操作人员能够直观地了解电路中的故障情况。

关键词：STC32G　　振荡电路　　自动检测

目录

1 总体设计方案

1.1 比较与选择

方案一：NE555 无稳态 RC 谐振电路。

NE555 集成电路是一种常见且价格低廉的集成电路，其可以被用来产生方波信号的无稳态 RC 谐振电路。NE555 集成电路具有简单的引脚布局和操作方式，使得使用起来非常方便。但是 NE555 的无稳态 RC 谐振电路精度较低：相对于一些专用的振荡器电路，NE555 无稳态电路的频率和幅值稳定性较差，容易受到外部条件的影响。

方案二：LC 振荡电路。

LC 振荡电路可以产生非常稳定的振荡信号，频率稳定度较高；通过调整电感和电容的数值，可以实现广泛的频率范围，适应不同的应用需求。通过调整电容和电感元件的数值，LC 振荡电路可以实现对电容和电感值的精确测量，满足高精度测量的要求。

在实验中，我们对 NE555 无稳态 RC 谐振电路进行了测试，发现它并不能有效地检测线路网络中的电容短路问题。然而，在实验 LC 振荡电路时，我们观察到它能够准确地检测到线路网络中出现的任何元件的单一问题。故我们选择了方案二。

1.2 总体方案描述

根据 1.1 分析，我们选择了方案二作为最终方案。LC 谐振电路可以检测线路网络中的电感和电容元件故障，但无法检测二极管故障。为了解决这个问题，我们考虑使用单片机 IO 口检测高低电平的方法来进行二极管的故障检测，并通过继电器来实现模式的切换，以确保线路之间不会相互干扰。此外，我们还将使用软件来进行两种模式之间的协调切换以做到题目要求的自动检测，总框图如图 1

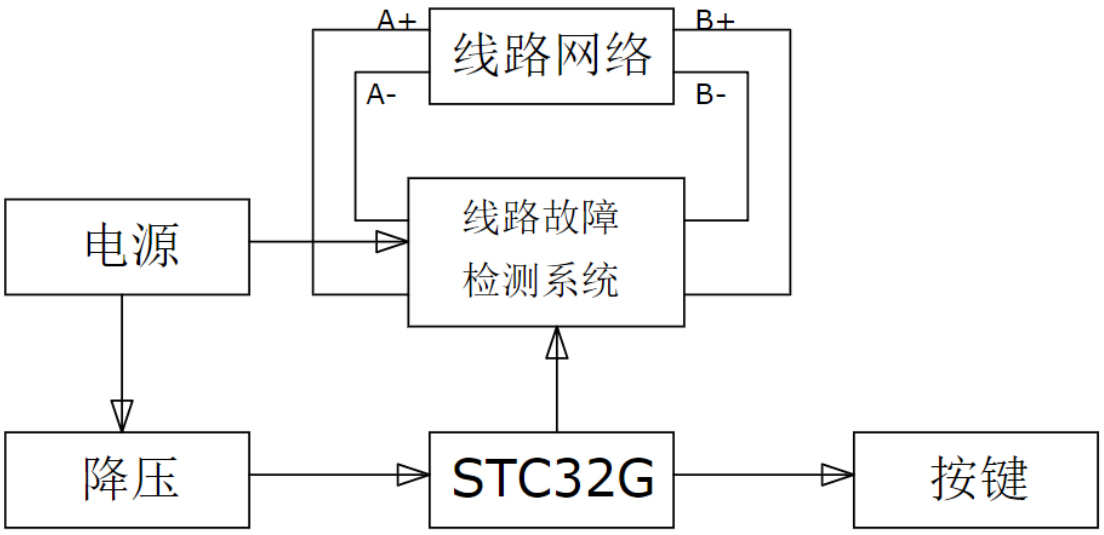

图 1: 总框图

2 理论分析与设计

2.1 检测原理分析

通过单片机采集检测系统输出的频率变化，我们可以判断线路网络中电感或电容是否正常；同时，通过单片机采集的电压信号，我们可以判断二极管是否存在故障。

2.1.1 检测电感电容故障原理分析

测量的原理是由单片机通过对 LM311 输出端进行频率的采集。我们将线路网络视为一个整体，将其作为 LC 振荡电路中的电感 L 和电容 C。当线路网络中的电感或电容出现故障时，LC 振荡网络就会发生改变，进而导致 LM311 的输出频率发生变化。

2.1.2 检测二极管故障原理分析

二极管开路故障检测的原理是将二极管的 B-或 A-引脚与单片机的 IO 口相连。当将输入端的 IO 口设置为高电平时，另外一边的 IO 口也会接收到高电平信号。通过这种连接方式，我们可以利用单片机的 IO 口来检测二极管是否正常工作。然而，如果二极管存在开路故障，输入端 IO 口的高电平信号将无法传递到另一端的 IO 口，导致另一端无法接收到高电平。这种现象表明二极管存在开路故障。

2.2 故障定位原理分析

通过单片机采集 LM311 输出端的频率，并与预设的正常频率进行比较，我们可以准确地判断是哪个电感或电容元件出现故障。另一方面，根据输入高电平，通过对接收信号的观察，我们可以确定二极管是否正常工作。

2.2.1 电感电容故障定位原理分析

当线路网络中的电感或电容出现故障时，这会导致 LM311 振荡电路的输出频率发生变化。不同的电感或电容元件故障会引起不同的输出频率变化。在电路正常工作的情况下，我们可以根据 LC 振荡公式得到一个预设的频率。当电感或电容元件出现故障时，振荡电路的参数发生改变，从而导致振荡频率的变化。当电路正常工作时，根据 LC 振荡公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 134KHz \tag{1}$$

当 C1 开路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 140KHz \tag{2}$$

当 C2 开路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 159KHz \tag{3}$$

当 C3 开路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 153KHz \tag{4}$$

当 C4 开路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 137KHz \tag{5}$$

当 L1 开路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 176KHz \tag{6}$$

当 L2 开路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 241KHz \tag{7}$$

当 L3 开路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 576KHz \tag{8}$$

当 C1 短路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 421KHz \tag{9}$$

当 C2 短路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 537KHz \tag{10}$$

当 C3 短路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 1408KHz \tag{11}$$

当 C4 短路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 820KHz \tag{12}$$

当 L1 短路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 135KHz \tag{13}$$

当 L2 短路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 141KHz \tag{14}$$

当 L3 短路时，由上述公式可得

$$f = \frac{1}{2\pi\sqrt{LC}} = 143KHz \tag{15}$$

综上所述，为了判断出具体是哪个电感电容元件故障，我们可以通过单片机采集 LM311 输出频率的变化并记录下来。将其与预设的正常频率进行对比，我们就能够准确地判断是哪个电感或电容元件故障。

2.2.2 二极管故障定位原理分析

二极管开路故障定位的原理是将二极管的 B-或 A-引脚与单片机的 IO 口相连。左右两个 IO 口分别设置为高电平，通过检测接收到的信号可以判断二极管的工作状态。如果接收到高电平信号，表示对应方向的二极管正常；而如果接收到低电平信号，则说明对应方向的二极管损坏。通过对左右两边的检测结果进行判断，可以确定二极管的故障位置或者判断二极管是否正常工作。检测图如图 2，电路图如图 3

	输入端值		输出端值		状态
IO 口	1	0	1	0	
IO_1	√		√		D2 正常
IO_1	√			√	D2 开路
IO_2	√		√		D1 正常
IO_2	√			√	D1 开路

图 2: 二极管故障定位简易图

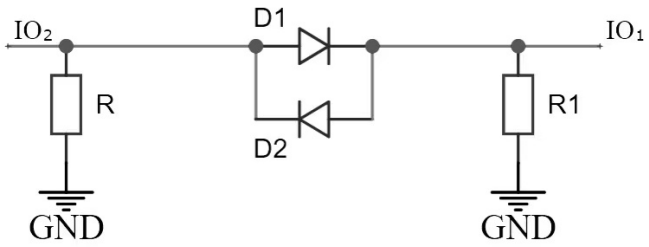

图 3: 二极管故障定位电路图

3 电路与程序设计

3.1 程序设计

3.1.1 程序框架设计

如图 4 该系统以 STC32G 单片机为控制核心，利用其强大的处理能力和丰富的接口资源，实现对故障频率的信号采集和处理。在系统中，首先对每种故障产生的不同频率进行信号采集，通过单片机的输入引脚将信号输入到系统中。接下来，系统将采集到的信号和原始程序进行校准。通过比较采集到的信号和预设的正常频率范围，系统可以判断是否存在故障，并进一步确定是哪种元件故障。在测试过程中，系统将根据不同频率的输入信号，实时在屏幕上反应相应的问题。以便操作者直观地了解故障情况。

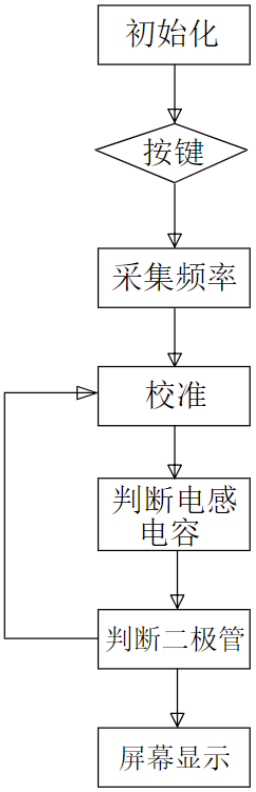

图 4: 程序框架图

3.1.2 程序功能设计

为实现题目要求的软件功能，并方便在测试过程中，保证操作者测试方便 软件实现如下功能：

1) 能自动检测线路网络中故障的元件。

2) 能显示故障元件编号及故障类型。

3.2 硬件电路设计

3.2.1 电路框架设计

电路框图如图 5所示。该系统采用了单电源供电的设计，经过降压处理后，为整套系统供电。在系统中，线路间的连接和切换是由单片机控制的继电器来实现的。通过单片机的精确控制，我们能够实现对各个部分的连接和切换，以便进行准确的故障检测操作。当系统工作时，单片机根据预设的程序和逻辑，控制继电器的开关状态，从而实现对线路的连接和切换。通过合理的连接和切换，以便单片机进行频率的采集和分析。通过单片机的控制，我们可以根据采集到的频率信息，与预设的正常频率进行比较和分析，从而准确地判断是哪个电感或电容元件出现故障。

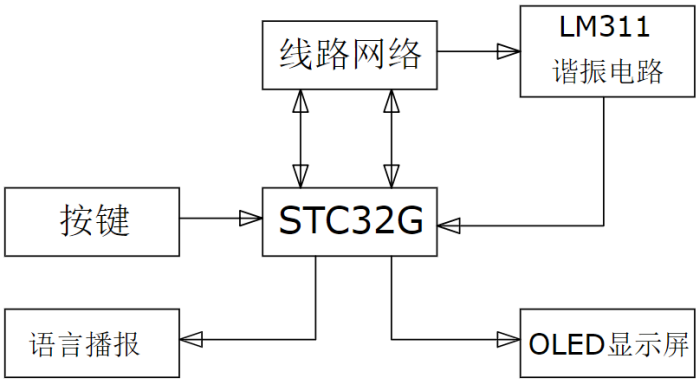

图 5: 硬件框架图

3.2.2 降压模块设计

本模块采用 78M05 芯片进行降压操作。78M05 是一款稳压器芯片，它可以将输入的高电压稳定地降压到 5V 输出。与其他降压芯片相比，78M05 在降压过程中消耗的电压较少，能够有效地提供稳定的电压输出。因此，使用 78M05 芯片进行降压操作可以确保系统获得稳定可靠的电源供应。模块如图 6

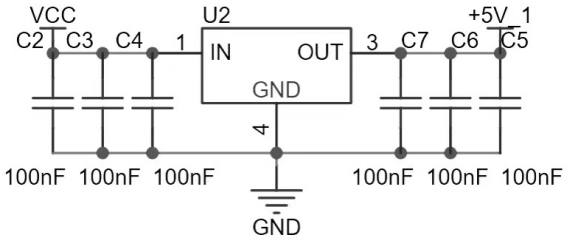

图 6: 降压模块图

3.2.3 检测模块设计

通过对 LM311 输出端进行频率的采样，然后依据采集的频率不同再计算出是哪个电容或电感出现故障。检测二极管的好坏时，用继电器控制线路的通断，保证互不干涉。模块大致如图 7

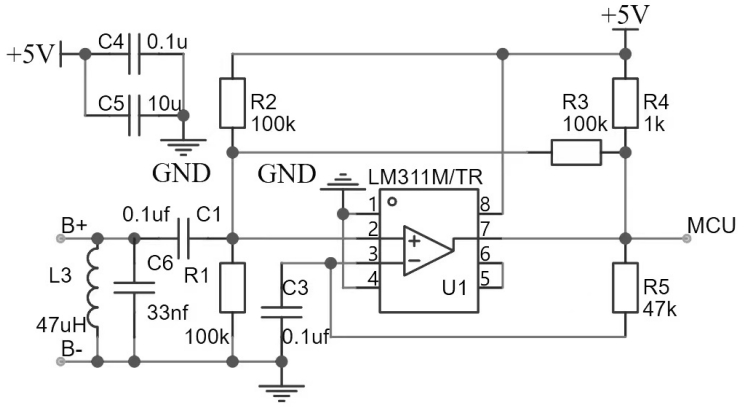

图 7: 检测模块图

4 测试与结果分析

4.1 测试环境及设备

测试在恒定温度下，使用电池供电。测试设备如表 1所示其中用示波器观察频率：

表 1: 测试设备

类型	万用表	示波器	直流稳压电源箱	数字交流毫伏表
型号	VC86E	DS1102E	QJ300-5S	TH1912A
规格	1/5	100MHz	30V/5A*2	5MHz

4.2 测试过程及结果

4.2.1 基本要求测试结果

1) 按基本要求，开关 S 为闭合状态。线路网络中存在一个电感元件或电容元件故障。按下启动键后，系统自动检测并显示故障元件编号及故障类型（断路和短路）。测试结果如表 2 所示设计能满足题目要求。

表 2: 基本要求测试结果

故障	L1 开	L2 开	L3 开	C1 开	C2 开	C3 开	C4 开
测试次数	50	50	50	50	50	50	50
成功率	100%	100%	100%	100%	100%	100%	100%
故障	L1 短	L2 短	L3 短	C1 短	C2 短	C3 短	C4 短
测试次数	50	50	50	50	50	50	50
成功率	96%	100%	100%	98%	92%	98%	100%

4.2.2 发挥要求（1）测试结果

2) 发挥部分 1，开关 S 为断开状态。线路网络中有一个二极管存在断路故障。同时，还存在一个电感或电容元件故障。按下启动键后，系统自动检测并显示故障元件编号及故障类型。测试结果如表 3 所示。

表 3: 发挥部分 1 测试结果

故障	L1,D1 开	L2,D1 开	L3,D1 开	C1,D1 开	C2,D1 开	C3,D1 开	C4,D1 开
测试次数	50	50	50	50	50	50	50
成功率	100%	100%	100%	100%	100%	100%	100%
故障	L1,D2 开	L2,D2 开	L3,D2 开	C1,D2 开	C2,D2 开	C3,D2 开	C4,D2 开
测试次数	50	50	50	50	50	50	50
成功率	100%	100%	100%	100%	100%	100%	100%
故障	L1,D1 短	L2,D1 短	L3,D1 短	C1,D1 短	C2,D1 短	C3,D1 短	C4,D1 短
测试次数	50	50	50	50	50	50	50
成功率	94%	100%	96%	96%	98%	100%	100%
故障	L1,D2 短	L2,D2 短	L3,D2 短	C1,D2 短	C2,D2 短	C3,D2 短	C4,D2 短
测试次数	50	50	50	50	50	50	50
成功率	92%	98%	96%	94%	100%	98%	100%

4.2.3 发挥要求（2）测试结果

3) 发挥部分 2，开关 S 为断开状态。线路网络或无故障，或只存在一个电感或电容元件故障，或二极管中有一个存在断路故障的同时还存在一个电感或电容元件故障。按下启动键后，系统自动检测并显示系统状态（是否有故障），以及在有故障时故障元件编号及故障类型。测试结果如表 4所示。

表 4: 发挥部分 2 测试结果

	L1	L2	L3	C1	C2	C3	C4
正常	正常	正常	正常	正常	正常	正常	正常
开路	100%成功	100%成功	100%成功	100%成功	100%成功	100%成功	100%成功
短路	100%成功	100%成功	100%成功	100%成功	100%成功	100%成功	100%成功
故障	L1,D1 开	L2,D1 开	L3,D1 开	C1,D1 开	C2,D1 开	C3,D1 开	C4,D1 开
测试次数	50	50	50	50	50	50	50
成功率	100%	100%	100%	100%	100%	100%	100%
故障	L1,D2 开	L2,D2 开	L3,D2 开	C1,D2 开	C2,D2 开	C3,D2 开	C4,D2 开
测试次数	50	50	50	50	50	50	50
成功率	100%	100%	100%	100%	100%	100%	100%
故障	L1,D1 短	L2,D1 短	L3,D1 短	C1,D1 短	C2,D1 短	C3,D1 短	C4,D1 短
测试次数	50	50	50	50	50	50	50
成功率	94%	96	100%	90%	98%	96%	100%
故障	L1,D2 短	L2,D2 短	L3,D2 短	C1,D2 短	C2,D2 短	C3,D2 短	C4,D2 短
测试次数	50	50	50	50	50	50	50
成功率	92%	98%	94%	98%	96%	94%	96%

5 结论

通过利用 LM311 比较器和 LC 振荡电路搭建基本的线路故障检测系统。让单片机接受 LM311 输出端频率并经过处理结合继电器实现自动检测，我们成功满足了设计要求。这个系统能够自动检测线路中的故障，并能够准确地判定和显示故障的位置和类型。

参考文献

[1] 赵巧妮. 基于单片机的电阻、电感、电容测试仪的设计. 湖南铁道职业技术学院. 2016.6.25.

[2] 蒋威，瓮嘉民，陈孝宗. 电感电容频率一体化简易测量仪设计. 中国电机工程学报, 2017.4.21.

[3] 刘树林，程红丽. 低频电子线路 [M]. 北京: 机械工业出版社.2007.8.

[4] Altera Corporation. Avalon Intreface Specification，May，2006.

[5] 韦炜，电容器和电感器交流参数测量方法的对比研究 [J]. 现代电子技术.2012(13).

[6] 何富运，罗晓曙，数字式电感电容测量仪的设计 [J]. 现代电子技术.2008（22）.

[7] 苏梁，便携式电容测试仪的设计与研究 [D]，华中科技大学.2007.

[8] 黄璞，黎会鹏. 简易电容测试仪的设计. 科技咨询.2013.8.13.

6 附录 1：电路原理图

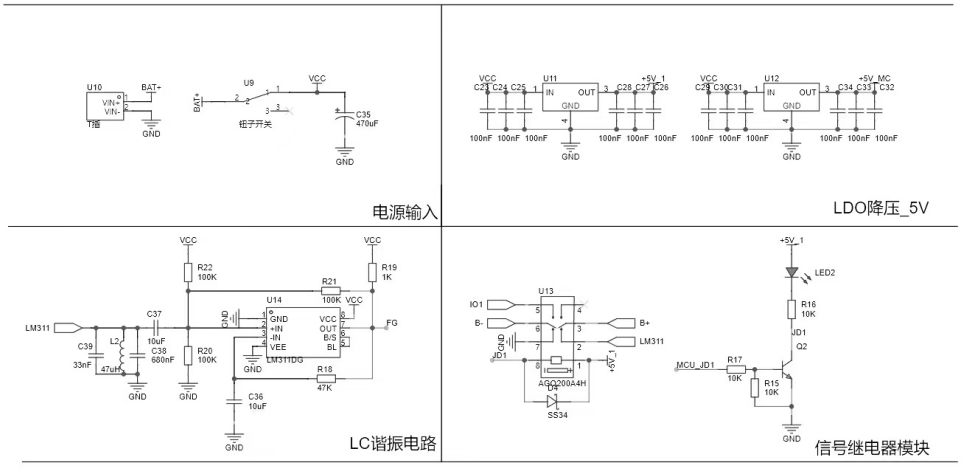

图 8: 电路原理图

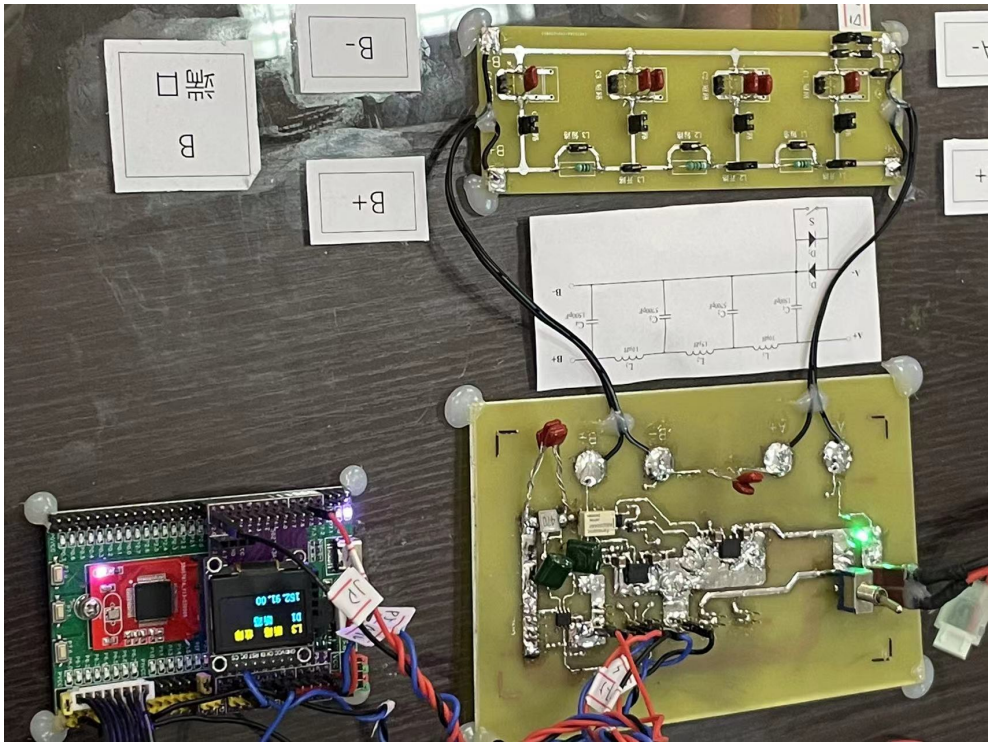

图 9: 实物图

7 附录 2：源代码

```
#include <config.h>
#include <fft.h>

#define PII  6.28318530717958647692528676655900
float INPUT_A[N];
float dataR_A[N];//实数
float dataI_A[N]={0};//虚数
float w_A[N];//总抗性
static float sin_tab_A[N];
static float cos_tab_A[N];

float INPUT_B[N];
float dataR_B[N];//实数
float dataI_B[N]={0};//虚数
float w_B[N];//总抗性
static float sin_tab_B[N];
static float cos_tab_B[N];

void InitForFFT_A(void)
{
unsigned int i;
for ( i=0;i<N;i++ )
{
sin_tab_A[i]=sin(PII*i/N);
cos_tab_A[i]=cos(PII*i/N);
}
}
void InitForFFT_B(void)
{
unsigned int i;
for ( i=0;i<N;i++ )
{
sin_tab_B[i]=sin(PII*i/N);
cos_tab_B[i]=cos(PII*i/N);
}
}

void FFT_A(void)
{
unsigned int x0,x1,x2,x3,x4,x5,x6,xx=0;
unsigned int i,j,k,b,p,L;
```

```
42                  float TR,TI,temp;
43                  float avg_w_value = 0;
44                  for (i=0;i<N;i++)
45                  {
46                  dataR_A[i]=(float)INPUT_A[i];
47                  dataI_A[i]=0.0f;
48                  w_A[i]=0;
49                  }
50                  /************** following code FFT *******************/
51                  for ( L=1;L<=7;L++ )
52                  {
53                  b=1; i=L-1;
54                  while ( i>0 )
55                  {
56                  b=b*2; i--;
57                  }
58                  for ( j=0;j<=b-1;j++ )
59                  {
60                  p=1; i=7-L;
61                  while ( i>0 )
62                  {
63                  p=p*2; i--;
64                  }
65                  p=p*j;
66                  for ( k=j;k<N;k=k+2*b )
67                  {
68                  TR=dataR_A[k]; TI=dataI_A[k]; temp=dataR_A[k+b];
69                  dataR_A[k]=dataR_A[k]+dataR_A[k+b]*cos_tab_A[p]+dataI_A[k+b
                        ]*sin_tab_A[p];
70                  dataI_A[k]=dataI_A[k]-dataR_A[k+b]*sin_tab_A[p]+dataI_A[k+b
                        ]*cos_tab_A[p];
71                  dataR_A[k+b]=TR-dataR_A[k+b]*cos_tab_A[p]-dataI_A[k+b]*
                        sin_tab_A[p];
72                  dataI_A[k+b]=TI+temp*sin_tab_A[p]-dataI_A[k+b]*cos_tab_A[p];
73                  }
74                  }
75                  }
76                  for ( i=0;i<N/2;i++ ){
77                  w_A[i]= sqrt(dataR_A[i]*dataR_A[i]+dataI_A[i]*dataI_A[i]);
78                  }
79                  }
```