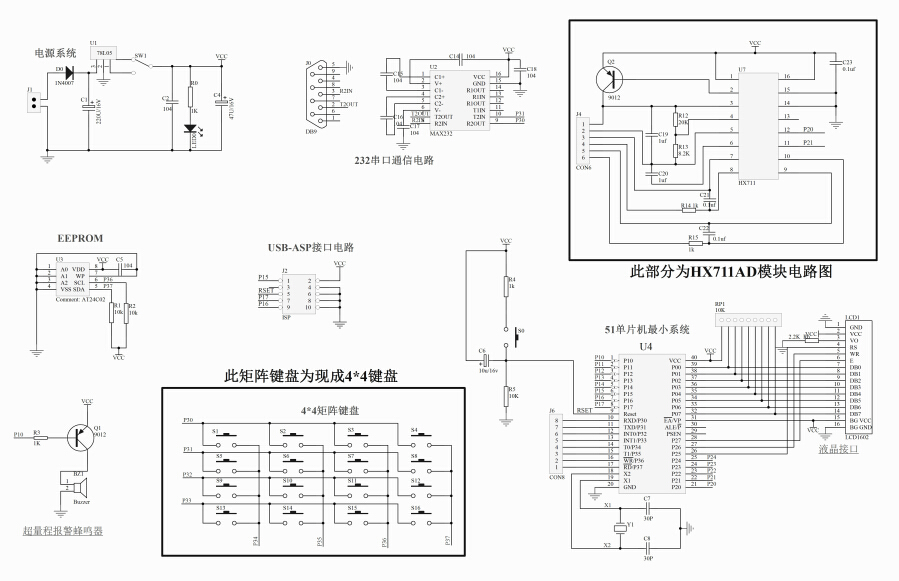
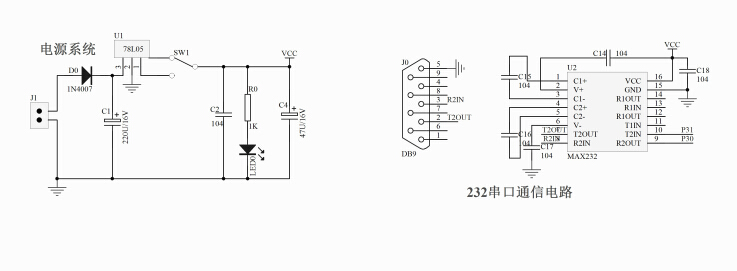
称重模块电路+程序（测试通过）

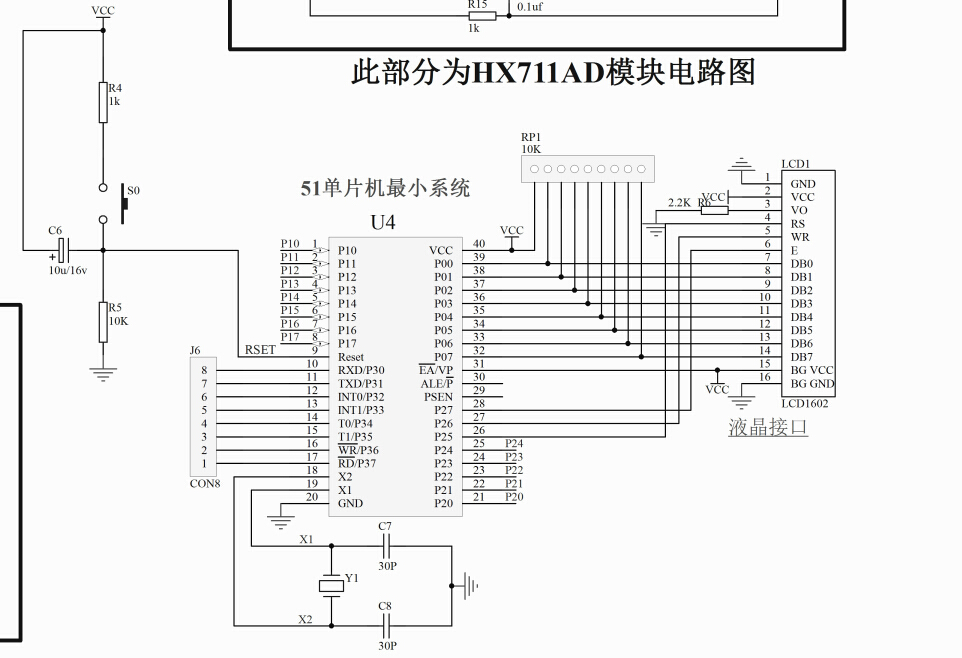


总体电路

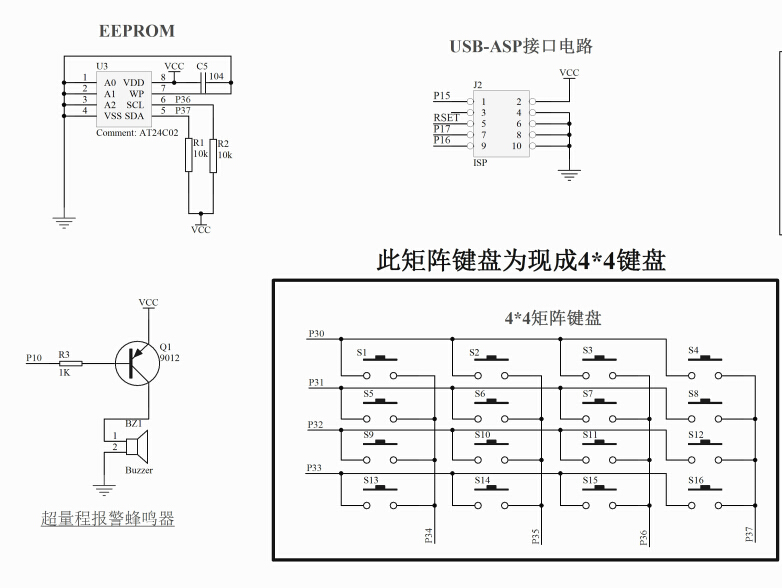
电源+串口通讯



单片机最小系统：



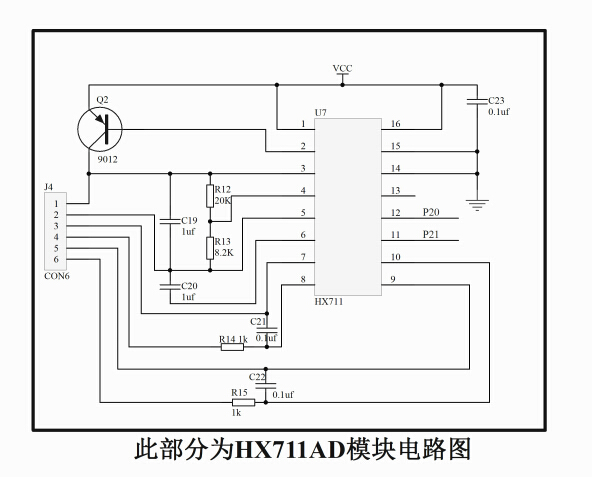
存储模块+下载模块+蜂鸣器+矩阵键盘



称重模块：

淘宝链接：

http://detail.tmall.com/item.htm?spm=a230r.1.0.0.tule1b&id=20046845949&ad\_id=&am\_id=&cm\_id=140105335569ed55e27b&pm\_id=&abbucket=5



主程序：

#include "main.h"

#include "LCD1602.h"

#include "HX711.h"

unsigned long HX711\_Buffer = 0;

unsigned long Weight\_Maopi = 0,Weight\_Shiwu = 0;

char Price\_Count = 0;

unsigned char KEY\_NUM = 0;

unsigned char Price\_Buffer[3] = {0x00,0x00,0x00};

unsigned long Money = 0;

bit Flag\_OK = 0;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//主函数

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void main()

{

Init\_LCD1602(); //初始化LCD1602

LCD1602\_write\_com(0x80); //指针设置

LCD1602\_write\_word("Welcome to use! "); //开机画面第一行

Delay\_ms(2000); //延时2s

loop:Price\_Count = 0;

Price\_Buffer[0] = 0;

Price\_Buffer[1] = 0;

Price\_Buffer[2] = 0;

Flag\_OK = 0;

LCD1602\_write\_com(0x80); //指针设置

LCD1602\_write\_word("+WEI |PRI | MON ");

LCD1602\_write\_com(0x80+0x40); //指针设置

LCD1602\_write\_word("0.000| . | . ");

Get\_Maopi(); //称毛皮重量

while(1)

{

if( Flag\_OK == 0)

{

Get\_Weight(); //称重

//显示当前重量

LCD1602\_write\_com(0x80+0x40);

LCD1602\_write\_data(Weight\_Shiwu/1000 + 0x30);

LCD1602\_write\_data('.');

LCD1602\_write\_data(Weight\_Shiwu%1000/100 + 0x30);

LCD1602\_write\_data(Weight\_Shiwu%100/10 + 0x30);

LCD1602\_write\_data(Weight\_Shiwu%10 + 0x30);

}

KEY\_NUM = KEY\_Scan();

if( KEY\_NUM != 0x55) //当返回的不是初值时候，确认按键按下。

{

if(KEY\_NUM == 16) //数字A键，去皮功能

{

Get\_Maopi(); //去皮

}

if(KEY\_NUM == 15) //数字B键清除键，二次测量

{

goto loop;

}

if(KEY\_NUM == 12) //数字C输入单价错误时返回上一步

{

Price\_Count--;

if( Price\_Count < 0)

{

Price\_Count = 0;

}

Price\_Buffer[Price\_Count] = 0; //清除上一个输入的数据

switch(Price\_Count)

{

case 0:

LCD1602\_write\_com(0x80+0x40+6);

LCD1602\_write\_data(' ');

break;

case 1:

LCD1602\_write\_com(0x80+0x40+7);

LCD1602\_write\_data(' ');

break;

case 2:

LCD1602\_write\_com(0x80+0x40+9);

LCD1602\_write\_data(' ');

break;

default : break;

}

}

if(KEY\_NUM == 13) //数字D键，计算总价

{

Money = Price\_Buffer[0] \* 100 + Price\_Buffer[1] \* 10 + Price\_Buffer[2];

Money = Money \* Weight\_Shiwu / 1000;

LCD1602\_write\_com(0x80+0x40+11);

LCD1602\_write\_data(Money/1000 + 0x30);

LCD1602\_write\_data(Money%1000/100 + 0x30);

LCD1602\_write\_data(Money%100/10 + 0x30);

LCD1602\_write\_data('.');

LCD1602\_write\_data(Money%10 + 0x30);

Flag\_OK = 1;

}

if(KEY\_NUM >= 0 && KEY\_NUM <= 9) //显示输入的价值

{

Price\_Buffer[Price\_Count] = KEY\_NUM;

switch(Price\_Count)

{

case 0:

LCD1602\_write\_com(0x80+0x40+6);

LCD1602\_write\_data(Price\_Buffer[0] + 0x30);

break;

case 1:

LCD1602\_write\_com(0x80+0x40+7);

LCD1602\_write\_data(Price\_Buffer[1] + 0x30);

break;

case 2:

LCD1602\_write\_com(0x80+0x40+9);

LCD1602\_write\_data(Price\_Buffer[2] + 0x30);

break;

default : break;

}

Price\_Count++;

if( Price\_Count >= 3)

{

Price\_Count = 3;

}

}

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//矩阵键盘扫描

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

unsigned char KEY\_Scan()

{

unsigned char temp = 0;

unsigned char com = 0x55 , com1 = 0 , com2 = 0;

P3=0xf0;

if(P3!=0xf0)

{

com1=P3;

P3=0x0f;

com2=P3;

}

P3=0xf0;

while(P3!=0xf0);

temp=com1|com2;

if(temp==0xee)com=1;//数字1

if(temp==0xed)com=4;//数字4

if(temp==0xeb)com=7;//数字7

if(temp==0xe7)com=11;//备用键\*号键

if(temp==0xd7)com=0;//数字0

if(temp==0xb7)com=14;//备用键\*号键

if(temp==0xde)com=2; //数字2

if(temp==0xdd)com=5; //数字5

if(temp==0xdb)com=8;//数字8

if(temp==0xbe)com=3;//数字3

if(temp==0xbd)com=6; //数字6

if(temp==0xbb)com=9;//数字9

if(temp==0x7e)com=16;//数字A键，去皮功能

if(temp==0x7d)com=15;//数字B键清除键，二次测量

if(temp==0x7b)com=12;//数字C输入单价错误时返回上一步

if(temp==0x77)com=13;//数字D键，计算总价

return(com);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//称重

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Get\_Weight()

{

HX711\_Buffer = HX711\_Read();

HX711\_Buffer = HX711\_Buffer/100;

if(HX711\_Buffer > Weight\_Maopi)

{

Weight\_Shiwu = HX711\_Buffer;

Weight\_Shiwu = Weight\_Shiwu - Weight\_Maopi; //获取实物的AD采样数值。

Weight\_Shiwu = (unsigned int)((float)Weight\_Shiwu/4.22+0.05); //计算实物的实际重量

//因为不同的传感器特性曲线不一样，因此，每一个传感器需要矫正这里的4.30这个除数。

//当发现测试出来的重量偏大时，增加该数值。

//如果测试出来的重量偏小时，减小改数值。

//该数值一般在4.0-5.0之间。因传感器不同而定。

//+0.05是为了四舍五入百分位

Buzzer = 1; //关闭警报

}

// else if(HX711\_Buffer < Weight\_Maopi - 30)

// {

// Buzzer = 0; //负重量报警

// }

// else if(HX711\_Buffer > Weight\_Maopi + 24970) //大于5Kg的最大量程，报警

// {

// Buzzer = 0;

// }

if(Weight\_Shiwu > 5000 || HX711\_Buffer < Weight\_Maopi - 30)

{

Buzzer = 0;

}

else

{

Buzzer = 1;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//获取毛皮重量

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Get\_Maopi()

{

HX711\_Buffer = HX711\_Read();

Weight\_Maopi = HX711\_Buffer/100;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//MS延时函数(12M晶振下测试)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Delay\_ms(unsigned int n)

{

unsigned int i,j;

for(i=0;i<n;i++)

for(j=0;j<123;j++);

}

1602.H文件

#ifndef \_\_LCD1602\_H\_\_

#define \_\_LCD1602\_H\_\_

#include <reg52.h>

//LCD1602 IO设置

#define LCD1602\_PORT P0

sbit LCD1602\_RS = P2^5;

sbit LCD1602\_RW = P2^6;

sbit LCD1602\_EN = P2^7;

//函数或者变量声明

extern void LCD1602\_delay\_ms(unsigned int n);

extern void LCD1602\_write\_com(unsigned char com);

extern void LCD1602\_write\_data(unsigned char dat);

extern void LCD1602\_write\_word(unsigned char \*s);

extern void Init\_LCD1602();

#endif

1602.C程序 ：

#include "LCD1602.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//MS延时函数(12M晶振下测试)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LCD1602\_delay\_ms(unsigned int n)

{

unsigned int i,j;

for(i=0;i<n;i++)

for(j=0;j<123;j++);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//写指令

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LCD1602\_write\_com(unsigned char com)

{

LCD1602\_RS = 0;

LCD1602\_delay\_ms(1);

LCD1602\_EN = 1;

LCD1602\_PORT = com;

LCD1602\_delay\_ms(1);

LCD1602\_EN = 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//写数据

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LCD1602\_write\_data(unsigned char dat)

{

LCD1602\_RS = 1;

LCD1602\_delay\_ms(1);

LCD1602\_PORT = dat;

LCD1602\_EN = 1;

LCD1602\_delay\_ms(1);

LCD1602\_EN = 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//连续写字符

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void LCD1602\_write\_word(unsigned char \*s)

{

while(\*s>0)

{

LCD1602\_write\_data(\*s);

s++;

}

}

void Init\_LCD1602()

{

LCD1602\_EN = 0;

LCD1602\_RW = 0; //设置为写状态

LCD1602\_write\_com(0x38); //显示模式设定

LCD1602\_write\_com(0x0c); //开关显示、光标有无设置、光标闪烁设置

LCD1602\_write\_com(0x06); //写一个字符后指针加一

LCD1602\_write\_com(0x01); //清屏指令

}

称重模块.H文件：

#ifndef \_\_HX711\_H\_\_

#define \_\_HX711\_H\_\_

#include <reg52.h>

#include <intrins.h>

//IO设置

sbit HX711\_DOUT=P2^0;

sbit HX711\_SCK=P2^1;

//函数或者变量声明

extern void Delay\_\_hx711\_us(void);

extern unsigned long HX711\_Read(void);

#endif

称重模块.C程序

#include "HX711.h"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//延时函数

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Delay\_\_hx711\_us(void)

{

\_nop\_();

\_nop\_();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//读取HX711

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

unsigned long HX711\_Read(void) //增益128

{

unsigned long count;

unsigned char i;

HX711\_DOUT=1;

Delay\_\_hx711\_us();

HX711\_SCK=0;

count=0;

while(HX711\_DOUT);

for(i=0;i<24;i++)

{

HX711\_SCK=1;

count=count<<1;

HX711\_SCK=0;

if(HX711\_DOUT)

count++;

}

HX711\_SCK=1;

count=count^0x800000;//第25个脉冲下降沿来时，转换数据

Delay\_\_hx711\_us();

HX711\_SCK=0;

return(count);

}