Wireless Interface for the "DANI" (Dynamic Automated Nutrition Innovations) Diet Aid Weight Scale

A Design Project Report
Presented to the Engineering Division of the Graduate School
of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering (Electrical)

by
Jsoon Kim

Project Advisor: Bruce R. Land

Degree Date: May, 2011

Abstract

Master of Electrical Engineering Program
Cornell University
Design Project Report

Project Title: Wireless Interface for the "DANI" (Dynamic Automated Nutrition Innovations) Diet Aid Weight Scale

Author: Jsoon Kim

Abstract:

This project entails the development of a microcontroller board and PC graphical user interface that together comprise a prototype of the “DANI” (Dynamic Automated Nutrition Innovations) weight scale, a portion control and diet aid scale designed by a group of Cornell students. The microcontroller board will process electrical signals from a strain gauge, convert the signals to weight data, and send the data wirelessly to a Bluetooth enabled device. A graphical user interface running on a remote PC will allow the user to interpret this weight data in terms of the nutritional value of the food item weighed. This project will be a key component in the development of the DANI scale, which if successfully commercialized, will be a revolutionary diet aid helping curtail the trend of rising obesity in the United States.

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

In this project, a working prototype of the "DANI" (Dynamic Automated Nutrition Innovations) diet aid weight scale was developed. The prototype consists of three parts: the circuitry to capture weight data as electrical signals and amplify them, the microprocessor and wireless transceiver module to process and transmit the collected data, and the computer software to receive and appropriately display the data.

1. Obtaining Weight Data

Weight data is obtained using a strain gauge mounted on a metal cantilever. Any weight applied on the metal cantilever causes it to bend and deflect a very tiny amount; this strain changes the resistances across the strain gauge, allowing a weight-dependent voltage signal to be extracted. This electrical signal is also tiny, requiring amplification using a low-noise high-accuracy instrumentation amplifier built using three op-amps and resistors.

2. Processing and Transmitting Weight Data

The amplified voltage is fed into an analog-to-digital converter (ADC) inside an ATmega microprocessor. To improve the accuracy of the measurement and to prevent the measurement from fluctuating up and down from noise, 4096 samples of the weight data are collected and averaged. Once the weight data is digitized, it can be sent out using a commercially available Bluetooth transceiver module using SPI mode operation.

3. Receiving and Displaying Weight Data

A Bluetooth-enabled computer connects to the remote Bluetooth transceiver module via an emulated serial (COM) port. The computer requests weight data or recalibration (taring) by sending signals to the remote microprocessor and Bluetooth module, which obliges by sending the appropriate data signals back. The locally downloaded USDA SR23 national nutrient database is paired with the weight data to generate the detailed nutritional value of the food item weighed.

The result of this project was a fully functional prototype of a wireless weight scale, complete with a user interface program running on a Windows computer. Resolution was approximately ±1 grams, with a current consumption of ~50 mA during active transmission and ~10 mA during idle, in this case mostly consumed through the power indicator LED. The time taken to collect weight data was on the order of seconds. Transmission was near-instantaneous. Using cheaper compatible parts, the unit cost per 100 units was calculated at

$19.64plus an estimated ~$20 for the cost of PCB manufacturing, strain gauge, and enclosure. A PCB layout for the design was also completed in both PCB123 and ExpressPCB.

2. INTRODUCTION

Obesity in the United States is on the rise – according to data from the Centers for Disease Control and Prevention, in 2009, all states but Colorado had obesity rates of 20% or more with nine states having rates of 30%, whereas only two decades earlier, no state had a rate of 15% or more.[1] Obesity has severe health consequences and puts a strain on the U.S. health care system; medical care costs attributable to obesity totaling $147 billion in 2009.[2] As such, people who are willing to lose weight and overcome their obesity should be given aid to make that as easy to do as possible. The reason why people become obese is because their caloric intake is greater than the amount of calories burned off by their bodies.[3] To help curtail this trend of obesity, a group of Cornell students have been designing a diet aid weight scale named "DANI," Dynamic Automated Nutrition Innovations. The goal of the weight scale is to measure the nutritional intake of a person in as simple and non-intrusive a manner as possible and to report the measured data so that the person using the scale becomes aware of exactly how many calories and other nutrients they are consuming over the course of a day. Because the weight scale reports very detailed nutritional data, including the amount of vitamins and minerals ingested, it can double as a nutritional assistant in addition to being a diet aid.

The goal of this project is to design and create a prototype for the DANI weight scale. The successful creation of this prototype will be an important step in actually commercializing the DANI weight scale.

2.1. DESIGN REQUIREMENTS

The prototype should serve as a proof of concept for the DANI weight scale. It should deliver the basic functionality of the envisioned DANI weight scale, including:

- Measure the weight of the food item accurately.

- Wirelessly transmit the weight data to a computer or smartphone. As the scale is a food scale, it will likely be placed away from a computer or smartphone. In addition, laptop computers and smartphones, by their nature, will get carried around, away from the scale. For the weight scale to be as easy to use and non-intrusive as possible, it should be wireless so that devices can connect to it without needing the user to physically attach and detach wires whenever they want to use the scale.

-Report the detailed nutritional data of the measured food item, based on its weight. This should include items normally found on a "nutrition facts" label, such as total calories and fat, carbohydrate, protein, vitamins, and mineral content.

-Have a method of logging measured nutritional data for later time look up.

Furthermore, it should:

- Use as little power as possible. In correspondence with the wireless functionality of the scale, it should be able to run off of batteries for an extended amount of time as a complete no wire solution.

- Cost as little as possible. For the scale to be successfully commercialized, it should be affordable.

- Respond as fast as possible. For the device to be non-intrusive, it should respond quickly so that the user can measure, log, and leave the device in a matter of seconds.

3. DESIGN

There are three main stages to the design: the circuitry to capture weight data as electrical signals and amplify them, the microprocessor and wireless transceiver module to process and transmit the collected data, and the computer software to receive and appropriately display the data.

3.1. OBTAINING WEIGHT DATA

The purpose of this stage is to convert weight into an electrical signal that is appropriate for the microcontroller in the subsequent stage to handle.

3.1.1. HARDWARE

Upon dismantling commercial digital weight scales, I discovered that the typical weight sensing mechanism involves a strain gauge. A strain gauge is a thin metal resistor that changes in resistance as strain is applied. Typically, they are mounted on metal cantilevers as a way of measuring weight: any weight on the cantilever will cause it to deflect a very small amount, applying strain on the mounted strain gauge, which changes in resistance. The change in resistance, proportional to the amount of weight applied to the cantilever, can be detected by an electrical circuit. For this project, cantilever-mounted strain gauges salvaged from commercial digital weight scales were used; similar cantilever-strain-gauge combinations were found online for less than $10. One of the salvaged cantilevers was completely dissected and multimeter-analyzed to find the wiring scheme, which was a simple

full-bridge Wheatstone bridge with 1 kΩ resistances, with an extra strain-independent 100 Ω resistance at ground.The differential output voltage of a Wheatstone bridge is equal to

$$V_{out+} - V_{out-} = \left[\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2}\right] V_{ex}$$

where V_{ex} is the voltage across the Wheatstone bridge.[4]

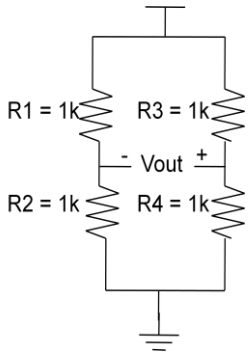


Figure 1: Full-bridge Wheatstone bridge

I am unsure as to the purpose of what appears to be an extra strain-independent (connected orthogonal to the cantilever beam) strain gauge with 100 Ω resistance that is connected between the tail of the Wheatstone bridge and ground in the salvaged cantilever. In any case, the extra 100 Ω resistance would reduce the voltage across the Wheatstone bridge by the tail current times 100 Ω , but otherwise not affect the operation of the bridge. The differential output voltage can be written as:

$$V_{out+} - V_{out-} = \left[\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2}\right] V_{cc} \left\{1 - \frac{100}{\frac{(R_1+R_2)(R_3+R_4)}{R_1+R_2+R_3+R_4} + 100}\right\}$$

When R1 = R2 = R3 = R4 = 1 kΩ, the voltage across the bridge is reduced to 1 - 100/1100 = 90.909% compared to if the 100 Ω resistance were not there. Changes in resistance from the operation of the strain gauge will change the voltage across the bridgeadditionally, but the additional change is small enough that the accuracy of the strain gauge is not compromised. For example, if R3 is increased by 1% (a change that is much larger than what R3 would normally experience), the voltage across the bridgewill increase by 0.02%, resulting in an 0.02% inaccuracy in the output measurement. This inaccuracy is likely much smaller than inaccuracies introduced by noise, temperature, etc., and thus can be ignored. In any case, as long as the strain gauge works, what is done inside the black box is irrelevant.

As mentioned earlier, the deflection of the metal cantilever is very tiny and the strain applied on the strain gauge equally small. This means that the change in resistance across a strain gauge mounted on the cantilever is also very small, typically much less than 1% at maximum tolerable deflection. This means that the output voltage of the strain gauge Wheatstone bridge is also small, on the order of millivolts for a supply voltage of 3.3V. This means that the output voltage must be amplified before it can be processed by the ADC on the ATmega microcontroller. Upon doing some research, the instrumentation amplifier described in [5] was found to be ideal for several reasons. First, it's double-ended input, single-ended output, which is ideal because the output from the Wheatstone bridge is double-ended and while the ATmega ADC tolerates differential inputs, it can provide much better accuracy for single-ended inputs (10 bits vs 8 bits). Second, the amplifier has inputs that are directly connected to an op-amp terminal with no feedback or additional connections on that node. This means that the input impedance of the amplifier is very high and the operation of the amplifier is strongly independent of the output impedance of the Wheatstone bridge, which may differ from strain gauge to strain gauge and also changes as the strain gauge resistances change. Third, the gain is a product of two resistance ratios,

$$V_{out} = (V_+ - V_-) * \left[\frac{R_4}{R_2}\left(\frac{2R_f}{R_g} + 1\right)\right]$$

meaning that high gain can be achieved without requiring resistors that have resistances that are orders of magnitude different.[5]

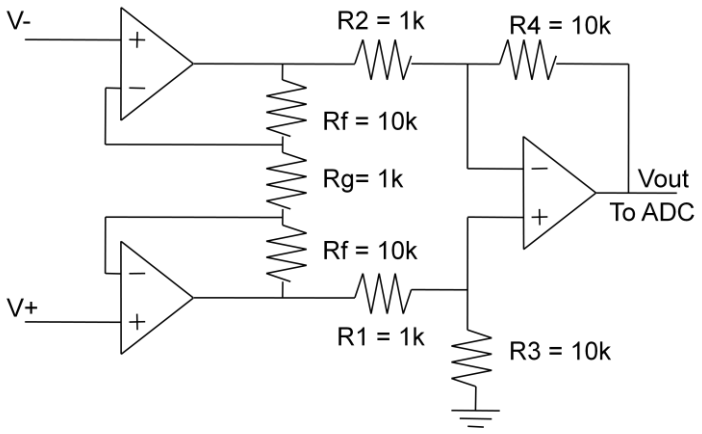


Figure 2: instrumentationamplifier

Additional advantages include a low noise factor and high common mode rejection ratio.For actual production, a self-contained instrumentationamplifier chip will likely replace the manually constructed one used in this project.

The gain, after some experimentation, was set at 210, but this can be easily adjustable by changing the resistance of Rg. Increasing the gain will decrease the maximum measurable weight, as any weights that after amplification result in voltages above 2.56 V (the reference voltage used in the ATmega ADC) cannot be resolved. Increasing the gain will also increase the output voltage for any particular weight (as long as the weight doesn't generate voltages above 2.56 V), meaning that the ADC will be able to resolve finer weight differences (i.e. the resolution is improved). However, noise also gets amplified by the amplifier, and thus, after a certain point, increasing the gain will have little effect on improving the resolution.

Finally, because the cantilever has weight of its own, and because a tray is typically placed on the cantilever in an actual weight scale, adding some weight, the differential output voltage from the cantilever is not zero even when nothing is being weighed. This means that the measured weight must be tared, or added an offset, as expanded upon in section 3.2.

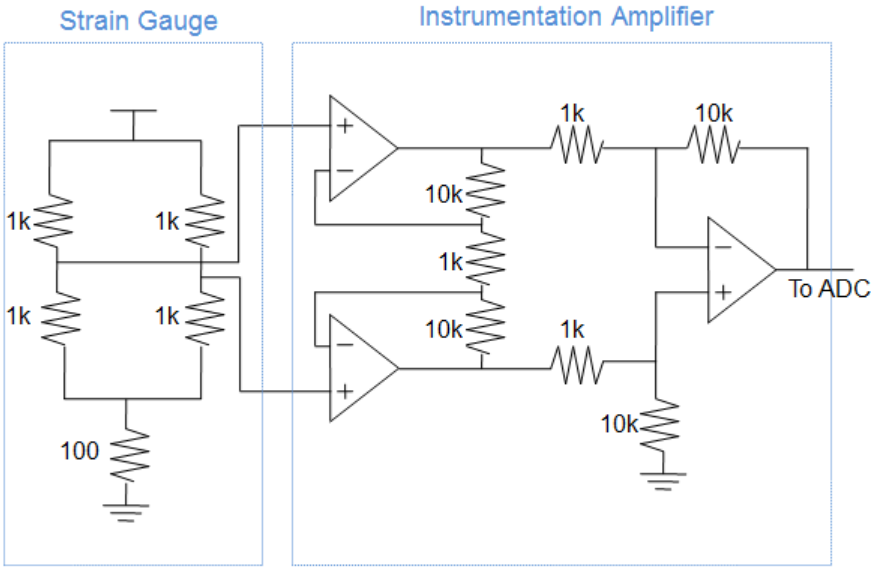


Figure 3: First stage overview

3.2. Processing and Transmitting Weight Data

This stage involves processing the weight-dependent voltage signal from the circuit in section 3.1 and transmitting that information via a wireless Bluetooth transceiver module.

3.2.1. Hardware

This circuit uses two pieces of hardware: an ATmega 644/48 microcontroller and a Roving Networks RN-41/RN-42 Bluetooth transceiver module. Preliminary breadboard prototyping was done using an ATmega 644 microcontroller paired with a RN-41 module. The PCB layout, which may be used for mass production, is based on the cheaper and smaller but functionally

compatible ATmega 48 microcontroller paired with a cheaper RN-42 module. In the following discussion, any mentions of ATmega 644 may be replaced with ATmega 48 and RN-41 with RN-42.

The Roving Networks RN-41 Bluetooth transceiver module is the most expensive part of this project. While I would have preferred to use CSR BlueCore modules, which are actually implemented in the RN-41, the expensive development kit and support seat and the machine-solder-only chips prevented me from doing so. In any case, I appreciate that Roving Networks is providing a low-volume solution for Bluetooth.

The placement of resistors and bypass capacitors in the ATmega 644 periphery was based on Bruce Land's protoboard.[6]

The voltage signal from the first stage is passed through a simple resistor-capacitor low pass filter into an ADC input port on the ATmega 644 microcontroller. The low pass filter cuts off high frequency noise, significantly reducing noise to about 6-8 mV peak-to-peak; without the low pass filter, noise would be on the order of 40-60 mV peak-to-peak.The downside to the low pass filter is that the capacitor takes time to charge and discharge proportional to the RC product, meaning that changes in the measured weight are not as rapidly reflected in the ADC input voltage. Increasing the RC constant will cut off more noise, but will also increase the time taken for the ADC input voltage to respond to changes in the applied weight. After experimentation, a 100 uF capacitor and a 1 kΩ resistor were chosen for the low pass filter for having a reasonable charge/discharge rate and for significantly reducing the noise, leading to more precise measurements.

The reference voltage for the ATmega 644 ADC was set to the internally generated 2.56 V reference. Since the gain of the previous stage is adjustable, the choice of reference voltage of the ADC has minimal effect on the resolution - if the reference voltage is doubled, the gain of the previous stage could be doubled and the ADC would see approximately the same signal with the same signal to noise ratio.

The ATmega 644's USART serial TX/RX ports are wired to a Roving Networks RN-41 Bluetooth transceiver module, which implements the SPI mode of Bluetooth to emulate a serial (COM port) connection with a Bluetooth enabled device. The serial communication settings were set to 9600 baud rate, 8 data bits, 1 stop bit, no parity, Xon/Xoff handshake, which are typical and often default settings. The ATmega 644, the RN-42, and the Bluetooth enabled device should all have the same serial communication settings.

The ATmega 644 is clocked by a 3.6864 MHz oscillator. This clock frequency is evenly divisible by most of the commonly used baud rates, including 9600, meaning that there is 0% error in generating the baud rate.[7]

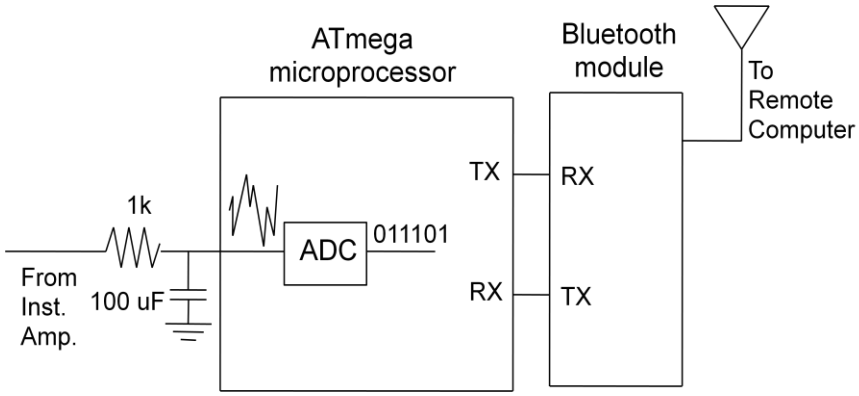


Figure 4: Second stage overview

3.2.2. SOFTWARE

To improve the resolution of the weight scale, the ATmega 644 was programmed to take 4096 samples from the ADCand shift right the sum by 8 bits, following the technique discussed in [8].This technique is basically averaging the samples, but also simultaneously extracting a few additional bits of resolution from the samples. Essentially, by oversampling by a factor of 4 and dividing the sum by 2, an additional bit of precision can be extracted from the ADC. Unlike when we increased the resolution by increasing the amplifier gain on the first stage, this method reduces the effects of noise and the amount of precision that can be obtained is not as harshly bounded by noise. However, each sample takes 65 - 260 μs time [7] (the sampling time is adjustable; setting it too low will reduce the precision of the ADC), meaning that to take 4096 samples with a sample time of 100 μs, 0.4 seconds plus processing overhead is required. This is a relatively large amount of time, probably the most that is tolerable for a usable weight scale. According to [8], 16 bits of precision can be extracted by sampling 4096 times and shifting the sum right by 6 bits; for this project, the sum had to be shifted an extra 2 bits for a total of 8 bits due to the fact that the ADC readings were already imprecise due to noise. Finally, the shifted sum was multiplied by a "grams per ADC unit" factor, which converted from digital ADC readings to actual grams. The gain in the first stage and the amount of shifting in this stage was adjusted so that the "grams per ADC unit" factor would be < 1 so that the weight scale will have the target resolution of 1g.

To keep the sum of 4096 bits without overflowing, the 8-bit chars and 16-bit integers that could be used in an ATmega C program was insufficient. Since each ADC reading is 10 bits, a total of $10 + \log_2(4096) = 22$ bits were required to store the sum. To accommodate this, four 8-bit variables were treated as a 32-bit variable that stores the sum. A simple 32-bit addition function that added two of these 32-bit variables was implemented. In this function, each of the 8-bit variables were added with the corresponding 8-bit variable in the addend, then

overflow bits were added appropriately. This allowed a running sum of 4096 samples to be kept without overflow.

The ATmega 644 runs in a loop that takes an ADC sample every loop cycle. The samples are taken using the sleep mode of the ATmega (but with USART kept on) to reduce internal electrical noise that affects the accuracy of the ADC and to reduce power usage. After accumulating 4096 samples, the weight measurement is calculated and stored in a variable, after which the ATmega repeats the process. During each cycle, the ATmega also checks to see if any USART transmissions have been received. If there is a received transmission, the received character is stored in a buffer that is two characters long. Whenever the buffer has "ww" (standing for weight request), the weight reading is pushed to the USART transmission buffer and the "ww" in the buffer is cleared. Whenever the buffer has "cc" (standing for recalibration/taring request), the calibration offset, which is subtracted from the weight that is reported in "ww" commands, is updated to be the current weight. The commands were chosen to be "ww" and "cc" for its simplicity and modest resistance to erroneous communications (it's easier for the receiver to detect in error a single character compared to a sequence of two specific characters). A few other extra commands were implemented, including "vv" (report version number), "tt" (return test message for communication testing), "11" (turn LED 1 on), "22" (turn LED 1 off), "33" (turn LED 2 on), and "44" (turn LED 2 off).

When the weight is requested, the ATmega controller complies with a string of the form "w=[xx]g w=[xx]g", where "[xx]" is a variable-length number that equals the weight of the weighed object in grams. The string can be easily parsed to extract just the weight of the object. The weight is repeated to add modest resistance to erroneous communications. When recalibration/taring is requested, the response is "c=[xx]g c=[xx]g" where "[xx]" is the calibration offset in grams.

3.3. Receiving and Displaying Weight Data

This part of the project is a pure software stage that involves receiving the weight data sent by the remote Bluetooth transceiver module and displaying the data on the user's computer.

3.3.1. Hardware

The hardware used in this stage is a Bluetooth-enabled device. For this project, a Bluetooth-equipped laptop computer was used.

3.3.2. Software

The software for this stage was written in C#. The software consists of two main parts. The first part, the API, provides a set of functions that allow for easy connection to the remote

Bluetooth transceiver and easy access of the weight data. The second part, the demo program, provides a sample user interface that may be written using the API. It provides an intuitive interface to configure the serial connection settings, connect to the Bluetooth module, disconnect from the Bluetooth module, and tare the weight scale. Once connected, weight readings are automatically updated every 200 ms. Furthermore, the program provides an interface to look up and search the locally downloaded USDA SR23 database (found at [9]) of nutrition facts, and use the nutrition facts data and the weight data to compute and display the detailed nutritional value of the food item weighed. Any weighed items can also be added to a log, which retains its memory when the program is closed then later reopened, for later perusal.

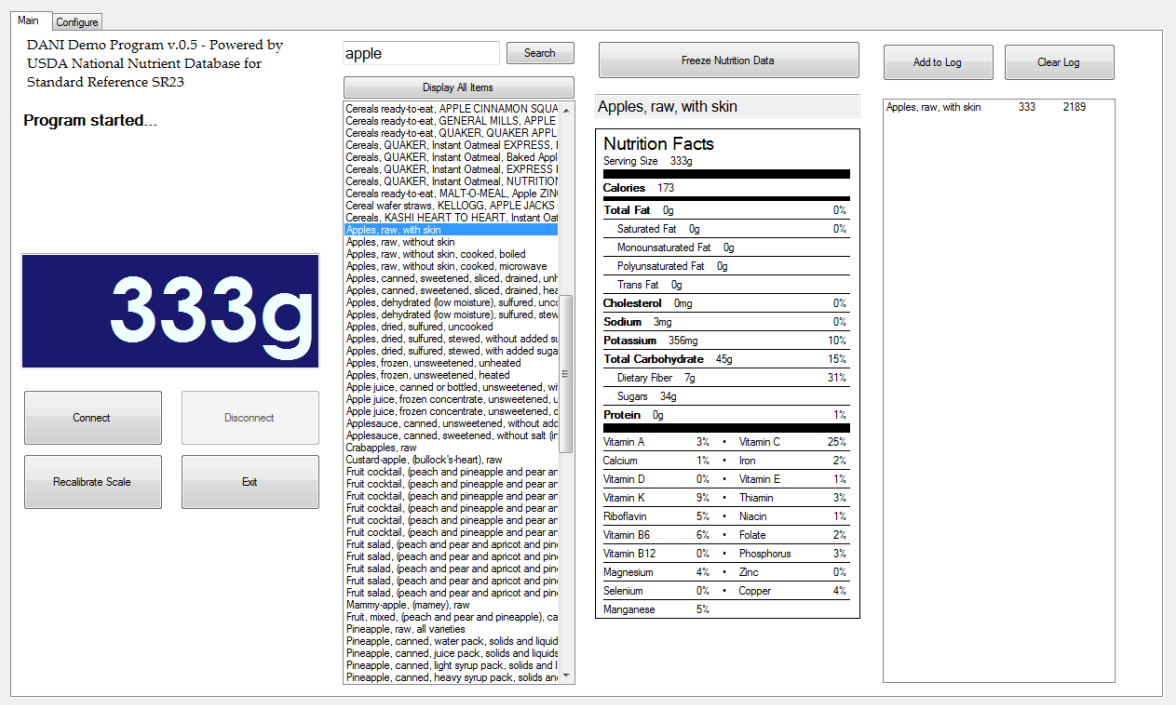


Figure 5: Screenshot of demo program

A simple summary of the workings of the code follows. Once connected, every 200 ms, the program sends out a weight request "ww" command over the Bluetooth serial connection and at the same time, checks for any response from the previous cycle's "ww" command. If the weight data is successfully received, the weight display is updated, as is the nutrition facts display, if any food item is selected in the list of food items obtained from the database. On program load, the contents of the database are loaded to an array of structs that allow for easy and fast access. Every time the weight updates, or whenever a different food item is selected, the selected food item's struct is read along with the weight to produce the nutrition facts information. Percentage daily values are based on the FDA recommendations for a 2000 calorie diet. Also, when a item on the log is clicked, the nutrition facts display shows the data

for the food item in the log, ignoring the weight and selected food item. A simple text search for the database is implemented; when a particular text is searched, all name fields in the struct array of food item entries are checked to see if it contains a substring equal to the searched text (ignoring uppercase/lowercase). If there is a match, the food item is added to the list of displayed items; if there is no match, the food item is not added. The commented code listing of the API and demo program should be referred to for a more detailed understanding of the code.

3.4. Additional Design

A simple power supply rail was constructed from a linear regulator and a couple of power sources (a 9V battery and a 12V AC adapter) connected to the input of the linear regulator through diodes. The diodes prevent current from one power source from flowing into the other power source. The AC adapter is chosen to have a voltage higher than that of the battery so that the battery would only be drained when the AC adapter is disconnected.

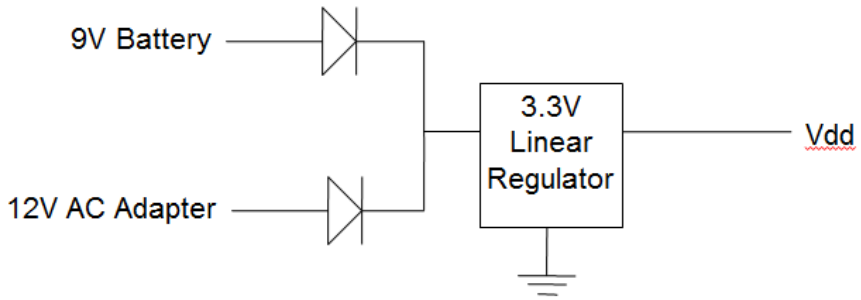


Figure 6: Power supply rail

Bypass capacitors are placed throughout the circuit to reduce high frequency noise and spikes. In particular, they are placed at the input and output of the linear regulator and near the op-amp as these parts handle voltages that are crucial to the proper operation of the circuit.

A PCB of the circuit was laid out in both PCB123 and ExpressPCB. These PCBs use the compatible ATmega 48 instead of the ATmega 644 to save layout space (and parts cost). Screenshots of the layout are in the appendix.

4. RESULTS

The result of this project was a fully-functional prototype of a wireless diet aid weight scale. The resolution was approximately ±1g, measured as the fluctuation of the weight scale reading over time. When recalibration/taring was properly performed, weight readings were reproducible between different sessions, though there was an error of at most ±2g. The wireless transmission was demonstrated to be working, with near-instantaneous transmission. To combat noise, the low-pass filter at the ADC input was set with an aggressively high RC constant. Similarly, the number of samples taken per measurement was a high 4096 samples. This meant that the weight acquisition time was relatively long, on the order of seconds. In this sense, the requirement of having a speedy response time was not completely met. The demo program was able to report the nutrition facts of the weighed food item and log the readings in an intuitive manner. Using a multimeter, the current usage was roughly measured at ~50 mA during active transmission, ~10 mA when idle. With a 9V alkaline battery with a typical capacity of 565 mAh, this device should operate for over 10 hours. While the current consumption is decent, a series of four rechargeable AA batteries with capacities of up to 3000 mAh might be a better power source choice. The unit cost for the weight scale in quantities of 100 and in single quantities is shown in table 1:

Table 1: Parts cost

Part	Unit cost per quantity of 100	Unit cost per single quantity
RN-42 Bluetooth module	15.44	17.65
ATmega48 microcontroller	1.62	2.30
3.6864 MHz clock oscillator	0.99	1.60
3.3V linear regulator	0.367	0.42
9V battery snap	0.17	0.38
28-DIP socket	0.17	0.31
Power slide switch	0.181	0.29
Quad op-amps	0.15	0.22
2-pin male headers	0.075*3	0.09*3
Green LED	0.08	0.09
Diodes	0.013*2	0.02*2
100 Ohm resistors	0.008*4	0.05*4
1 kOhm resistors	0.008*4	0.05*4
10 kOhm resistors	0.008*4	0.05*4
100 kOhm resistor	0.008	0.05
0.1 uF ceramic capacitor	0.015*5	0.07*5
100 uF electrolytic capacitor	0.04	0.04
PCB manufacturing	varies	varies
Strain gauge	varies	varies
Total	19.638 + PCB + strain gauge	24.61 + PCB + strain gauge

For a quantity of 100, the unit cost is $19.64 plus the cost for the PCB and strain gauge, estimated at around $20. For single quantities, the cost is $24.61 plus the cost for the PCB and strain gauge. The majority of the cost is from the RN-42 Bluetooth module; if these could be replaced with CSR BlueCore chips (which the RN-42 also uses), the cost could be brought down significantly. However, the development kit and support seat required to implement the BlueCore chip directly would add a significant fixed cost, which would not be desirable for small scale production.

5. CONCLUSIONS

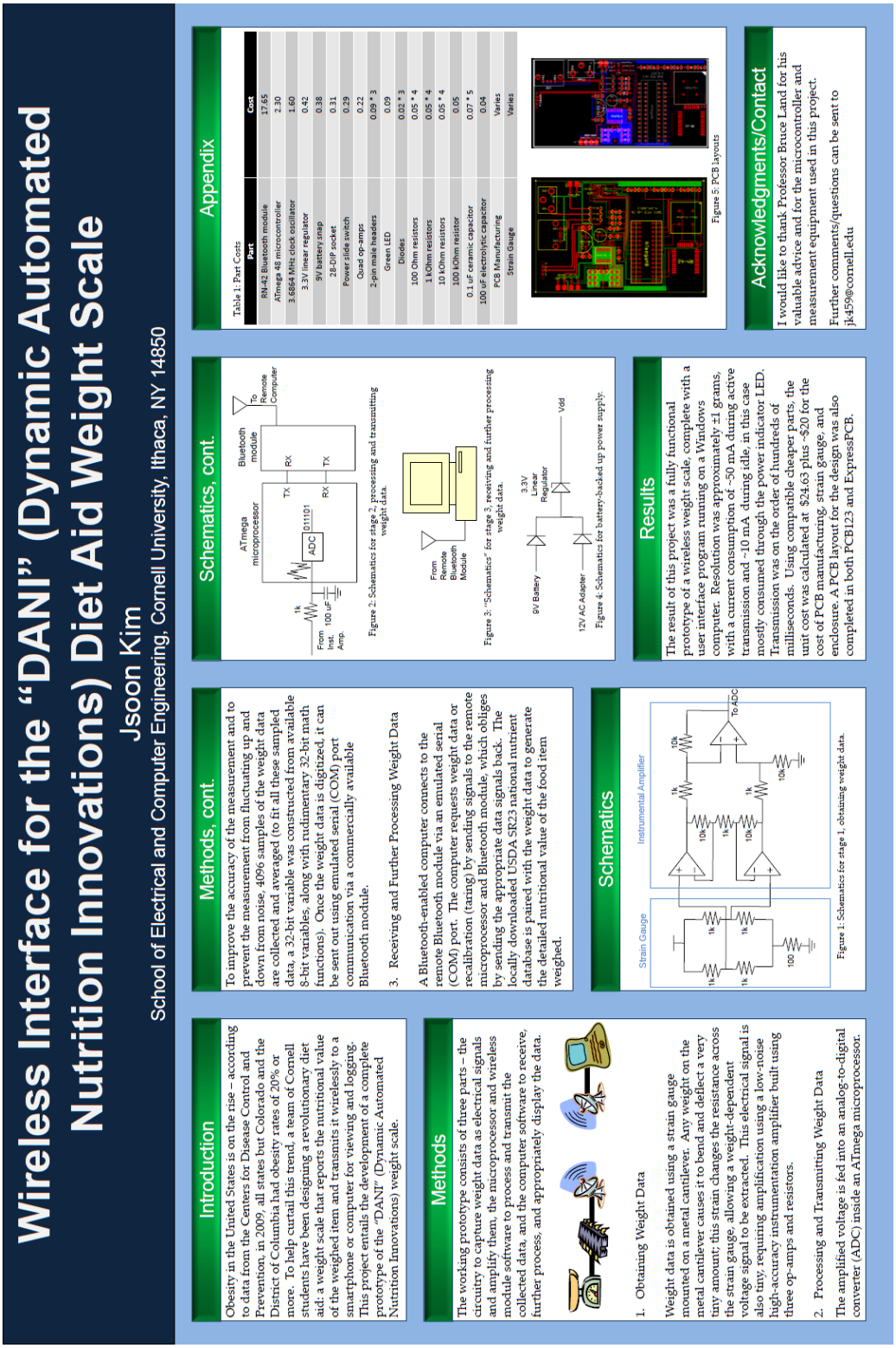
In this project, a fully functional wireless diet aid weight scale was implemented. Reasonably good specs were achieved, though there remains room for improvement. In addition to improving the specs, several ideas were encountered during this project that were outside the initial scope of this project. One often requested feature was to have the scale automatically detect the nutritional content of the weighed item without further user input. Another possible feature is to implement a barcode reader that will automatically pull up nutrition facts based on the barcode scanned from the food item. The data logging features could be expanded from the simple one implemented in the demo program to record a full, easy to access diary of foods consumed each day. Finally, the user interface could be implemented on a smartphone for more convenient access.

6. REFERENCES

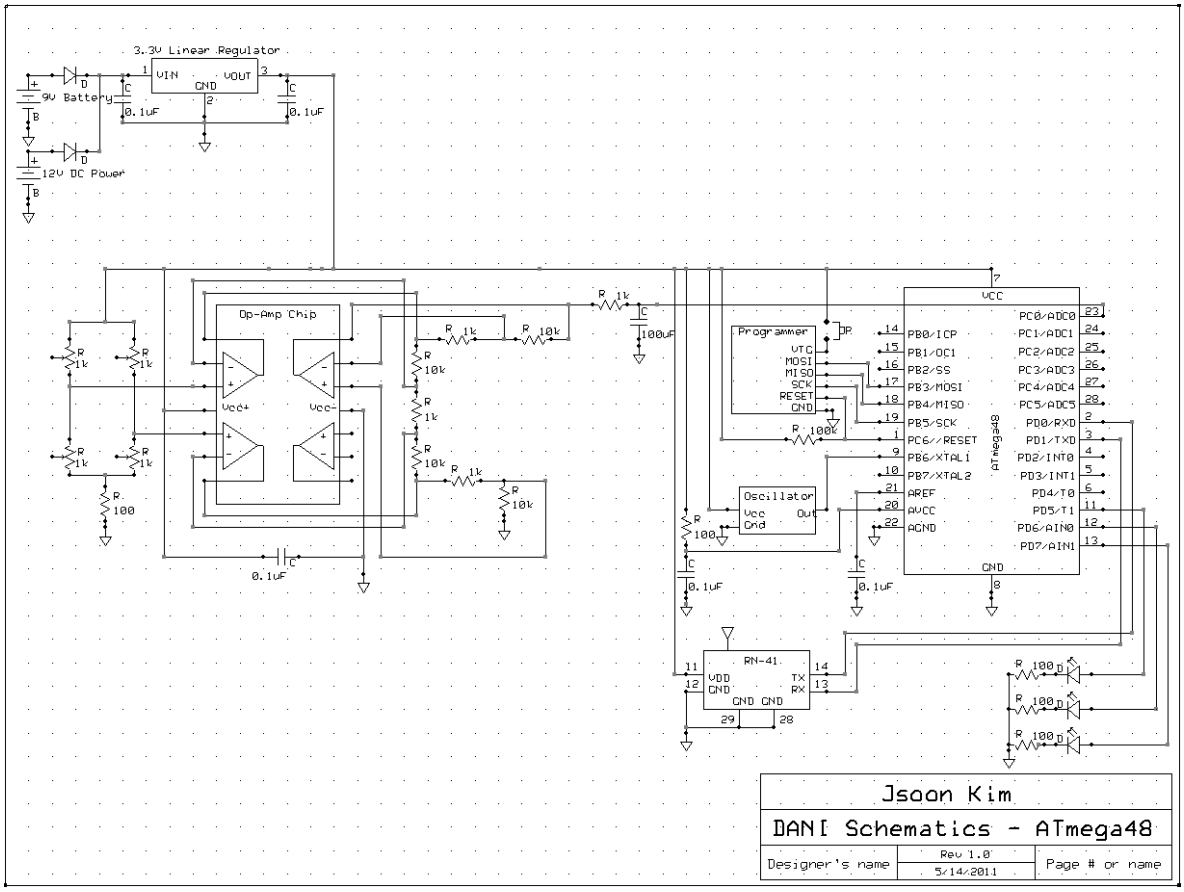
[1] "U.S. Obesity Trends," http://www.cdc.gov/obesity/data/trends.html, accessed 5-15-2011.
[2] "Economic Consequences," http://www.cdc.gov/obesity/causes/economics.html, accessed 5-15-2011.
[3] "Causes and Consequences," http://www.cdc.gov/obesity/causes/index.html, accessed 5-15-2011.
[4] "Measuring Strain with Strain Gages," http://zone.ni.com/devzone/cda/tut/p/id/3642, accessed 5-15-2011.
[5] "Signal Conditioning Wheatstone Resistive Bridge Sensors," http://focus.ti.com/lit/an/sloa034/sloa034.pdf, accessed 5-15-2011.
[6] http://people.ece.cornell.edu/land/PROJECTS/ProtoBoard476/v11_2011/Schematic.png, accessed 5-15-2011.

[7] "ATmega 644/V," http://www.atmel.com/dyn/resources/prod_documents/doc2593.pdf, accessed 5-15-2011.
[8] "AVR121: Enhancing ADC resolution by oversampling," http://www.atmel.com/dyn/resources/prod_documents/doc8003.pdf, accessed 5-15-2011.
[9] "USDA National Nutrient Database for Standard Reference," http://www.ars.usda.gov/Services/docs.htm?docid=8964, accessed 5-15-2011.

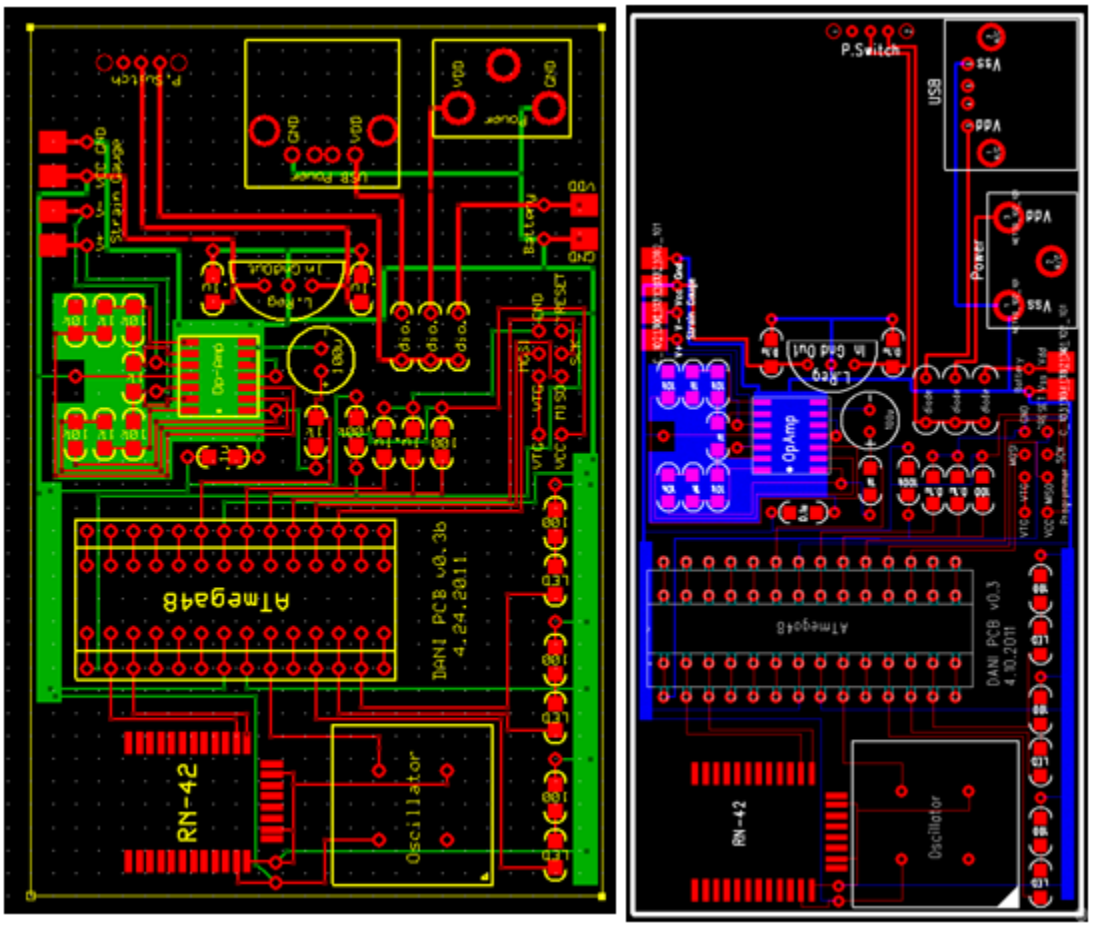
APPENDIX A: ECE DAY POSTER



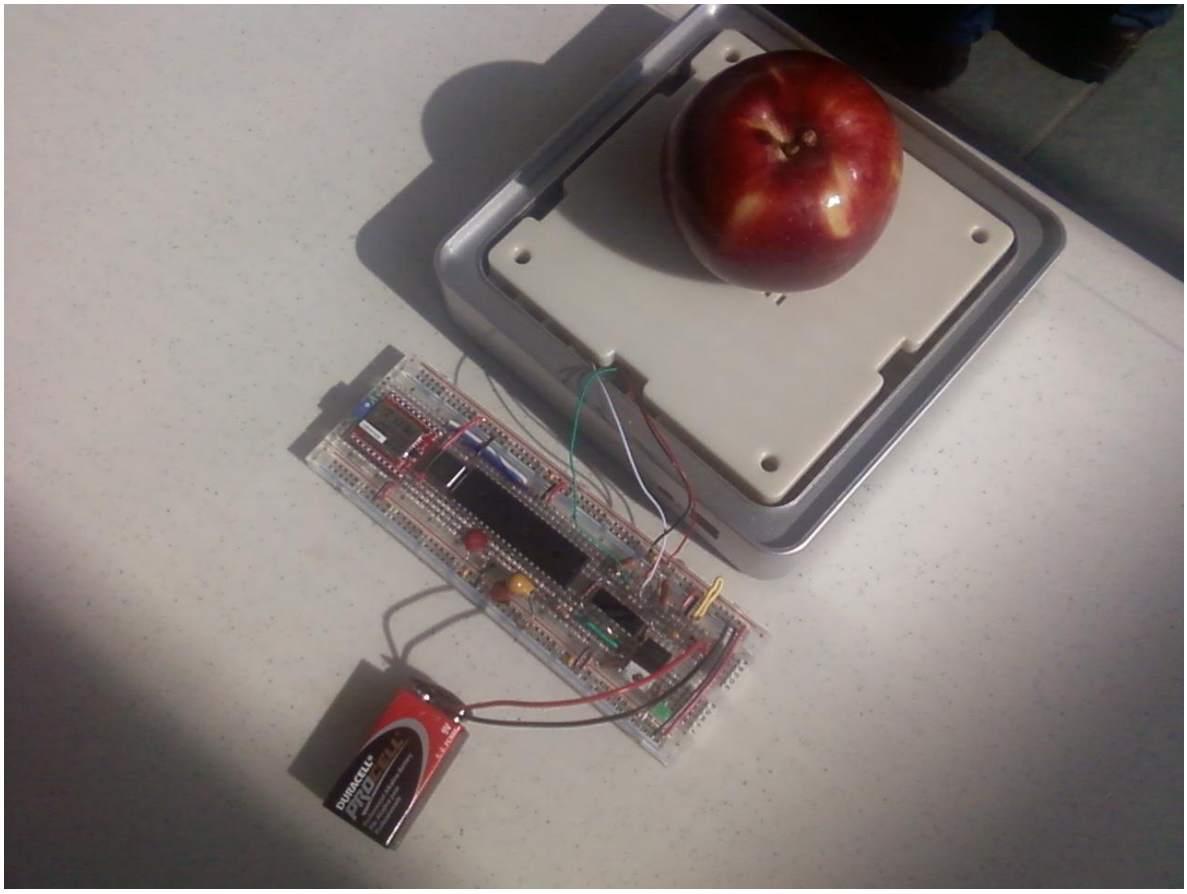
APPENDIX B: SCHEMATICS



APPENDIX C: PCB123 AND EXPRESSPCB LAYOUTS



APPENDIX D: BREADBOARD PHOTO



APPENDIX E: CODE LISTING

ATMEGA 644/48 MICROCONTROLLER CODE:

```
//Written by Jsoon Kim
//Updated 5.15.11

// Mega644 version

#include <inttypes.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include <stdio.h>

//String ops
#include <string.h>

//set up the debugging utility ASSERT
//#define __ASSERT_USE_STDERR
//#include <assert.h>

// serial communication library
//#include "uart.h"

#define begin {
#define end }
#define t1 1

void initialize(void); //all the usual mcu stuff
void main_routine(void); //main loop routine
void USART_Transmit( unsigned char data ); //transmit one char over USART
unsigned char USART_Receive(void); //receive one char over USART
//32-bit adddition
void THIRTY_TWO_BIT_UNSIGNED_PLUS(unsigned char* a3,
                                  unsigned char* a2,
                                  unsigned char* a1,
                                  unsigned char* a0,
                                  unsigned char* b3,
                                  unsigned char* b2,
                                  unsigned char* b1,
                                  unsigned char* b0,
                                  unsigned char* o3,
                                  unsigned char* o2,
                                  unsigned char* o1,
                                  unsigned char* o0);

char str[80]; //buffer for string to transmit over USART
int strlength; //length of string in buffer
unsigned char recv_temp; //temporary storage for received character
unsigned char recv[2]; //currently received char in USART
unsigned int calibration; //calibration offset
volatile unsigned char time1;
unsigned int weight; //current measured weight
```

```
unsigned int weight_temp;
unsigned long long oversample_counter; //number of samples taken
unsigned char oversample_accum_3; //running sum of samples
unsigned char oversample_accum_2; //running sum of samples
unsigned char oversample_accum_1; //running sum of samples
unsigned char oversample_accum_0; //running sum of samples
unsigned char empty_char; //empty 8-bit variable
unsigned int debug;
unsigned int overflow_detection; //used in adding 32 bit numbers
unsigned char temp_a3; //temporary storage
unsigned char temp_a2; //temporary storage
unsigned char temp_a1; //temporary storage
unsigned char temp_a0; //temporary storage
unsigned char temp_b3; //temporary storage
unsigned char temp_b2; //temporary storage
unsigned char temp_b1; //temporary storage
unsigned char temp_b0; //temporary storage
unsigned char temp_ADCH; //temporary storage
unsigned char temp_ADCL; //temporary storage

const float GRAMS_PER_ADC_UNIT = 3.000; //the actual grams per adc unit is this divided by 4
                                                                                    //needs calibration

//1ms ISR
ISR (TIMER0_COMPA_vect)
begin
 //Decrement the three times if they are not already zero
 if (time1>0)          --time1;
end

//Main entry point
int main(void)
begin
 initialize();

 //Main task scheduler loop
 while(1)
 begin
        main_routine();
 end
end

//Main loop routine
void main_routine(void)
begin
        //Go to sleep, wake up when ADC finished
        asm("sleep");

        //4096 samples have been taken
        if(oversample_counter == 4095)
        begin
                oversample_counter = 0;

                //Equivalent to shifting right the sum by 8 bits, then
                //multiplying by GRAMS_PER_ADC_UNIT / 4
                weight = (unsigned int)
```

```
                    ((((float)oversample_accum_0) * GRAMS_PER_ADC_UNIT +
                    ((float)oversample_accum_1) * GRAMS_PER_ADC_UNIT * 256.0 +
                    ((float)oversample_accum_2) * GRAMS_PER_ADC_UNIT * 65536.0 +
                    ((float)oversample_accum_3) * GRAMS_PER_ADC_UNIT * 16777216.0) / 1024.0);

            temp_ADCL = ADCL;
            temp_ADCH = ADCH;
            THIRTY_TWO_BIT_UNSIGNED_PLUS(
                    &empty_char,
                    &empty_char,
                    &temp_ADCH,
                    &temp_ADCL,
                    &empty_char,
                    &empty_char,
                    &empty_char,
                    &empty_char,
                    &oversample_accum_3,
                    &oversample_accum_2,
                    &oversample_accum_1,
                    &oversample_accum_0);

            if (calibration == -1)
            begin
                    calibration = weight;
            end
end
else
begin
            oversample_counter++;
            temp_ADCL = ADCL;
            temp_ADCH = ADCH;
            THIRTY_TWO_BIT_UNSIGNED_PLUS(
                    &empty_char,
                    &empty_char,
                    &temp_ADCH,
                    &temp_ADCL,
                    &oversample_accum_3,
                    &oversample_accum_2,
                    &oversample_accum_1,
                    &oversample_accum_0,
                    &oversample_accum_3,
                    &oversample_accum_2,
                    &oversample_accum_1,
                    &oversample_accum_0);
end

recv_temp = USART_Receive(); //Get received char if there exists any
//There was an incoming character on USART
if (recv_temp != 0)
begin
            recv[1] = recv[0];
            recv[0] = recv_temp;
end

//Process commands
```

```
//Check for "ww" input; if so, start transmission
if (recv[1] == 'w' && recv[0] == 'w')
begin
        if (calibration > weight) //weight is negative
                weight_temp = 0;
        else
                weight_temp = weight - calibration;
        sprintf(str,"w=%dg w=%dg\r\n",
                (unsigned int)(weight_temp),
                (unsigned int)(weight_temp));
        strlength = 0;
        recv[0] = 0;
end
//Check for "cc" input; if so, reset calibration
else if (recv[1] == 'c' && recv[0] == 'c')
begin
        calibration = weight;
        sprintf(str,"c=%dg c=%dg\r\n",
                (unsigned int)calibration,
                (unsigned int)calibration);
        strlength = 0;
        recv[0] = 0;
end
//Check for "tt" input; if so, return "Test" (test to check connection)
else if (recv[1] == 't' && recv[0] == 't')
begin
        sprintf(str,"test success test success\r\n");
        strlength = 0;
        recv[0] = 0;
end
//Check for "vv" input; if so, return version number
else if (recv[1] == 'v' && recv[0] == 'v')
begin
        sprintf(str,"1h 1h");
        strlength = 0;
        recv[0] = 0;
end
//Check for "11" input; if so, turn light 1 (PC1) on
else if (recv[1] == '1' && recv[0] == '1')
begin
        sprintf(str,"Light 1 On Light 1 On\r\n");
        PORTC = PORTC & 0xfd;
        strlength = 0;
        recv[0] = 0;
end
//Check for "22" input; if so, turn light 1 (PC1) off
else if (recv[1] == '2' && recv[0] == '2')
begin
        sprintf(str,"Light 1 Off Light 1 Off\r\n");
        PORTC = PORTC | 0x02;
        strlength = 0;
        recv[0] = 0;
end
//Check for "33" input; if so, turn light 2 (PC2) on
else if (recv[1] == '3' && recv[0] == '3')
begin
```

```
                sprintf(str,"Light 2 On Light 2 On\r\n");
                PORTC = PORTC & 0xfb;
                strlength = 0;
                recv[0] = 0;
        end
        //Check for "44" input; if so, turn light 2 (PC2) off
        else if (recv[1] == '4' && recv[0] == '4')
        begin
                sprintf(str,"Light 2 Off Light 2 Off\r\n");
                PORTC = PORTC | 0x04;
                strlength = 0;
                recv[0] = 0;
        end
        //Debug command
        else if (recv[1] == 'd' && recv[0] == 'd')
        begin
                debug = 0;
                sprintf(str,"debug=%d,%d,%d,%d,%d,%d\r\n", (unsigned int)oversample_accum_3,
                        (unsigned int)oversample_accum_2,
                        (unsigned int)oversample_accum_1,
                        (unsigned int)oversample_accum_0,
                        (unsigned int)temp_ADCH,
                        (unsigned int)temp_ADCL);
                strlength = 0;
                recv[0] = 0;
        end

        if (strlength < strlen(str))
        begin
                USART_Transmit(str[strlength]);
                strlength++;
        end
end

void initialize(void)
begin
        DDRB=0x0f;  // PORT B is an output
        PORTB=0;
        DDRD=0x00;          // PORT D is an input

        //Set up timer 0 for 1 mSec ticks -- UNUSED
        TIMSK0 = 2;                     //turn on timer 0 cmp match ISR
        OCR0A = 250;        //set the compare reg to 250 time ticks
        TCCR0A = 0b00000010; // turn on clear-on-match
        TCCR0B = 0b00000011;        // clock prescalar to 64

        //Crank up the ISRs
        sei();

        //Init USART
        // Disable PRR
        PRR = 0x00;
        // Set baud rate
        UBRR0L = (unsigned char)23;
        UBRR0H = (unsigned char)0;
        UCSR0A = 0;
```

```
	// Enable receiver and transmitter
	UCSR0B = (1<<RXEN0)|(1<<TXEN0);
	// Set frame format: 8data, 1stop bit
	UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);

	//Init ADC
	//Internal 2.56V reference
	ADMUX = (0<<REFS1)|(1<<REFS0);
	//ADC sleep interrupt on
	ADCSRA = (1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS0);
	//Set sleep to idle mode (allows USART to keep working), sleep enabled
	SMCR = (1<<SE);

	weight = 0;
	strlength = 0;
	calibration = -1;
	oversample_counter = 0;
	oversample_accum_3 = 0;
	oversample_accum_2 = 0;
	oversample_accum_1 = 0;
	oversample_accum_0 = 0;
	empty_char = 0;
end


void USART_Transmit( unsigned char data )
{
	// Wait for empty transmit buffer
	while ( !( UCSR0A & (1<<UDRE0)) ) ;
	// Put data into buffer, sends the data
	UDR0 = data;
}

unsigned char USART_Receive(void)
{
	// Wait for data to be received
	//while ( !(UCSR0A & (1 << RXC0)) ) ;

	// Check if received data exists
	if ( !(UCSR0A & (1 << RXC0)) )
		return 0;

	// Get and return received data from buffer
	return UDR0;
}

//Does A + B = O
void THIRTY_TWO_BIT_UNSIGNED_PLUS(unsigned char* a3,
                                  unsigned char* a2,
                                  unsigned char* a1,
                                  unsigned char* a0,
                                  unsigned char* b3,
                                  unsigned char* b2,
                                  unsigned char* b1,
                                  unsigned char* b0,
                                  unsigned char* o3,
```

```
                                                unsigned char* o2,
                                                unsigned char* o1,
                                                unsigned char* o0)
{
        //Copy variables first in case O has the same address as A or B
        temp_a3 = *a3;
        temp_a2 = *a2;
        temp_a1 = *a1;
        temp_a0 = *a0;
        temp_b3 = *b3;
        temp_b2 = *b2;
        temp_b1 = *b1;
        temp_b0 = *b0;

        //First ignore all overflow
        *o0 = temp_a0 + temp_b0;
        *o1 = temp_a1 + temp_b1;
        *o2 = temp_a2 + temp_b2;
        *o3 = temp_a3 + temp_b3;

        //Then process all overflow bits
        overflow_detection = ((unsigned int)temp_a0) + ((unsigned int)temp_b0);
        if ((overflow_detection >> 8) > 0) //overflow in lowest byte
        {
                if (*o1 == 255)
                {
                        *o1 = 0;
                        if (*o2 == 255)
                        {
                                *o2 = 0;
                                *o3 = *o3 + 1; //overflow may occur when 32 bits can't hold A+B
                        }
                        else
                                *o2 = *o2 + 1;
                }
                else
                        *o1 = *o1 + 1;
        }
        overflow_detection = ((unsigned int)temp_a1) + ((unsigned int)temp_b1);
        if ((overflow_detection >> 8) > 0) //overflow in second lowest byte
        {
                if (*o2 == 255)
                {
                        *o2 = 0;
                        *o3 = *o3 + 1; //overflow may occur when 32 bits can't hold A+B
                }
                else
                        *o2 = *o2 + 1;
        }
        overflow_detection = ((unsigned int)temp_a2) + ((unsigned int)temp_b2);
        if ((overflow_detection >> 8) > 0) //overflow in third lowest byte
        {
                *o3 = *o3 + 1; //overflow may occur when 32 bits can't hold A+B
        }
}
```

```
ISR(ADC_vect)
begin
        //nothing
end
```

C# API CODE:

API.cs

```
//DANI CSharp API
//Written by Jsoon Kim
//2.18.2011
//Ver 0.1

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using System.IO.Ports;

namespace DANI_CSharp
{
publicclassAPI
    {
privateSerialPort port;
privateString error;
privateChar[] responseBuffer;
privateInt32 responseOffset;
privateconstInt32 RESPONSE_BUFFER_SIZE = 1024;

//Constructor
public API()
        {
this.port = newSerialPort(
"NONE",
            9600,
Parity.None,
            8,
StopBits.One);
this.port.Handshake = Handshake.XOnXOff;
this.port.Encoding = Encoding.ASCII;

this.error = "";
this.responseBuffer = newChar[RESPONSE_BUFFER_SIZE];
this.responseOffset = 0;
        }

//Constructor
public API(
String portName,
Int32 baudRate,
            System.IO.Ports.Parity parity,
Int32 dataBits,
            System.IO.Ports.StopBits stopBits,
            System.IO.Ports.Handshake handshake)
        {
this.port = newSerialPort(
                portName,
                baudRate,
                parity,
                dataBits,
                stopBits);
this.port.Handshake = handshake;
this.port.Encoding = Encoding.ASCII;

this.error = "";
this.responseBuffer = newChar[RESPONSE_BUFFER_SIZE];
this.responseOffset = 0;
        }

//Set port name setting
publicInt32 setPortName(String portName)
        {
if (this.port.IsOpen)
return -1000;
else
this.port.PortName = portName;
```

```
return 0;
        }

//Get port name setting
publicString getPortName()
        {
returnthis.port.PortName;
        }

//Set baud rate setting
publicInt32 setBaudRate(Int32 baudRate)
        {
if (this.port.IsOpen)
return -1000;
else
this.port.BaudRate = baudRate;
return 0;
        }

//Get baud rate setting
publicInt32 getBaudRate()
        {
returnthis.port.BaudRate;
        }

//Set parity setting
publicInt32 setParity(System.IO.Ports.Parity parity)
        {
if (this.port.IsOpen)
return -1000;
else
this.port.Parity = parity;
return 0;
        }

//Get parity setting
public System.IO.Ports.Parity getParity()
        {
returnthis.port.Parity;
        }

//Set data bits setting
publicInt32 setDataBits(Int32 dataBits)
        {
if (this.port.IsOpen)
return -1000;
else
this.port.DataBits = dataBits;
return 0;
        }

//Get data bits setting
publicInt32 getDataBits()
        {
returnthis.port.DataBits;
        }

//Set stop bits setting
publicInt32 setStopBits(System.IO.Ports.StopBits stopBits)
        {
if (this.port.IsOpen)
return -1000;
else
this.port.StopBits = stopBits;
return 0;
        }

//Get stop bits setting
public System.IO.Ports.StopBits getStopBits()
        {
returnthis.port.StopBits;
        }

//Set handshake setting
publicInt32 setHandshake(System.IO.Ports.Handshake handshake)
        {
if (this.port.IsOpen)
return -1000;
else
this.port.Handshake = handshake;
return 0;
        }

//Get handshake setting
public System.IO.Ports.Handshake getHandshake()
        {
returnthis.port.Handshake;
        }
```

```
//Get port open status
publicBoolean getPortOpen()
        {
returnthis.port.IsOpen;
        }

//Return string containing last special error
publicString getLastError()
        {
returnthis.error;
        }

//Connect to remote module
publicInt32 Connect()
        {
if (this.port.IsOpen)
return -1001;
elseif (this.port.PortName == "")
return -1002;
else
            {
try
                {
this.port.Open();
                }
catch (Exception ex)
                {
this.error = ex.Message;
return -4000;
                }
            }

return 0;
        }

//Disconnect from remote module
publicInt32 Disconnect()
        {
if (!this.port.IsOpen)
return -1003;
else
            {
try
                {
this.port.Close();
                }
catch (Exception ex)
                {
this.error = ex.Message;
return -4000;
                }
            }

return 0;
        }

//Send weight read command to remote module.  Weight data can be checked using CheckResponse
publicInt32 SendWeightReadCommand()
        {
if (!this.port.IsOpen)
return -1004;
else
            {
this.responseOffset = 0;
this.port.WriteLine("ww");
            }

return 0;
        }

//Send recalibrate (set current weight to zero) command to remote module
publicInt32 SendRecalibrateCommand()
        {
if (!this.port.IsOpen)
return -1004;
else
            {
this.responseOffset = 0;
this.port.WriteLine("cc");
            }

return 0;
        }

//Check the response from the remote module
publicInt32 CheckResponse(refString response)
        {
```

```
//Read any bytes in serial buffer
if (!this.port.IsOpen)
return -1005;
elseif (this.port.BytesToRead > 0)
        {
if (this.port.BytesToRead > RESPONSE_BUFFER_SIZE)
return -2001;
else
            {
//Read bytes to response array
Array.Clear(this.responseBuffer, 0, RESPONSE_BUFFER_SIZE);
this.port.Read(this.responseBuffer, 0, this.port.BytesToRead);
                response = newString(this.responseBuffer);
return 0;
            }
        }
else
return -2000;
    }

//Convert error code returned from any of the API functions to a string message
publicString ErrorCodeToMessage(Int32 errorCode)
    {
switch (errorCode)
        {
case 0:
return"No error.";
case -1000:
return"Port is open; Cannot write to parameters.";
case -1001:
return"Port is already open.";
case -1002:
return"Port name undefined.";
case -1003:
return"Port is already closed.";
case -1004:
return"Port is closed; Cannot send commands.";
case -1005:
return"Port is closed.";
case -2000:
return"No response yet.  If there is no response for a while, try resending command.";
case -2001:
return"Response exceeded maximum response length.  Try resending command.";
case -2002:
return"Response cannot be parsed; may be corrupt or only partially transmitted.  Try resending command.";
case -2003:
return"Corruption detected in response.  Try resending command.";
case -4000:
return"Check returned string in getLastError().";
default:
return"Undefined Error Code";
        }

return"";
    }

//Returns direct handle to serial port for advanced programming
publicSerialPort GetSerialPort()
    {
returnthis.port;
    }

//DEBUG - Write line to serial buffer
publicInt32 DEBUG_SendLine(String command)
    {
if (this.port.IsOpen)
this.port.WriteLine(command);
else
return -1;

return 0;
    }

//DEBUG - Read line from serial buffer
publicString DEBUG_ReadLine()
    {
if (this.port.IsOpen &&
this.port.BytesToRead > 0 &&
this.port.BytesToRead <= RESPONSE_BUFFER_SIZE)
        {
Array.Clear(this.responseBuffer, 0, RESPONSE_BUFFER_SIZE);
this.port.Read(this.responseBuffer, 0, this.port.BytesToRead);
returnnewString(this.responseBuffer);
        }

return"";
    }
  }
```

```
}
```

C# DEMO PROGRAM CODE:

PROGRAM.CS

```
//DANI CSharp Demo Program 1
//Written by Jsoon Kim
//2.18.2011
//Ver 0.1

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
staticclassProgram
    {
///<summary>
/// The main entry point for the application.
///</summary>
        [STAThread]
staticvoid Main()
        {
Application.EnableVisualStyles();
Application.SetCompatibleTextRenderingDefault(false);
Application.Run(newForm1());
        }
    }
}
```

SR23_FOOD_DES.CS

```
//DANI CSharp Demo Program 1
//Written by Jsoon Kim
//2.18.2011
//Ver 0.1

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace WindowsFormsApplication1
{
publicstructSR23_FOOD_DES
    {
publicInt32 NDB_No;
publicInt32 FdGrp_Cd;
publicString Long_Desc;
publicDouble Refuse;
publicDouble N_Factor;
publicDouble Pro_Factor;
publicDouble Fat_Factor;
publicDouble CHO_Factor;

public SR23_FOOD_DES(
Int32 NDB_No,
Int32 FdGrp_Cd,
String Long_Desc,
Double Refuse,
Double N_Factor,
Double Pro_Factor,
Double Fat_Factor,
Double CHO_Factor)
        {
this.NDB_No = NDB_No;
this.FdGrp_Cd = FdGrp_Cd;
this.Long_Desc = Long_Desc;
this.Refuse = Refuse;
this.N_Factor = N_Factor;
this.Pro_Factor = Pro_Factor;
this.Fat_Factor = Fat_Factor;
this.CHO_Factor = CHO_Factor;
        }
    }
```

```
}
```

SR23_NUT_DATA.cs

```
//DANI CSharp Demo Program 1
//Written by Jsoon Kim
//2.18.2011
//Ver 0.1

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace WindowsFormsApplication1
{
classSR23_NUT_DATA
    {
publicInt32 NDB_No;
publicInt32 Nutr_No;
publicDouble Nutr_Val;

public SR23_NUT_DATA(
Int32 NDB_No,
Int32 Nutr_No,
Double Nutr_Val)
        {
this.NDB_No = NDB_No;
this.Nutr_No = Nutr_No;
this.Nutr_Val = Nutr_Val;
        }

    }
}
```

SR23_NUTR_DEF.cs

```
//DANI CSharp Demo Program 1
//Written by Jsoon Kim
//2.18.2011
//Ver 0.1

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace WindowsFormsApplication1
{
publicstructSR23_NUTR_DEF
    {
publicInt32 Nutr_No;
publicString Units;
publicString NutrDesc;

public SR23_NUTR_DEF(
Int32 Nutr_No,
String Units,
String NutrDesc)
        {
this.Nutr_No = Nutr_No;
this.Units = Units;
this.NutrDesc = NutrDesc;
        }
    }
}
```

Form1.cs

```
//DANI CSharp Demo Program 1
//Written by Jsoon Kim
//5.12.2011
//Ver 0.5

using System;
using System.Collections.Generic;
using System.ComponentModel;
```

```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.IO.Ports;
using System.IO;
using DANI_CSharp;

namespace WindowsFormsApplication1
{
publicpartialclassForm1 : Form
    {
privateAPI api;
privateString timerString; //temp variable used in timer routine
privateInt32 timerInt, timerInt2; //temp variable used in timer routine
privateBoolean recalibrateFlag; //flag that is raised when recalibrate button is pressed
privateBoolean freezeNutritionDisplay; //true when the nutrition display is frozen
privateSerialPort port; //handle to port; api is functional without handling this
privateDouble weight; //weight value of last measurement

privateSR23_NUTR_DEF[] sr23_nutr_def; //Put nutrition definition in array for O(1) access.
//The array is about 20% full, wasting some memory,
//but nutr_def.txt is small enough that it won't hurt.
privateList<SR23_FOOD_DES> sr23_food_des;
privateList<SR23_NUT_DATA> sr23_nut_data;
privateInt32[] lsb_FoodList_to_sr23_nutr_def; //Mapping between index of lsb_FoodList to sr23_nutr_def
privateTextWriter logWriter;
privateBoolean logDisplayMode; //If true, the nutrition data will display data from the log

public Form1()
        {
            InitializeComponent();
            Initialize();
            InitializeTab2();
            InitializeSR23();
            InitializeLogWriter();
        }

privatevoid Initialize()
        {
this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None;
this.WindowState = FormWindowState.Maximized;
this.lbl_Status.Text = "Program started...";
this.btn_Disconnect.Enabled = false;
this.api = newAPI(); //Use default config settings for now
this.timer1.Start();
this.txt_Search.KeyDown += newKeyEventHandler(txt_Search_KeyDown);
this.recalibrateFlag = false;
this.freezeNutritionDisplay = false;
this.logDisplayMode = false;
        }

privatevoid Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
this.logWriter.Close();
        }

privatevoid InitializeSR23()
        {
//Define some temporary variables
TextReader txtRdr;
String line;
String[] decLine;
Char[] delim = newChar[] { '^' };
Char[] trim = newChar[] { '~' };
Int32 nutr_no;

try
            {
//Load sr23_nutr_def
this.sr23_nutr_def = newSR23_NUTR_DEF[1000];
                txtRdr = newStreamReader(Application.StartupPath + "\\NUTR_DEF.txt");
                line = txtRdr.ReadLine();
while (line != null)
                {
                    decLine = line.Split(delim);
                    nutr_no = Int32.Parse(decLine[0].Trim(trim));
this.sr23_nutr_def[nutr_no] = newSR23_NUTR_DEF(
                        nutr_no, decLine[1].Trim(trim), decLine[3].Trim(trim));
                    line = txtRdr.ReadLine();
                }
                txtRdr.Close();

//Load sr23_food_des
this.sr23_food_des = newList<SR23_FOOD_DES>();
                txtRdr = newStreamReader(Application.StartupPath + "\\FOOD_DES.txt");
```

```
            line = txtRdr.ReadLine();
while (line != null)
            {
               decLine = line.Split(delim);

this.sr23_food_des.Add(
newSR23_FOOD_DES(
Int32.Parse(decLine[0].Trim(trim)),
Int32.Parse(decLine[1].Trim(trim)),
                     decLine[2].Trim(trim),
                     (decLine[8].Length > 0) ? Double.Parse(decLine[8]) : 0,
                     (decLine[10].Length > 0) ? Double.Parse(decLine[10]) : 0,
                     (decLine[11].Length > 0) ? Double.Parse(decLine[11]) : 0,
                     (decLine[12].Length > 0) ? Double.Parse(decLine[12]) : 0,
                     (decLine[13].Length > 0) ? Double.Parse(decLine[13]) : 0));
               line = txtRdr.ReadLine();
            }
            txtRdr.Close();

//Load sr23_nut_data
this.sr23_nut_data = newList<SR23_NUT_DATA>();
            txtRdr = newStreamReader(Application.StartupPath + "\\NUT_DATA.txt");
            line = txtRdr.ReadLine();
while (line != null)
            {
               decLine = line.Split(delim);

this.sr23_nut_data.Add(
newSR23_NUT_DATA(
Int32.Parse(decLine[0].Trim(trim)),
Int32.Parse(decLine[1].Trim(trim)),
Double.Parse(decLine[2])));
               line = txtRdr.ReadLine();
            }
            txtRdr.Close();

//Copy over log data from file
            txtRdr = newStreamReader(Application.StartupPath + "\\log.txt");
            line = txtRdr.ReadLine();
while (line != null)
            {
this.lsb_Log.Items.Add(line);
               line = txtRdr.ReadLine();
            }
            txtRdr.Close();

//Display data in list
for (Int32 i = 0; i <this.sr23_food_des.Count; i++)
this.lsb_FoodList.Items.Add(this.sr23_food_des[i].Long_Desc);

//Initialize FoodList_to_nutr_def
this.lsb_FoodList_to_sr23_nutr_def = newInt32[this.lsb_FoodList.Items.Count];
for (Int32 i = 0; i <this.lsb_FoodList_to_sr23_nutr_def.Length; i++)
this.lsb_FoodList_to_sr23_nutr_def[i] = i; //Initially, the nutr_def index is the same as the FoodList index
         }
catch (Exception ex)
         {
MessageBox.Show(ex.Message);
         }
      }

privatevoid InitializeLogWriter()
      {
try
         {
this.logWriter = newStreamWriter(Application.StartupPath + "\\log.txt", true);
         }
catch (Exception ex)
         {
MessageBox.Show(ex.Message);
         }
      }

privatevoid btn_Exit_Click(object sender, EventArgs e)
      {
this.Close();
      }

privatevoid btn_Connect_Click(object sender, EventArgs e)
      {
this.lbl_Status.Text = "Attempting to connect...";
this.btn_Connect.Enabled = false;
this.Refresh();
if (api.Connect() == 0)
         {
this.lbl_Status.Text = "Connection Successful.";
this.btn_Connect.Enabled = false;
this.btn_Disconnect.Enabled = true;
```

```
            }
else
            {
this.lbl_Status.Text = "Connection Unsuccessful.";
this.btn_Connect.Enabled = true;
this.btn_Disconnect.Enabled = false;
            }
        }

privatevoid btn_Disconnect_Click(object sender, EventArgs e)
        {
this.lbl_Status.Text = "Disconnecting...";
if (api.Disconnect() == 0)
            {
this.lbl_Status.Text = "Disconnected.";
this.btn_Connect.Enabled = true;
this.btn_Disconnect.Enabled = false;
this.txt_Display.Text = "";
this.weight = 0;
            }
else
            {
this.lbl_Status.Text = "Disconnection Unsuccessful.";
this.btn_Connect.Enabled = false;
this.btn_Disconnect.Enabled = true;
            }
        }

privatevoid btn_RecalibrateScale_Click(object sender, EventArgs e)
        {
this.recalibrateFlag = true;
        }

privatevoid timer1_Tick(object sender, EventArgs e)
        {
if (this.recalibrateFlag)
            {
                api.SendRecalibrateCommand();
this.recalibrateFlag = false;
            }
else
            {
try
                {
//Send request weight command
                    api.SendWeightReadCommand();
                }
catch (Exception ex)
                {
//MessageBox.Show(ex.Message);
                }

//Read response from last request weight command
//timerString = api.DEBUG_ReadLine();
                api.CheckResponse(ref timerString);
if (timerString != null)
                {
                    timerInt = timerString.IndexOf("w=");
                    timerInt2 = timerString.IndexOf("g");
if (timerInt == 0)
                    {
try
                        {
this.weight = Double.Parse(timerString.Substring(timerInt + 2, timerInt2 - (timerInt + 2)));
                        }
catch (Exception ex)
                        {
//MessageBox.Show(ex.Message);
                        }
this.txt_Display.Text = this.weight.ToString() + "g";
                        UpdateNutritionDisplay();
                    }
                }
            }
        }

privatevoid InitializeTab2()
        {
this.port = api.GetSerialPort();

String[] ports = SerialPort.GetPortNames();
            cmb_PortName.Items.Add("NONE");
for (Int32 i = 0; i < ports.Length; i++)
                cmb_PortName.Items.Add(ports[i]);

            cmb_PortName.SelectedIndex = 0;
for (Int32 i2 = 0; i2 < cmb_PortName.Items.Count; i2++)
            {
```

```
if (cmb_PortName.Items[i2].ToString() == port.PortName)
                cmb_PortName.SelectedIndex = i2;
        }

        cmb_BaudRate.Items.AddRange(newString[] {
"2400",
"4800",
"9600",
"14.4k",
"19.2k",
"28.8k",
"38.4k",
"57.6k",
"76.8k",
"115.2k",
"230.4k"});

switch (port.BaudRate)
        {
case 2400:
                cmb_BaudRate.SelectedIndex = 0;
break;
case 4800:
                cmb_BaudRate.SelectedIndex = 1;
break;
case 9600:
                cmb_BaudRate.SelectedIndex = 2;
break;
case 14400:
                cmb_BaudRate.SelectedIndex = 3;
break;
case 19200:
                cmb_BaudRate.SelectedIndex = 4;
break;
case 28800:
                cmb_BaudRate.SelectedIndex = 5;
break;
case 38400:
                cmb_BaudRate.SelectedIndex = 6;
break;
case 57600:
                cmb_BaudRate.SelectedIndex = 7;
break;
case 76800:
                cmb_BaudRate.SelectedIndex = 8;
break;
case 115200:
                cmb_BaudRate.SelectedIndex = 9;
break;
case 230400:
                cmb_BaudRate.SelectedIndex = 10;
break;
        }

        txt_DataBits.Text = port.DataBits.ToString();

        cmb_StopBits.Items.AddRange(newString[] {
"None",
"One",
"OnePointFive",
"Two"
            });

if (port.StopBits == StopBits.None)
            cmb_StopBits.SelectedIndex = 0;
elseif (port.StopBits == StopBits.One)
            cmb_StopBits.SelectedIndex = 1;
elseif (port.StopBits == StopBits.OnePointFive)
            cmb_StopBits.SelectedIndex = 2;
elseif (port.StopBits == StopBits.Two)
            cmb_StopBits.SelectedIndex = 3;

        cmb_Parity.Items.AddRange(newString[] {
"Even",
"Mark",
"Odd",
"None",
"Space"
            });

if (port.Parity == Parity.Even)
            cmb_Parity.SelectedIndex = 0;
elseif (port.Parity == Parity.Mark)
            cmb_Parity.SelectedIndex = 1;
elseif (port.Parity == Parity.Odd)
            cmb_Parity.SelectedIndex = 2;
elseif (port.Parity == Parity.None)
            cmb_Parity.SelectedIndex = 3;
```

```
elseif (port.Parity == Parity.Space)
                cmb_Parity.SelectedIndex = 4;

            cmb_Handshake.Items.AddRange(newString[] {
"None",
"RequestToSend",
"RequestToSendXOnXOff",
"XOnXOff"});

if (port.Handshake == Handshake.None)
                cmb_Handshake.SelectedIndex = 0;
elseif (port.Handshake == Handshake.RequestToSend)
                cmb_Handshake.SelectedIndex = 1;
elseif (port.Handshake == Handshake.RequestToSendXOnXOff)
                cmb_Handshake.SelectedIndex = 2;
elseif (port.Handshake == Handshake.XOnXOff)
                cmb_Handshake.SelectedIndex = 3;

this.btn_OK.Enabled = false;
        }

privatevoid btn_OK_Click(object sender, EventArgs e)
        {
this.port.PortName = this.cmb_PortName.SelectedItem.ToString();

switch (cmb_BaudRate.SelectedIndex)
            {
case 0:
                    port.BaudRate = 2400;
break;
case 1:
                    port.BaudRate = 4800;
break;
case 2:
                    port.BaudRate = 9600;
break;
case 3:
                    port.BaudRate = 14400;
break;
case 4:
                    port.BaudRate = 19200;
break;
case 5:
                    port.BaudRate = 28800;
break;
case 6:
                    port.BaudRate = 38400;
break;
case 7:
                    port.BaudRate = 57600;
break;
case 8:
                    port.BaudRate = 76800;
break;
case 9:
                    port.BaudRate = 115200;
break;
case 10:
                    port.BaudRate = 230400;
break;
            }

try
            {
                port.DataBits = Convert.ToInt32(txt_DataBits.Text);
            }
catch (FormatException ex)
            {
MessageBox.Show("Data bits not a valid value.");
            }

switch (cmb_StopBits.SelectedIndex)
            {
case 0:
                    port.StopBits = StopBits.None;
break;
case 1:
                    port.StopBits = StopBits.One;
break;
case 2:
                    port.StopBits = StopBits.OnePointFive;
break;
case 3:
                    port.StopBits = StopBits.Two;
break;
            }

switch (cmb_Parity.SelectedIndex)
```

```
            {
case 0:
                port.Parity = Parity.Even;
break;
case 1:
                port.Parity = Parity.Mark;
break;
case 2:
                port.Parity = Parity.Odd;
break;
case 3:
                port.Parity = Parity.None;
break;
case 4:
                port.Parity = Parity.Space;
break;
            }

switch (cmb_Handshake.SelectedIndex)
            {
case 0:
                port.Handshake = Handshake.None;
break;
case 1:
                port.Handshake = Handshake.RequestToSend;
break;
case 2:
                port.Handshake = Handshake.RequestToSendXOnXOff;
break;
case 3:
                port.Handshake = Handshake.XOnXOff;
break;
            }

this.btn_OK.Enabled = false;
//this.Close();
        }

privatevoid btn_Cancel_Click(object sender, EventArgs e)
        {
//this.Close();
        }

privatevoid lsb_FoodList_SelectedIndexChanged(object sender, EventArgs e)
        {
this.logDisplayMode = false;
if (!freezeNutritionDisplay)
                UpdateNutritionDisplay();
        }

privatevoid btn_FreezeUnfreezeDisplay_Click(object sender, EventArgs e)
        {
if (freezeNutritionDisplay)
            {
                freezeNutritionDisplay = false;
this.btn_FreezeUnfreezeDisplay.Text = "Freeze Nutrition Data";
            }
else
            {
                freezeNutritionDisplay = true;
this.btn_FreezeUnfreezeDisplay.Text = "Unfreeze Nutrition Data";
            }
        }

privatevoid bt_AddToLog_Click(object sender, EventArgs e)
        {
//Extract the name, weight, and food_des index of currently displaying item, delimit with tabs
String temp = this.txt_Name.Text + "\t" + this.weight.ToString() + "\t" +
this.lsb_FoodList_to_sr23_nutr_def[this.lsb_FoodList.SelectedIndex];
this.lsb_Log.Items.Add(temp);
this.logWriter.WriteLine(temp);
        }

privatevoid bt_ClearLog_Click(object sender, EventArgs e)
        {
this.lsb_Log.Items.Clear();
this.logWriter.Close();
this.logWriter = newStreamWriter(Application.StartupPath + "\\log.txt", false);
        }

privatevoid lsb_Log_SelectedIndexChanged(object sender, EventArgs e)
        {
this.logDisplayMode = true;
            UpdateNutritionDisplay();
        }

privatevoid UpdateNutritionDisplay()
        {
```

```
Int32 ndb_no, food_des_index = 0;
Double serving_weight = 0;
Int32 curIndex, lowBound, highBound;

//No items selected yet
if (this.lsb_FoodList.SelectedIndex < 0 &&this.lsb_Log.SelectedIndex < 0)
return;
elseif (this.lsb_FoodList.SelectedIndex >= 0 && !logDisplayMode)
        {
           serving_weight = this.weight;
           food_des_index = this.lsb_FoodList_to_sr23_nutr_def[this.lsb_FoodList.SelectedIndex];
        }
elseif (this.lsb_Log.SelectedIndex >= 0 && logDisplayMode)
        {
String temp = this.lsb_Log.Items[this.lsb_Log.SelectedIndex].ToString();
Int32 delim_Index1 = temp.IndexOf("\t");
Int32 delim_Index2 = temp.IndexOf("\t", delim_Index1 + 1);
try
           {
              serving_weight = Double.Parse(temp.Substring(delim_Index1 + 1, delim_Index2 - delim_Index1 - 1));
              food_des_index = Int32.Parse(temp.Substring(delim_Index2 + 1, temp.Length - delim_Index2 - 1));
           }
catch (Exception ex)
           {
MessageBox.Show(ex.Message);
return;
           }
        }

        ndb_no = this.sr23_food_des[food_des_index].NDB_No;

//Find first index of ndb_no
        lowBound = 0;
        highBound = this.sr23_nut_data.Count;
        curIndex = (highBound - lowBound) / 2;
while (true)
        {
if (this.sr23_nut_data[curIndex].NDB_No == ndb_no)
           {
//Check if this index really is the first index
while (curIndex != 0 &&
this.sr23_nut_data[curIndex - 1].NDB_No == ndb_no)
              {
                 curIndex = curIndex - 1;
              }
break;
           }
elseif (this.sr23_nut_data[curIndex].NDB_No > ndb_no)
           {
//Check for item not found
if (curIndex == lowBound)
              {
                 curIndex = -1;
break;
              }
              highBound = curIndex;
              curIndex = lowBound + (highBound - lowBound) / 2;
           }
else
           {
//Check for item not found
if (curIndex == lowBound)
              {
                 curIndex = -1;
break;
              }
              lowBound = curIndex;
              curIndex = lowBound + (highBound - lowBound) / 2;
           }
        }

//If first index is found, spit data to text display
if (curIndex != -1)
        {
this.txt_Name.Text = this.sr23_food_des[food_des_index].Long_Desc;
this.lbl_ServingSize_g.Text = serving_weight.ToString() + "g";

while (curIndex <this.sr23_nut_data.Count &&
this.sr23_nut_data[curIndex].NDB_No == ndb_no)
           {
if (this.sr23_nut_data[curIndex].Nutr_No == 208) // Calories
              {
this.lbl_Calories_kcal.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString();
              }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 204) // Total Fat
              {
this.lbl_TotalFat_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
this.lbl_TotalFat_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 65)).ToString() + "%";
```

```
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 606) // Saturated Fat
            {
this.lbl_SaturatedFat_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
this.lbl_SaturatedFat_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 20)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 645) // Monounsaturated Fat
            {
this.lbl_MonounsaturatedFat_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 646) // Polyunsaturated Fat
            {
this.lbl_PolyunsaturatedFat_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 605) // Trans Fat
            {
this.lbl_TransFat_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 601) // Cholesterol
            {
this.lbl_Cholesterol_mg.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "mg";
this.lbl_Cholesterol_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 300)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 307) // Sodium
            {
this.lbl_Sodium_mg.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "mg";
this.lbl_Sodium_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 2400)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 306) // Potassium
            {
this.lbl_Potassium_mg.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "mg";
this.lbl_Potassium_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 3500)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 205) // Total Carbohydrate
            {
this.lbl_TotalCarbohydrate_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
this.lbl_TotalCarbohydrate_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 300)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 291) // Fiber
            {
this.lbl_DietaryFiber_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
this.lbl_DietaryFiber_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 25)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 269) // Sugars
            {
this.lbl_Sugars_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 203) // Protein
            {
this.lbl_Protein_g.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 100)).ToString() + "g";
this.lbl_Protein_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 50)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 318) // Vitamin A
            {
this.lbl_VitaminA_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 5000)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 401) // Vitamin C
            {
this.lbl_VitaminC_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 60)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 301) // Calcium
            {
this.lbl_Calcium_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 1000)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 303) // Iron
            {
this.lbl_Iron_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 18)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 324) // Vitamin D
            {
this.lbl_VitaminD_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 400)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 323) // Vitamin E
            {
this.lbl_VitaminE_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 30)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 430) // Vitamin K
            {
this.lbl_VitaminK_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 80)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 404) // Thiamin
            {
this.lbl_Thiamin_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 1.5)).ToString() + "%";
            }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 405) // Riboflavin
            {
this.lbl_Riboflavin_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 1.7)).ToString() + "%";
            }
```

```
elseif (this.sr23_nut_data[curIndex].Nutr_No == 406) // Niacin
                {
this.lbl_Niacin_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 20)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 415) // Vitamin B6
                {
this.lbl_VitaminB6_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 2.0)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 417) // Folate
                {
this.lbl_Folate_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 400)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 418) // Vitamin B12
                {
this.lbl_VitaminB12_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 6.0)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 305) // Phosphorus
                {
this.lbl_Phosphorus_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 1000)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 304) // Magnesium
                {
this.lbl_Magnesium_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 400)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 309) // Zinc
                {
this.lbl_Zinc_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 15)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 317) // Selenium
                {
this.lbl_Selenium_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 70)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 312) // Copper
                {
this.lbl_Copper_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 2.0)).ToString() + "%";
                }
elseif (this.sr23_nut_data[curIndex].Nutr_No == 315) // Manganese
                {
this.lbl_Manganese_pct.Text = ((Int32)(this.sr23_nut_data[curIndex].Nutr_Val * serving_weight / 2.0)).ToString() + "%";
                }

////Nutrients Detailed List
//this.rtb_Nutrients.Text += this.sr23_nutr_def[this.sr23_nut_data[curIndex].Nutr_No].NutrDesc + ": "
//    + (this.sr23_nut_data[curIndex].Nutr_Val * this.weight / 100).ToString() + " "
//    + this.sr23_nutr_def[this.sr23_nut_data[curIndex].Nutr_No].Units + "\n";
                curIndex++;
            }
        }
    }

privatevoid bt_Search_Click(object sender, EventArgs e)
    {
        SearchFoodData();
    }

privatevoid txt_Search_KeyDown(object sender, KeyEventArgs e)
    {
if (e.KeyCode == Keys.Enter)
        {
            SearchFoodData();
        }
    }

privatevoid SearchFoodData()
    {
//Reload data in list
this.lsb_FoodList.Items.Clear();
for (Int32 i = 0; i <this.sr23_food_des.Count; i++)
this.lsb_FoodList.Items.Add(this.sr23_food_des[i].Long_Desc);

//Remove all list items that do not match the search term
Int32 itemsRemoved = 0;
for (Int32 i = 0; i <this.sr23_food_des.Count; i++)
        {
if (this.sr23_food_des[i].Long_Desc.ToLower().Contains(this.txt_Search.Text.ToLower()))
            {
this.lsb_FoodList_to_sr23_nutr_def[i - itemsRemoved] = i;
            }
else
            {
this.lsb_FoodList.Items.RemoveAt(i - itemsRemoved);
                itemsRemoved++;
            }
        }

return;

//Int32 curIndex, firstIndex = this.lsb_FoodList.SelectedIndex + 1;
```

```
//if (firstIndex == this.sr23_food_des.Count || firstIndex == -1)
//    curIndex = 0;
//else
//    curIndex = firstIndex;
//for (Int32 i = firstIndex; i < this.sr23_food_des.Count; i++)
//{
//    if (this.sr23_food_des[i].Long_Desc.ToLower().Contains(this.txt_Search.Text.ToLower()))
//    {
//        this.lsb_FoodList.SelectedIndex = i;
//        return;
//    }
//}
//for (Int32 i = 0; i < curIndex; i++)
//{
//    if (this.sr23_food_des[i].Long_Desc.ToLower().Contains(this.txt_Search.Text.ToLower()))
//    {
//        this.lsb_FoodList.SelectedIndex = i;
//        return;
//    }
//}
//MessageBox.Show("Search term not found.");
//return;
        }

privatevoid btn_DisplayAllItems_Click(object sender, EventArgs e)
        {
//Reload data in list
this.lsb_FoodList.Items.Clear();
for (Int32 i = 0; i <this.sr23_food_des.Count; i++)
this.lsb_FoodList.Items.Add(this.sr23_food_des[i].Long_Desc);
        }

privatevoid cmb_PortName_SelectedIndexChanged(object sender, EventArgs e)
        {
this.btn_OK.Enabled = true;
        }

privatevoid cmb_BaudRate_SelectedIndexChanged(object sender, EventArgs e)
        {
this.btn_OK.Enabled = true;
        }

privatevoid txt_DataBits_TextChanged(object sender, EventArgs e)
        {
this.btn_OK.Enabled = true;
        }

privatevoid cmb_StopBits_SelectedIndexChanged(object sender, EventArgs e)
        {
this.btn_OK.Enabled = true;
        }

privatevoid cmb_Parity_SelectedIndexChanged(object sender, EventArgs e)
        {
this.btn_OK.Enabled = true;
        }

privatevoid cmb_Handshake_SelectedIndexChanged(object sender, EventArgs e)
        {
this.btn_OK.Enabled = true;
        }
    }
}
```

FORM1.DESIGNER.CS

```
namespace WindowsFormsApplication1
{
partialclassForm1
    {
///<summary>
/// Required designer variable.
///</summary>
private System.ComponentModel.IContainer components = null;

///<summary>
/// Clean up any resources being used.
///</summary>
///<param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
protectedoverridevoid Dispose(bool disposing)
        {
if (disposing && (components != null))
            {
                components.Dispose();
```

```
        }
base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

///<summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
///</summary>
privatevoid InitializeComponent()
    {
this.components = new System.ComponentModel.Container();
this.btn_Connect = new System.Windows.Forms.Button();
this.timer1 = new System.Windows.Forms.Timer(this.components);
this.btn_RecalibrateScale = new System.Windows.Forms.Button();
this.lbl_Status = new System.Windows.Forms.Label();
this.btn_Disconnect = new System.Windows.Forms.Button();
this.btn_Exit = new System.Windows.Forms.Button();
this.txt_Display = new System.Windows.Forms.TextBox();
this.tab_Main = new System.Windows.Forms.TabControl();
this.tab_1 = new System.Windows.Forms.TabPage();
this.bt_ClearLog = new System.Windows.Forms.Button();
this.bt_AddToLog = new System.Windows.Forms.Button();
this.lsb_Log = new System.Windows.Forms.ListBox();
this.txt_Name = new System.Windows.Forms.TextBox();
this.lbl_ServingSize_g = new System.Windows.Forms.Label();
this.lbl_Protein_g = new System.Windows.Forms.Label();
this.lbl_Sugars_g = new System.Windows.Forms.Label();
this.lbl_Sodium_mg = new System.Windows.Forms.Label();
this.lbl_TransFat_g = new System.Windows.Forms.Label();
this.lbl_PolyunsaturatedFat_g = new System.Windows.Forms.Label();
this.lbl_Calories_kcal = new System.Windows.Forms.Label();
this.lbl_DietaryFiber_g = new System.Windows.Forms.Label();
this.lbl_TotalCarbohydrate_g = new System.Windows.Forms.Label();
this.lbl_SaturatedFat_g = new System.Windows.Forms.Label();
this.lbl_Potassium_mg = new System.Windows.Forms.Label();
this.lbl_Cholesterol_mg = new System.Windows.Forms.Label();
this.lbl_MonounsaturatedFat_g = new System.Windows.Forms.Label();
this.lbl_TotalFat_g = new System.Windows.Forms.Label();
this.lbl_Cholesterol_pct = new System.Windows.Forms.Label();
this.lbl_SaturatedFat_pct = new System.Windows.Forms.Label();
this.lbl_TotalFat_pct = new System.Windows.Forms.Label();
this.lbl_DietaryFiber_pct = new System.Windows.Forms.Label();
this.lbl_Potassium_pct = new System.Windows.Forms.Label();
this.lbl_Sodium_pct = new System.Windows.Forms.Label();
this.lbl_VitaminC_pct = new System.Windows.Forms.Label();
this.lbl_VitaminD_pct = new System.Windows.Forms.Label();
this.lbl_Protein_pct = new System.Windows.Forms.Label();
this.lbl_TotalCarbohydrate_pct = new System.Windows.Forms.Label();
this.lbl_VitaminE_pct = new System.Windows.Forms.Label();
this.lbl_Iron_pct = new System.Windows.Forms.Label();
this.lbl_Calcium_pct = new System.Windows.Forms.Label();
this.lbl_VitaminA_pct = new System.Windows.Forms.Label();
this.lbl_Magnesium_pct = new System.Windows.Forms.Label();
this.lbl_VitaminB6_pct = new System.Windows.Forms.Label();
this.lbl_Riboflavin_pct = new System.Windows.Forms.Label();
this.lbl_VitaminK_pct = new System.Windows.Forms.Label();
this.lbl_Folate_pct = new System.Windows.Forms.Label();
this.lbl_Manganese_pct = new System.Windows.Forms.Label();
this.lbl_Selenium_pct = new System.Windows.Forms.Label();
this.lbl_VitaminB12_pct = new System.Windows.Forms.Label();
this.lbl_Zinc_pct = new System.Windows.Forms.Label();
this.lbl_Phosphorus_pct = new System.Windows.Forms.Label();
this.lbl_Niacin_pct = new System.Windows.Forms.Label();
this.lbl_Thiamin_pct = new System.Windows.Forms.Label();
this.lbl_Copper_pct = new System.Windows.Forms.Label();
this.pictureBox11 = new System.Windows.Forms.PictureBox();
this.lbl_Dot9 = new System.Windows.Forms.Label();
this.lbl_Dot8 = new System.Windows.Forms.Label();
this.lbl_Dot7 = new System.Windows.Forms.Label();
this.lbl_Dot6 = new System.Windows.Forms.Label();
this.lbl_Dot5 = new System.Windows.Forms.Label();
this.lbl_Dot2 = new System.Windows.Forms.Label();
this.lbl_Dot4 = new System.Windows.Forms.Label();
this.lbl_Dot3 = new System.Windows.Forms.Label();
this.pictureBox40 = new System.Windows.Forms.PictureBox();
this.pictureBox26 = new System.Windows.Forms.PictureBox();
this.pictureBox27 = new System.Windows.Forms.PictureBox();
this.pictureBox28 = new System.Windows.Forms.PictureBox();
this.pictureBox29 = new System.Windows.Forms.PictureBox();
this.pictureBox30 = new System.Windows.Forms.PictureBox();
this.pictureBox31 = new System.Windows.Forms.PictureBox();
this.pictureBox32 = new System.Windows.Forms.PictureBox();
this.pictureBox33 = new System.Windows.Forms.PictureBox();
this.pictureBox34 = new System.Windows.Forms.PictureBox();
this.pictureBox35 = new System.Windows.Forms.PictureBox();
this.pictureBox21 = new System.Windows.Forms.PictureBox();
```

```
this.pictureBox10 = new System.Windows.Forms.PictureBox();
this.pictureBox12 = new System.Windows.Forms.PictureBox();
this.pictureBox13 = new System.Windows.Forms.PictureBox();
this.pictureBox14 = new System.Windows.Forms.PictureBox();
this.pictureBox15 = new System.Windows.Forms.PictureBox();
this.pictureBox7 = new System.Windows.Forms.PictureBox();
this.pictureBox8 = new System.Windows.Forms.PictureBox();
this.pictureBox9 = new System.Windows.Forms.PictureBox();
this.pictureBox6 = new System.Windows.Forms.PictureBox();
this.pictureBox5 = new System.Windows.Forms.PictureBox();
this.pictureBox4 = new System.Windows.Forms.PictureBox();
this.pictureBox3 = new System.Windows.Forms.PictureBox();
this.pictureBox2 = new System.Windows.Forms.PictureBox();
this.pictureBox1 = new System.Windows.Forms.PictureBox();
this.lbl_Manganese = new System.Windows.Forms.Label();
this.lbl_Copper = new System.Windows.Forms.Label();
this.lbl_Selenium = new System.Windows.Forms.Label();
this.lbl_Zinc = new System.Windows.Forms.Label();
this.lbl_Magnesium = new System.Windows.Forms.Label();
this.lbl_Phosphorus = new System.Windows.Forms.Label();
this.lbl_VitaminB12 = new System.Windows.Forms.Label();
this.lbl_Folate = new System.Windows.Forms.Label();
this.lbl_VitaminB6 = new System.Windows.Forms.Label();
this.lbl_Niacin = new System.Windows.Forms.Label();
this.lbl_Riboflavin = new System.Windows.Forms.Label();
this.lbl_Thiamin = new System.Windows.Forms.Label();
this.lbl_VitaminK = new System.Windows.Forms.Label();
this.lbl_VitaminE = new System.Windows.Forms.Label();
this.lbl_VitaminD = new System.Windows.Forms.Label();
this.lbl_Iron = new System.Windows.Forms.Label();
this.lbl_Calcium = new System.Windows.Forms.Label();
this.lbl_Dot1 = new System.Windows.Forms.Label();
this.lbl_VitaminC = new System.Windows.Forms.Label();
this.lbl_VitaminA = new System.Windows.Forms.Label();
this.lbl_Protein = new System.Windows.Forms.Label();
this.lbl_Sugars = new System.Windows.Forms.Label();
this.lbl_DietaryFiber = new System.Windows.Forms.Label();
this.lbl_TotalCarbohydrate = new System.Windows.Forms.Label();
this.lbl_Potassium = new System.Windows.Forms.Label();
this.lbl_Sodium = new System.Windows.Forms.Label();
this.lbl_Cholesterol = new System.Windows.Forms.Label();
this.lbl_TransFat = new System.Windows.Forms.Label();
this.lbl_PolyunsaturatedFat = new System.Windows.Forms.Label();
this.lbl_MonounsaturatedFat = new System.Windows.Forms.Label();
this.lbl_SaturatedFat = new System.Windows.Forms.Label();
this.lbl_Calories = new System.Windows.Forms.Label();
this.lbl_ServingSize = new System.Windows.Forms.Label();
this.lbl_NutritionFacts = new System.Windows.Forms.Label();
this.lbl_TotalFat = new System.Windows.Forms.Label();
this.btn_FreezeUnfreezeDisplay = new System.Windows.Forms.Button();
this.btn_DisplayAllItems = new System.Windows.Forms.Button();
this.btn_Search = new System.Windows.Forms.Button();
this.txt_Search = new System.Windows.Forms.TextBox();
this.lsb_FoodList = new System.Windows.Forms.ListBox();
this.lbl_Caption = new System.Windows.Forms.Label();
this.tab_2 = new System.Windows.Forms.TabPage();
this.btn_Cancel = new System.Windows.Forms.Button();
this.btn_OK = new System.Windows.Forms.Button();
this.cmb_Handshake = new System.Windows.Forms.ComboBox();
this.cmb_Parity = new System.Windows.Forms.ComboBox();
this.txt_DataBits = new System.Windows.Forms.TextBox();
this.cmb_StopBits = new System.Windows.Forms.ComboBox();
this.cmb_BaudRate = new System.Windows.Forms.ComboBox();
this.lbl_PortName = new System.Windows.Forms.Label();
this.lbl_HandShake = new System.Windows.Forms.Label();
this.lbl_Parity = new System.Windows.Forms.Label();
this.lbl_StopBits = new System.Windows.Forms.Label();
this.lbl_DataBits = new System.Windows.Forms.Label();
this.lbl_BaudRate = new System.Windows.Forms.Label();
this.cmb_PortName = new System.Windows.Forms.ComboBox();
this.tab_Main.SuspendLayout();
this.tab_1.SuspendLayout();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox11)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox40)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox26)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox27)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox28)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox29)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox30)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox31)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox32)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox33)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox34)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox35)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox21)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox10)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox12)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox13)).BeginInit();
```

```
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox14)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox15)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox7)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox8)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox9)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox6)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox5)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox4)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
this.tab_2.SuspendLayout();
this.SuspendLayout();
//
// btn_Connect
//
this.btn_Connect.Location = new System.Drawing.Point(10, 385);
this.btn_Connect.Name = "btn_Connect";
this.btn_Connect.Size = new System.Drawing.Size(150, 60);
this.btn_Connect.TabIndex = 3;
this.btn_Connect.Text = "Connect";
this.btn_Connect.UseVisualStyleBackColor = true;
this.btn_Connect.Click += new System.EventHandler(this.btn_Connect_Click);
//
// timer1
//
this.timer1.Interval = 200;
this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
//
// btn_RecalibrateScale
//
this.btn_RecalibrateScale.Location = new System.Drawing.Point(10, 454);
this.btn_RecalibrateScale.Name = "btn_RecalibrateScale";
this.btn_RecalibrateScale.Size = new System.Drawing.Size(150, 60);
this.btn_RecalibrateScale.TabIndex = 8;
this.btn_RecalibrateScale.Text = "Recalibrate Scale";
this.btn_RecalibrateScale.UseVisualStyleBackColor = true;
this.btn_RecalibrateScale.Click += new System.EventHandler(this.btn_RecalibrateScale_Click);
//
// lbl_Status
//
this.lbl_Status.AutoSize = true;
this.lbl_Status.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_Status.Location = new System.Drawing.Point(6, 85);
this.lbl_Status.Name = "lbl_Status";
this.lbl_Status.Size = new System.Drawing.Size(78, 20);
this.lbl_Status.TabIndex = 8;
this.lbl_Status.Text = "STATUS";
//
// btn_Disconnect
//
this.btn_Disconnect.Location = new System.Drawing.Point(179, 385);
this.btn_Disconnect.Name = "btn_Disconnect";
this.btn_Disconnect.Size = new System.Drawing.Size(150, 60);
this.btn_Disconnect.TabIndex = 9;
this.btn_Disconnect.Text = "Disconnect";
this.btn_Disconnect.UseVisualStyleBackColor = true;
this.btn_Disconnect.Click += new System.EventHandler(this.btn_Disconnect_Click);
//
// btn_Exit
//
this.btn_Exit.Location = new System.Drawing.Point(179, 454);
this.btn_Exit.Name = "btn_Exit";
this.btn_Exit.Size = new System.Drawing.Size(150, 60);
this.btn_Exit.TabIndex = 10;
this.btn_Exit.Text = "Exit";
this.btn_Exit.UseVisualStyleBackColor = true;
this.btn_Exit.Click += new System.EventHandler(this.btn_Exit_Click);
//
// txt_Display
//
this.txt_Display.BackColor = System.Drawing.Color.MidnightBlue;
this.txt_Display.Font = new System.Drawing.Font("Century Gothic", 72F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txt_Display.ForeColor = System.Drawing.Color.Azure;
this.txt_Display.Location = new System.Drawing.Point(7, 238);
this.txt_Display.Name = "txt_Display";
this.txt_Display.Size = new System.Drawing.Size(322, 125);
this.txt_Display.TabIndex = 11;
this.txt_Display.Text = "0g";
this.txt_Display.TextAlign = System.Windows.Forms.HorizontalAlignment.Right;
//
// tab_Main
//
this.tab_Main.Controls.Add(this.tab_1);
this.tab_Main.Controls.Add(this.tab_2);
this.tab_Main.Location = new System.Drawing.Point(12, 12);
this.tab_Main.Name = "tab_Main";
this.tab_Main.SelectedIndex = 0;
```

```
this.tab_Main.Size = new System.Drawing.Size(1240, 738);
this.tab_Main.TabIndex = 12;
//
// tab_1
//
this.tab_1.Controls.Add(this.bt_ClearLog);
this.tab_1.Controls.Add(this.bt_AddToLog);
this.tab_1.Controls.Add(this.lsb_Log);
this.tab_1.Controls.Add(this.txt_Name);
this.tab_1.Controls.Add(this.lbl_ServingSize_g);
this.tab_1.Controls.Add(this.lbl_Protein_g);
this.tab_1.Controls.Add(this.lbl_Sugars_g);
this.tab_1.Controls.Add(this.lbl_Sodium_mg);
this.tab_1.Controls.Add(this.lbl_TransFat_g);
this.tab_1.Controls.Add(this.lbl_PolyunsaturatedFat_g);
this.tab_1.Controls.Add(this.lbl_Calories_kcal);
this.tab_1.Controls.Add(this.lbl_DietaryFiber_g);
this.tab_1.Controls.Add(this.lbl_TotalCarbohydrate_g);
this.tab_1.Controls.Add(this.lbl_SaturatedFat_g);
this.tab_1.Controls.Add(this.lbl_Potassium_mg);
this.tab_1.Controls.Add(this.lbl_Cholesterol_mg);
this.tab_1.Controls.Add(this.lbl_MonounsaturatedFat_g);
this.tab_1.Controls.Add(this.lbl_TotalFat_g);
this.tab_1.Controls.Add(this.lbl_Cholesterol_pct);
this.tab_1.Controls.Add(this.lbl_SaturatedFat_pct);
this.tab_1.Controls.Add(this.lbl_TotalFat_pct);
this.tab_1.Controls.Add(this.lbl_DietaryFiber_pct);
this.tab_1.Controls.Add(this.lbl_Potassium_pct);
this.tab_1.Controls.Add(this.lbl_Sodium_pct);
this.tab_1.Controls.Add(this.lbl_VitaminC_pct);
this.tab_1.Controls.Add(this.lbl_VitaminD_pct);
this.tab_1.Controls.Add(this.lbl_Protein_pct);
this.tab_1.Controls.Add(this.lbl_TotalCarbohydrate_pct);
this.tab_1.Controls.Add(this.lbl_VitaminE_pct);
this.tab_1.Controls.Add(this.lbl_Iron_pct);
this.tab_1.Controls.Add(this.lbl_Calcium_pct);
this.tab_1.Controls.Add(this.lbl_VitaminA_pct);
this.tab_1.Controls.Add(this.lbl_Magnesium_pct);
this.tab_1.Controls.Add(this.lbl_VitaminB6_pct);
this.tab_1.Controls.Add(this.lbl_Riboflavin_pct);
this.tab_1.Controls.Add(this.lbl_VitaminK_pct);
this.tab_1.Controls.Add(this.lbl_Folate_pct);
this.tab_1.Controls.Add(this.lbl_Manganese_pct);
this.tab_1.Controls.Add(this.lbl_Selenium_pct);
this.tab_1.Controls.Add(this.lbl_VitaminB12_pct);
this.tab_1.Controls.Add(this.lbl_Zinc_pct);
this.tab_1.Controls.Add(this.lbl_Phosphorus_pct);
this.tab_1.Controls.Add(this.lbl_Niacin_pct);
this.tab_1.Controls.Add(this.lbl_Thiamin_pct);
this.tab_1.Controls.Add(this.lbl_Copper_pct);
this.tab_1.Controls.Add(this.pictureBox11);
this.tab_1.Controls.Add(this.lbl_Dot9);
this.tab_1.Controls.Add(this.lbl_Dot8);
this.tab_1.Controls.Add(this.lbl_Dot7);
this.tab_1.Controls.Add(this.lbl_Dot6);
this.tab_1.Controls.Add(this.lbl_Dot5);
this.tab_1.Controls.Add(this.lbl_Dot2);
this.tab_1.Controls.Add(this.lbl_Dot4);
this.tab_1.Controls.Add(this.lbl_Dot3);
this.tab_1.Controls.Add(this.pictureBox40);
this.tab_1.Controls.Add(this.pictureBox26);
this.tab_1.Controls.Add(this.pictureBox27);
this.tab_1.Controls.Add(this.pictureBox28);
this.tab_1.Controls.Add(this.pictureBox29);
this.tab_1.Controls.Add(this.pictureBox30);
this.tab_1.Controls.Add(this.pictureBox31);
this.tab_1.Controls.Add(this.pictureBox32);
this.tab_1.Controls.Add(this.pictureBox33);
this.tab_1.Controls.Add(this.pictureBox34);
this.tab_1.Controls.Add(this.pictureBox35);
this.tab_1.Controls.Add(this.pictureBox21);
this.tab_1.Controls.Add(this.pictureBox10);
this.tab_1.Controls.Add(this.pictureBox12);
this.tab_1.Controls.Add(this.pictureBox13);
this.tab_1.Controls.Add(this.pictureBox14);
this.tab_1.Controls.Add(this.pictureBox15);
this.tab_1.Controls.Add(this.pictureBox7);
this.tab_1.Controls.Add(this.pictureBox8);
this.tab_1.Controls.Add(this.pictureBox9);
this.tab_1.Controls.Add(this.pictureBox6);
this.tab_1.Controls.Add(this.pictureBox5);
this.tab_1.Controls.Add(this.pictureBox4);
this.tab_1.Controls.Add(this.pictureBox3);
this.tab_1.Controls.Add(this.pictureBox2);
this.tab_1.Controls.Add(this.pictureBox1);
this.tab_1.Controls.Add(this.lbl_Manganese);
this.tab_1.Controls.Add(this.lbl_Copper);
this.tab_1.Controls.Add(this.lbl_Selenium);
```

```
this.tab_1.Controls.Add(this.lbl_Zinc);
this.tab_1.Controls.Add(this.lbl_Magnesium);
this.tab_1.Controls.Add(this.lbl_Phosphorus);
this.tab_1.Controls.Add(this.lbl_VitaminB12);
this.tab_1.Controls.Add(this.lbl_Folate);
this.tab_1.Controls.Add(this.lbl_VitaminB6);
this.tab_1.Controls.Add(this.lbl_Niacin);
this.tab_1.Controls.Add(this.lbl_Riboflavin);
this.tab_1.Controls.Add(this.lbl_Thiamin);
this.tab_1.Controls.Add(this.lbl_VitaminK);
this.tab_1.Controls.Add(this.lbl_VitaminE);
this.tab_1.Controls.Add(this.lbl_VitaminD);
this.tab_1.Controls.Add(this.lbl_Iron);
this.tab_1.Controls.Add(this.lbl_Calcium);
this.tab_1.Controls.Add(this.lbl_Dot1);
this.tab_1.Controls.Add(this.lbl_VitaminC);
this.tab_1.Controls.Add(this.lbl_VitaminA);
this.tab_1.Controls.Add(this.lbl_Protein);
this.tab_1.Controls.Add(this.lbl_Sugars);
this.tab_1.Controls.Add(this.lbl_DietaryFiber);
this.tab_1.Controls.Add(this.lbl_TotalCarbohydrate);
this.tab_1.Controls.Add(this.lbl_Potassium);
this.tab_1.Controls.Add(this.lbl_Sodium);
this.tab_1.Controls.Add(this.lbl_Cholesterol);
this.tab_1.Controls.Add(this.lbl_TransFat);
this.tab_1.Controls.Add(this.lbl_PolyunsaturatedFat);
this.tab_1.Controls.Add(this.lbl_MonounsaturatedFat);
this.tab_1.Controls.Add(this.lbl_SaturatedFat);
this.tab_1.Controls.Add(this.lbl_Calories);
this.tab_1.Controls.Add(this.lbl_ServingSize);
this.tab_1.Controls.Add(this.lbl_NutritionFacts);
this.tab_1.Controls.Add(this.lbl_TotalFat);
this.tab_1.Controls.Add(this.btn_RecalibrateScale);
this.tab_1.Controls.Add(this.btn_FreezeUnfreezeDisplay);
this.tab_1.Controls.Add(this.btn_DisplayAllItems);
this.tab_1.Controls.Add(this.txt_Display);
this.tab_1.Controls.Add(this.btn_Search);
this.tab_1.Controls.Add(this.txt_Search);
this.tab_1.Controls.Add(this.lsb_FoodList);
this.tab_1.Controls.Add(this.lbl_Caption);
this.tab_1.Controls.Add(this.lbl_Status);
this.tab_1.Controls.Add(this.btn_Exit);
this.tab_1.Controls.Add(this.btn_Disconnect);
this.tab_1.Controls.Add(this.btn_Connect);
this.tab_1.Location = new System.Drawing.Point(4, 22);
this.tab_1.Name = "tab_1";
this.tab_1.Padding = new System.Windows.Forms.Padding(3);
this.tab_1.Size = new System.Drawing.Size(1232, 712);
this.tab_1.TabIndex = 0;
this.tab_1.Text = "Main";
this.tab_1.UseVisualStyleBackColor = true;
//
// bt_ClearLog
//
this.bt_ClearLog.Location = new System.Drawing.Point(1063, 13);
this.bt_ClearLog.Name = "bt_ClearLog";
this.bt_ClearLog.Size = new System.Drawing.Size(120, 40);
this.bt_ClearLog.TabIndex = 155;
this.bt_ClearLog.Text = "Clear Log";
this.bt_ClearLog.UseVisualStyleBackColor = true;
this.bt_ClearLog.Click += new System.EventHandler(this.bt_ClearLog_Click);
//
// bt_AddToLog
//
this.bt_AddToLog.Location = new System.Drawing.Point(933, 13);
this.bt_AddToLog.Name = "bt_AddToLog";
this.bt_AddToLog.Size = new System.Drawing.Size(120, 40);
this.bt_AddToLog.TabIndex = 154;
this.bt_AddToLog.Text = "Add to Log";
this.bt_AddToLog.UseVisualStyleBackColor = true;
this.bt_AddToLog.Click += new System.EventHandler(this.bt_AddToLog_Click);
//
// lsb_Log
//
this.lsb_Log.FormattingEnabled = true;
this.lsb_Log.Location = new System.Drawing.Point(933, 72);
this.lsb_Log.Name = "lsb_Log";
this.lsb_Log.Size = new System.Drawing.Size(250, 628);
this.lsb_Log.TabIndex = 153;
this.lsb_Log.SelectedIndexChanged += new System.EventHandler(this.lsb_Log_SelectedIndexChanged);
//
// txt_Name
//
this.txt_Name.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txt_Name.Location = new System.Drawing.Point(624, 67);
this.txt_Name.Name = "txt_Name";
this.txt_Name.ReadOnly = true;
this.txt_Name.Size = new System.Drawing.Size(285, 26);
```

```
this.txt_Name.TabIndex = 152;
//
// lbl_ServingSize_g
//
this.lbl_ServingSize_g.AutoSize = true;
this.lbl_ServingSize_g.Location = new System.Drawing.Point(702, 132);
this.lbl_ServingSize_g.Name = "lbl_ServingSize_g";
this.lbl_ServingSize_g.Size = new System.Drawing.Size(13, 13);
this.lbl_ServingSize_g.TabIndex = 151;
this.lbl_ServingSize_g.Text = "g";
//
// lbl_Protein_g
//
this.lbl_Protein_g.AutoSize = true;
this.lbl_Protein_g.Location = new System.Drawing.Point(683, 405);
this.lbl_Protein_g.Name = "lbl_Protein_g";
this.lbl_Protein_g.Size = new System.Drawing.Size(19, 13);
this.lbl_Protein_g.TabIndex = 150;
this.lbl_Protein_g.Text = "0g";
//
// lbl_Sugars_g
//
this.lbl_Sugars_g.AutoSize = true;
this.lbl_Sugars_g.Location = new System.Drawing.Point(691, 385);
this.lbl_Sugars_g.Name = "lbl_Sugars_g";
this.lbl_Sugars_g.Size = new System.Drawing.Size(19, 13);
this.lbl_Sugars_g.TabIndex = 149;
this.lbl_Sugars_g.Text = "0g";
//
// lbl_Sodium_mg
//
this.lbl_Sodium_mg.AutoSize = true;
this.lbl_Sodium_mg.Location = new System.Drawing.Point(684, 305);
this.lbl_Sodium_mg.Name = "lbl_Sodium_mg";
this.lbl_Sodium_mg.Size = new System.Drawing.Size(27, 13);
this.lbl_Sodium_mg.TabIndex = 148;
this.lbl_Sodium_mg.Text = "0mg";
//
// lbl_TransFat_g
//
this.lbl_TransFat_g.AutoSize = true;
this.lbl_TransFat_g.Location = new System.Drawing.Point(703, 265);
this.lbl_TransFat_g.Name = "lbl_TransFat_g";
this.lbl_TransFat_g.Size = new System.Drawing.Size(19, 13);
this.lbl_TransFat_g.TabIndex = 147;
this.lbl_TransFat_g.Text = "0g";
//
// lbl_PolyunsaturatedFat_g
//
this.lbl_PolyunsaturatedFat_g.AutoSize = true;
this.lbl_PolyunsaturatedFat_g.Location = new System.Drawing.Point(752, 245);
this.lbl_PolyunsaturatedFat_g.Name = "lbl_PolyunsaturatedFat_g";
this.lbl_PolyunsaturatedFat_g.Size = new System.Drawing.Size(19, 13);
this.lbl_PolyunsaturatedFat_g.TabIndex = 146;
this.lbl_PolyunsaturatedFat_g.Text = "0g";
//
// lbl_Calories_kcal
//
this.lbl_Calories_kcal.AutoSize = true;
this.lbl_Calories_kcal.Location = new System.Drawing.Point(688, 161);
this.lbl_Calories_kcal.Name = "lbl_Calories_kcal";
this.lbl_Calories_kcal.Size = new System.Drawing.Size(13, 13);
this.lbl_Calories_kcal.TabIndex = 145;
this.lbl_Calories_kcal.Text = "0";
//
// lbl_DietaryFiber_g
//
this.lbl_DietaryFiber_g.AutoSize = true;
this.lbl_DietaryFiber_g.Location = new System.Drawing.Point(717, 365);
this.lbl_DietaryFiber_g.Name = "lbl_DietaryFiber_g";
this.lbl_DietaryFiber_g.Size = new System.Drawing.Size(19, 13);
this.lbl_DietaryFiber_g.TabIndex = 143;
this.lbl_DietaryFiber_g.Text = "0g";
//
// lbl_TotalCarbohydrate_g
//
this.lbl_TotalCarbohydrate_g.AutoSize = true;
this.lbl_TotalCarbohydrate_g.Location = new System.Drawing.Point(751, 345);
this.lbl_TotalCarbohydrate_g.Name = "lbl_TotalCarbohydrate_g";
this.lbl_TotalCarbohydrate_g.Size = new System.Drawing.Size(19, 13);
this.lbl_TotalCarbohydrate_g.TabIndex = 142;
this.lbl_TotalCarbohydrate_g.Text = "0g";
//
// lbl_SaturatedFat_g
//
this.lbl_SaturatedFat_g.AutoSize = true;
this.lbl_SaturatedFat_g.Location = new System.Drawing.Point(722, 205);
this.lbl_SaturatedFat_g.Name = "lbl_SaturatedFat_g";
```

```
this.lbl_SaturatedFat_g.Size = new System.Drawing.Size(19, 13);
this.lbl_SaturatedFat_g.TabIndex = 141;
this.lbl_SaturatedFat_g.Text = "0g";
//
// lbl_Potassium_mg
//
this.lbl_Potassium_mg.AutoSize = true;
this.lbl_Potassium_mg.Location = new System.Drawing.Point(700, 325);
this.lbl_Potassium_mg.Name = "lbl_Potassium_mg";
this.lbl_Potassium_mg.Size = new System.Drawing.Size(27, 13);
this.lbl_Potassium_mg.TabIndex = 140;
this.lbl_Potassium_mg.Text = "0mg";
//
// lbl_Cholesterol_mg
//
this.lbl_Cholesterol_mg.AutoSize = true;
this.lbl_Cholesterol_mg.Location = new System.Drawing.Point(706, 285);
this.lbl_Cholesterol_mg.Name = "lbl_Cholesterol_mg";
this.lbl_Cholesterol_mg.Size = new System.Drawing.Size(27, 13);
this.lbl_Cholesterol_mg.TabIndex = 139;
this.lbl_Cholesterol_mg.Text = "0mg";
//
// lbl_MonounsaturatedFat_g
//
this.lbl_MonounsaturatedFat_g.AutoSize = true;
this.lbl_MonounsaturatedFat_g.Location = new System.Drawing.Point(759, 225);
this.lbl_MonounsaturatedFat_g.Name = "lbl_MonounsaturatedFat_g";
this.lbl_MonounsaturatedFat_g.Size = new System.Drawing.Size(19, 13);
this.lbl_MonounsaturatedFat_g.TabIndex = 138;
this.lbl_MonounsaturatedFat_g.Text = "0g";
//
// lbl_TotalFat_g
//
this.lbl_TotalFat_g.AutoSize = true;
this.lbl_TotalFat_g.Location = new System.Drawing.Point(694, 185);
this.lbl_TotalFat_g.Name = "lbl_TotalFat_g";
this.lbl_TotalFat_g.Size = new System.Drawing.Size(19, 13);
this.lbl_TotalFat_g.TabIndex = 137;
this.lbl_TotalFat_g.Text = "0g";
//
// lbl_Cholesterol_pct
//
this.lbl_Cholesterol_pct.Location = new System.Drawing.Point(843, 285);
this.lbl_Cholesterol_pct.Name = "lbl_Cholesterol_pct";
this.lbl_Cholesterol_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Cholesterol_pct.TabIndex = 136;
this.lbl_Cholesterol_pct.Text = "%";
this.lbl_Cholesterol_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_SaturatedFat_pct
//
this.lbl_SaturatedFat_pct.Location = new System.Drawing.Point(843, 205);
this.lbl_SaturatedFat_pct.Name = "lbl_SaturatedFat_pct";
this.lbl_SaturatedFat_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_SaturatedFat_pct.TabIndex = 134;
this.lbl_SaturatedFat_pct.Text = "%";
this.lbl_SaturatedFat_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_TotalFat_pct
//
this.lbl_TotalFat_pct.Location = new System.Drawing.Point(843, 185);
this.lbl_TotalFat_pct.Name = "lbl_TotalFat_pct";
this.lbl_TotalFat_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_TotalFat_pct.TabIndex = 133;
this.lbl_TotalFat_pct.Text = "%";
this.lbl_TotalFat_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_DietaryFiber_pct
//
this.lbl_DietaryFiber_pct.Location = new System.Drawing.Point(843, 365);
this.lbl_DietaryFiber_pct.Name = "lbl_DietaryFiber_pct";
this.lbl_DietaryFiber_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_DietaryFiber_pct.TabIndex = 132;
this.lbl_DietaryFiber_pct.Text = "%";
this.lbl_DietaryFiber_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Potassium_pct
//
this.lbl_Potassium_pct.Location = new System.Drawing.Point(843, 325);
this.lbl_Potassium_pct.Name = "lbl_Potassium_pct";
this.lbl_Potassium_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Potassium_pct.TabIndex = 131;
this.lbl_Potassium_pct.Text = "%";
this.lbl_Potassium_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Sodium_pct
//
this.lbl_Sodium_pct.Location = new System.Drawing.Point(843, 305);
```

```
this.lbl_Sodium_pct.Name = "lbl_Sodium_pct";
this.lbl_Sodium_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Sodium_pct.TabIndex = 130;
this.lbl_Sodium_pct.Text = "%";
this.lbl_Sodium_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_VitaminC_pct
//
this.lbl_VitaminC_pct.Location = new System.Drawing.Point(843, 434);
this.lbl_VitaminC_pct.Name = "lbl_VitaminC_pct";
this.lbl_VitaminC_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_VitaminC_pct.TabIndex = 128;
this.lbl_VitaminC_pct.Text = "%";
this.lbl_VitaminC_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_VitaminD_pct
//
this.lbl_VitaminD_pct.Location = new System.Drawing.Point(696, 474);
this.lbl_VitaminD_pct.Name = "lbl_VitaminD_pct";
this.lbl_VitaminD_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_VitaminD_pct.TabIndex = 127;
this.lbl_VitaminD_pct.Text = "%";
this.lbl_VitaminD_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Protein_pct
//
this.lbl_Protein_pct.Location = new System.Drawing.Point(843, 405);
this.lbl_Protein_pct.Name = "lbl_Protein_pct";
this.lbl_Protein_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Protein_pct.TabIndex = 126;
this.lbl_Protein_pct.Text = "%";
this.lbl_Protein_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_TotalCarbohydrate_pct
//
this.lbl_TotalCarbohydrate_pct.Location = new System.Drawing.Point(843, 345);
this.lbl_TotalCarbohydrate_pct.Name = "lbl_TotalCarbohydrate_pct";
this.lbl_TotalCarbohydrate_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_TotalCarbohydrate_pct.TabIndex = 125;
this.lbl_TotalCarbohydrate_pct.Text = "%";
this.lbl_TotalCarbohydrate_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_VitaminE_pct
//
this.lbl_VitaminE_pct.Location = new System.Drawing.Point(843, 474);
this.lbl_VitaminE_pct.Name = "lbl_VitaminE_pct";
this.lbl_VitaminE_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_VitaminE_pct.TabIndex = 124;
this.lbl_VitaminE_pct.Text = "%";
this.lbl_VitaminE_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Iron_pct
//
this.lbl_Iron_pct.Location = new System.Drawing.Point(843, 454);
this.lbl_Iron_pct.Name = "lbl_Iron_pct";
this.lbl_Iron_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Iron_pct.TabIndex = 123;
this.lbl_Iron_pct.Text = "%";
this.lbl_Iron_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Calcium_pct
//
this.lbl_Calcium_pct.Location = new System.Drawing.Point(696, 454);
this.lbl_Calcium_pct.Name = "lbl_Calcium_pct";
this.lbl_Calcium_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Calcium_pct.TabIndex = 122;
this.lbl_Calcium_pct.Text = "%";
this.lbl_Calcium_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_VitaminA_pct
//
this.lbl_VitaminA_pct.Location = new System.Drawing.Point(696, 434);
this.lbl_VitaminA_pct.Name = "lbl_VitaminA_pct";
this.lbl_VitaminA_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_VitaminA_pct.TabIndex = 121;
this.lbl_VitaminA_pct.Text = "%";
this.lbl_VitaminA_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Magnesium_pct
//
this.lbl_Magnesium_pct.Location = new System.Drawing.Point(696, 574);
this.lbl_Magnesium_pct.Name = "lbl_Magnesium_pct";
this.lbl_Magnesium_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Magnesium_pct.TabIndex = 120;
this.lbl_Magnesium_pct.Text = "%";
this.lbl_Magnesium_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_VitaminB6_pct
```

```
//
this.lbl_VitaminB6_pct.Location = new System.Drawing.Point(696, 534);
this.lbl_VitaminB6_pct.Name = "lbl_VitaminB6_pct";
this.lbl_VitaminB6_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_VitaminB6_pct.TabIndex = 119;
this.lbl_VitaminB6_pct.Text = "%";
this.lbl_VitaminB6_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Riboflavin_pct
//
this.lbl_Riboflavin_pct.Location = new System.Drawing.Point(696, 514);
this.lbl_Riboflavin_pct.Name = "lbl_Riboflavin_pct";
this.lbl_Riboflavin_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Riboflavin_pct.TabIndex = 118;
this.lbl_Riboflavin_pct.Text = "%";
this.lbl_Riboflavin_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_VitaminK_pct
//
this.lbl_VitaminK_pct.Location = new System.Drawing.Point(696, 494);
this.lbl_VitaminK_pct.Name = "lbl_VitaminK_pct";
this.lbl_VitaminK_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_VitaminK_pct.TabIndex = 117;
this.lbl_VitaminK_pct.Text = "%";
this.lbl_VitaminK_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Folate_pct
//
this.lbl_Folate_pct.Location = new System.Drawing.Point(843, 534);
this.lbl_Folate_pct.Name = "lbl_Folate_pct";
this.lbl_Folate_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Folate_pct.TabIndex = 116;
this.lbl_Folate_pct.Text = "%";
this.lbl_Folate_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Manganese_pct
//
this.lbl_Manganese_pct.Location = new System.Drawing.Point(696, 614);
this.lbl_Manganese_pct.Name = "lbl_Manganese_pct";
this.lbl_Manganese_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Manganese_pct.TabIndex = 115;
this.lbl_Manganese_pct.Text = "%";
this.lbl_Manganese_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Selenium_pct
//
this.lbl_Selenium_pct.Location = new System.Drawing.Point(696, 594);
this.lbl_Selenium_pct.Name = "lbl_Selenium_pct";
this.lbl_Selenium_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Selenium_pct.TabIndex = 114;
this.lbl_Selenium_pct.Text = "%";
this.lbl_Selenium_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_VitaminB12_pct
//
this.lbl_VitaminB12_pct.Location = new System.Drawing.Point(696, 554);
this.lbl_VitaminB12_pct.Name = "lbl_VitaminB12_pct";
this.lbl_VitaminB12_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_VitaminB12_pct.TabIndex = 113;
this.lbl_VitaminB12_pct.Text = "%";
this.lbl_VitaminB12_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Zinc_pct
//
this.lbl_Zinc_pct.Location = new System.Drawing.Point(843, 574);
this.lbl_Zinc_pct.Name = "lbl_Zinc_pct";
this.lbl_Zinc_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Zinc_pct.TabIndex = 112;
this.lbl_Zinc_pct.Text = "%";
this.lbl_Zinc_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Phosphorus_pct
//
this.lbl_Phosphorus_pct.Location = new System.Drawing.Point(843, 554);
this.lbl_Phosphorus_pct.Name = "lbl_Phosphorus_pct";
this.lbl_Phosphorus_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Phosphorus_pct.TabIndex = 111;
this.lbl_Phosphorus_pct.Text = "%";
this.lbl_Phosphorus_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Niacin_pct
//
this.lbl_Niacin_pct.Location = new System.Drawing.Point(843, 514);
this.lbl_Niacin_pct.Name = "lbl_Niacin_pct";
this.lbl_Niacin_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Niacin_pct.TabIndex = 110;
this.lbl_Niacin_pct.Text = "%";
this.lbl_Niacin_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
```

```
//
// lbl_Thiamin_pct
//
this.lbl_Thiamin_pct.Location = new System.Drawing.Point(843, 494);
this.lbl_Thiamin_pct.Name = "lbl_Thiamin_pct";
this.lbl_Thiamin_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Thiamin_pct.TabIndex = 109;
this.lbl_Thiamin_pct.Text = "%";
this.lbl_Thiamin_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// lbl_Copper_pct
//
this.lbl_Copper_pct.Location = new System.Drawing.Point(843, 594);
this.lbl_Copper_pct.Name = "lbl_Copper_pct";
this.lbl_Copper_pct.Size = new System.Drawing.Size(55, 13);
this.lbl_Copper_pct.TabIndex = 105;
this.lbl_Copper_pct.Text = "%";
this.lbl_Copper_pct.TextAlign = System.Drawing.ContentAlignment.TopRight;
//
// pictureBox11
//
this.pictureBox11.BackColor = System.Drawing.Color.Black;
this.pictureBox11.Location = new System.Drawing.Point(908, 104);
this.pictureBox11.Name = "pictureBox11";
this.pictureBox11.Size = new System.Drawing.Size(1, 527);
this.pictureBox11.TabIndex = 104;
this.pictureBox11.TabStop = false;
//
// lbl_Dot9
//
this.lbl_Dot9.AutoSize = true;
this.lbl_Dot9.Location = new System.Drawing.Point(757, 594);
this.lbl_Dot9.Name = "lbl_Dot9";
this.lbl_Dot9.Size = new System.Drawing.Size(13, 13);
this.lbl_Dot9.TabIndex = 103;
this.lbl_Dot9.Text = "•";
//
// lbl_Dot8
//
this.lbl_Dot8.AutoSize = true;
this.lbl_Dot8.Location = new System.Drawing.Point(757, 574);
this.lbl_Dot8.Name = "lbl_Dot8";
this.lbl_Dot8.Size = new System.Drawing.Size(13, 13);
this.lbl_Dot8.TabIndex = 102;
this.lbl_Dot8.Text = "•";
//
// lbl_Dot7
//
this.lbl_Dot7.AutoSize = true;
this.lbl_Dot7.Location = new System.Drawing.Point(757, 554);
this.lbl_Dot7.Name = "lbl_Dot7";
this.lbl_Dot7.Size = new System.Drawing.Size(13, 13);
this.lbl_Dot7.TabIndex = 101;
this.lbl_Dot7.Text = "•";
//
// lbl_Dot6
//
this.lbl_Dot6.AutoSize = true;
this.lbl_Dot6.Location = new System.Drawing.Point(757, 534);
this.lbl_Dot6.Name = "lbl_Dot6";
this.lbl_Dot6.Size = new System.Drawing.Size(13, 13);
this.lbl_Dot6.TabIndex = 100;
this.lbl_Dot6.Text = "•";
//
// lbl_Dot5
//
this.lbl_Dot5.AutoSize = true;
this.lbl_Dot5.Location = new System.Drawing.Point(757, 514);
this.lbl_Dot5.Name = "lbl_Dot5";
this.lbl_Dot5.Size = new System.Drawing.Size(13, 13);
this.lbl_Dot5.TabIndex = 99;
this.lbl_Dot5.Text = "•";
//
// lbl_Dot2
//
this.lbl_Dot2.AutoSize = true;
this.lbl_Dot2.Location = new System.Drawing.Point(757, 454);
this.lbl_Dot2.Name = "lbl_Dot2";
this.lbl_Dot2.Size = new System.Drawing.Size(13, 13);
this.lbl_Dot2.TabIndex = 98;
this.lbl_Dot2.Text = "•";
//
// lbl_Dot4
//
this.lbl_Dot4.AutoSize = true;
this.lbl_Dot4.Location = new System.Drawing.Point(757, 494);
this.lbl_Dot4.Name = "lbl_Dot4";
this.lbl_Dot4.Size = new System.Drawing.Size(13, 13);
```

```
this.lbl_Dot4.TabIndex = 97;
this.lbl_Dot4.Text = "•";
//
// lbl_Dot3
//
this.lbl_Dot3.AutoSize = true;
this.lbl_Dot3.Location = new System.Drawing.Point(757, 474);
this.lbl_Dot3.Name = "lbl_Dot3";
this.lbl_Dot3.Size = new System.Drawing.Size(13, 13);
this.lbl_Dot3.TabIndex = 96;
this.lbl_Dot3.Text = "•";
//
// pictureBox40
//
this.pictureBox40.BackColor = System.Drawing.Color.Black;
this.pictureBox40.Location = new System.Drawing.Point(633, 421);
this.pictureBox40.Name = "pictureBox40";
this.pictureBox40.Size = new System.Drawing.Size(265, 10);
this.pictureBox40.TabIndex = 95;
this.pictureBox40.TabStop = false;
//
// pictureBox26
//
this.pictureBox26.BackColor = System.Drawing.Color.Black;
this.pictureBox26.Location = new System.Drawing.Point(624, 104);
this.pictureBox26.Name = "pictureBox26";
this.pictureBox26.Size = new System.Drawing.Size(1, 527);
this.pictureBox26.TabIndex = 90;
this.pictureBox26.TabStop = false;
//
// pictureBox27
//
this.pictureBox27.BackColor = System.Drawing.Color.Black;
this.pictureBox27.Location = new System.Drawing.Point(624, 104);
this.pictureBox27.Name = "pictureBox27";
this.pictureBox27.Size = new System.Drawing.Size(285, 1);
this.pictureBox27.TabIndex = 89;
this.pictureBox27.TabStop = false;
//
// pictureBox28
//
this.pictureBox28.BackColor = System.Drawing.Color.Black;
this.pictureBox28.Location = new System.Drawing.Point(624, 630);
this.pictureBox28.Name = "pictureBox28";
this.pictureBox28.Size = new System.Drawing.Size(285, 1);
this.pictureBox28.TabIndex = 88;
this.pictureBox28.TabStop = false;
//
// pictureBox29
//
this.pictureBox29.BackColor = System.Drawing.Color.Black;
this.pictureBox29.Location = new System.Drawing.Point(633, 610);
this.pictureBox29.Name = "pictureBox29";
this.pictureBox29.Size = new System.Drawing.Size(265, 1);
this.pictureBox29.TabIndex = 87;
this.pictureBox29.TabStop = false;
//
// pictureBox30
//
this.pictureBox30.BackColor = System.Drawing.Color.Black;
this.pictureBox30.Location = new System.Drawing.Point(633, 590);
this.pictureBox30.Name = "pictureBox30";
this.pictureBox30.Size = new System.Drawing.Size(265, 1);
this.pictureBox30.TabIndex = 86;
this.pictureBox30.TabStop = false;
//
// pictureBox31
//
this.pictureBox31.BackColor = System.Drawing.Color.Black;
this.pictureBox31.Location = new System.Drawing.Point(633, 570);
this.pictureBox31.Name = "pictureBox31";
this.pictureBox31.Size = new System.Drawing.Size(265, 1);
this.pictureBox31.TabIndex = 85;
this.pictureBox31.TabStop = false;
//
// pictureBox32
//
this.pictureBox32.BackColor = System.Drawing.Color.Black;
this.pictureBox32.Location = new System.Drawing.Point(633, 550);
this.pictureBox32.Name = "pictureBox32";
this.pictureBox32.Size = new System.Drawing.Size(265, 1);
this.pictureBox32.TabIndex = 84;
this.pictureBox32.TabStop = false;
//
// pictureBox33
//
this.pictureBox33.BackColor = System.Drawing.Color.Black;
this.pictureBox33.Location = new System.Drawing.Point(633, 530);
```

```
this.pictureBox33.Name = "pictureBox33";
this.pictureBox33.Size = new System.Drawing.Size(265, 1);
this.pictureBox33.TabIndex = 83;
this.pictureBox33.TabStop = false;
//
// pictureBox34
//
this.pictureBox34.BackColor = System.Drawing.Color.Black;
this.pictureBox34.Location = new System.Drawing.Point(633, 510);
this.pictureBox34.Name = "pictureBox34";
this.pictureBox34.Size = new System.Drawing.Size(265, 1);
this.pictureBox34.TabIndex = 82;
this.pictureBox34.TabStop = false;
//
// pictureBox35
//
this.pictureBox35.BackColor = System.Drawing.Color.Black;
this.pictureBox35.Location = new System.Drawing.Point(633, 490);
this.pictureBox35.Name = "pictureBox35";
this.pictureBox35.Size = new System.Drawing.Size(265, 1);
this.pictureBox35.TabIndex = 81;
this.pictureBox35.TabStop = false;
//
// pictureBox21
//
this.pictureBox21.BackColor = System.Drawing.Color.Black;
this.pictureBox21.Location = new System.Drawing.Point(633, 470);
this.pictureBox21.Name = "pictureBox21";
this.pictureBox21.Size = new System.Drawing.Size(265, 1);
this.pictureBox21.TabIndex = 71;
this.pictureBox21.TabStop = false;
//
// pictureBox10
//
this.pictureBox10.BackColor = System.Drawing.Color.Black;
this.pictureBox10.Location = new System.Drawing.Point(633, 450);
this.pictureBox10.Name = "pictureBox10";
this.pictureBox10.Size = new System.Drawing.Size(265, 1);
this.pictureBox10.TabIndex = 70;
this.pictureBox10.TabStop = false;
//
// pictureBox12
//
this.pictureBox12.BackColor = System.Drawing.Color.Black;
this.pictureBox12.Location = new System.Drawing.Point(633, 401);
this.pictureBox12.Name = "pictureBox12";
this.pictureBox12.Size = new System.Drawing.Size(265, 1);
this.pictureBox12.TabIndex = 68;
this.pictureBox12.TabStop = false;
//
// pictureBox13
//
this.pictureBox13.BackColor = System.Drawing.Color.Black;
this.pictureBox13.Location = new System.Drawing.Point(633, 381);
this.pictureBox13.Name = "pictureBox13";
this.pictureBox13.Size = new System.Drawing.Size(265, 1);
this.pictureBox13.TabIndex = 67;
this.pictureBox13.TabStop = false;
//
// pictureBox14
//
this.pictureBox14.BackColor = System.Drawing.Color.Black;
this.pictureBox14.Location = new System.Drawing.Point(633, 361);
this.pictureBox14.Name = "pictureBox14";
this.pictureBox14.Size = new System.Drawing.Size(265, 1);
this.pictureBox14.TabIndex = 66;
this.pictureBox14.TabStop = false;
//
// pictureBox15
//
this.pictureBox15.BackColor = System.Drawing.Color.Black;
this.pictureBox15.Location = new System.Drawing.Point(633, 341);
this.pictureBox15.Name = "pictureBox15";
this.pictureBox15.Size = new System.Drawing.Size(265, 1);
this.pictureBox15.TabIndex = 65;
this.pictureBox15.TabStop = false;
//
// pictureBox7
//
this.pictureBox7.BackColor = System.Drawing.Color.Black;
this.pictureBox7.Location = new System.Drawing.Point(633, 321);
this.pictureBox7.Name = "pictureBox7";
this.pictureBox7.Size = new System.Drawing.Size(265, 1);
this.pictureBox7.TabIndex = 64;
this.pictureBox7.TabStop = false;
//
// pictureBox8
//
```

```
this.pictureBox8.BackColor = System.Drawing.Color.Black;
this.pictureBox8.Location = new System.Drawing.Point(633, 301);
this.pictureBox8.Name = "pictureBox8";
this.pictureBox8.Size = new System.Drawing.Size(265, 1);
this.pictureBox8.TabIndex = 63;
this.pictureBox8.TabStop = false;
//
// pictureBox9
//
this.pictureBox9.BackColor = System.Drawing.Color.Black;
this.pictureBox9.Location = new System.Drawing.Point(633, 281);
this.pictureBox9.Name = "pictureBox9";
this.pictureBox9.Size = new System.Drawing.Size(265, 1);
this.pictureBox9.TabIndex = 62;
this.pictureBox9.TabStop = false;
//
// pictureBox6
//
this.pictureBox6.BackColor = System.Drawing.Color.Black;
this.pictureBox6.Location = new System.Drawing.Point(633, 261);
this.pictureBox6.Name = "pictureBox6";
this.pictureBox6.Size = new System.Drawing.Size(265, 1);
this.pictureBox6.TabIndex = 61;
this.pictureBox6.TabStop = false;
//
// pictureBox5
//
this.pictureBox5.BackColor = System.Drawing.Color.Black;
this.pictureBox5.Location = new System.Drawing.Point(633, 241);
this.pictureBox5.Name = "pictureBox5";
this.pictureBox5.Size = new System.Drawing.Size(265, 1);
this.pictureBox5.TabIndex = 60;
this.pictureBox5.TabStop = false;
//
// pictureBox4
//
this.pictureBox4.BackColor = System.Drawing.Color.Black;
this.pictureBox4.Location = new System.Drawing.Point(633, 221);
this.pictureBox4.Name = "pictureBox4";
this.pictureBox4.Size = new System.Drawing.Size(265, 1);
this.pictureBox4.TabIndex = 59;
this.pictureBox4.TabStop = false;
//
// pictureBox3
//
this.pictureBox3.BackColor = System.Drawing.Color.Black;
this.pictureBox3.Location = new System.Drawing.Point(633, 201);
this.pictureBox3.Name = "pictureBox3";
this.pictureBox3.Size = new System.Drawing.Size(265, 1);
this.pictureBox3.TabIndex = 58;
this.pictureBox3.TabStop = false;
//
// pictureBox2
//
this.pictureBox2.BackColor = System.Drawing.Color.Black;
this.pictureBox2.Location = new System.Drawing.Point(633, 177);
this.pictureBox2.Name = "pictureBox2";
this.pictureBox2.Size = new System.Drawing.Size(265, 5);
this.pictureBox2.TabIndex = 57;
this.pictureBox2.TabStop = false;
//
// pictureBox1
//
this.pictureBox1.BackColor = System.Drawing.Color.Black;
this.pictureBox1.Location = new System.Drawing.Point(633, 148);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(265, 10);
this.pictureBox1.TabIndex = 56;
this.pictureBox1.TabStop = false;
//
// lbl_Manganese
//
this.lbl_Manganese.AutoSize = true;
this.lbl_Manganese.Location = new System.Drawing.Point(630, 614);
this.lbl_Manganese.Name = "lbl_Manganese";
this.lbl_Manganese.Size = new System.Drawing.Size(63, 13);
this.lbl_Manganese.TabIndex = 55;
this.lbl_Manganese.Text = "Manganese";
//
// lbl_Copper
//
this.lbl_Copper.AutoSize = true;
this.lbl_Copper.Location = new System.Drawing.Point(776, 594);
this.lbl_Copper.Name = "lbl_Copper";
this.lbl_Copper.Size = new System.Drawing.Size(41, 13);
this.lbl_Copper.TabIndex = 54;
this.lbl_Copper.Text = "Copper";
//
```

```
// lbl_Selenium
//
this.lbl_Selenium.AutoSize = true;
this.lbl_Selenium.Location = new System.Drawing.Point(630, 594);
this.lbl_Selenium.Name = "lbl_Selenium";
this.lbl_Selenium.Size = new System.Drawing.Size(50, 13);
this.lbl_Selenium.TabIndex = 53;
this.lbl_Selenium.Text = "Selenium";
//
// lbl_Zinc
//
this.lbl_Zinc.AutoSize = true;
this.lbl_Zinc.Location = new System.Drawing.Point(776, 574);
this.lbl_Zinc.Name = "lbl_Zinc";
this.lbl_Zinc.Size = new System.Drawing.Size(28, 13);
this.lbl_Zinc.TabIndex = 52;
this.lbl_Zinc.Text = "Zinc";
//
// lbl_Magnesium
//
this.lbl_Magnesium.AutoSize = true;
this.lbl_Magnesium.Location = new System.Drawing.Point(630, 574);
this.lbl_Magnesium.Name = "lbl_Magnesium";
this.lbl_Magnesium.Size = new System.Drawing.Size(61, 13);
this.lbl_Magnesium.TabIndex = 51;
this.lbl_Magnesium.Text = "Magnesium";
//
// lbl_Phosphorus
//
this.lbl_Phosphorus.AutoSize = true;
this.lbl_Phosphorus.Location = new System.Drawing.Point(776, 554);
this.lbl_Phosphorus.Name = "lbl_Phosphorus";
this.lbl_Phosphorus.Size = new System.Drawing.Size(63, 13);
this.lbl_Phosphorus.TabIndex = 50;
this.lbl_Phosphorus.Text = "Phosphorus";
//
// lbl_VitaminB12
//
this.lbl_VitaminB12.AutoSize = true;
this.lbl_VitaminB12.Location = new System.Drawing.Point(630, 554);
this.lbl_VitaminB12.Name = "lbl_VitaminB12";
this.lbl_VitaminB12.Size = new System.Drawing.Size(63, 13);
this.lbl_VitaminB12.TabIndex = 49;
this.lbl_VitaminB12.Text = "Vitamin B12";
//
// lbl_Folate
//
this.lbl_Folate.AutoSize = true;
this.lbl_Folate.Location = new System.Drawing.Point(776, 534);
this.lbl_Folate.Name = "lbl_Folate";
this.lbl_Folate.Size = new System.Drawing.Size(36, 13);
this.lbl_Folate.TabIndex = 48;
this.lbl_Folate.Text = "Folate";
//
// lbl_VitaminB6
//
this.lbl_VitaminB6.AutoSize = true;
this.lbl_VitaminB6.Location = new System.Drawing.Point(630, 534);
this.lbl_VitaminB6.Name = "lbl_VitaminB6";
this.lbl_VitaminB6.Size = new System.Drawing.Size(57, 13);
this.lbl_VitaminB6.TabIndex = 47;
this.lbl_VitaminB6.Text = "Vitamin B6";
//
// lbl_Niacin
//
this.lbl_Niacin.AutoSize = true;
this.lbl_Niacin.Location = new System.Drawing.Point(776, 514);
this.lbl_Niacin.Name = "lbl_Niacin";
this.lbl_Niacin.Size = new System.Drawing.Size(37, 13);
this.lbl_Niacin.TabIndex = 46;
this.lbl_Niacin.Text = "Niacin";
//
// lbl_Riboflavin
//
this.lbl_Riboflavin.AutoSize = true;
this.lbl_Riboflavin.Location = new System.Drawing.Point(630, 514);
this.lbl_Riboflavin.Name = "lbl_Riboflavin";
this.lbl_Riboflavin.Size = new System.Drawing.Size(54, 13);
this.lbl_Riboflavin.TabIndex = 45;
this.lbl_Riboflavin.Text = "Riboflavin";
//
// lbl_Thiamin
//
this.lbl_Thiamin.AutoSize = true;
this.lbl_Thiamin.Location = new System.Drawing.Point(776, 494);
this.lbl_Thiamin.Name = "lbl_Thiamin";
this.lbl_Thiamin.Size = new System.Drawing.Size(44, 13);
this.lbl_Thiamin.TabIndex = 44;
```

```
this.lbl_Thiamin.Text = "Thiamin";
//
// lbl_VitaminK
//
this.lbl_VitaminK.AutoSize = true;
this.lbl_VitaminK.Location = new System.Drawing.Point(630, 494);
this.lbl_VitaminK.Name = "lbl_VitaminK";
this.lbl_VitaminK.Size = new System.Drawing.Size(51, 13);
this.lbl_VitaminK.TabIndex = 43;
this.lbl_VitaminK.Text = "Vitamin K";
//
// lbl_VitaminE
//
this.lbl_VitaminE.AutoSize = true;
this.lbl_VitaminE.Location = new System.Drawing.Point(776, 474);
this.lbl_VitaminE.Name = "lbl_VitaminE";
this.lbl_VitaminE.Size = new System.Drawing.Size(51, 13);
this.lbl_VitaminE.TabIndex = 42;
this.lbl_VitaminE.Text = "Vitamin E";
//
// lbl_VitaminD
//
this.lbl_VitaminD.AutoSize = true;
this.lbl_VitaminD.Location = new System.Drawing.Point(630, 474);
this.lbl_VitaminD.Name = "lbl_VitaminD";
this.lbl_VitaminD.Size = new System.Drawing.Size(52, 13);
this.lbl_VitaminD.TabIndex = 41;
this.lbl_VitaminD.Text = "Vitamin D";
//
// lbl_Iron
//
this.lbl_Iron.AutoSize = true;
this.lbl_Iron.Location = new System.Drawing.Point(776, 454);
this.lbl_Iron.Name = "lbl_Iron";
this.lbl_Iron.Size = new System.Drawing.Size(25, 13);
this.lbl_Iron.TabIndex = 40;
this.lbl_Iron.Text = "Iron";
//
// lbl_Calcium
//
this.lbl_Calcium.AutoSize = true;
this.lbl_Calcium.Location = new System.Drawing.Point(630, 454);
this.lbl_Calcium.Name = "lbl_Calcium";
this.lbl_Calcium.Size = new System.Drawing.Size(44, 13);
this.lbl_Calcium.TabIndex = 39;
this.lbl_Calcium.Text = "Calcium";
//
// lbl_Dot1
//
this.lbl_Dot1.AutoSize = true;
this.lbl_Dot1.Location = new System.Drawing.Point(757, 434);
this.lbl_Dot1.Name = "lbl_Dot1";
this.lbl_Dot1.Size = new System.Drawing.Size(13, 13);
this.lbl_Dot1.TabIndex = 38;
this.lbl_Dot1.Text = "•";
//
// lbl_VitaminC
//
this.lbl_VitaminC.AutoSize = true;
this.lbl_VitaminC.Location = new System.Drawing.Point(776, 434);
this.lbl_VitaminC.Name = "lbl_VitaminC";
this.lbl_VitaminC.Size = new System.Drawing.Size(51, 13);
this.lbl_VitaminC.TabIndex = 37;
this.lbl_VitaminC.Text = "Vitamin C";
//
// lbl_VitaminA
//
this.lbl_VitaminA.AutoSize = true;
this.lbl_VitaminA.Location = new System.Drawing.Point(630, 434);
this.lbl_VitaminA.Name = "lbl_VitaminA";
this.lbl_VitaminA.Size = new System.Drawing.Size(51, 13);
this.lbl_VitaminA.TabIndex = 36;
this.lbl_VitaminA.Text = "Vitamin A";
//
// lbl_Protein
//
this.lbl_Protein.AutoSize = true;
this.lbl_Protein.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_Protein.Location = new System.Drawing.Point(630, 405);
this.lbl_Protein.Name = "lbl_Protein";
this.lbl_Protein.Size = new System.Drawing.Size(47, 13);
this.lbl_Protein.TabIndex = 35;
this.lbl_Protein.Text = "Protein";
//
// lbl_Sugars
//
this.lbl_Sugars.AutoSize = true;
this.lbl_Sugars.Location = new System.Drawing.Point(645, 385);
```

```
this.lbl_Sugars.Name = "lbl_Sugars";
this.lbl_Sugars.Size = new System.Drawing.Size(40, 13);
this.lbl_Sugars.TabIndex = 34;
this.lbl_Sugars.Text = "Sugars";
//
// lbl_DietaryFiber
//
this.lbl_DietaryFiber.AutoSize = true;
this.lbl_DietaryFiber.Location = new System.Drawing.Point(645, 365);
this.lbl_DietaryFiber.Name = "lbl_DietaryFiber";
this.lbl_DietaryFiber.Size = new System.Drawing.Size(66, 13);
this.lbl_DietaryFiber.TabIndex = 33;
this.lbl_DietaryFiber.Text = "Dietary Fiber";
//
// lbl_TotalCarbohydrate
//
this.lbl_TotalCarbohydrate.AutoSize = true;
this.lbl_TotalCarbohydrate.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_TotalCarbohydrate.Location = new System.Drawing.Point(630, 345);
this.lbl_TotalCarbohydrate.Name = "lbl_TotalCarbohydrate";
this.lbl_TotalCarbohydrate.Size = new System.Drawing.Size(115, 13);
this.lbl_TotalCarbohydrate.TabIndex = 32;
this.lbl_TotalCarbohydrate.Text = "Total Carbohydrate";
//
// lbl_Potassium
//
this.lbl_Potassium.AutoSize = true;
this.lbl_Potassium.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_Potassium.Location = new System.Drawing.Point(630, 325);
this.lbl_Potassium.Name = "lbl_Potassium";
this.lbl_Potassium.Size = new System.Drawing.Size(64, 13);
this.lbl_Potassium.TabIndex = 31;
this.lbl_Potassium.Text = "Potassium";
//
// lbl_Sodium
//
this.lbl_Sodium.AutoSize = true;
this.lbl_Sodium.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_Sodium.Location = new System.Drawing.Point(630, 305);
this.lbl_Sodium.Name = "lbl_Sodium";
this.lbl_Sodium.Size = new System.Drawing.Size(48, 13);
this.lbl_Sodium.TabIndex = 30;
this.lbl_Sodium.Text = "Sodium";
//
// lbl_Cholesterol
//
this.lbl_Cholesterol.AutoSize = true;
this.lbl_Cholesterol.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_Cholesterol.Location = new System.Drawing.Point(630, 285);
this.lbl_Cholesterol.Name = "lbl_Cholesterol";
this.lbl_Cholesterol.Size = new System.Drawing.Size(70, 13);
this.lbl_Cholesterol.TabIndex = 29;
this.lbl_Cholesterol.Text = "Cholesterol";
//
// lbl_TransFat
//
this.lbl_TransFat.AutoSize = true;
this.lbl_TransFat.Location = new System.Drawing.Point(645, 265);
this.lbl_TransFat.Name = "lbl_TransFat";
this.lbl_TransFat.Size = new System.Drawing.Size(52, 13);
this.lbl_TransFat.TabIndex = 28;
this.lbl_TransFat.Text = "Trans Fat";
//
// lbl_PolyunsaturatedFat
//
this.lbl_PolyunsaturatedFat.AutoSize = true;
this.lbl_PolyunsaturatedFat.Location = new System.Drawing.Point(645, 245);
this.lbl_PolyunsaturatedFat.Name = "lbl_PolyunsaturatedFat";
this.lbl_PolyunsaturatedFat.Size = new System.Drawing.Size(101, 13);
this.lbl_PolyunsaturatedFat.TabIndex = 27;
this.lbl_PolyunsaturatedFat.Text = "Polyunsaturated Fat";
//
// lbl_MonounsaturatedFat
//
this.lbl_MonounsaturatedFat.AutoSize = true;
this.lbl_MonounsaturatedFat.Location = new System.Drawing.Point(645, 225);
this.lbl_MonounsaturatedFat.Name = "lbl_MonounsaturatedFat";
this.lbl_MonounsaturatedFat.Size = new System.Drawing.Size(108, 13);
this.lbl_MonounsaturatedFat.TabIndex = 26;
this.lbl_MonounsaturatedFat.Text = "Monounsaturated Fat";
//
// lbl_SaturatedFat
//
this.lbl_SaturatedFat.AutoSize = true;
this.lbl_SaturatedFat.Location = new System.Drawing.Point(645, 205);
this.lbl_SaturatedFat.Name = "lbl_SaturatedFat";
this.lbl_SaturatedFat.Size = new System.Drawing.Size(71, 13);
this.lbl_SaturatedFat.TabIndex = 25;
```

```
this.lbl_SaturatedFat.Text = "Saturated Fat";
//
// lbl_Calories
//
this.lbl_Calories.AutoSize = true;
this.lbl_Calories.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_Calories.Location = new System.Drawing.Point(630, 161);
this.lbl_Calories.Name = "lbl_Calories";
this.lbl_Calories.Size = new System.Drawing.Size(52, 13);
this.lbl_Calories.TabIndex = 24;
this.lbl_Calories.Text = "Calories";
//
// lbl_ServingSize
//
this.lbl_ServingSize.AutoSize = true;
this.lbl_ServingSize.Location = new System.Drawing.Point(630, 132);
this.lbl_ServingSize.Name = "lbl_ServingSize";
this.lbl_ServingSize.Size = new System.Drawing.Size(66, 13);
this.lbl_ServingSize.TabIndex = 22;
this.lbl_ServingSize.Text = "Serving Size";
//
// lbl_NutritionFacts
//
this.lbl_NutritionFacts.AutoSize = true;
this.lbl_NutritionFacts.Font = new System.Drawing.Font("Microsoft Sans Serif", 14.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_NutritionFacts.Location = new System.Drawing.Point(629, 108);
this.lbl_NutritionFacts.Name = "lbl_NutritionFacts";
this.lbl_NutritionFacts.Size = new System.Drawing.Size(129, 24);
this.lbl_NutritionFacts.TabIndex = 21;
this.lbl_NutritionFacts.Text = "Nutrition Facts";
//
// lbl_TotalFat
//
this.lbl_TotalFat.AutoSize = true;
this.lbl_TotalFat.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_TotalFat.Location = new System.Drawing.Point(630, 185);
this.lbl_TotalFat.Name = "lbl_TotalFat";
this.lbl_TotalFat.Size = new System.Drawing.Size(58, 13);
this.lbl_TotalFat.TabIndex = 20;
this.lbl_TotalFat.Text = "Total Fat";
//
// btn_FreezeUnfreezeDisplay
//
this.btn_FreezeUnfreezeDisplay.Location = new System.Drawing.Point(627, 10);
this.btn_FreezeUnfreezeDisplay.Name = "btn_FreezeUnfreezeDisplay";
this.btn_FreezeUnfreezeDisplay.Size = new System.Drawing.Size(282, 41);
this.btn_FreezeUnfreezeDisplay.TabIndex = 19;
this.btn_FreezeUnfreezeDisplay.Text = "Freeze Nutrition Data";
this.btn_FreezeUnfreezeDisplay.UseVisualStyleBackColor = true;
this.btn_FreezeUnfreezeDisplay.Click += new System.EventHandler(this.btn_FreezeUnfreezeDisplay_Click);
//
// btn_DisplayAllItems
//
this.btn_DisplayAllItems.Location = new System.Drawing.Point(353, 47);
this.btn_DisplayAllItems.Name = "btn_DisplayAllItems";
this.btn_DisplayAllItems.Size = new System.Drawing.Size(250, 26);
this.btn_DisplayAllItems.TabIndex = 18;
this.btn_DisplayAllItems.Text = "Display All Items";
this.btn_DisplayAllItems.UseVisualStyleBackColor = true;
this.btn_DisplayAllItems.Click += new System.EventHandler(this.btn_DisplayAllItems_Click);
//
// btn_Search
//
this.btn_Search.Location = new System.Drawing.Point(528, 10);
this.btn_Search.Name = "btn_Search";
this.btn_Search.Size = new System.Drawing.Size(75, 26);
this.btn_Search.TabIndex = 16;
this.btn_Search.Text = "Search";
this.btn_Search.UseVisualStyleBackColor = true;
this.btn_Search.Click += new System.EventHandler(this.bt_Search_Click);
//
// txt_Search
//
this.txt_Search.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.txt_Search.Location = new System.Drawing.Point(353, 10);
this.txt_Search.Name = "txt_Search";
this.txt_Search.Size = new System.Drawing.Size(169, 26);
this.txt_Search.TabIndex = 15;
//
// lsb_FoodList
//
this.lsb_FoodList.FormattingEnabled = true;
this.lsb_FoodList.Location = new System.Drawing.Point(353, 74);
this.lsb_FoodList.Name = "lsb_FoodList";
this.lsb_FoodList.Size = new System.Drawing.Size(250, 628);
this.lsb_FoodList.TabIndex = 14;
this.lsb_FoodList.SelectedIndexChanged += new System.EventHandler(this.lsb_FoodList_SelectedIndexChanged);
//
```

```
// lbl_Caption
//
this.lbl_Caption.Font = new System.Drawing.Font("Palatino Linotype", 11.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_Caption.Location = new System.Drawing.Point(10, 3);
this.lbl_Caption.Name = "lbl_Caption";
this.lbl_Caption.Size = new System.Drawing.Size(318, 70);
this.lbl_Caption.TabIndex = 12;
this.lbl_Caption.Text = "DANI Demo Program v.0.5 - Powered by USDA National Nutrient Database for Standard" +
" Reference SR23";
//
// tab_2
//
this.tab_2.Controls.Add(this.btn_Cancel);
this.tab_2.Controls.Add(this.btn_OK);
this.tab_2.Controls.Add(this.cmb_Handshake);
this.tab_2.Controls.Add(this.cmb_Parity);
this.tab_2.Controls.Add(this.txt_DataBits);
this.tab_2.Controls.Add(this.cmb_StopBits);
this.tab_2.Controls.Add(this.cmb_BaudRate);
this.tab_2.Controls.Add(this.lbl_PortName);
this.tab_2.Controls.Add(this.lbl_HandShake);
this.tab_2.Controls.Add(this.lbl_Parity);
this.tab_2.Controls.Add(this.lbl_StopBits);
this.tab_2.Controls.Add(this.lbl_DataBits);
this.tab_2.Controls.Add(this.lbl_BaudRate);
this.tab_2.Controls.Add(this.cmb_PortName);
this.tab_2.Location = new System.Drawing.Point(4, 22);
this.tab_2.Name = "tab_2";
this.tab_2.Padding = new System.Windows.Forms.Padding(3);
this.tab_2.Size = new System.Drawing.Size(1232, 712);
this.tab_2.TabIndex = 1;
this.tab_2.Text = "Configure";
this.tab_2.UseVisualStyleBackColor = true;
//
// btn_Cancel
//
this.btn_Cancel.Location = new System.Drawing.Point(139, 179);
this.btn_Cancel.Name = "btn_Cancel";
this.btn_Cancel.Size = new System.Drawing.Size(119, 28);
this.btn_Cancel.TabIndex = 27;
this.btn_Cancel.Text = "Cancel";
this.btn_Cancel.UseVisualStyleBackColor = true;
this.btn_Cancel.Click += new System.EventHandler(this.btn_Cancel_Click);
//
// btn_OK
//
this.btn_OK.Location = new System.Drawing.Point(4, 179);
this.btn_OK.Name = "btn_OK";
this.btn_OK.Size = new System.Drawing.Size(119, 28);
this.btn_OK.TabIndex = 26;
this.btn_OK.Text = "Apply";
this.btn_OK.UseVisualStyleBackColor = true;
this.btn_OK.Click += new System.EventHandler(this.btn_OK_Click);
//
// cmb_Handshake
//
this.cmb_Handshake.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmb_Handshake.FormattingEnabled = true;
this.cmb_Handshake.Location = new System.Drawing.Point(108, 140);
this.cmb_Handshake.Name = "cmb_Handshake";
this.cmb_Handshake.Size = new System.Drawing.Size(150, 21);
this.cmb_Handshake.TabIndex = 25;
this.cmb_Handshake.SelectedIndexChanged += new System.EventHandler(this.cmb_Handshake_SelectedIndexChanged);
//
// cmb_Parity
//
this.cmb_Parity.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmb_Parity.FormattingEnabled = true;
this.cmb_Parity.Location = new System.Drawing.Point(108, 113);
this.cmb_Parity.Name = "cmb_Parity";
this.cmb_Parity.Size = new System.Drawing.Size(150, 21);
this.cmb_Parity.TabIndex = 24;
this.cmb_Parity.SelectedIndexChanged += new System.EventHandler(this.cmb_Parity_SelectedIndexChanged);
//
// txt_DataBits
//
this.txt_DataBits.Location = new System.Drawing.Point(108, 60);
this.txt_DataBits.Name = "txt_DataBits";
this.txt_DataBits.Size = new System.Drawing.Size(150, 20);
this.txt_DataBits.TabIndex = 23;
this.txt_DataBits.TextChanged += new System.EventHandler(this.txt_DataBits_TextChanged);
//
// cmb_StopBits
//
this.cmb_StopBits.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmb_StopBits.FormattingEnabled = true;
this.cmb_StopBits.Location = new System.Drawing.Point(108, 86);
this.cmb_StopBits.Name = "cmb_StopBits";
```

```
this.cmb_StopBits.Size = new System.Drawing.Size(150, 21);
this.cmb_StopBits.TabIndex = 22;
this.cmb_StopBits.SelectedIndexChanged += new System.EventHandler(this.cmb_StopBits_SelectedIndexChanged);
//
// cmb_BaudRate
//
this.cmb_BaudRate.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmb_BaudRate.FormattingEnabled = true;
this.cmb_BaudRate.Location = new System.Drawing.Point(108, 33);
this.cmb_BaudRate.Name = "cmb_BaudRate";
this.cmb_BaudRate.Size = new System.Drawing.Size(150, 21);
this.cmb_BaudRate.TabIndex = 21;
this.cmb_BaudRate.SelectedIndexChanged += new System.EventHandler(this.cmb_BaudRate_SelectedIndexChanged);
//
// lbl_PortName
//
this.lbl_PortName.AutoSize = true;
this.lbl_PortName.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_PortName.Location = new System.Drawing.Point(4, 9);
this.lbl_PortName.Name = "lbl_PortName";
this.lbl_PortName.Size = new System.Drawing.Size(66, 15);
this.lbl_PortName.TabIndex = 20;
this.lbl_PortName.Text = "Port Name";
//
// lbl_HandShake
//
this.lbl_HandShake.AutoSize = true;
this.lbl_HandShake.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_HandShake.Location = new System.Drawing.Point(4, 143);
this.lbl_HandShake.Name = "lbl_HandShake";
this.lbl_HandShake.Size = new System.Drawing.Size(70, 15);
this.lbl_HandShake.TabIndex = 19;
this.lbl_HandShake.Text = "Handshake";
//
// lbl_Parity
//
this.lbl_Parity.AutoSize = true;
this.lbl_Parity.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_Parity.Location = new System.Drawing.Point(4, 116);
this.lbl_Parity.Name = "lbl_Parity";
this.lbl_Parity.Size = new System.Drawing.Size(37, 15);
this.lbl_Parity.TabIndex = 18;
this.lbl_Parity.Text = "Parity";
//
// lbl_StopBits
//
this.lbl_StopBits.AutoSize = true;
this.lbl_StopBits.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_StopBits.Location = new System.Drawing.Point(4, 89);
this.lbl_StopBits.Name = "lbl_StopBits";
this.lbl_StopBits.Size = new System.Drawing.Size(55, 15);
this.lbl_StopBits.TabIndex = 17;
this.lbl_StopBits.Text = "Stop Bits";
//
// lbl_DataBits
//
this.lbl_DataBits.AutoSize = true;
this.lbl_DataBits.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_DataBits.Location = new System.Drawing.Point(4, 63);
this.lbl_DataBits.Name = "lbl_DataBits";
this.lbl_DataBits.Size = new System.Drawing.Size(59, 15);
this.lbl_DataBits.TabIndex = 16;
this.lbl_DataBits.Text = "Data Bits ";
//
// lbl_BaudRate
//
this.lbl_BaudRate.AutoSize = true;
this.lbl_BaudRate.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
this.lbl_BaudRate.Location = new System.Drawing.Point(4, 36);
this.lbl_BaudRate.Name = "lbl_BaudRate";
this.lbl_BaudRate.Size = new System.Drawing.Size(65, 15);
this.lbl_BaudRate.TabIndex = 15;
this.lbl_BaudRate.Text = "Baud Rate";
//
// cmb_PortName
//
this.cmb_PortName.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
this.cmb_PortName.FormattingEnabled = true;
this.cmb_PortName.Location = new System.Drawing.Point(108, 6);
this.cmb_PortName.Name = "cmb_PortName";
this.cmb_PortName.Size = new System.Drawing.Size(150, 21);
this.cmb_PortName.TabIndex = 14;
this.cmb_PortName.SelectedIndexChanged += new System.EventHandler(this.cmb_PortName_SelectedIndexChanged);
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
```

```
this.ClientSize = new System.Drawing.Size(1264, 762);
this.Controls.Add(this.tab_Main);
this.Name = "Form1";
this.Text = "DANI Demo";
this.FormClosing += new System.Windows.Forms.FormClosingEventHandler(this.Form1_FormClosing);
this.tab_Main.ResumeLayout(false);
this.tab_1.ResumeLayout(false);
this.tab_1.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox11)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox40)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox26)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox27)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox28)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox29)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox30)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox31)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox32)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox33)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox34)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox35)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox21)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox10)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox12)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox13)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox14)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox15)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox7)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox8)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox9)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox6)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox5)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox4)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
this.tab_2.ResumeLayout(false);
this.tab_2.PerformLayout();
this.ResumeLayout(false);

    }

    #endregion

private System.Windows.Forms.Button btn_Connect;
private System.Windows.Forms.Timer timer1;
private System.Windows.Forms.Button btn_RecalibrateScale;
private System.Windows.Forms.Label lbl_Status;
private System.Windows.Forms.Button btn_Disconnect;
private System.Windows.Forms.Button btn_Exit;
private System.Windows.Forms.TextBox txt_Display;
private System.Windows.Forms.TabControl tab_Main;
private System.Windows.Forms.TabPage tab_1;
private System.Windows.Forms.TabPage tab_2;
private System.Windows.Forms.Button btn_Cancel;
private System.Windows.Forms.Button btn_OK;
private System.Windows.Forms.ComboBox cmb_Handshake;
private System.Windows.Forms.ComboBox cmb_Parity;
private System.Windows.Forms.TextBox txt_DataBits;
private System.Windows.Forms.ComboBox cmb_StopBits;
private System.Windows.Forms.ComboBox cmb_BaudRate;
private System.Windows.Forms.Label lbl_PortName;
private System.Windows.Forms.Label lbl_HandShake;
private System.Windows.Forms.Label lbl_Parity;
private System.Windows.Forms.Label lbl_StopBits;
private System.Windows.Forms.Label lbl_DataBits;
private System.Windows.Forms.Label lbl_BaudRate;
private System.Windows.Forms.ComboBox cmb_PortName;
private System.Windows.Forms.Label lbl_Caption;
private System.Windows.Forms.ListBox lsb_FoodList;
private System.Windows.Forms.Button btn_Search;
private System.Windows.Forms.TextBox txt_Search;
private System.Windows.Forms.Button btn_DisplayAllItems;
private System.Windows.Forms.Button btn_FreezeUnfreezeDisplay;
private System.Windows.Forms.Label lbl_Selenium;
private System.Windows.Forms.Label lbl_Zinc;
private System.Windows.Forms.Label lbl_Magnesium;
private System.Windows.Forms.Label lbl_Phosphorus;
private System.Windows.Forms.Label lbl_VitaminB12;
private System.Windows.Forms.Label lbl_Folate;
private System.Windows.Forms.Label lbl_VitaminB6;
private System.Windows.Forms.Label lbl_Niacin;
private System.Windows.Forms.Label lbl_Riboflavin;
private System.Windows.Forms.Label lbl_Thiamin;
private System.Windows.Forms.Label lbl_VitaminK;
private System.Windows.Forms.Label lbl_VitaminE;
private System.Windows.Forms.Label lbl_VitaminD;
private System.Windows.Forms.Label lbl_Iron;
private System.Windows.Forms.Label lbl_Calcium;
```

```
private System.Windows.Forms.Label lbl_Dot1;
private System.Windows.Forms.Label lbl_VitaminC;
private System.Windows.Forms.Label lbl_VitaminA;
private System.Windows.Forms.Label lbl_Protein;
private System.Windows.Forms.Label lbl_Sugars;
private System.Windows.Forms.Label lbl_DietaryFiber;
private System.Windows.Forms.Label lbl_TotalCarbohydrate;
private System.Windows.Forms.Label lbl_Potassium;
private System.Windows.Forms.Label lbl_Sodium;
private System.Windows.Forms.Label lbl_Cholesterol;
private System.Windows.Forms.Label lbl_TransFat;
private System.Windows.Forms.Label lbl_PolyunsaturatedFat;
private System.Windows.Forms.Label lbl_MonounsaturatedFat;
private System.Windows.Forms.Label lbl_SaturatedFat;
private System.Windows.Forms.Label lbl_Calories;
private System.Windows.Forms.Label lbl_ServingSize;
private System.Windows.Forms.Label lbl_NutritionFacts;
private System.Windows.Forms.Label lbl_TotalFat;
private System.Windows.Forms.PictureBox pictureBox26;
private System.Windows.Forms.PictureBox pictureBox27;
private System.Windows.Forms.PictureBox pictureBox28;
private System.Windows.Forms.PictureBox pictureBox29;
private System.Windows.Forms.PictureBox pictureBox30;
private System.Windows.Forms.PictureBox pictureBox31;
private System.Windows.Forms.PictureBox pictureBox32;
private System.Windows.Forms.PictureBox pictureBox33;
private System.Windows.Forms.PictureBox pictureBox34;
private System.Windows.Forms.PictureBox pictureBox35;
private System.Windows.Forms.PictureBox pictureBox21;
private System.Windows.Forms.PictureBox pictureBox10;
private System.Windows.Forms.PictureBox pictureBox12;
private System.Windows.Forms.PictureBox pictureBox13;
private System.Windows.Forms.PictureBox pictureBox14;
private System.Windows.Forms.PictureBox pictureBox15;
private System.Windows.Forms.PictureBox pictureBox7;
private System.Windows.Forms.PictureBox pictureBox8;
private System.Windows.Forms.PictureBox pictureBox9;
private System.Windows.Forms.PictureBox pictureBox6;
private System.Windows.Forms.PictureBox pictureBox5;
private System.Windows.Forms.PictureBox pictureBox4;
private System.Windows.Forms.PictureBox pictureBox3;
private System.Windows.Forms.PictureBox pictureBox2;
private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.Label lbl_Manganese;
private System.Windows.Forms.Label lbl_Copper;
private System.Windows.Forms.PictureBox pictureBox40;
private System.Windows.Forms.Label lbl_Dot8;
private System.Windows.Forms.Label lbl_Dot7;
private System.Windows.Forms.Label lbl_Dot6;
private System.Windows.Forms.Label lbl_Dot5;
private System.Windows.Forms.Label lbl_Dot2;
private System.Windows.Forms.Label lbl_Dot4;
private System.Windows.Forms.Label lbl_Dot3;
private System.Windows.Forms.PictureBox pictureBox11;
private System.Windows.Forms.Label lbl_Dot9;
private System.Windows.Forms.Label lbl_Cholesterol_pct;
private System.Windows.Forms.Label lbl_SaturatedFat_pct;
private System.Windows.Forms.Label lbl_TotalFat_pct;
private System.Windows.Forms.Label lbl_DietaryFiber_pct;
private System.Windows.Forms.Label lbl_Potassium_pct;
private System.Windows.Forms.Label lbl_Sodium_pct;
private System.Windows.Forms.Label lbl_VitaminC_pct;
private System.Windows.Forms.Label lbl_VitaminD_pct;
private System.Windows.Forms.Label lbl_Protein_pct;
private System.Windows.Forms.Label lbl_TotalCarbohydrate_pct;
private System.Windows.Forms.Label lbl_VitaminE_pct;
private System.Windows.Forms.Label lbl_Iron_pct;
private System.Windows.Forms.Label lbl_Calcium_pct;
private System.Windows.Forms.Label lbl_VitaminA_pct;
private System.Windows.Forms.Label lbl_Magnesium_pct;
private System.Windows.Forms.Label lbl_VitaminB6_pct;
private System.Windows.Forms.Label lbl_Riboflavin_pct;
private System.Windows.Forms.Label lbl_VitaminK_pct;
private System.Windows.Forms.Label lbl_Folate_pct;
private System.Windows.Forms.Label lbl_Manganese_pct;
private System.Windows.Forms.Label lbl_Selenium_pct;
private System.Windows.Forms.Label lbl_VitaminB12_pct;
private System.Windows.Forms.Label lbl_Zinc_pct;
private System.Windows.Forms.Label lbl_Phosphorus_pct;
private System.Windows.Forms.Label lbl_Niacin_pct;
private System.Windows.Forms.Label lbl_Thiamin_pct;
private System.Windows.Forms.Label lbl_Copper_pct;
private System.Windows.Forms.Label lbl_Protein_g;
private System.Windows.Forms.Label lbl_Sugars_g;
private System.Windows.Forms.Label lbl_Sodium_mg;
private System.Windows.Forms.Label lbl_TransFat_g;
private System.Windows.Forms.Label lbl_PolyunsaturatedFat_g;
private System.Windows.Forms.Label lbl_Calories_kcal;
```

```
private System.Windows.Forms.Label lbl_DietaryFiber_g;
private System.Windows.Forms.Label lbl_TotalCarbohydrate_g;
private System.Windows.Forms.Label lbl_SaturatedFat_g;
private System.Windows.Forms.Label lbl_Potassium_mg;
private System.Windows.Forms.Label lbl_Cholesterol_mg;
private System.Windows.Forms.Label lbl_MonounsaturatedFat_g;
private System.Windows.Forms.Label lbl_TotalFat_g;
private System.Windows.Forms.Label lbl_ServingSize_g;
private System.Windows.Forms.TextBox txt_Name;
private System.Windows.Forms.ListBox lsb_Log;
private System.Windows.Forms.Button bt_ClearLog;
private System.Windows.Forms.Button bt_AddToLog;
    }
}
```