AC 2007-1319: DESIGNING A PORTABLE SURFACE ROUGHNESS INSPECTION PROBE

Saeid Moslehpour, University of Hartford

SAEID MOSLEHPOUR is an Assistant Professor in the Electrical and Computer Engineering Department in the College of Engineering, Technology, and Architecture at the University of Hartford. He holds PhD from Iowa State University and BS MS and EdSp degrees from Central Missouri State University. His areas of interest are logic design, CPLDs, FPGAs and distance learning.

Greg Cloutier, University of Hartford

Greg Cloutier is currently employed as a Hardware Development and Research Engineer for JDS Uniphase, an optical solutions company. He expects to graduate in the Spring of 2007 from the University of Hartford with a Bachelors degree in Computer Engineering Technology. He enjoys experimenting with electronics and microcontrollers.

Matthew Brown, University of Hartford

Matt Brown is a senior at the University of Hartford. He is expected to Graduate in the Summer of 2007 with a Bachelor of Science in Computer Engineering Technology. He is currently working on his senior project involving designing a control system for a robotic arm.

Determining Surface Roughness by Laser Light Backscatter

Abstract

The University of Hartford has been involved with the measurement of surface roughness using non-contact methods. The type of surface finish under study is created by a grinding operation in industrial applications. This surface must meet a specified pre-determined finish. Surface roughness is a measure of localized small-scale height variations or imperfections on the surface of a material. The measurement parameter of surface roughness is described by the term Ra, which is the average roughness expressed in micro-inches. One method used for non-contact measurement of a ground surface is to measure the back-scatter of a laser light source. The amount of back-scatter is directly proportional to the Ra value.

One drawback to the current design is that it is not portable. A computer handles data acquisition and provides a user interface. The inspection probe is attached to the computer via a limited length of wire. In an industrial setting, it would be more efficient and flexible to bring the inspection probe to the surface in question rather than to bring the surface to the inspection probe. This design will eliminate the need for the wire tether and the computer altogether. The functionality for both data acquisition and user interface will be replaced by an embedded system. This embedded system will consist of a microcontroller, a multi-channel analog to digital converter, an LCD display, and a keypad input device.

Introduction

As essential part of quality control in many industrial manufacturing operations is the measurement of the surface roughness in a metal. Surface roughness is a measure of localized small-scale height variations or imperfections on the surface of a material. These variations are caused during the manufacturing process. These variations consist of indentations, scratches, gouges, and any imperfections in the metal.

In recent years the need for more precisely cut machines has increased. The measurement parameter of surface roughness is described as Ra, which is the average roughness expressed in micro-inches. The standard method is to physically measure the surface. The device for measuring roughness is called a contact profilometer. This is considered to be the baseline method. A profilometer consists of a flat inspection table and a small stylus attached to an adjustable holder. The stylus is then dragged across the grain of the work piece. As the stylus moves across the work piece, it moves up and down generating an analog signal. By evaluating this analog signal and by comparing it to signals created by set standards, an average roughness can be determined. The problem with this method is that the stylus may scratch the surface when the surface is being measured. Also the surface may not be readily accessible. A non contact method would solve the issues related to destructive testing and accessibility while still providing a valid measurement.

For these reasons several non-contact methods for determining surface roughness have also been developed. All of these methods operate under the principle that a reflection of a surface carries information about that surface[1]. An example is shown in the figure.

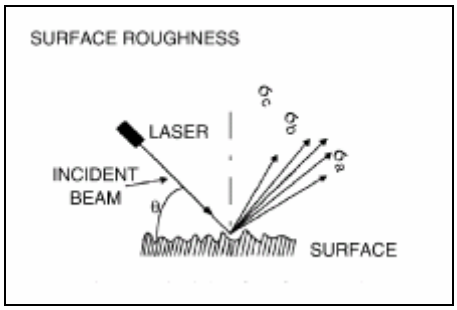

Figure 1 Reflected laser

There are many variations of using light to perform non-contact surface roughness measurement. One apparatus capable of accomplishing this has been developed by the Engineering Applications Center at the University of Hartford. This design uses a solid state red laser to provide a monochromatic collimated light source which is aimed perpendicularly to the surface in question. The backscatter or diffuse reflection[2] from the surface is reflected coaxially back towards the source to the light. In this system, 4 photo detectors are distributed evenly around the laser to measure this backscatter. The amount of backscatter is directly proportional to the magnitude of the surface roughness. To use this apparatus, a computer with a data acquisition card is used to collect the voltages. The voltages are added together and the sum total is used as the measurement value. The computer also provides a graphical user interface which leads the operator through the calibration and measurement procedures.

This project is intended to cut the umbilical cord from the computer and design and implement a stand alone device capable of remote operation of the measurement system. This will allow for the measurement system to be mobile without sacrificing accuracy or precision of the measurement. In this stand alone system, a 4 line by 20 character LCD display will replace the graphical user interface. The data acquisition card will be replaced with two 2-channel 16-bit sigma-delta analog to digital converters. An 8-bit microcontroller will be sufficient to control the entire system. An on-board EEPROM will be used to store calibration data. A custom circuit board will be designed to tie the whole system together. The software will be written in the C language for potential portability in the future.

Methodology

The goal of the project is to create a stand alone surface roughness measurement system. The project box contains all that is needed to run the project including the measurement system and user interface. Power is supplied to the inspection probe to drive the laser and photo detectors as well. The following is a block diagram to show how all the hardware ties together.

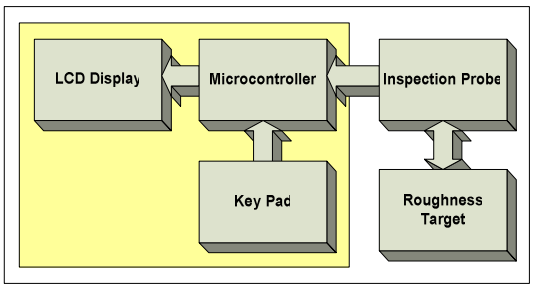

Figure 2- Inspection Probe (a)

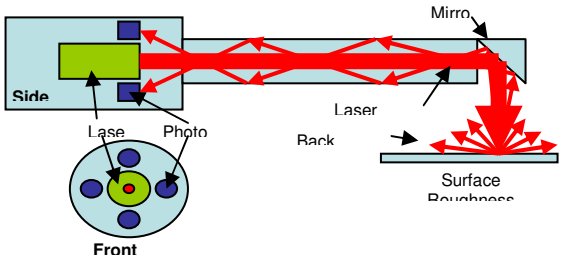

Figure 3- Inspection Probe (b)

The inspection probe consists of a visible laser light source surrounded by four photo detectors. The idea of the apparatus is that laser light is transmitted along a hollow tube and reflects 90° off of a mirror and onto the surface of interest. If the surface is a mirror finish with no roughness, the light would be reflected back along the same path directly towards the laser source. If the surface contains a rough finish the laser light will still reflect most of its light back towards the source, but a portion of it will become scattered. Some of the scattered light will find its way back down the tube, but not collinear with the original beam. There is a relationship between the roughness of the surface and the amount of scattered light finding its way back down the tube. This scattered light will reach the four photo detectors and be converted into a voltage. It is the sum of the voltages that is considered the measurement value for the surface roughness sensor.

Figure 4- Surface Roughness Standard

The surface roughness standard is what will be used for both calibration and measurement. Once the measurement system is calibrated, success will be determined if the known values are then correctly identified. On this standard, there are six types of surface finishes available. The only surface finish of interest is the grinding finish. Within the grinding finish, there are six samples ranging from 2 to 63 AA (micro-inches).

Microcontroller

As a stand alone measurement system with user interaction, the project needs some sort of intelligence. A Microchip brand dsPIC microcontroller was chosen to provide this intelligence. A microcontroller can be described as a microprocessor, but with a high integration of memory and peripherals. The dsPIC30F3013 had all the memory and peripheral needed to control this project on-board. Each of the 28 pins of this device was consumed in the development and deployment of this project. The dsPIC has the capability to run at a 120 MHz clock rate. This project will run at an 80 MHz clock rate due to the availability of a 20 MHz resonator (which is up-converted to 80 MHz internally). The instruction execution rate of all Microchip devices is always 4 times less than the clock rate. This means that the processor will be running 20 Million Instructions per Second (MIPS). From this, all timing for the peripherals can be derived. [5]

The USART is used to provide data and commands to the LCD display. Only one data line is needed to pass this information. As a built in peripheral, the USART was configured to pass the data at 19,200 Kbps. As each byte of data is placed into the USART buffer, it is passed to the LCD Display automatically. An alternate use for the USART is that it is the standard port for In-Circuit Serial Programming (ICSP). A Microchip compatible programmer is used to load the program code into the dsPIC's flash memory through this port.[5]

Each of the four photo detectors was read by four Analog to Digital (ADC) inputs. In reality, the dsPIC only has one ADC, but it is multiplexed between many pins (in this case, four were used). In fact, one of the main reasons the dsPICF3013 was the device chosen from the family was that it contains a 12-bit ADC. Most other microcontrollers on the market are limited to 10-bits or less. A 12-bit ADC provides 4096 steps of granularity between voltage readings. Initial trials of the inspection probe indicated that there would not be a large voltage difference between each of the six surface roughness levels, so fine granularity was needed. With a voltage reference of 5 Volts and a baseline of 0 Volts, each bit represents 1.22 mV.

The majority of the rest of the pins were configured as general purpose digital input pins. Each of the 12 input pins was connected to the individual terminals of the Key Pad. In an idle state where a key is not pressed, 5 Volts is provided to the input pin and read as a logical high. When a key is pressed, that line is shorted to ground, providing 0 Volts to the input pin and read as a logical low.

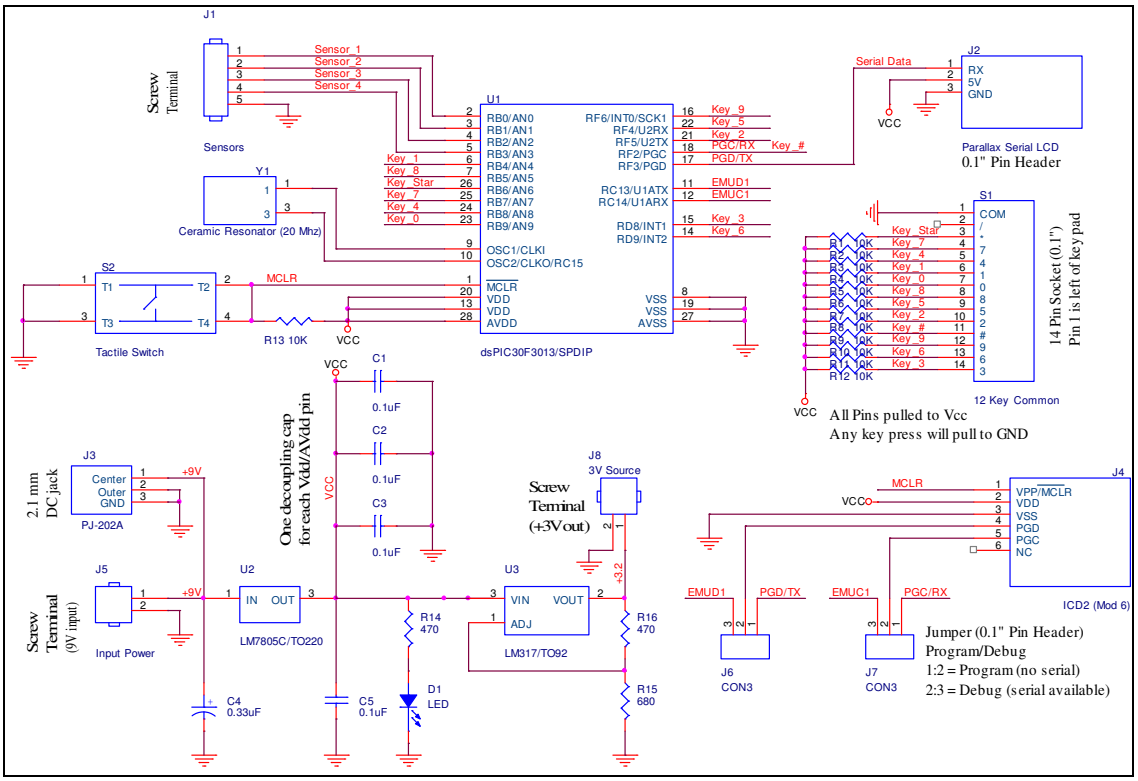

Figure 5- Schematic

The schematic was created in Orcad Capture 10.5. The schematic ties all the hardware components together. The dsPIC microcontroller has just enough pins to handle all the functionality. Not one pin was left over. There is one power supply input for the project. It must be greater than 5 Volts DC. From there, the voltage is regulated to 5 Volts and distributed to the microcontroller, LCD display and Key Pad pull-up resistors. That 5 Volts is then further regulated and reduced to 3 Volts to drive the Laser and Photo Detectors. Each of the four Photo Detector analog signals is fed to four ADC pins on the microcontroller. Each of the 12 dedicated lines from the Key Pad is fed to 12 digital input pins on the microcontroller. These pins are arbitrary, and the keys are defined in software. The communication to the LCD display is a single line running from the USART TX. There is an ICD2 receptacle used to attach the ICD2 compatible programmer/debugger. Note the set of jumpers used to switch back and forth between programmer and debugger modes. These operations take place on different pins in this case. If the USART was not in use, the operations could take place on the same pins

Algorithm Development

The first trial application to see if the hardware was working was a voltage display. Each of the 4 Photo Detectors were read and displayed on the LCD as a bit-count along with the sum total.

The bit-count could range from 0 to 4095 as the voltage ranged from 0 to 5 volts. The first goal of the algorithm development was to derive the relationship between the 6 surface roughness template standard values and the voltages read from the measurement probe. The second goal was to see if there was to see if this relationship was repeatable and reproducible.

Historically, users of the measurement probe have used an average of 3 readings at 3 locations on any measurement target. This project did not deviate from this approach and did not investigate any alternative methods. The data below shows the results from a sample trial run.

Table 1 Measured results

Index	Roughness	Trial 1
1	2	3151.5
2	4	3333
3	8	3767
4	16	4177
5	32	4366
6	63	4733.5

The goal of any statistical evaluation is to try to get your data to fit a known model. Knowing that the microcontroller code would need to be developed to handle the statistics of this project, a linear fit would be ideal. As show below, sensor output vs. surface roughness is not linear.

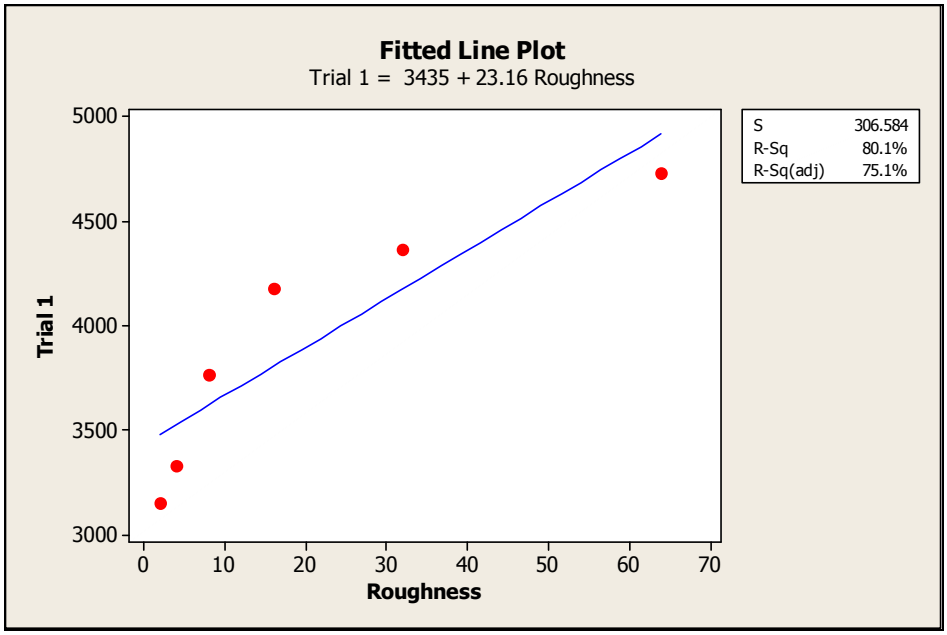

Figure 6- Measured results graph (1)

If the same plot and analysis is performed between the measurements and the index number of the surface, the result can be modeled by a line.

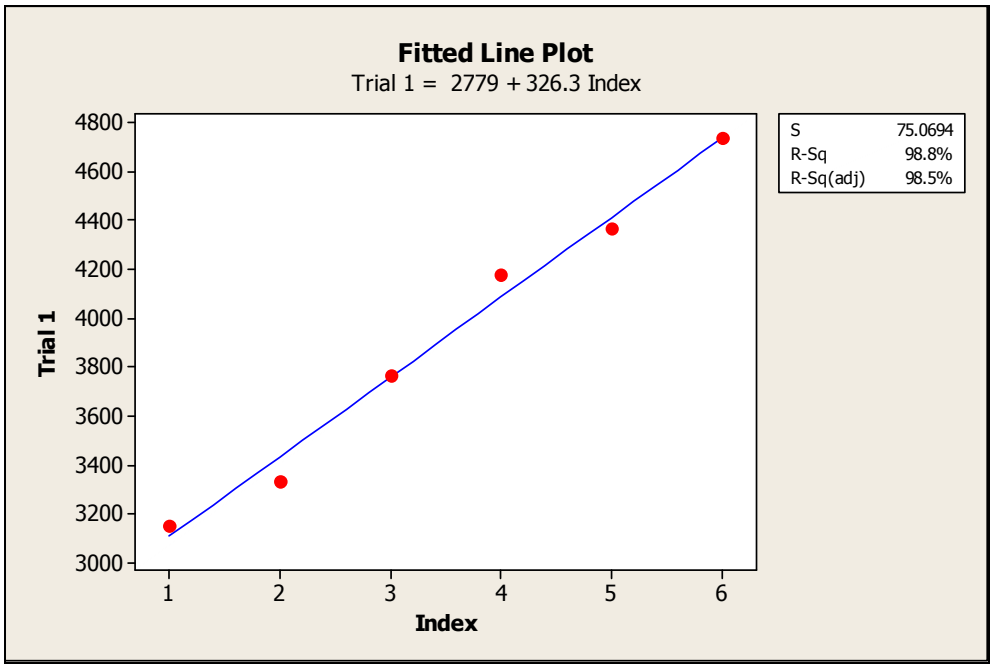

Figure 7- Measured results graph (2)

Besides just being lucky that the index has a high correlation with the measured values, there should be some sort of mathematical transformation between the roughness value and the index. There is such a transformation in that $SurfaceRoughness = 2^{index}$. This transformation validates the use of index and allows the algorithm development to continue. It is this relationship which will be used for calibration of the system and in the measurement routines.

Calibration Algorithm

To calibrate the measurement system, sample data will be collected from the known surface roughness target. This data will then be fit to a line and the equation of the line will be stored for later use. To judge the quality of a linear fit without being able to see a graph of the data, an R-Squared (R^2) value will be calculated as well. The R^2 value represents the proportion of variation in the response data explained by the predictors. This means that the higher the R^2 value, the greater the chance that the index alone will describe the measured value. Mathematically, the R^2 value is literally the square of the correlation coefficient value R. The correlation coefficient can be derived using the following equation:

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}$$

Where the standard deviation of the population is estimated using

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$ with an average of $$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^{n} x_i$$.

The line equation that is required is of the form y = mx + b where m is the slope and b is the intercept. The slope is found first using $m = \frac{rS_y}{S_x}$. The r is the same one that was calculated as the correlation coefficient earlier. The slope is then used in the calculation to find the intercept using $b = \bar{y} - m\bar{x}$. This linear equation works when the measurement value is in bits like the trial or as volts like what will be used from this point forward.

Measurement Algorithm

As would be assumed, the measurement algorithm is nearly the reciprocal of the calibration routine. The goal is to use the measured value to predict the surface roughness. This could be done simply by feeding the measured value in to the calibrated line equation and identifying the index value. The index value would then be transformed back into a surface roughness value. The algorithm adds just a little bit of intelligence to this approach in order to restrict the output of interpolated values. This project was not intended to give results that had higher resolution than the calibration standard, so only the six calibrated values are reportable. The way that this is achieved is that the index value is rounded to the nearest whole index value. The six roughness targets indexes are mapped to their given surface roughness values in micro-inches.

To ensure that one trial could predict the outcome of another trial (such as a calibration and then a measurement), a second set of trial data was collected. Using the same approach as the linear fit above, the correlation between the two data sets was very high as can be seen in the chart below.

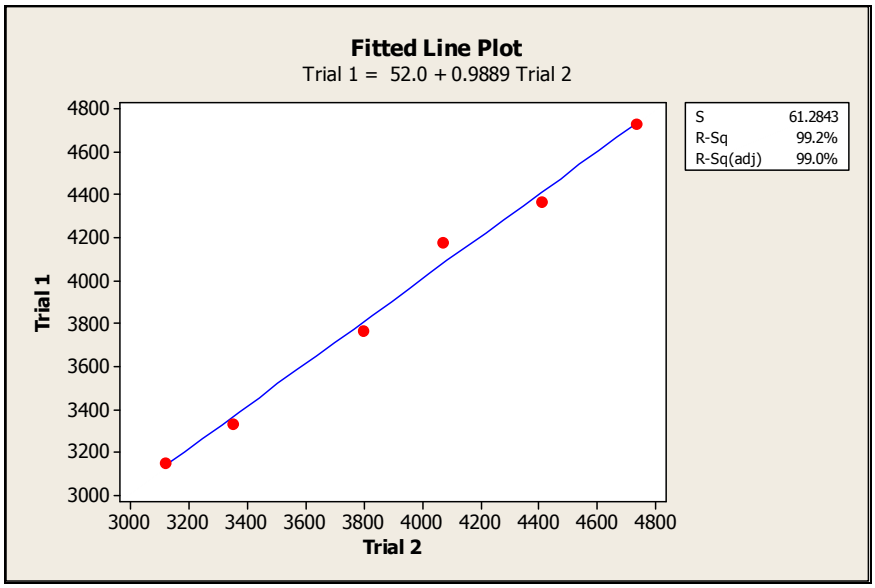

Figure 8- Fitted Line Plot

Software

As the dsPIC microcontroller is a Microchip brand device, the development environment was provided by Microchip as well. The entire project was developed in MPLAB. This environment handles the functionality of the source code editor, project manager, simulation, debugging, downloading of the code, and calling the compiler and linker. It is also MPLAB that runs the ICD2 programmer/debugger. MPLAB is free to download and use and contains all that is needed to develop code in assembly language. Although assembly language is required for some situations, a more universal approach was to use 'C' for this project. This allowed for focus on developing routines that were good enough to demonstrate a successful project. Assembly is used sometimes to optimize speed and size of the code when it is absolutely needed. For most applications, the benefits of 'C' simply out weigh any performance loss due to its use. The 'C' compiler used in this case is also provided by Microchip in the form of the MPLAB C30 C Compiler Student Edition. This compiler is free, with the only limitations being the lack of optimization routines.

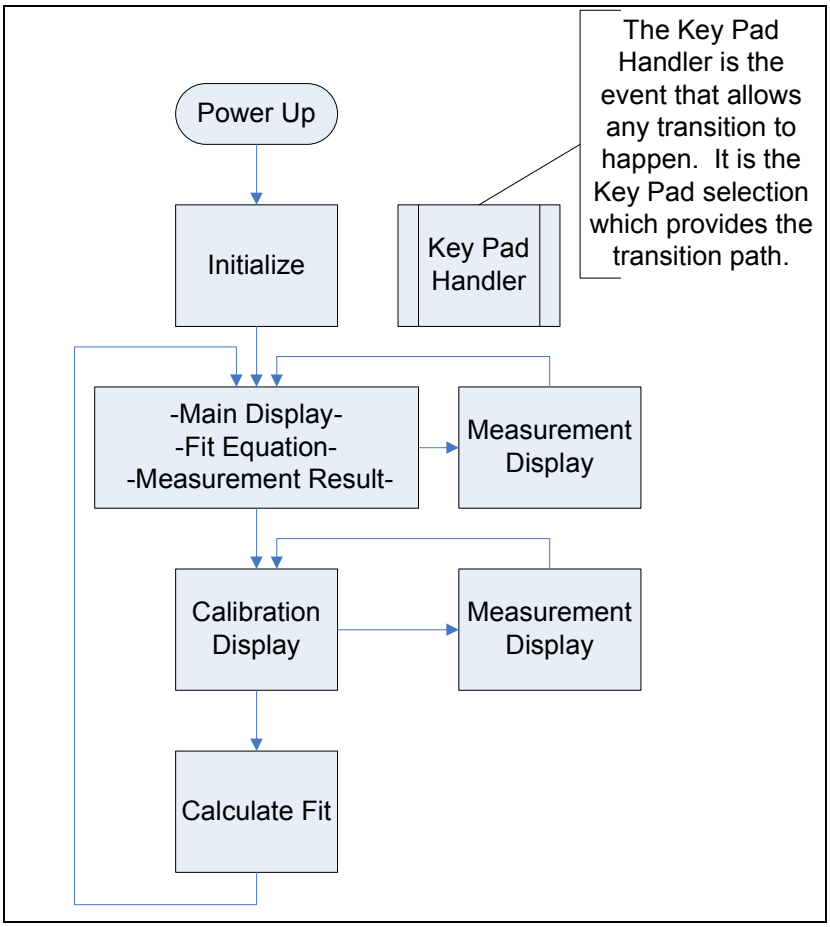

Figure 9- Program Algorithm

This flow chart describes the basic approach used for the user interface. A set of options are described on the LCD display at any given time. A Key Pad selection is required to transition from the current screen to the next one. Some of the selections capture data while others trigger

a calculation of a value to be used or displayed later on. A calibration must be performed before a measurement can be made, so the user is forced to the calibration screen at power up. The Measurement display is used in both the calibration and measurement routines. A common look and feel is provided throughout the project for ease of use and ease of training.

LCD

The LCD Display routine was developed in a stand-alone modular fashion. From top, just 3 parameters are required. The parameters are row, column, and the text to display. There is also a routine to clear the display to prepare for the next state of information. Within the display routine, the Print Formatted Data (printf) routine is called from the standard input/output (stdio.h) library. Since USART1 is where printf directs its output, nothing more needs to be done here with the exception of setting up the desired baud rate during initialization. The row and column information are used to calculate a command which sets the cursor location before the text data is sent. Since the text sometimes needs to include information contained in variables, a text string is created using the String Print Formatted Data (sprintf) function which is also from the stdio.h library. Rather than sending its data to the USART like printf does, sprintf fills an array of characters with the ASCII representation of human readable lines of text. Both routines have the capability of converting data contained in variables to a human readable format. This includes floating point values as well.

Key Pad

The key pad is read through 12 individual digital input lines. In software, each pin can be assigned to its physical key pad button. Each key pad button can be assigned a reference to its display numeric value or symbol. The reference used in this program is a bit location in a 16-bit wide binary value. The 4 highest bits are always assigned a value of zero. The lower 12 bits are assigned to the key pad buttons as described in the table below.

```
#define KEY_1 1
#define KEY_2 2
#define KEY_3 4
#define KEY_4 8
#define KEY_5 16
#define KEY_6 32
#define KEY_7 64
#define KEY_8 128
#define KEY_9 256
#define KEY_0 512
#define KEY_STAR 1024
#define KEY_POUND 2048
```

This way, the key pad value can just be read back as a value in a variable. The value can then be used to make branching decisions to provide the transition path to the next task. This method is flexible enough to identify any arbitrary key pad combinations as well.

The Key Pad Handler routine was developed in a stand-alone modular fashion. When the value of the key pad is required, this routine is called with no parameters. What is returned it the key pad value as described in the map above. To ensure de-bounce functionality, no key pad value is returned until it is stable for 100ms. This is done by reading the key pad value, waiting 100ms, and then reading it again. If the value is the same both times, the value is returned to the calling function. It the values are not the same, it will keep trying until it is. A sample of a routine which requires a key pad value to branch is given below. Note that if an alternate value is given, the function is skipped.

```
keyPress = getKeypadValue();
switch(keyPress)
{
     case KEY_1:
          {
               lastMeasurementVal = getMeasurement();
               lcdClear();
          }
          break;
     case KEY_2:
          {
               calibrateSensors();
               calculateFit();
               lcdClear();
          }
          break;
}
```

Measure

The measurement function was developed in a stand-alone modular fashion. The function is called without and parameters and the measurement value is returned to the calling function. The 12-bit ADC peripheral handles the conversion of an analog voltage to a 12-bit representation of the voltage. The peripheral handles the timing and sequencing required to set the multiplexer to the correct input pin, sample the voltage onto an internal capacitor, and then convert the voltage into bits. The sequence is started by flipping a control bit in the ADC control register. When the sample is done, the ADC peripheral will flip a control bit in the ADC control register. The software routine will poll the 'done' bit until it is flipped. Next the software will read the 12-bit value from the ADC location into a variable. This can be done up to a rate of 200 thousand times per second using the dsPIC30F3013 microcontroller. At this rate, each of the 4 ADC voltages read from the 4 photo detectors was read 16 times each and then averaged. From there, the 4 voltages from the 4 photo detectors were added together into a sum total which was then returned back to the calling function.

Calculations

Given the statistics and the ability to solve for variables described previously, all the equations needed to be developed for use on a microcontroller. For the ease of development, and to provide proper user interaction, floating point math was used exclusively. The dsPIC family of

microcontroller does not have native floating point capability, but the MPLAB C30 compiler will translate it into functional code with no particular knowledge needed by the developer. With the aid of high clock speed, a 16-bit wide bus, and a hardware multiplier, any delays created in the floating point calculations are imperceptible to the user. All high level calculation were placed in their own file of linear fit routines. To prove the value of 'C' as a development language, all the routines were developed first in Microsoft Visual C++ due to its speed and debugging capability. Once the routines were established, the linear fit routine files were moved to the MPLAB project. The routines were called directly without modification for use in an embedded application.

Results

As a general comment, the project worked as expected and as desired. As a portable surface roughness inspection probe, the umbilical cord to the computer was truly cut. This is a stand-alone unit with a complete user interface that only requires a single source of power. It is portable and can be plugged in anywhere. It would also be possible to add a battery to the project box to eliminate any outlet restrictions, although it would be unlikely that this would be used away from a power source.

During two functional trials and three live demonstrations, the displayed result was correct all but one time. The user interface is clear and intuitive and provides all relevant information as the user needs it. There is some setup which helps to achieve the desired results. The inspection probe must be adjusted so that the reflected beam is as co-linear as possible. This can be done by holding one of the buttons on the measurement screen and viewing the measurement value in real-time. While viewing the measurement, the inspection probe should be twisted and turned so that the value is as low as possible. Distance from the target is also of concern. It was recommended that a distance of about 15mm was ideal and the demonstrated setup was set as close to this as possible. Given the mechanical setup as was just described, the differentiation between samples goes up. Any deviations from the setup raise the measurement noise floor and can dilute the results.

With the setup involved to obtain the desired results, it is difficult to determine if this design truly is portable. As it stands, it is not. If the surface being tested was of a known shape and size, the inspection probe certainly could be set up and calibrated for specific tasks. Dedicated fixturing would go a long way here.

During initial trials, it was discovered that the measurement values were wandering around and were not stable. The root cause was that feedback to the 3 Volt voltage regulator was left open. Although this was correct at the schematic level, the trace was unintentionally left out of the PCB design. This is one of the risks you run when a low-end software package is used. There are no design-rules or netlist verification tools built in. The trace simply went un-noticed. This was easily fixed by adding a piece of wire to the circuit where the trace would have gone. The PCB design was also updated to reflect this.

As described above, the measurement system did make one wrong reading. This was only off by one index on the measurement standard, and a human would likely make the same mistake once

in a while. It may be OK for a human to make a mistake due to perception, but a measurement system should not be wrong. One contributor to the problem could be caused by the measurement standard itself. It is old, pitted, corroded, and slightly warped. All of these things would lead to improper amounts of back scatter. The success of the system is due to the average value taken from 3 locations on the target. The more locations on the target that are averaged, the more consistent the results would likely be. In fact, if the locations were not arbitrary, but built into the fixturing, this would be the best case. Aside from anomalies in the surface, it needs to be understood that this is an average roughness measurement system. The laser light is very precise and may be on the edge of measuring local roughness that would not be considered consistent across the entire surface.

Bibliography

1. Reference: H.Y. Kim, Y. F. Shen, J.H. Ahn, "Development of a surfrace roughness measurement system using reflected laser beam," Journal of Materials Processing Technology (2002)
2. Bennet J M and Mattsson L *Introduction to Surface Roughness and Scattering* (Washington, DC: Optical Society of America, 1993)
3. Shetty, Devdas. Neault, Henry. 2007 Method and apparatus for surface roughness measurement using laser diffraction pattern. US Patent 5,189,490, filed September 27, 1991, and issued February 23, 1993
4. Larry Adams, "Laser-Based Tool Maps Surface Roughness," Quality Innovations February 2001
5. DSPIC30F Family Overview, Retrieved December 6, 2006 from: http://ww1.microchip.com/downloads/en/devicedoc/70043F.pdf
6. DSPIC30F Reference Manual, Retrieved December 6, 2006 from: http://ww1.microchip.com/downloads/en/DeviceDoc/80169E.pdf
7. SerialLCD-RevE.pdf, Retrieved December 6, 2006 from: http://www.parallax.com/dl/docs/prod/audiovis/SerialLCD-RevE.pdf