Capillary Detection During Calibration Process

By

Yifei Jin

3/18/2018

Chapter 1: Introduction

1.1 Abstract

Medical Electric Systems(MES) is an Israel based technology company specializing in rapid, automated semen analysis for the human and the veterinary markets. Its most important product, Sperm quality analyzer, needs a tedious and manual process for its calibration. My goal was to improve the calibration. By using barcodes and barcode scanners, I developed an efficient and automatic way to calibrate.

1.2 Acknowledgements

I would like to thank Mes Medical for offering me the great major qualifying project and the resources related to the project. Especially I want to thank Beni Cohen and Nirit Touboul for overseeing my project and providing invaluable and helpful advices. In addition, I would like to thank my advisors, Professor Isa Bar-On and Micha Hofri, for their guidance and assistance during the project.

1.3 Table of Contents

1.4 Table of Figures

1.5 Glossary

Term	**Description**
MES	Medical Electric System
SQA-V	Sperm Quality Analyzer
ISO	International Organization for Standardization
FDA	US Food & Drug Administration
CEE	Consumption for Energy Efficiency
IVF	In Vitro Fertilization
PC	Personal Computer
HID	Human Interface Device
API	Application programming interface

1.6 Executive Summary

A sperm analysis is an important part of a male infertility diagnosis. It is most commonly used in medical domain and human sperm donation tests. For animals, it is generally used in animal breeding and physical examination. The primary purpose of sperm analyses is to determine whether sperm parameters are related to male fertility. Other reasons for performing it includes verification of the success of vasectomy, confirmation of the presence of sperm after a reverse vasectomy, and fertility evaluation. The analysis can be relatively expensive and time-consuming. In the US, the cost varies from $50 up to $300 and in most cases, it is not covered by insurance.

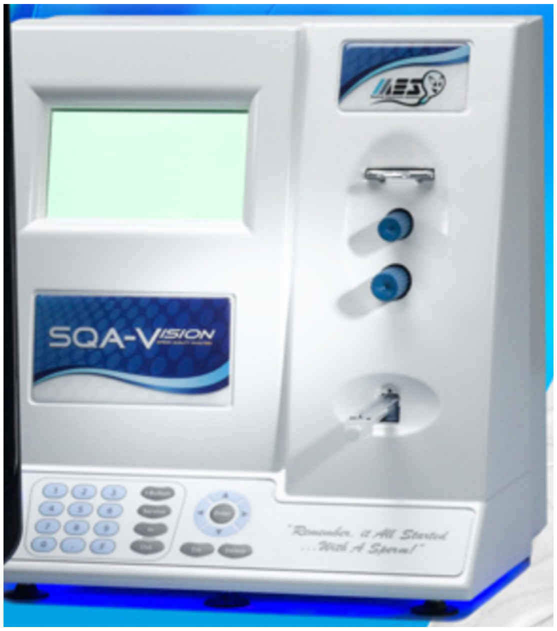


Figure 1 - Sperm Quality Analyzer - Vision

One of the main products from Mes Medical is Sperm Quality Analyzer – Vision (SQA-V), which aims to provide rapid, objective, and standardized semen services. The SQA-V device makes the analysis much easier and more affordable. However, every SQA-V device needs to be calibrated approximately every two months and the calibration process is fairly complicated and time-consuming.

In practice, the calibration process would involve multiple capillaries and devices that are tested in parallel. As a result, it is difficult to distinguish the capillaries and there might be confusion about the capillaries. For example, it would be hard to tell which capillary has been tested in a device. Mes Medical has decided to enhance its process by automating several of the manual steps. One of the most crucial step that needs to be improved is the identification of capillaries.

My task was to improve the calibration process by setting up the basis for automatic calibration. First of all, I needed to develop a solution that allows tracking the capillaries inserted into each device. The solution should include an automatic capillary identification mechanism. My approach is to create a barcode marked on each capillary and there will be a barcode reader that is attached to every device during the calibration. The capillary should be scan in each device before using it. In this way, it is able to keep track of the capillaries and avoid confusion.

Chapter 2: Background

2.1 Company

Figure 2 - Mes Medical Logo

The project was completed as requirement of Major Qualifying Project during a seven-week period at Mes Medical in Israel. Mes Medical is a leading-edge technology company specializing in automated semen analysis. Today it has placed over 2,500 automated semen analyzers worldwide. The company is ISO (International Organization for Standardization) certified as a medical manufacturer and all products are FDA (US Food & Drug Administration) approved and CEE (Consumption for Energy Efficiency) certified. In addition to Israel, Mes also

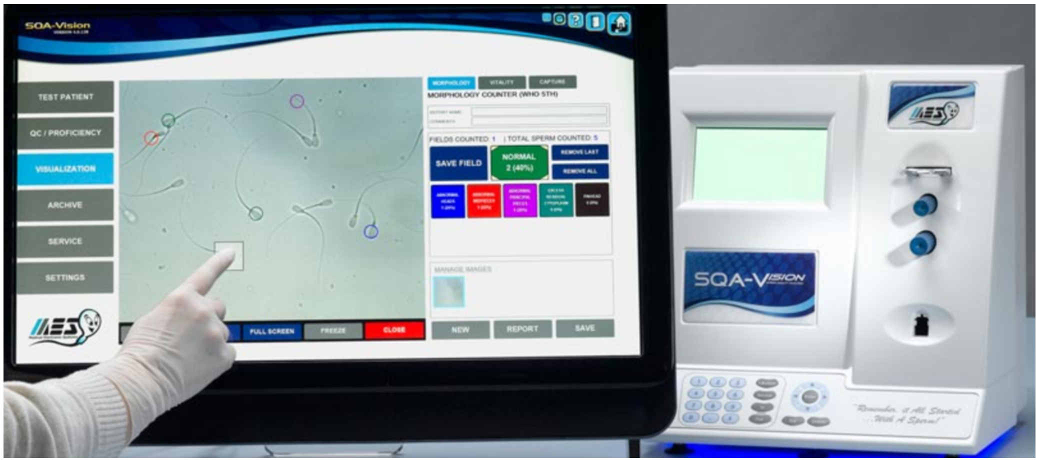


Figure 3 - Visual Interface and SQA-V device

has offices in Los Angeles, Austria, Hong Kong, and Shanghai. It sells the instruments through a worldwide network of distributors.

Its best-selling instrument is SQA-V, which delivers a complete quantitative semen analysis in seventy-five seconds. According to Mes, the device is developed as an intuitive tool for fertility screening, post-vasectomy follow-up, sperm banking, research or detailed IVF (In vitro fertilization) assessment. The SQA-V device includes a high resolution visual interface, bar code scanner and 'touch-to-mark' technology for assessing both sperm vitality and conducting full differential morphology.

Sperm analysis can be extremely subjective, time-consuming, and lacking in standardization. The instrument makes life much easier both in the lab and in the hospitals. The instrument performs a five-minute calibration when it is first turned on and a 30-second self-check before each test. The next step is for the operator to enter some basic patient information, load the disposable testing capillary and insert it into the machine. Seventy-five seconds later the results including sperm concentration motility breaking into rapid, slow, and non-progressive morphology functional sperm velocity and more will be shown on the screen. There will be sixteen clinical parameters for quality control including average length, count in a certain area, and diameter. Overall, the SQA-V provides a solution in standardizing semen analysis and offers a lower cost per test with accuracy and precision.

2.2 Calibration

The machine needs to be calibrated for the first time use and every two months to ensure its accuracy. The whole calibration process will take around five minutes. If there is something wrong with the sperm sample, more time is expected to be spent in the process. I talked to a lab technician at Mes Medical who is responsible for the calibration. She revealed that it normally takes up to more than one work day to calibrate a dozen devices. The process involves recording data manually. Consequently, in order to avoid error, the calibration is done more than once per machine.

During the calibration process, the devices are calibrated against reference devices. The reference devices are the ones that have been calibrated already against microscope, which is the golden standard. The process is also fairly complicated and time-consuming. The figure shows three devices that need to be calibrated and two reference devices. Specifically, the process can be described in the following six steps:

Figure 4 - Calibration Schematic

1. A capillary, or plastic container with known material, is inserted into a device and a test is performed.

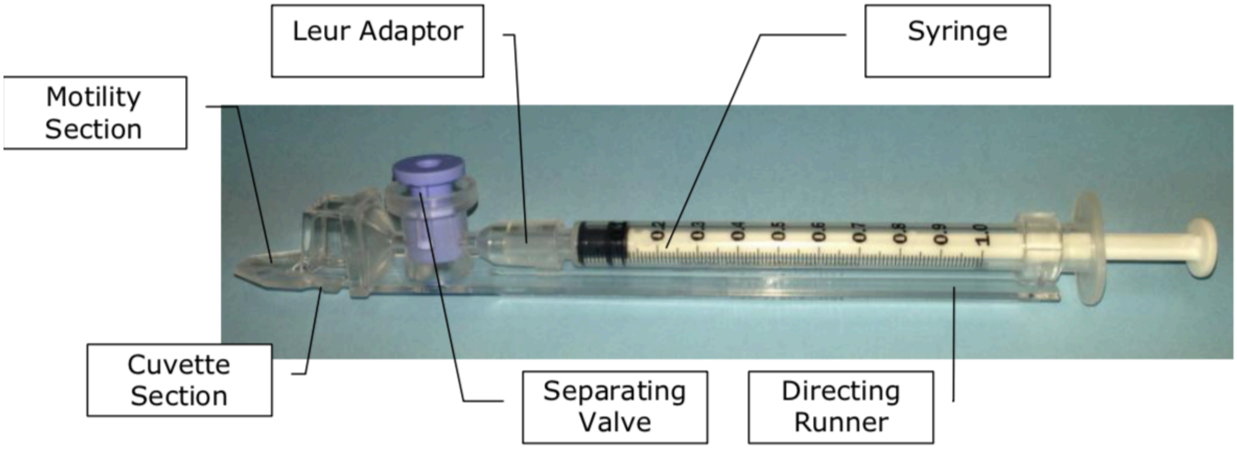


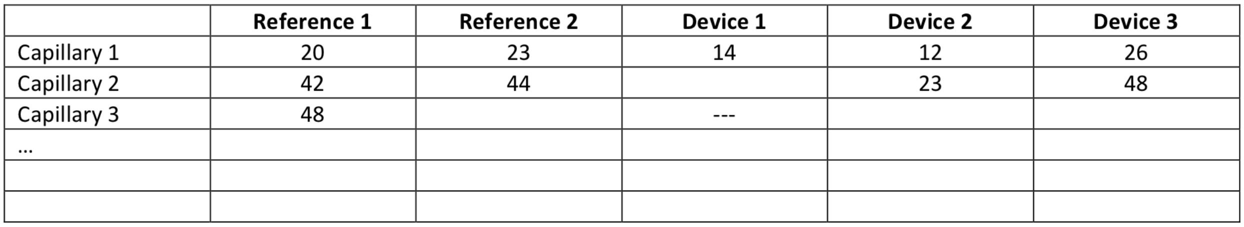
Figure 5 - Capillary

2. The results for the test for the specific device are written manually.

3. The same capillary is inserted to the following device and a test is performed.
4. In the same way, the results are written and collected for all devices.
5. This process continues with the same capillary and includes the reference devices as well.
6. For each device the results are compared to the reference results and if needed the calibration parameters are adjusted to bring the reading of the device to be the same as the reference devices.

The whole process, obviously, is long and error prone, especially with the numerical parameters recorded by hand. As many capillaries are calibrated together, there might be a confusion about the capillaries and devices. it is difficult to follow up which capillary has run already and in which device.

In order to improve the productivity of the calibration process, Mes Medical decided to create a more automatic and efficient way to execute the calibration. The company wanted a solution that allows tracking the capillaries that inserted in each device. It was required that, when a capillary is inserted into a device, an identification mechanism should send the information to the PC. When a test is completed in a specific device with a specific capillary the results should be sent to the PC. The PC should always present a table with all the measurements that are taken while each row represents different capillary and each column represent a device. The reference devices should always be at the first couple columns of the table. Each time a new capillary was tested, a new row was supposed to be automatically created. The table below shows an example of the data that is collected. The whole outcome was supposed to be presented on Windows.



	Reference 1	**Reference 2**	**Device 1**	**Device 2**	**Device 3**
Capillary 1	20	23	14	12	26
Capillary 2	42	44		23	48
Capillary 3	48		---		
…					

Figure 6 - Required Table

Chapter 3: Methodology

3.1 Introduction

Two approaches came up. One was to mark every capillary with a unique barcode and there should be a barcode reader attached to each device during the calibration. The capillary should be scanned in each device before using. The second approach was to mark each capillary with a radio frequency low cost component which will transmit to the PC the Capillary ID each time a capillary is used in a device. The first one finally stood out due to its simplicity and people were generally more familiar with barcode. The other reason was that a wireless barcode reader provided remote capability. With the wireless barcode reader, it was no longer necessary to connect to the PC that it is working with. The laboratory could be much cleaner and more orderly.

After the decision on the barcode, there were four phases needed to be considered. The first phase was to create a barcode based on the given number and save it as a certain size that fit the capillary. The second was to connect the device and a scanner. After the connection, a table should be created automatically. Then the third was to distinguish the different scanners. The key here was to obtain a unique ID from each scanner. The final phase was to send the results from the device to the PC. WinForms was used because of its extreme simplicity. Especially for a

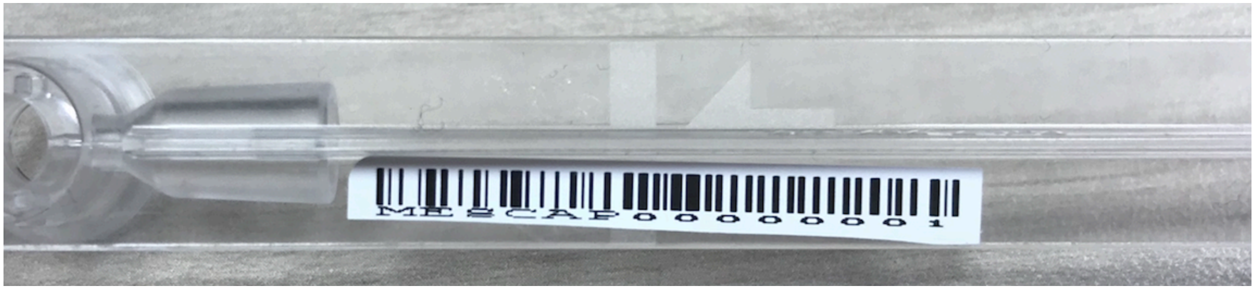


Figure 7 - Barcode on a Capillary

short time, WinFroms was able to develop an application in a fast and efficient manner. It was also easy to use and new user friendly.

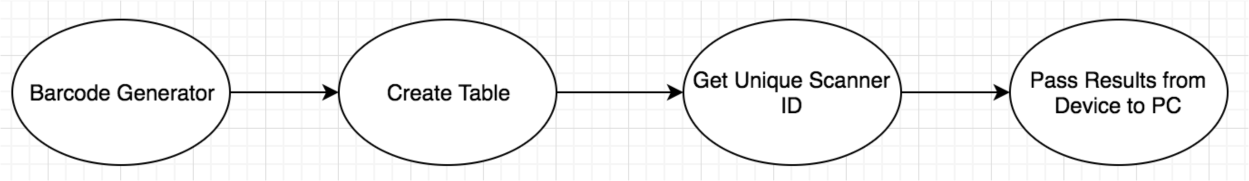


Figure 8 - Work Phases

3.2 Barcode Generator

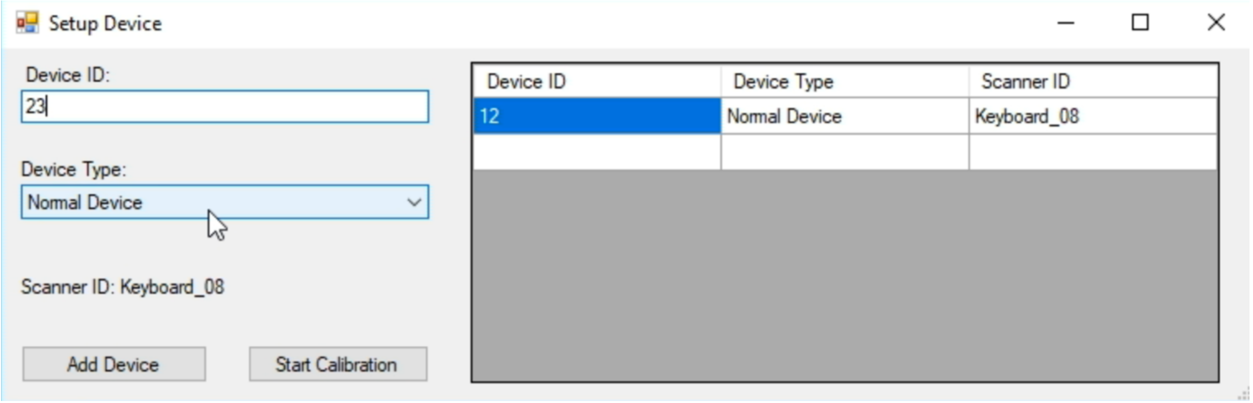
Barcode is one of the most commonly used tools in daily life. By varying the widths between the parallel lines, the barcode is able to represent different data including numbers and characters. Due to its wide usage, there are many existing libraries in C# that could generate barcode. The one that I used was called BarCodeLib, which included many different types of barcodes. To get the image representing the given data, I simply called one of the encode methods. Based on the parameters, it will generate different types of barcode with the given string, width, height, and color.

```
public Image Encode(TYPE iType, string StringToEncode, int Width, int Height)
public Image Encode(TYPE iType, string StringToEncode,
        Color ForeColor, Color BackColor, int Width, int Height)
public Image Encode(TYPE iType, string StringToEncode, Color ForeColor, Color BackColor)
public Image Encode(TYPE iType, string StringToEncode)
```

Figure 9 – Encode Methods

3.3 Create Table

To create a table, I needed the information from both the scanner and the SQA-V device. The unique IDs from those two needed to be connected together. After that, the information should be shown in the table. In a Windows application, a DataGridView was the layout element most commonly used for visual table. According to Microsoft library, it displayed data in a customizable grid and offers various ability to customize.



Device ID: 23

Device Type: Normal Device

Scanner ID: Keyboard_08

Device ID	Device Type	Scanner ID
12	Normal Device	Keyboard_08

Figure 10 - Create Table/Setup Device Interface

In the application, once I clicked the add device button, another window would popup asking the user to type the device ID and device type. The scanner ID showed the scanner or keyboard just used for input. The device type was either normal device or reference device. After setting up the device, the Add Device button sent the data to the table on the right-hand side. When all the devices and scanners had been associated, the Start Calibration button would return to the main interface.

Every device ID was connected to a unique scanner ID. Any duplicate ones would lead to warming. A C# Dictionary instance, which represented a collection of keys and values, was used for recording the existing devices and scanners. Before each device addition, the dictionary would be checked for duplication. The scanner ID was obtained by raw input, which would be explained in the next section.

3.4 Get Unique Scanner ID

All the Bluetooth scanners that I owned worked as standard inputs like keyboards. That was convenient in most cases like in market and warehouse. However, for me it caused some problems. There was no obvious way to distinguish them from each other. It was also difficult to where the input was from. No matter the input was from the scanner or the keyboard, they all showed the same feature and characteristic on the PC. I tried to find a unique ID from each scanner such as device ID to solve the problem.

My first attempt was to use a library called 32feet, which was “a shared-source project to make personal area networking technologies such as Bluetooth” according to its official web page. It supported most functions a Bluetooth device would need such as connection, control of the device, pairing, and object exchange. An exhaustive documentation was also provided. However, after about one week’s work, I realized that the library was no longer under maintenance and there was no update since 2014. Unfortunately, one crucial and necessary connection function did not work well within my code. After a discussion with my sponsors, I decided to move to the second approach.

Figure 11 - 32feet and .NET logo

Without the help from the existing library, it would be hard and complicated to have direct control over the Bluetooth scanners because of the lack of understanding how exactly Bluetooth worked. Then I tried to convert every keyboard stroke to a raw input.

Nowadays it is very common to have a wide variation of Human Interface Devices (HIDs) such as touching screens and remote controllers. All the scanners as well worked like HIDs. In .NET framework, it offered no obvious way to differentiate one input from another. applications handling keypress event would receive the inputs from all different devices as if they were from a single same device.

Fortunately, Windows XP and above supported a "raw input" API that allowed the programs to interact with and handle the Human Interface Devices directly. By intercepting and filtering the input, it enabled the program to identify which device actually triggered the input. Using the raw input API to handle keypress event allowed the program to get essential information about the keypress including ASCII code, virtual key, key press state, message in hex, handle number, device type, name, source, and more. It successfully solved the issue and gets the unique ID from each scanner.

3.5 Pass Results from Device to PC

After inserting the capillary and starting the test for a while, the results of the sample, which included a group of numerical data, would be shown on the screen of the SQA-V device. Three most important data used in the laboratory were diameter, average, and count. The transmission of the data from the device to the PC mostly depended on the SQA-V API (Application programming interface). Mes Medical will finish it later.

Chapter 4: Results

I successfully identified the capillaries and connected the scanners with the SQA-V devices. The major functionality was implemented. Some other features required by Mes Medical were well developed as well. Those features comprised exporting the data, importing from an excel file, checking whether a line was filled out, and warming the user after a certain time period.

4.1 Demonstration

Once you enter the application, the main window consists of an add device button, a Generate Barcode button, a Load button, an Export button, a Check button, and an empty table.

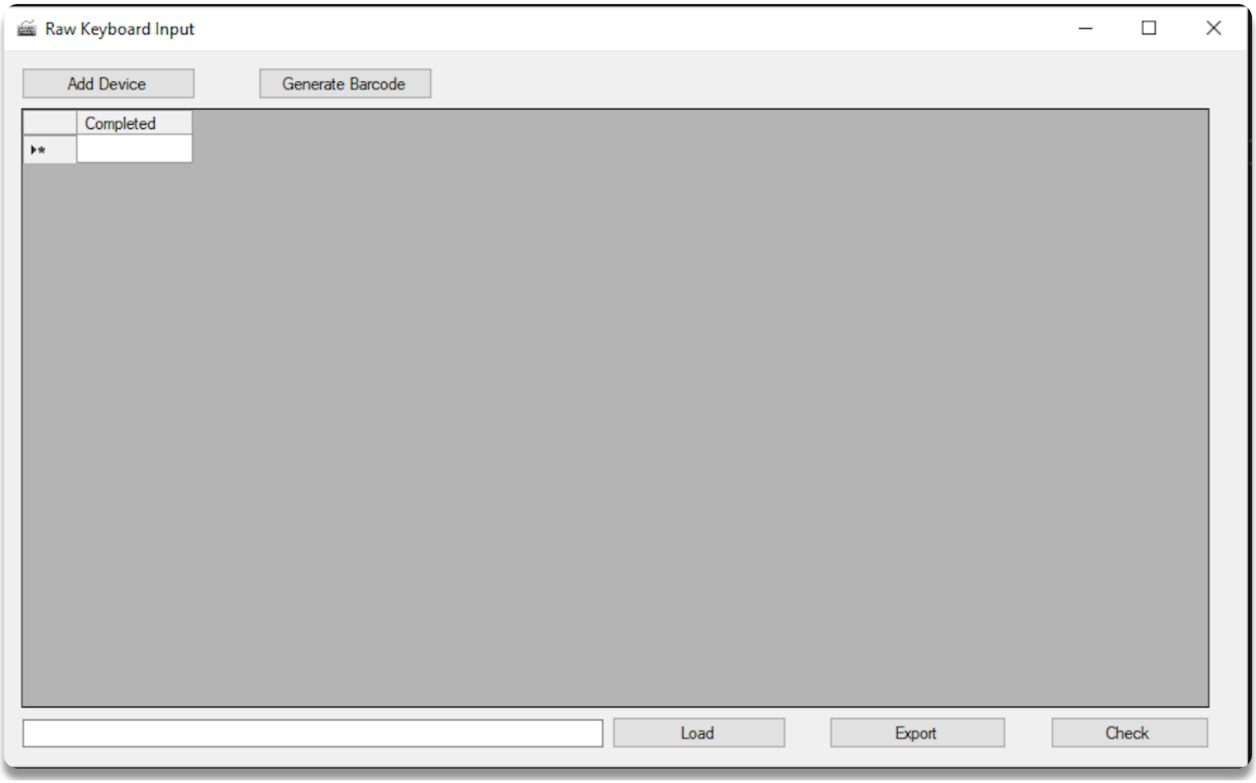
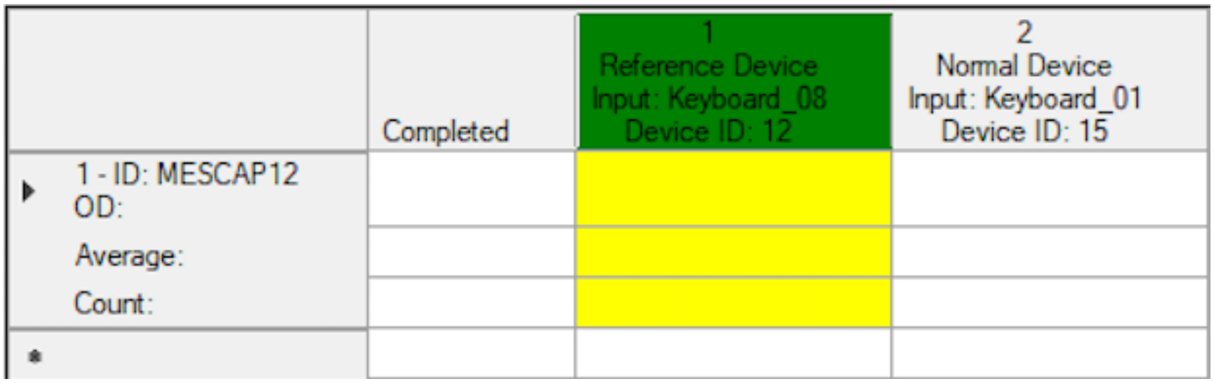


Figure 12 – Main Interface of the Program

The first thing is to connect the SQA-V devices with the scanners. The setting up window has been shown before.

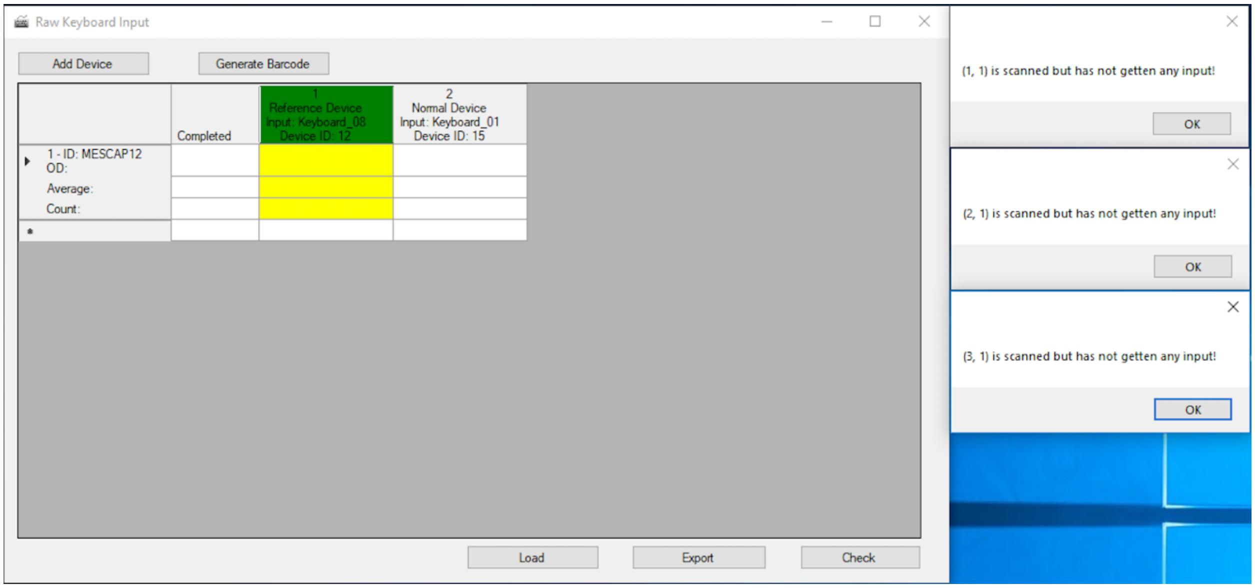
When the devices have been set up and a new capillary is scanned, a new row will be added, and the scanner just used will be marked with yellow. Note that the reference device will always be highlighted with green. The new row will show the capillary number. For example, in the table below, the capillary is with barcode MESCAP12.



	Completed	1 Reference Device Input: Keyboard_08 Device ID: 12	2 Normal Device Input: Keyboard_01 Device ID: 15
▸ 1 - ID: MESCAP12 OD:		🟨	
Average:		🟨	
Count:		🟨	
*			

Figure 13 – New Row

In each field, you are allowed to manually enter some data or get the data from the device automatically after a while. However, if one field has been scanned, marked as yellow, and has not gotten any input back after two minutes, an alert will pop up to warm the user.



	Completed	1 Reference Device Input: Keyboard_08 Device ID: 12	2 Normal Device Input: Keyboard_01 Device ID: 15
▸ 1 - ID: MESCAP12 OD:		🟨	
Average:		🟨	
Count:		🟨	
*			

(1, 1) is scanned but has not getten any input!

OK

(2, 1) is scanned but has not getten any input!

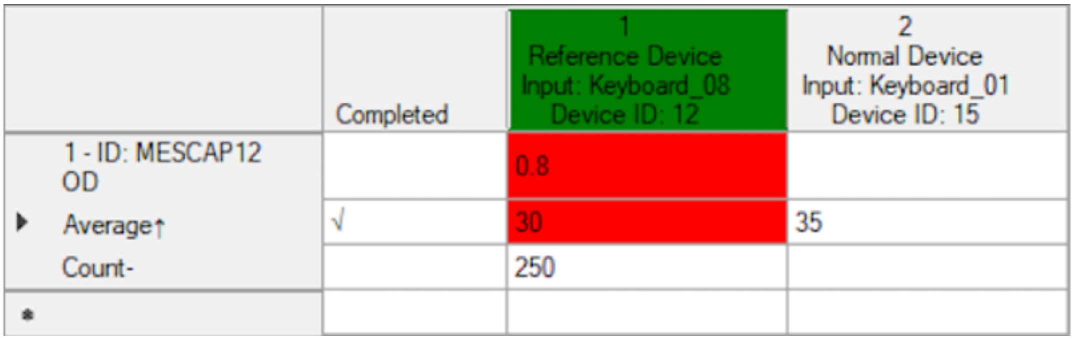
OK

(3, 1) is scanned but has not getten any input!

OK

Figure 14 - Popup warming after a certain time frame

Mes Medical also has a range, either high, low, or normal, for each parameter. If the received data is out of the given range, the cell with the data will be highlighted with red. If the row is filled out after receiving a new data, the completed row will be marked with a small check symbol.



	Completed	1 Reference Device Input: Keyboard_08 Device ID: 12	2 Normal Device Input: Keyboard_01 Device ID: 15
1 - ID: MESCAP12 OD		0.8	
▶ Average↑	√	30	35
Count-		250	
*			

Figure 15 – Red Cell

4.2 Other Features

The program also has some other useful features based on Mes Medical's need. It could export the existing data to an excel file and save it anywhere on the PC. On the opposite, load an excel file to the table is also implemented. There is also a check button, which shows the user whether all the cells have been filled out. If not, all the empty cells with their coordinates will be shown in a new window.

Chapter 5: Conclusion

Mes Medical would like to improve its calibration process. Based on its plan, it could save a couple of hours of work every day in the laboratory. The capillary detection during calibration process was definitely one of the most important function for the plan. My program basically solved the problem by using the barcode and scanners.

As for time improvement, the current process used to take up to two days depending on the quality of the sample. If bubble or other errors always occurred in the sample, it would take an even longer time. In addition, because of the manual record, the tests might be done multiple times to reduce human mistakes. The new process saved time. More importantly, because of its automatic mechanism, errors in the data were avoided and ignored. The lab technicians were also able to work on other things during the tests. The only thing they needed to do was to move the capillaries around and scan the capillaries before tests.

Due to the time constrains, a few features could not be fetishized. The program definitely could be improved. One major missing function was to pass the data from the SQA-V device to the PC. Hopefully this part will be implemented by Mes Medical soon. Also, the interface layout could be enhanced to bring the user a better visual experience.

During the project, I no doubt learned and improved a lot. One most important thing I learned was to manage my schedule based on a full-time job. The way to work every day was about three hours and I definitely knew better how to use the interstice of time. As for programming aspect, I learned how to use .NET framework and its C# language well, which I am sure will help me in my career in the future.

In conclusion, I really appreciated the opportunity to work at Mes Medical and I enjoyed the time there.

Works Cited

Medical Electronic Systems Global | Remember it all started with a Sperm. (n.d.). Retrieved March 15, 2018, from http://www.mes-global.com/

The Universities of Europe in the Middle Ages: Salerno. Bologna. Paris. (n.d.). Retrieved March 15, 2018, from https://books.google.com/books?id=3oFKAAAAYAAJ&pg=PA150#v=onepage&q&f=false

I. (2018, March 13). Inthehand/32feet. Retrieved March 15, 2018, from https://github.com/inthehand/32feet

Messer, E. B. (n.d.). Using Raw Input from C# to handle multiple keyboards. Retrieved March 15, 2018, from https://www.codeproject.com/Articles/17123/Using-Raw-Input-from-C-to-handle-multiple-keyboard

Semen Analysis: Sperm Count Test Procedure and Results. (n.d.). Retrieved March 15, 2018, from https://www.healthline.com/health/semen-analysis

Semen Analysis: Sperm Count Test Procedure and Results. (n.d.). Retrieved March 15, 2018, from https://www.healthline.com/health/semen-analysis#outlook