Advanced Laser Control System V2.1 Beta

Developed By: Yen Goh
Thusitha Mabotuwana
Jimmy Chen
Kasun Talwatte

Date of Report: 05.05.2003

Executive Summary

Development of a low-cost triangulation sensor prototype to measure the thickness of a metal sheet was the main objective of this project and this was to be implemented using the development board for the Mitsubishi M16C controller and a laser triangulation sensor. The provided laser triangulation system successfully triangulates the distance of the white reflective surface using the laser and the light sensor. The thickness of a sheet of metal can be determined by taking two distance measurements and then taking the difference of the two values. An active triangulation technique is used to measure the distance and lookup tables are used to determine the distance depending on the position of the pixel with the lowest generated voltage.

The M16C board is used to control the laser power and this power can be changed through the graphical user interface from 0% to a 100% duty cycle. The integration time can also be varied from 4 – 20ms. The M16C continuously sends the pixel line data to the user interface at a rate of 0.5 frames per second which is plotted in real time on the PC.

A total of 497 person-hours and NZ$15,062.70 were spent in the development of the required final prototype which meets all the design specifications as requested by the client. Since the development was initially scheduled to take only around 160 person-hours there was a substantial delay before completion.

This report gives a brief overview of the project, development stages involved with the final solution and describes how the required functionalities were implemented in the solution to meet the project specifications. It further includes a discussion on testing and performance of the M16C board and the GUI followed by merits and demerits of the whole system. Detailed algorithms and other technical details can be found in the appendices attached as referenced. A detailed project schedule that was developed using Microsoft Project is included in appendix IV.

Glossary of Terms

GUI	Graphical User Interface
PWM	Pulse Width Modulation
M16C	Mitsubishi M16C/62 Group M30624FGLFP Microcontroller
LUT	Lookup Table
CLK	Clock Signal
ROG	Read-Out Gate (Signal)
ADC	Analog to Digital Convertor / Conversion
COM1	Communication Port 1
COM2	Communication Port 2
UART	Universal Asynchronous Receiver Transmitter
LED	Light Emitting Diode
ILX521A	Sony’s 256-pixel CCD Linear Image Sensor
CCD	Charged Couple Device
EXE	Executable File
MSA0654	The development board where the M16C is located
LTS	Laser Triangulation Sensor
PC	Personal Computer

Table of Contents

List of Illustrations

1.0 Introduction

Auckland Sheet Manufacturing Company is a leading company in the business of producing large sheets of metal on a mass scale. The company wants a prototype of a low-cost triangulation sensor to be developed to monitor their production process so that the sheet metal thickness can be measured during production.. The demand for high precision measurements has become simply inevitable with the growing demand for some form of technology for controlling the production quality of many heavy industries thus giving rise to this requirement.

The project can be split up into two main sections, namely data acquisition of a linear sensor using the serial port and designing an algorithm to determine the laser pulse position on the linear sensor and displaying the actual distance value in centimeters using a lookup table.

The implementation of the final design was to be done using an MSA0654 board with an M16C microcontroller which would control the laser and receive data from the sensor which senses the light from the laser. The laser pulse is reflected off a white surface which generates appropriate voltages on the pixels of the laser depending on the distance the pulse was reflected off at. By determining the pixel that corresponds to the maximum voltage the distance to the white board can be obtained with the help of a lookup table. Although the actual distance can be calculated using an active triangulation technique, the lookup table method is used to compensate for the non-linear relationship between the pixel number and the distance value thereby eliminating errors such as spherical aberrations and production tolerances. The actual thickness of the metal sheet is calculated by measuring the distance at two positions and then getting the difference of the two distances measured.

This report discusses the development stages of the final solution of the above prototype and describes how the required functionalities were implemented in the solution. It further includes a discussion on testing and performance of the M16C board and the GUI followed by merits and demerits of the system. Detailed algorithms and other technical details can be found in the appendices attached as referenced. A detailed project schedule that was developed using Microsoft Project is included in appendix IV.

2.0 Development of the design

The prototype was developed under two main subsections:

1. Development of software for the MSA0654 board to meet the following project specifications:
    - Generation of the two trigger signals, ROG and CLK for the sensor as specified in the data sheets. ROG was to be adjustable in the range from 4 – 20ms.
    - Digitalising the analog output of the LTS by using the internal ADC of the M16C chip.
    - Transfer of line data to the PC via the serial connection.
    - The maximum laser pulse amplitude is to be clamped at half the peak-to-peak intensity value by adjusting the PWM signal for the laser.
    - Development of a laser pulse detection algorithm
    - Pixel position in the measuring range is to be displayed on the hex display of the development board.
    - Development of an automatic measurement algorithm to generate the pixel vs. distance characteristics of the LTS using at least 8 samples.

- Transfer of the distance values to the PC via the serial connection.

2. Development of a Graphical User Interface to communicate with the MSA0654 board to control the LTS that would meet the following project specifications:
    - Displaying the line data
    - Implementation of a zoom function for better observation of the laser pulse
    - Ability to change integration time and laser power
    - Displaying the distance value online

3.0 Development of software for the MSA0654 board

An overview of the components interacting with the M16C microcontroller is shown in Appendix i.

3.1 The Sensor

The sensor used in this project is Sony's ILX521A. There are three signals that the M16C microcontroller outputs. Two of these signals, CLK and ROG are used with the sensor. The M16C receives one input from the sensor in the form of an analog voltage signal ("V_{out}").

The sensor captures light similar to a camera with its in-built CCD. There are 256 real light-capturing pixels and 17 dummy pixels within each picture frame. The pixels can be extracted in the form of an analog voltage between 0 to 5 volts on the V_{out} line. The ROG pulse causes the picture frame on the "film" to be copied into the internal shift register, ready to be shifted out. The CLK signal then shifts one pixel out of the internal shift register onto the V_{out} line at every period.

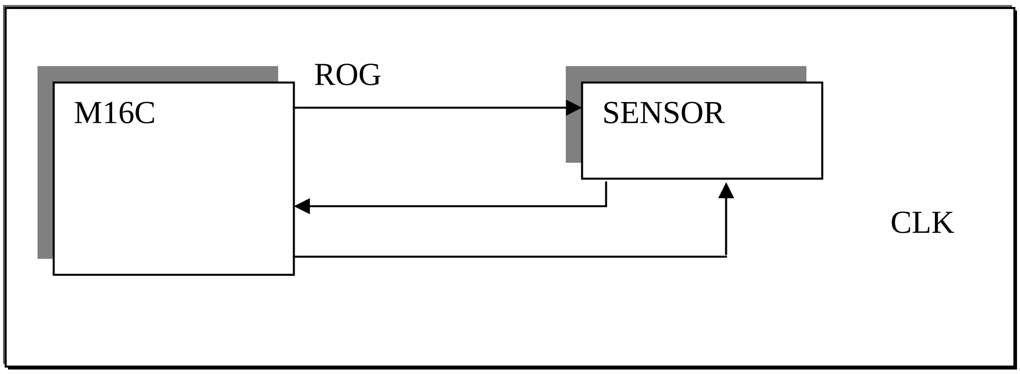


Figure 01: The lines of interaction between the microcontroller and the sensor.

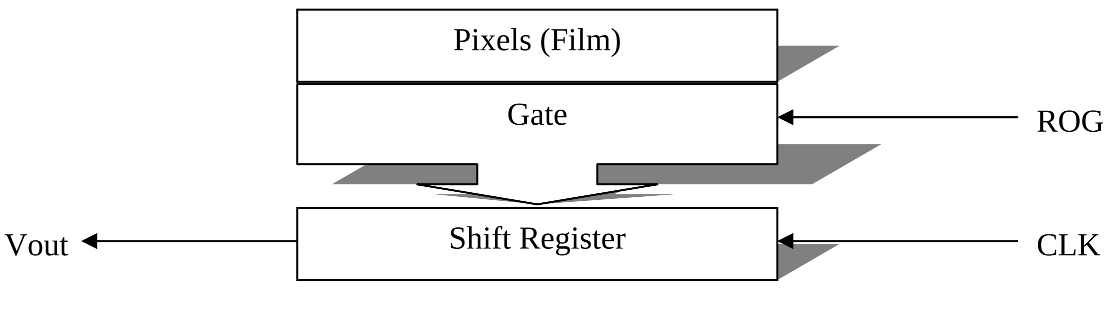


Figure 02: Illustration of how the ROG and CLK signals work together to put the appropriate pixel voltage values on the Vout line.

3.1.1 Sensor Signal Generation

Timer A2 is used to automatically generate the CLK signal since the this signal is connected to pin TA2out on the M16C. The ROG pulse is generated manually by pulling pin 7_2 low and then pulling pin 7_2 high again. Timer A0 is used to ensure that the timing for ROG is correct.

The ROG pulse is triggered every 271 CLK cycles, 3.8ms by default, just giving sufficient time for the CLK signals to shift out all the useful values of the real pixels for the M16C. The delay between the two ROG pulses can be changed from the GUI. The delay between the ROG pulses also acts as the integration or exposure time for the pixels.

Timer A2 toggles pin TA2out every 7µs giving a CLK period of 14µs. Timer A0 interrupts every 14µs and each time Timer A0 interrupts, a variable counter increments by one signifying that one period of CLK has just passed. This value is then used to determine if ROG or ADC should be done.

The time value 14µs is chosen to give enough time for the other subsystems in the M16C to work.

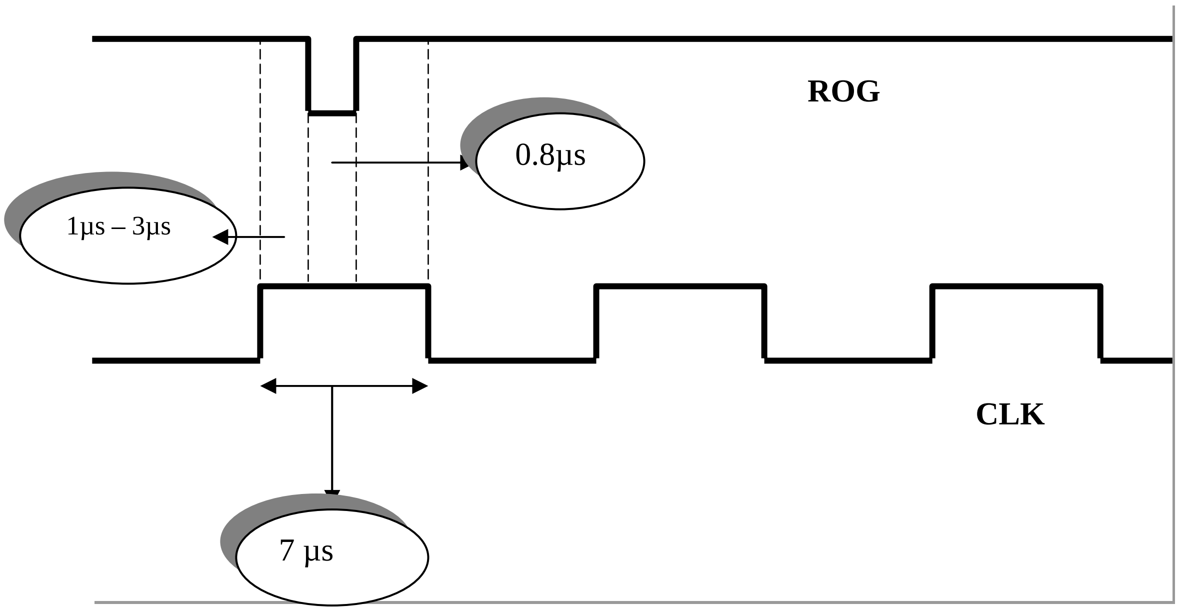


Figure 03: The synchronized timing diagram of the ROG and CLK signals.

3.1.2 Analog to Digital Conversion (ADC)

The V_{out} line from the sensor connects to pin AN1 on the M16C. Analog to digital conversion is thus performed on this pin, using the internal ADC subsystem in the M16C to obtain the digital values of the pixel. Pixels which have been exposed to much light, a low voltage will be present on the V_{out} line which will give a low digital value, and vice versa.

These digital pixel values are stored temporarily in a memory array while ADC has being done on the whole array of 256 pixels. When all the pixels of the array have been received they are sent one by one to the GUI via the UART0 subsystem. The array of pixels is also checked to determine which of these pixels holds the minimum value. The pixel that corresponds to the lowest input value is then passed through a distance lookup table, which displays the distance on the LEDs.

3.2 The Laser

The M16C outputs an 8-bit PWM signal to the laser in order to control the laser power. Timer 3A is used to automatically do this since pin TA3out is connected to the laser. The laser generating device used is iC-Haus's iC-WJ laser diode driver. The PWM duty cycle is set to 54% and the PWM period is set to 100µs by default as per the project specification. This duty cycle can also be changed through the GUI.

3.3 Displaying values on the LED

The LED on the MSA0654 board displays the distance of the laser reflector based on the input values obtained from the ADC. The pixel that corresponds with the lowest input value is passed through a distance lookup table, which is then displayed on the LED through Port 0.

The distance lookup table is defined initially as the hex value of the centre pixel.

3.4 Serial Communications with the GUI

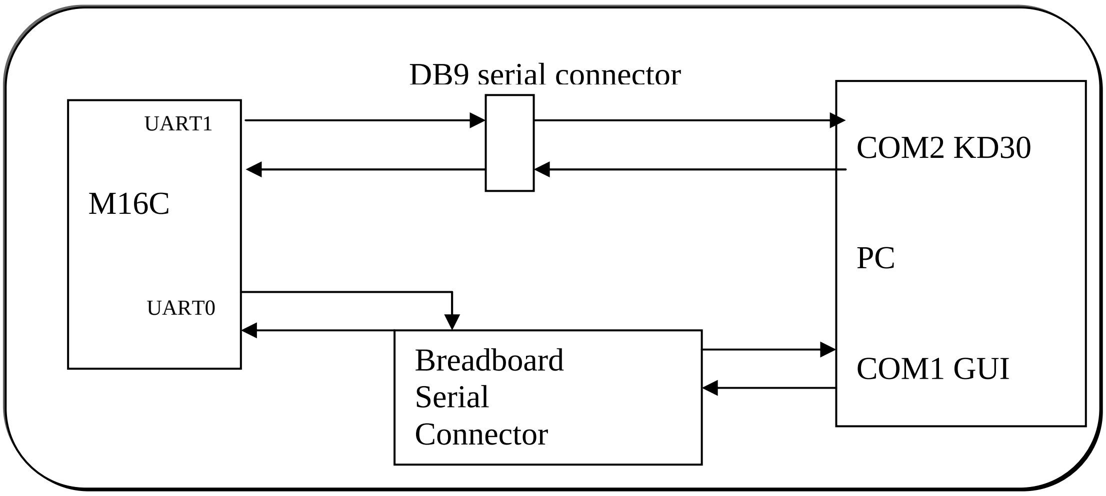


Figure 04: The GUI is communicating with the M16C through a specially built serial connector, connecting through the UART0 subsystem of the microcontroller instead of the ready-built DB9 connector on the MSA0654 board.

The UART0 subsystem of the M16C is used to send and receive data from the GUI on the PC. In order to do this, a new serial connector had to be built. The components of the second serial connector are built on a breadboard (Schematic of this is shown below in Figure 05). With prior permission from the Course Coordinator, Dr Stephan Hussmann, two wires were soldered onto the Adaptor Board. The UART0 subsystem is given the default settings of 9600 baud, no harware flow control, no software flow control, no parity and 1 stop bit.

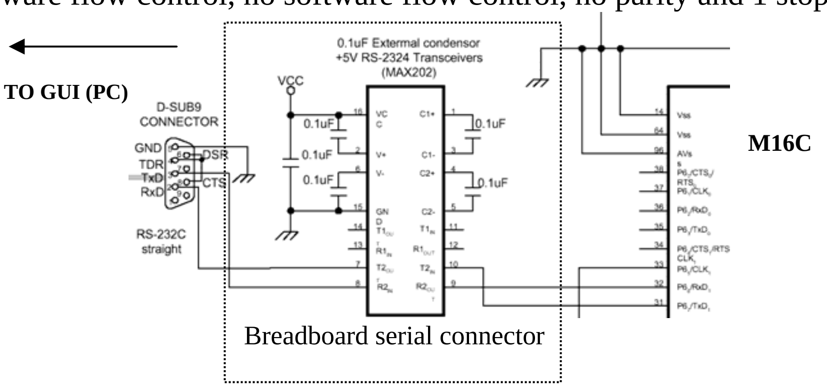


Figure 05: Schematic of the special purpose serial connector. (*source*: Flash Start manual.)

There is already one existing serial connector on the MSA0654 that can be used to communicate with the GUI. Using this existing serial connection, on the UART1 subsystem, proved to be a big hassle during the development of the final solution because of software development problems. A new serial connection completely bypasses those problems allowing the project to be developed smoothly and quickly. In addition, using a separate serial connector allows KD30 to debug the program at the same time the microcontroller is sending or receiving data.

3.4.1 UART Receiving Data

The microcontroller can receive the following commands from the GUI at any time:

axx : Change the laser PWM to xx in hexadecimal, 00 is 0% duty cycle, FF is 100% duty cycle.
bxxx : Change the integration time to xxx in hexadecimal, xxx is the number of clock cycles for each integration time.
cxxxxx... : Change the distance lookup table as specified by xxxxx.... (512 xs) (eg. c12121212 signifies 12 cm for the first four pixels)
d : Return the distance lookup table to its default values.

where the 'x's correspond to hexadecimal characters.

3.4.2 UART Transmitting Data

The microcontroller continously sends the most up-to-date pixel line data to the GUI. When transmitting data, ADC has to stop or else the line data that is being sent will be corrupted. The following data is continuously sent in sequence:

'*' : signifies that line data will be next
delay approximately 0.5s
"xxx...": the 256 bytes of line data sent as 512 hexadecimal characters
delay approximately 1.0s. This takes approximately 0.5s
According to these timings, one picture frame is sent to the GUI every 2 seconds. Originally when there were no delays incorporated into transmission, the GUI became flooded with too much data.

4.0 Development of the Graphical User Interface to communicate with the MSA0654 board to control the LTS

The GUI was developed using Microsoft Visual Basic and the final version of the project was compiled into a .exe file which could be run on any computer with a Microsoft Windows platform without any additional software. It meets all the specifications set out in the COMPSYS301AFC Project1 handout while incorporating additional user-friendly features. This section briefly describes how the main project specifications were implemented in the final design followed by a brief discussion on the additional features provided.

4.1 Plotting data in real time

Line data's sent from the M16C board every 2 seconds giving a graph refreshing speed of 0.5 frames per second. As described in section 3.4.2 the M16C board sends the character "*" first, and then the array of 512 characters with appropriate delays which are stored in a 256-element array in the GUI after getting the voltage values that correspond to the line data sent. These values are plotted in real time and the 'Data Table' updated. This table also highlights the pixel value that corresponds to the minimum voltage.

Figure 4.1 shows a snapshot of a graph plotted using a 10ms integration time at 53% laser power.

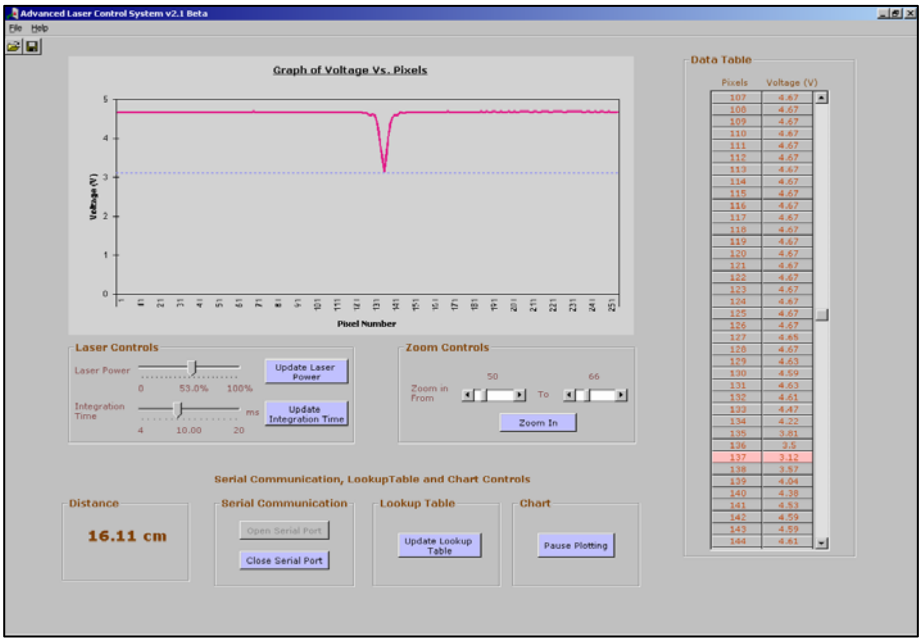


Figure 06: Snapshot of a graph plotted using a 10ms integration time at 53% laser power

4.2 Changing Pulse Width Modulation (PWM)

A slider is provided to change the laser power (or PWM) between 0% - 100% with a step size of 0.2. The unit step change in laser power is 100% / 256 which is approximately 0.39, therefore if a step size greater than this minimum value was chosen, getting all the values for laser power would not be possible. Hence a step size of 0.2% was decided on.

When the user wants to change laser power the percent value will be converted using the formula (entered value * 255 / 100) and this number will be transferred to the M16C board after converting to hexadecimal. These hexadecimal values will be taken care of by the microprocessor as described in section 3.2

4.3 Changing Integration Time

Selecting a value between 4ms – 20ms on the slider provided and pressing the "Update Integration Time" button allows the user to change the integration time from the GUI. The selected value is converted to the number of cycles using the formula:

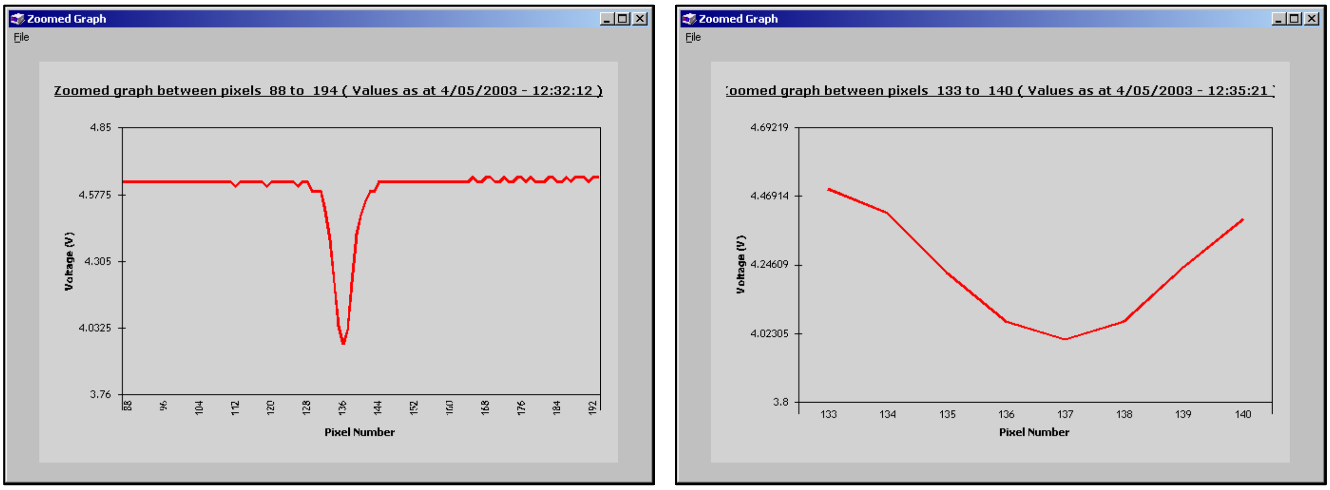
Number of cycles for required integration time = Entered value / 0.014ms.

The entered value has to be divided by 0.014ms since the minimum clock period allowed is 0.014ms.

This number of clock cycles calculated will then be converted to their corresponding hexadecimal values and then transferred to the M16C board. The way these hexadecimal numbers are used to change the integration time is explained in section 3.4.1

4.4 Zoom in function for better observation of the laser pulse

The user is allowed to select the range of pixels to be zoomed after opening the serial port and once the 'Zoom In' button is pressed a new window will show up with the zoomed in graph. As shown in Figure 4.2 this graph shows the required portion of the original graph with appropriate axis scaling.



Figures 07, 08: Zoomed in graphs with appropriate axis scaling.

4.5 Updating the default LookUp Table (LUT) and transferring values to the M16C board

The user is given the option to update the default lookup table by pressing the "Update Lookup Table" button. Values can either be entered manually or else by following the steps given below.

- First the distance has to be adjusted according to one of the values provided in the "Distance Lookup Table" window (Figure 4.4). The distance can be set up to any value between 12 – 22cm. The actual value can be determined by looking at the "Distance" frame in the main window.
- Then the button corresponding to the measuring distance has to be pressed which will automatically find the pixel that corresponds to the minimum voltage and enter the distance value in the appropriate pixel box in the "Distance Lookup Table" window.
- The user can do this for all the 11 value given and then adjust the values manually if necessary. As soon as all the buttons are pressed the table will automatically get updated using a point-to-point average algorithm (For more details on this algorithm please refer to Appendix III). This updated lookup table can then be downloaded to the M16C board by pressing the "Transfer Table to M16C" button.
- Pressing "Software Default" button allows the user to go back to the default lookup table which has been created using the following algorithm.
    - Total number of points is 11 (12 - 22 cm).
    - Therefore a suitable range has to be selected that would reasonably cover the whole range of pixels, thus the selected range was from pixel 40 - 240 with a step size of 20.
    - The initial position at 22cm is assumed to correspond to pixel number 240.
    - Rest of the positions in the "Distance Lookup Table" are updated by calculating the difference in adjacent distances and then dividing it by 20.

For example if pixel 240 had distance 22cm and pixel 220 had distance 21cm, then pixels between 220 and 240 will be updated using the calculation:

(22-21)cm / (240-220)

- Pressing the "Hardware Default" button makes the distance values displayed on the LED's of the M16C board appear in hexadecimal values which is the default displaying mode of the board.

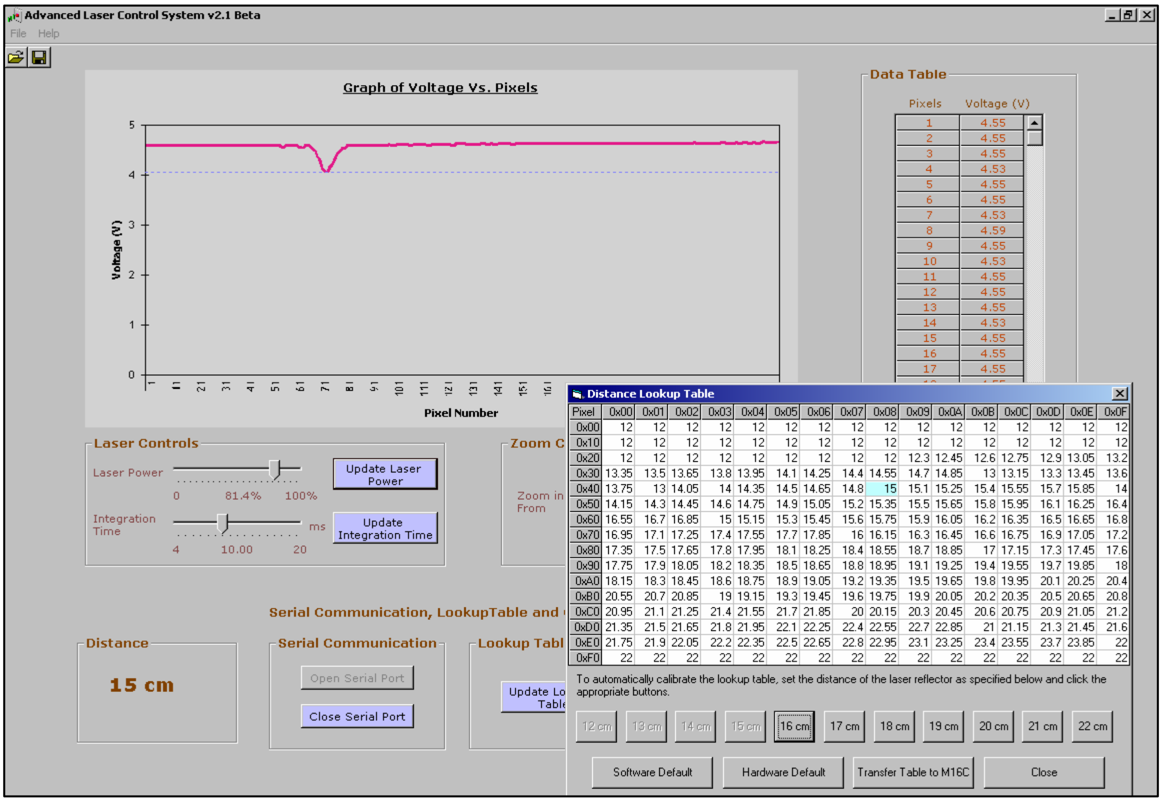


Figure 09: "Distance Lookup Table" window that enables the user to change the default lookup table

4.6 Displaying distance

The distance value is updated every time the graph refreshes. This is done by finding the pixel value that corresponds to the minimum voltage and then looking up the distance value that corresponds to that particular pixel from the LUT. The LUT can also be updated through the GUI, thus the distance will always follow the current values entered in the LUT.

Apart from these main project specifications, the following additional features were also provided in the graphical user interface.

4.7 Additional features incorporated in the GUI

4.7.1 Splash screen at startup

The splash screen (Figure 4.5) appears at startup and displays the name of the software, version of the software, developers' names and the institute the product is licensed to (The University of Auckland). It also includes a progress bar which shows the progress of opening the application.

Figure 10: Splash screen at startup

4.7.2 Letting the user save and open plotted graphs with the plotted date and time

The "Save" feature allows the user to save the original graph or a zoomed graph with the plotted date and time in .bmp format. The "Open" feature lets the user open an already saved graph. These options can be made use of by using short-cut keys, toolbar buttons or by clicking on "File" menu provided.

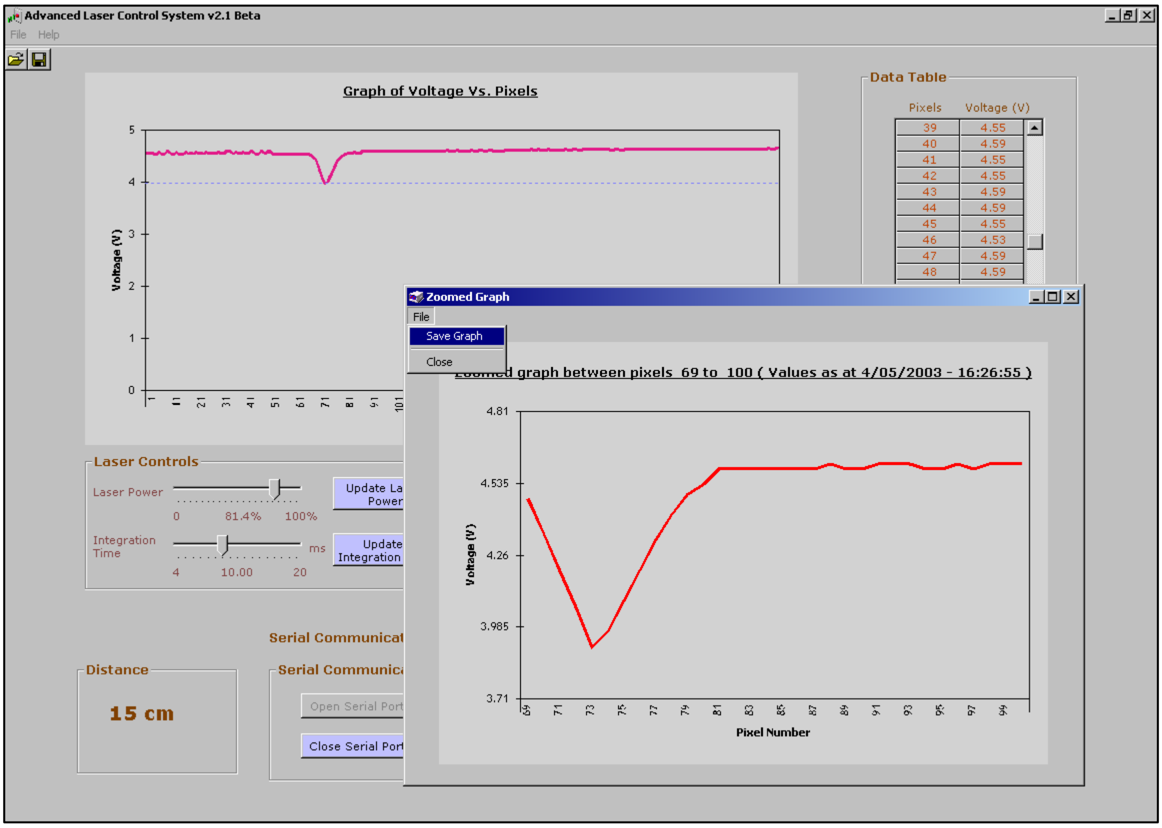


Figure 11: Implementation of "Save" and "Open" features

4.7.3 Link to the Project Report on the University Internet

This report is included on the University Internet at http://www.ele.auckland.ac.nz/~hussmann/LTSED3CSfinal.doc and can be accessed from anywhere in the world, provided that the computer has access to the internet. Thus product documentation and coding does not need to be separately included with the released software.

4.7.4 Pause Plotting / Resume Plotting button allowing the user to pause continuous plotting and look at the actual values

This option allows the user to pause or resume continuous plotting. When plotting is paused the user has the option of looking at the exact voltage values the graph was plotted for in the 'Data Table'. Figure 4.7 shows a snapshot of this feature.

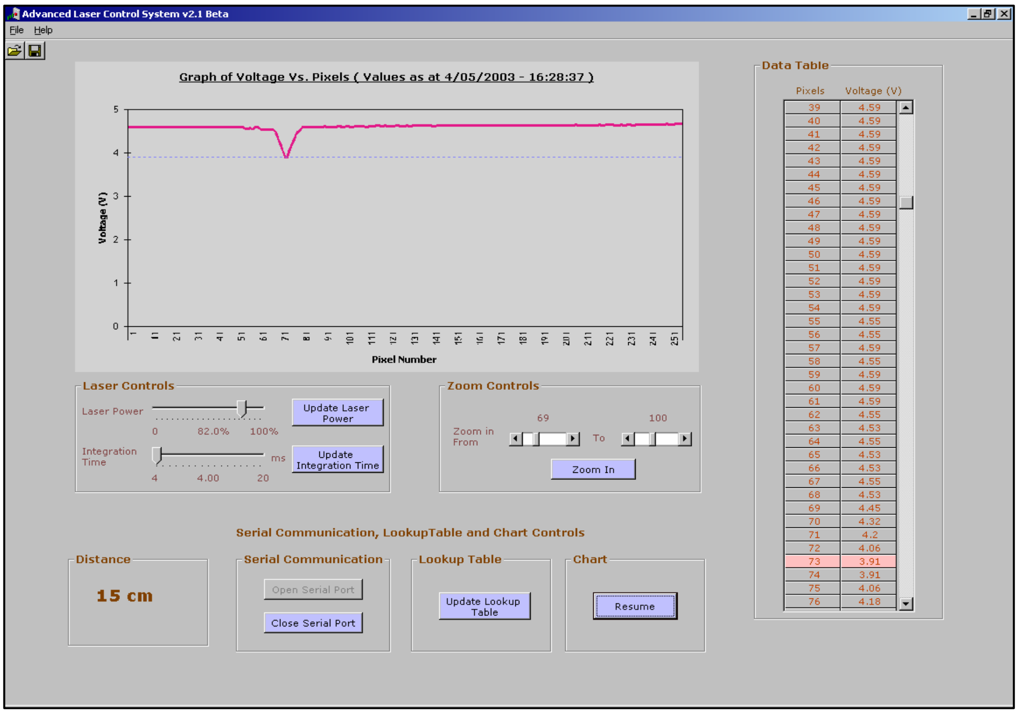


Figure 12: Pause Plotting / Resume Plotting option provided

5.0 Testing and Performance

Testing of the shape of the output signals was carried out using the oscilloscope by connecting the input probes to the appropriate pins on the M16C Board. Signals on the oscilloscope were then observed, analysed and compared with the project specifications and this process was repeated until the required shapes were obtained.

Testing the serial communication between the M16C and the GUI required the ingenious use of a handmade connection between DB9 connectors of the PC, effectively creating a loopback connection between COM1 and COM2. From this testing phase, a connection problem in the MSComm ActiveX Control was found to be handling control characters improperly which was the root cause of a very elusive serial communication problem.

6.0 Record of number of hours contributed by each group member itemised by task and the calculated cost for the whole project.

No.	Task Name	Each members' working hours			
		Yen	Thusitha	Kasun	Jimmy
	(a) Hardware :				
1	Research	13	10	8	9
2	Programming of the Clock Signal	16	14	11	13
3	Programming of the Read Out Gate	17	13	13	12
4	Programming of the Pulse Width Modulation	4	1	2	3
5	Send and receive data from and to the M16C (UART)	21	17	16	13
6	Programming of the Led Display	8	4	4	4
7	Programming of the Analog to Digital Conversion	6	4	3	2
8	Development of the laser pulse detection algorithm	2	2	2	2
	(b) Software (GUI) :				
1	Research on Visual Basic material	4	8	9	5
2	Designing the GUI layout	2	4	4	4
3	Plotting on MsChart using random values	4	7	6	5
4	Designing the zoom function	3	4	4	3
5	Designing the "Data Table" to display pixels and voltages	2	3	4	4
6	Getting line data through the serial line	3	4	2	3
7	Plotting and updating "Data Table" in real time	5	4	3	4
8	Transmitting data to M16C board (laser power and integration time)	1	1	1	1
9	Finding minimum voltage values and plotting statistical minimum	2	3	3	3
10	Optional features				
	- Splash Screen	0.5	1	1	0
	- Open and Save	1	2	2	1
	- Pause / Resume Plotting	0	2	0	1
	- Link to online documentation	1	2	3	0.5
	- Designing menus	0	2	2	1
11	Updating default lookup table and transmitting data	3	2	1	2
12	Debugging and implementing error handling capabilities	9	12	12	8
	(c) General:				
	Writing project report	6	6	6	6

Total number of hours spent	133.5	132	122	109.5
Cost per hour for each employee	$30.00			
Total labour cost (Hours x Cost per hour)	$4,005.00	$3,960.00	$3,660.00	$3,285.00

Instrument and equipment costs :

Oscilloscope	$25.00
DC Power supply	$20.00
M16C Board	$42.00

Serial Connector :

Bread board	$40.00
MAX 232 chip	$20.00

DB 9 Female connector head	$5.00
0.1 uF Capacitors (X 5)	$0.50
Wire crimps	$0.10
Heat shrinks	$0.10

Total actual costs incurred for the whole project = $15,062.70

7.0 Merits and Demerits

7.1 Merits

- The M16C microcontroller continuously sends the pixel line data, allowing for real time display of the line data.
- The M16C program is written in C language and has been optimised for performance.
- The GUI is a stand-alone executable and can be run on any computer with a Microsoft Windows platform. Many other GUIs are dependant on applications such as MATLAB and Microsoft Excel to run.
- The GUI is a user-friendly and logically designed application which focuses on stability and functionality instead of mere appearance.
- Multiple features accompany the GUI allowing a simple and detailed analysis of the line data.

7.2 Demerits

- The GUI is written in Visual Basic, which is too slow for the M16C microcontroller. Therefore data transmission had to be slowed down to one frame of pixels every 2 seconds.
- The LED display on the MSA0654 board can only display 2 digits which may not be accurate enough to display the distance. This deprives the software of displaying decimal values and any distance greater than 255 centimeters if displayed as hexadecimal and 99 centimeters if displayed in decimal.

8.0 Conclusions

The final solution described in this report is a solution to a real-world problem. For most of the developers in this project, it was the first time an application was developed that is useful in real life and had a monetary value. In addition, the highly professional manner in which this project was conducted created a brilliant working environment for the developers to enhance their skills and build their self confidence, trust within the group and team spirit.

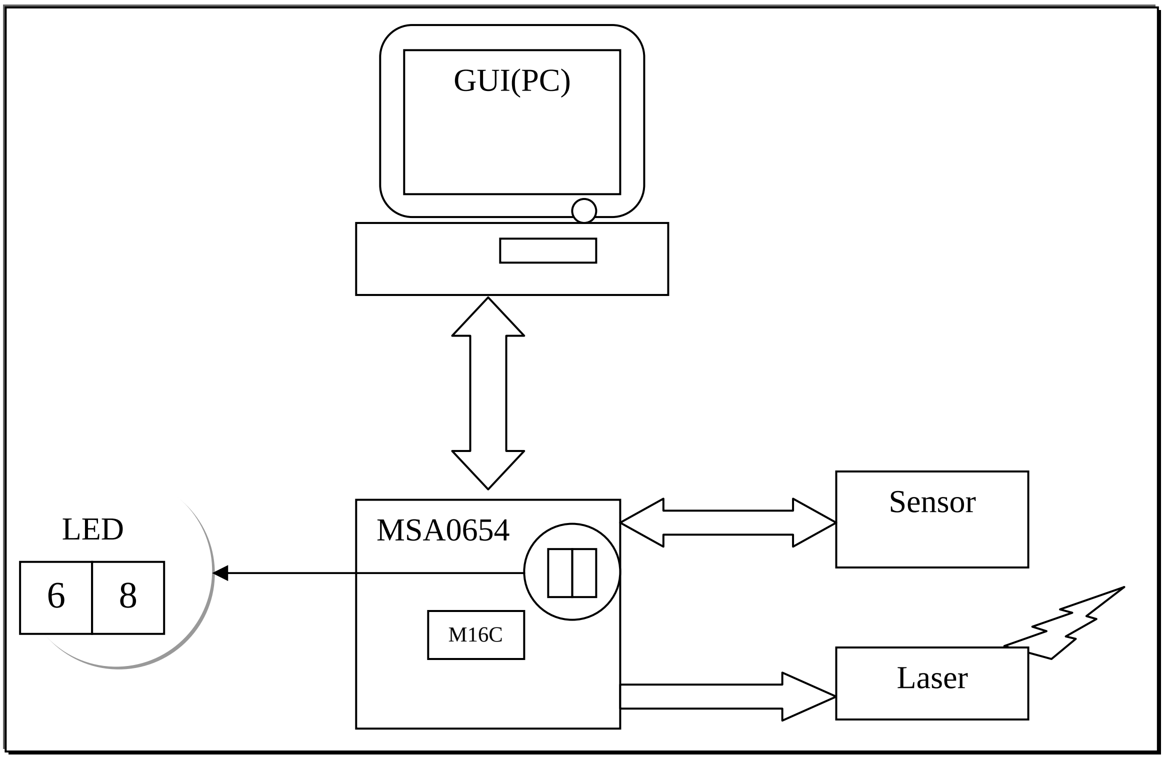
Using the M16C/62 Group microcontroller in this project was quite unusual since it is a too powerful and an expensive processor for the purpose it was used. Perhaps it was to get the developers familiar with working with such powerful microcontrollers although today’s microcontrollers will be weak and obsolete in future years.

This project required a lot of research in order to understand the whole system, proper planning and time management to make time to do the research work involved and a considerable amount of teamwork to execute the project plan successfully. A single line of code could not and was not written down before the whole system was discussed and understood.

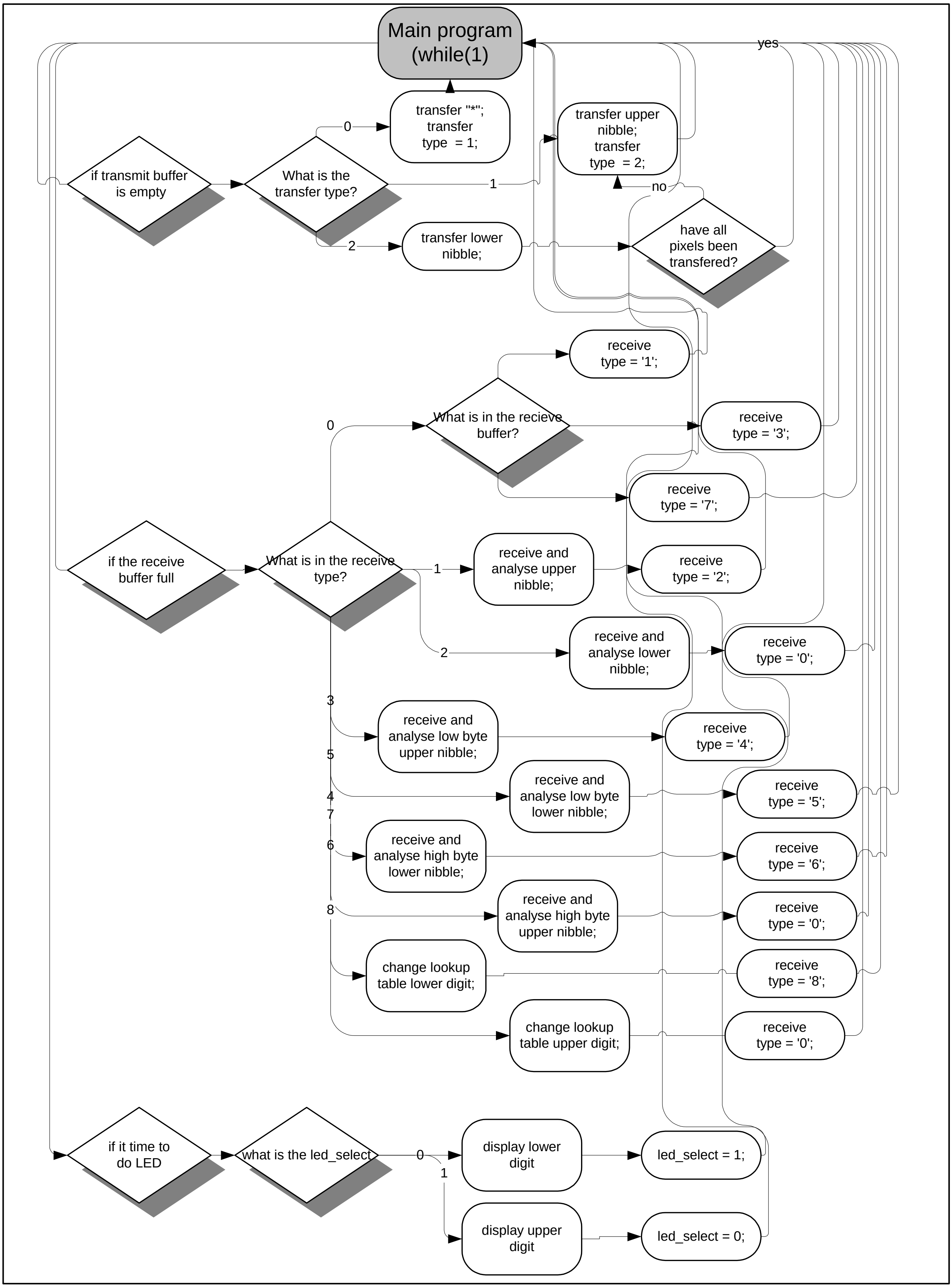
References :

1. Schneider, D. (1999) *An Introduction to Programming Using Visual Basic*. Prentice Hall, New Jersey.
2. *Mscomm control and Visual BASIC*. Retrieved April 25, 2003, from http://www.ontrak.net/visual.htm
3. *Serial Communication Components at Shareware Junction : Free , Beta , and Shareware Downloads !*. Retrieved April 25, 2003, from http://www.sharewarejunction.com/info.asp?ProductID=6733
4. Serial/Parallel communication in Visual Basic. Retrieved April 26, 2003, from http://members.aol.com/easyio/easyins3.html
5. VB2theMax:. Retrieved April 26, 2003, from http://www.vb2themax.com/HtmlDoc.asp?Table=Books&ID=4600
6. Visual Basic and the Basic Stamp. Retrieved April 26, 2003, from http://www.picobay.com/VB.html
7. visual basic vb C# serial communications: rs232, rs422, rs485 training tutorials example source code. Retrieved April 26, 2003, from http://www.htservices.com/Tools/VBandC/SerialCommunications.htm
8. STATISTICA Visual Basic Programs. Retrieved April 28, 2003 from http://www.statsoftinc.com/svbprograms.html#graph
9. Visual Basic example of drawing pie and bar charts and line graphs with csXGraph.Retrieved April 20, 2003 from http://www.chestysoft.com/xgraph/demo1.asp

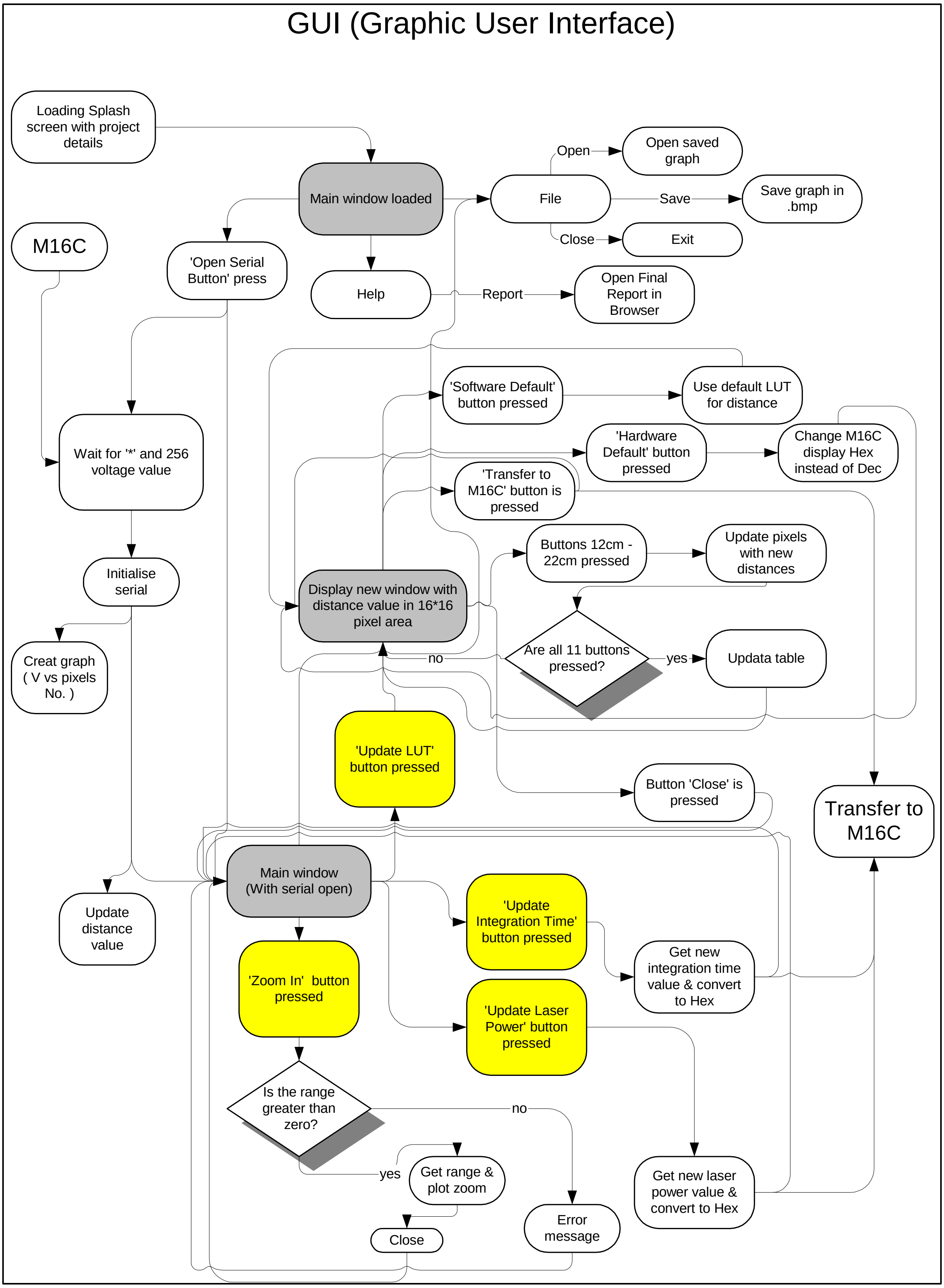
Appendix I : An overview of the components interacting with the M16C microcontroller



Appendix II : Flowchart for the functionality of the M16C Board



Appendix III : Flowchart for the functionality of the GUI



Appendix IV : Project Schedule