2-Dimensional Finger Gesture Based Mobile Robot Control Using Touch Screen

O. Ejale, N.B. Siddique, and R. Seals

***Abstract*—**The purpose of this study was to present a reliable mean for human-computer interfacing based on finger gestures made in two dimensions, which could be interpreted and adequately used in controlling a remote robot's movement. The gestures were captured and interpreted using an algorithm based on trigonometric functions, in calculating the angular displacement from one point of touch to another as the user's finger moved within a time interval; thereby allowing for pattern spotting of the captured gesture. In this paper the design and implementation of such a gesture based user interface was presented, utilizing the aforementioned algorithm. These techniques were then used to control a remote mobile robot's movement. A resistive touch screen was selected as the gesture sensor, then utilizing a programmed microcontroller to interpret them respectively.

***Keywords*—**2-Dimensional interface, finger gesture, mobile robot control, touch screen.

I. Introduction

OVER the years there have been a significant increase in intelligent computer based systems in everyday life; as a result the need for better ways of implementing human-computer interaction has also significantly increased. This was driven by the need to add more sophistication to everyday machinery being used; therefore it is of no surprise that the ability to easily control these equipments has a big role to play [1]. This has lead towards the need to create and implement user friendly interfaces on such machines, thereby making the machines easy and its usage intuitive. In order to create such intuitive interfaces, there is need to develop such interfaces to suit into the natural behaviour of the user [2]. This has given rise to the use of the gesture based interfacing techniques. In this paper was demonstrated how a two dimensional interface (resistive touch screen) was effectively used in the control of a mobile robot by interpreting finger movements made over it [7]. The technique used in analyzing the gesture was what was referred to as angular displacement method, which calculated the angle of displacement from one point of touch to another within a given time interval; as demonstrated by Fig. 1.

Several systems today are designed to utilize the intuitiveness of gesture based interfaces like smart phones, ATM, computers [11] and so on. After a gesture made had been captured, there must exist a way or algorithm for the gesture to be interpreted into meaningful command or instruction for human-robot interaction. A reliable way of interpreting gestures was a technique called pattern spotting [3]. In their work, Hyeon-Kyu L. and Jin-Hyung K made use of this technique; but that was after the gesture had been broken down into smaller bits of information using a process called segmentation [4]. The major problem with segmentation was the spatio-temporal variabilities; which has to do with the ability to effectively separate the gestures, and to compensate for the variations in inputted gestures [5]. In another work, a technique called time-varying-based spotting algorithm, was used in analyzing the gestures made towards a video camera [6]. In this work, the location, orientation and angular velocity were used in the analysis of the gesture being done.

II. Requirements

For the system to interpret the gestures, it had to be passed through the following processes:

A. Detect when the screen had been touched
B. Detect when the finger was in motion
C. Being capable of analyzing the direction of the moving finger.

A. Detecting a Touch

Since the touch screen was a resistive touch, which depended on the mechanical force exerted on it; the system was to be designed to sense the contact between the top and bottom layer of the touch screen within a time interval.

B. Detect when the Finger was in Motion

Since motion is a function of distance travelled and time [6], that is it has to do with the change of the position of an object from a reference point to another within a time interval (δt). The system was to be designed to check if there was a change in the position of the user's finger along both the X and Y axis (Cartesian coordinates), within a time interval (δt). This way it could detect if there was movement and the possible distance travelled.

C. Detecting the Direction of the Finger Movement

After a touch and the motion of the touch had been detected, the next step was to determine the direction the finger was moved. As earlier stated, the angular displacement

O. Ejale was in the University of Greenwich, Medway, ME4 4TB UK. He is now with SBE Limited, Ashford, TN23 7SH, UK as Technical Development Engineer (phone: +44-7424548028, e-mail: mariatony.ejale@gmail.com).

N. B. Siddique was in the University of Greenwich, Medway, ME4 4TB UK. He is now with SBE Limited, Ashford, TN23 7SH, UK as Technical Development Engineer (phone: +44-7411830818, e-mail: nafiz_siddique@ymail.com).

R. Seals is with the University of Greenwich, Medway, ME4 4TB UK. (phone: +44 (0)1634883953, e-mail: r.seals@gre.ac.uk).

of a point of contact from a previous reference point of contact was used within a predefined time interval was used. The angle of displacement used was the angle in reference to the positive Y axis, following a clockwise movement round the centre point as shown in Fig. 2. This was because in analyzing the angle, a 360º degrees round the centre point was needed; and since the Y axis was used as the forward/backward movement for the system, this made it to have the 0º degrees mark if the sensed movement was along the positive Y axis only. So since the Y axis was 0º, a 90º to the right (a straight line along the X axis alone) was to be a right movement and 90º to the left (270º) would mean a left movement. As shown below in Fig. 1. below, based on the point readings taken along the X and Y axis, the system could calculate the distance and angle from point P(x,y) to Q(x,y) to R(x,y) (θ_1 and θ_2) respectively.

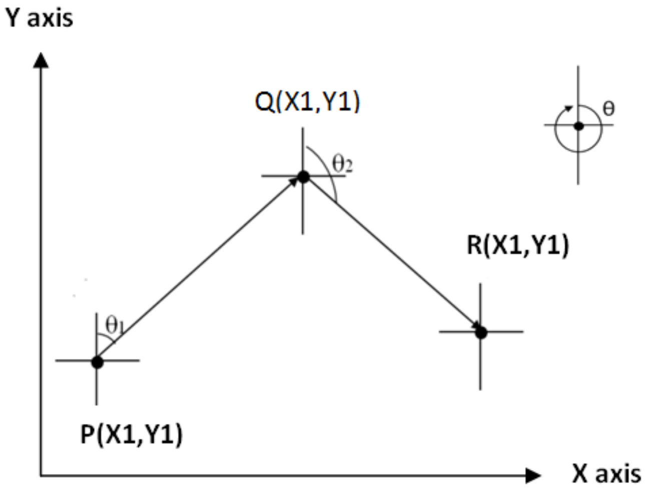


Fig. 1 Some points of reading and angles along the X and Y axis

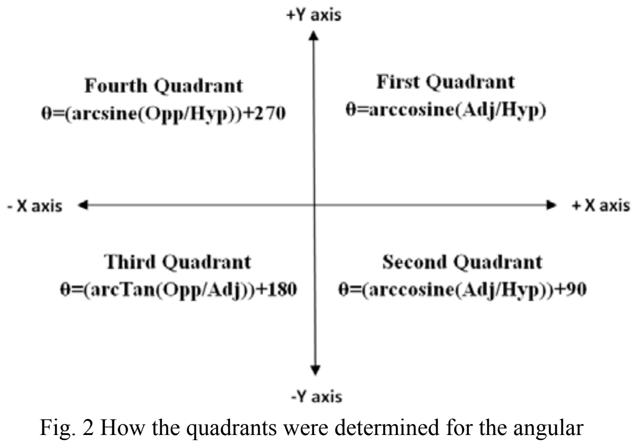


Fig. 2 How the quadrants were determined for the angular displacement analysis

By using Pythagoras's theorem and trigonometric functions (Fig. 1), the distance between points PQ and QR could be calculated and the angular displacements. For example to calculate the distance between points PQ using Pythagoras's theorem;

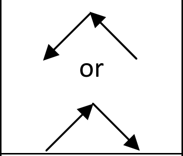
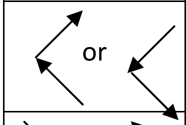
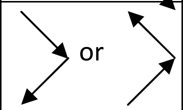
$$PQ = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2}$$

And the angle of displacement from point P to Q will be (since the movement was in the first quadrant (Fig. 2).

$$\theta 1 = \cos^{-1}\frac{(Y1 - Y2)}{PQ}$$

where, PQ was the calculated distance between points P and Q. The gestures the system was developed to interpret can be seen below:

TABLE I
SHOWING THE DIFFERENT GESTURES THAT WAS TO BE INTERPRETED BY THE SYSTEM FOR CONTROLS

Gesture	Code Outputted
↑	Fn: where "n" depends on distance moved by the finger
↗	Fn>: where "n" depends on distance moved by the finger
→	R
↘	Bn>: where "n" depends on distance moved by the finger
↓	Bn: where "n" depends on distance moved by the finger
↙	Bn<: where "n" depends on distance moved by the finger
←	L
↖	Fn<: where "n" depends on distance moved by the finger
	2
	1
	3

III. IMPLEMENTATION

In this project, we proposed a reliable means via which gestures made by the finger on a touch screen, can be adequately interpreted to control the movement of a mobile robot. The implementation of the project can be seen below:

A. Hardware

A block diagram of physical architecture of the system can be seen in Fig. 3. It consisted of the microcontroller, a resistive touch screen, a wireless digital link working on the

International Science Index Vol:6, No:12, 2012 waset.org/Publication/6451

433MHz range, and finally the mobile robot which had been built to be used in testing the system.

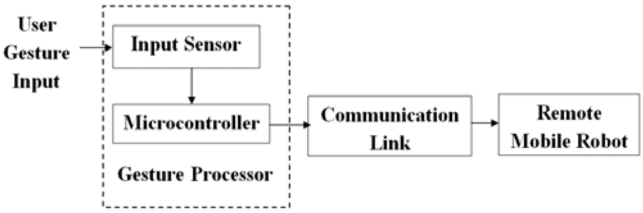


Fig. 3 System block diagram

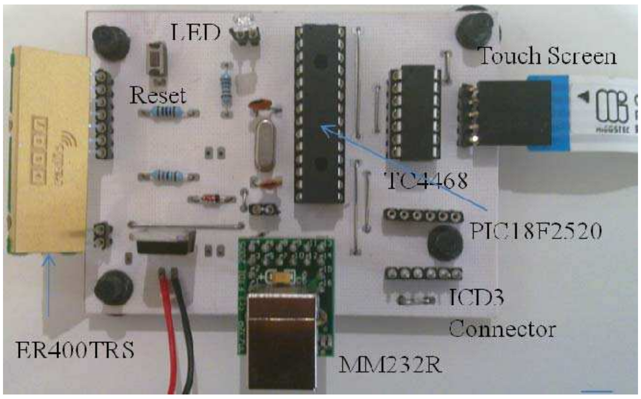


Fig. 4 Top view of the completed soldered PCB

B. Touch Screen

For this project, a five wire resistive touch screen which uses a mechanical contact to output an analogue voltage was selected. This was used because it was the most stable among the different types (4, 8 wire and so on), thereby making it less susceptible to temperature changes [7]. The position of the touch point was determined by the outputted analogue voltages from it; representing the position of the finger along the X and Y axis depending on configuration.

C. Micro Controller

A standalone Microchip® microcontroller (PIC18F2520) was used; it also had a lot of inbuilt peripherals needed to execute the project like the Analogue to Digital converter (ADC), power conservation mode, inbuilt oscillator (8MHz max), Universal Synchronous Asynchronous Receive Transmit (USART), Low voltage power capabilities, internal EEPROM (256 bytes) [8] and so on.

D. MOSFET Drivers

When the current drawn by the touch screen was measure independently with a supply voltage of 5 volts, it was noticed it drew a current more than what the PIC18F2520 microcontroller could supply which was limited to 25mA [8]. For this reason a MOSFET driver which could deliver enough current had to be used. The Microchip® TC4468 Quad AND MOSFET driver was selected. This was because it could supply the current needed by the system; for according to its data-sheet, it could supply a current of about 300mA from each of its pins, and 500mA from the total chip as a whole [8].

E. USB to Serial Converter

It was required that the serial protocol be used, for the microcontroller to communicate with the PC for testing purposes, and especially when giving calibration instructions to the user; but in most systems this port does not exist. Also on the other hand, it was desired that the converter be able to draw power from the USB host; for this will help in reducing the amount of power drawn from the system's independent power supply. The MM232R USB to serial converter was chosen to be used for the task [10].

F. Radio Frequency Transceiver Module

The mobile robot in Fig. 3.3 was to be controlled over a wireless link and therefore an RF transceiver module was needed to carry out the linking function. An RF transceiver is an electronic module capable of both receiving and transmitting. The ER400TRS was selected as the transceiver module to be used in setting up the link, as it could cover up to 10m in distance [9], and it operated within the licence-free frequency range of 433MHz. Lastly it could also support serial communication to and fro the microcontroller.

G. Mobile Robot

A pre-constructed mobile robot which could be controlled using the predefined codes and functions shown in Table I sent over the wireless channel was selected (Fig. 5). The mobile robot's direction and distance of movement could be controlled but not its speed. The mobile robot used can be seen below:

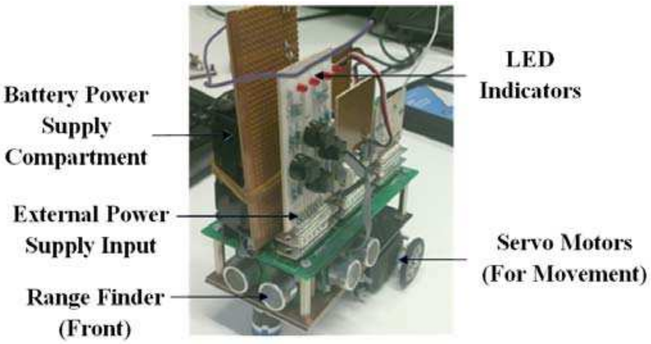


Fig. 5 The top view of the completed soldered PCB

H. Software

After the hardware had been developed, the software which was to be programmed into the microcontroller was then developed. The flowchart diagram of the system, upon which the software was built, can be seen in Fig. 6.

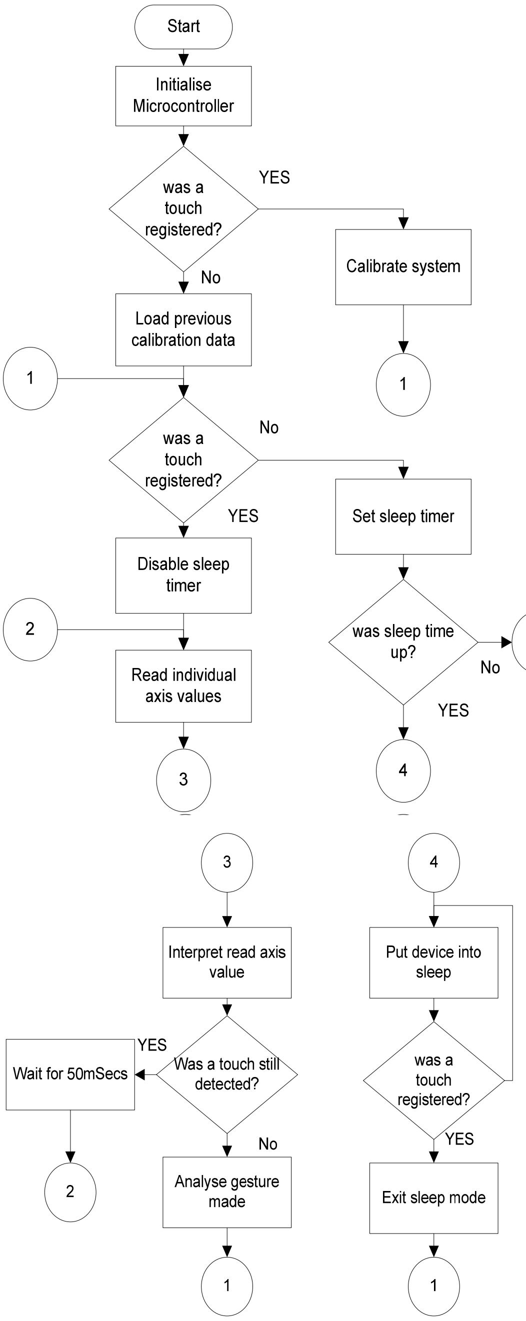


Fig. 6 The system’s flow chart for detecting and reading touch points

The software was developed using a mixture of both the C and assembler language, to aid in code efficiency; and the algorithm for the software that was implemented can be broken down into different tasks as outlined below:

A. Touch detection
B. Configuration of the Sampled Axis
C. Conversion of the Analogue Voltage into Digital
D. System Calibration
E. Reading of the Touch Point
F. Finger Motion and Gesture Analysis
G. Outputting of the Control Codes based on gestures

I. *Touch Detection*

This task was designed to be executed continuously, and was used to check if the touch screen was being touched by the user. Kit was a task which returned a binary value of either “1” or “0” representing it being touched or not respectively.

J. *Configuration of the Sampled Axis*

This task was used to configure the touch screen to take the required axis to be read; since the different axis (y and x) could only be read one at a time. A table can be seen below, showing the pin configuration used based on the touch screen’s datasheet for the X and Y axis value to be taken:

TABLE II
SHOWING HOW THE PINS WERE CONFIGURED TO READ THE AXIS

Pin\Axis	Y axis	X axis
Upper left	Low	Low
Lower left	High	Low
Upper right	Low	High
Lower right	High	High

K. *Conversion of the Analogue Voltage into Digital Values*

This was a task designed to be executed continuously as long as a touch was detected. It was used to check the point of contact along the X and Y axis. To achieve this; the pin used to sense the analogue voltage was converted to an analogue input then three readings of the inputted analogue voltage was taken and then added together. Finally the average of these readings was taken and the value stored.

L. *System Calibration*

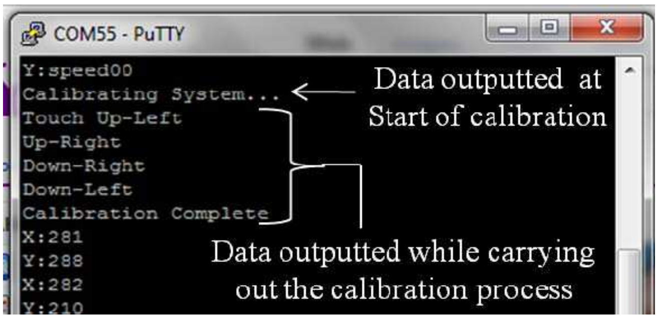
This was a task designed only to be executed if at start up or system reset, the touch screen was being touched by the user. It was used to calibrate the system to ensure that the touch points were properly read and analysed. It was done by first checking if the touch screen was being touched, and if touched give instructions to the user on the points to touch as shown in Fig. 3.5 below. It then calculated the “slope” of the touch screen's outputted data based on the linearity of the screen by using the formula shown below in Fig. 7. Next the data obtained was then stored in the system's EEPROM for usage.

$$\boldsymbol{X\ axis\ slope} = \frac{\boldsymbol{Xmax-Xmin}}{\boldsymbol{Length\ of\ X\ axis\ (obtained\ from\ the\ touch\ screen)}}$$

$$\boldsymbol{X\ axis\ slope} = \frac{\boldsymbol{Xmax-Xmin}}{\boldsymbol{Length\ of\ Y\ axis\ (obtained\ from\ the\ touch\ screen)}}$$

Fig. 7 Formula used for calculating slope of the touch screen’s outputted data

International Science Index Vol:6, No:12, 2012 waset.org/Publication/6451


COM55 - PuTTY
Data outputted at Start of calibration
Data outputted while carrying out the calibration process

```
Y:speed00
Calibrating System...
Touch Up-Left
Up-Right
Down-Right
Down-Left
Calibration Complete
X:281
Y:288
X:282
Y:210
```

Fig. 8 A screen dump, showing the data outputted by the system during calibration

M. Reading of the Touch Point

This task was responsible for analyzing the touch point, and then outputting the touch point in millimetres which was to be used for the touch analysis. Once a touch was detected the system then read the analogue voltage sensed on each of the axis depending on the one configured to be read.

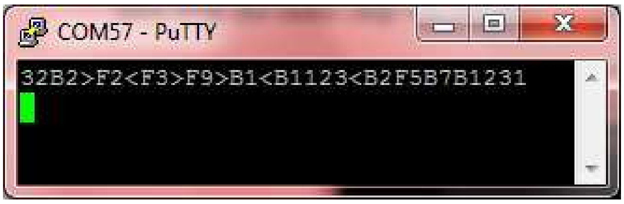
N. Finger Motion and Gesture Analysis

This task was used to detect when a motion was made on the touch screen. To compensate for any possible errors, the system ensured a minimum distance was moved along the X or Y axis before the motion analysis was done. In other to recognise if a motion had been properly made across the touch screen, the system was designed to take samples every 50mSecs; then based on the read samples, the system could determine the magnitude of the motion. Next the system analysed the gestures made on it using the displacement algorithm method (as outlined above) coupled with pattern spotting algorithm.

The pattern spotting algorithm worked by analyzing the "shape" of the obtained gesture based on the various angles of the and then

O. Outputting of the Control Codes based on Gestures

After the gestures have been analyzed, the next step was to send out the required code as specified in the Table I. A look-up table was used to check what code was to be sent out, for a particular gesture. The look-up table was designed using arrays which have the ASCII representatives, and addressing them to be sent out when needed. A screen shot can be seen below, showing the codes that can be sent over the serial port when the gestures were being interpreted in Fig. 8.

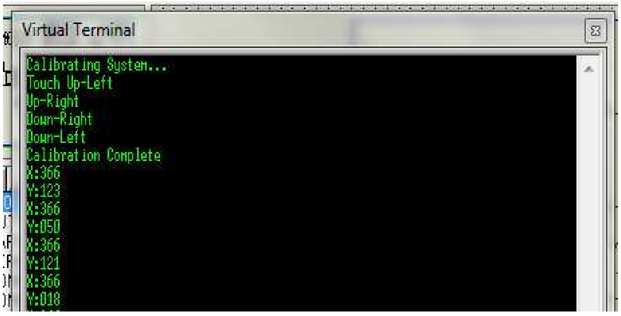


```
32B2>F2<F3>F9>B1<B1123<B2F5B7B1231
```

Fig. 9 A screen dump showing the data sent to the Robot

IV. Results

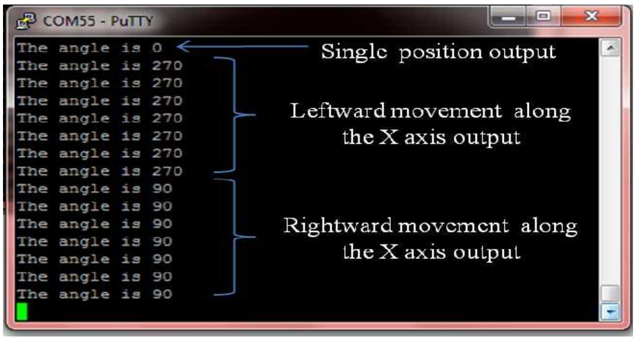
This section of the paper presents the obtained results of the project after it was completed. The various gestures as outlined in Table I was tested and the output were studied to check if the right codes were transmitted. The results from the ISIS simulation of the hardware design can be seen below via a screen dump in Fig. 10. A test code written to output data via the virtual terminals depending on the inputted signal was used in the test.



```
Calibrating System...
Touch Up-Left
Up-Right
Down-Right
Down-Left
Calibration Complete
X:366
Y:123
X:366
Y:050
X:366
Y:121
X:366
Y:018
```

Fig. 10 A screen dump, showing the virtual terminal output in ISIS

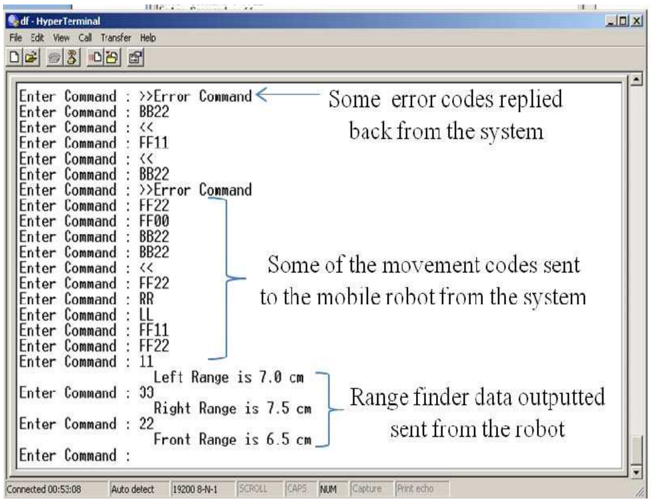
The system was tested to be able to properly analyze the angle of movement of the user's finger across the touch screen from point to point and made to output its analyzed angle via the COM port. A screen dump below showing what was outputted when the user's finger was on a spot, moved left and right along the X axis in Fig. 11.


COM55 - PuTTY
Single position output
Leftward movement along the X axis output
Rightward movement along the X axis output

```
The angle is 0
The angle is 270
The angle is 270
The angle is 270
The angle is 270
The angle is 270
The angle is 270
The angle is 270
The angle is 90
The angle is 90
The angle is 90
The angle is 90
The angle is 90
The angle is 90
The angle is 90
```

Fig. 11 Screen dump showing some of outputted angular data via the COM port

A snapshot of the hyper terminal window, which was used to view the transmitted data between both the touch screen and the robot can be seen below in Fig. 12. The mobile robot was capable of sending over the air how close it was to the nearest object at its left, right and centre.


df - HyperTerminal
File Edit View Call Transfer Help
Some error codes replied back from the system
Some of the movement codes sent to the mobile robot from the system
Range finder data outputted sent from the robot
Connected 00:53:08
Auto detect
19200 8-N-1
SCROLL
CAPS
NUM
Capture
Print echo

```
Enter Command : >>Error Command
Enter Command : BB22
Enter Command : <<
Enter Command : FF11
Enter Command : <<
Enter Command : BB22
Enter Command : >>Error Command
Enter Command : FF22
Enter Command : FF00
Enter Command : BB22
Enter Command : BB22
Enter Command : <<
Enter Command : FF22
Enter Command : RR
Enter Command : LL
Enter Command : FF11
Enter Command : FF22
Enter Command : 11
                  Left Range is 7.0 cm
Enter Command : 33
                  Right Range is 7.5 cm
Enter Command : 22
                  Front Range is 6.5 cm
Enter Command :
```

Fig. 12 A screen dump showing the data sent to the Robot, and the one received from it

V. Conclusion

A two dimensional touch interface, which is capable of controlling a remote mobile robot based on finger gestures made over the interface, was developed in this paper. The use of a resistive touch screen made it a two dimensional system; then by using the angular displacement method the system could adequately determine the movement of the finger within a predefined time interval. The combination of the angular displacement technique used in capturing then pattern spotting for analyzing the gesture was well suited for the system. The intuitiveness of the control system by using gestures was tested across 20 different users, with 95% of them learning to control the robot in less than 30 seconds. This method allowed for an effective and efficient way of interpreting the gestures, using them in controlling a mobile robot.

Acknowledgment

The authors express thanks to their parents whose continuous support, inspiration and love facilitated them reaching this stage successfully. O. Ejale and N. B. Siddique are intensely grateful to their supervisor Dr. Richard Seals from the University of Greenwich whose proper guidance, suggestion and inspiration in time made them accomplish this project successfully. They would also like to have the privilege of giving their heartiest thanks to all the helpful lab technicians specially Mr. Dodson and all the staffs of School of Engineering of the University of Greenwich for their kind cooperation. N.B. Siddique would also like to thank his beloved fiancée Dr. Fariha Hussain for performing a careful proof reading of this paper.

References

[1] K. Hinckley, *Input Technologies and Techniques*, in Handbook of Human-Computer Interaction, ed. By A. Sears and J. Jacko., L. Erlbaum Associates Inc., USA, 2003, pp. 1-11. ISBN:0-8058-3838-4

[2] Hudlicka, E., To feel or not to feel: The role of affect in human–computer interaction, *International Journal of Human-Computer Studies*, 2003, vol. 59 (1-2), pp. 1-32. ISSN: 10715819.

[3] Hong-Ki, K., Seunggun, L., Kwang-Seok Y., Capacitive tactile sensor array for touch screen application, *Sensors and Actuators A: Physical*, January 2011, vol. 165 (1), pp. 2-7. ISSN: 0924-4247

[4] Pisacreta, R., Rilling, M., Infrared touch technology as a response detector in animal research, *Behaviour Research Methods*, July 1987, vol. 19 (4), pp. 389-396. ISSN: 1554-351X

[5] Hyeon-Kyu L., Jin-Hyung K., Gesture spotting from continuous hand motion, *Pattern Recognition Letters*, April 1998, vol. 19 (5-6), pp. 513-520. ISSN: 0167-8655

[6] Ho-Sub Y. Et al, Hand gesture recognition using combined features of location, angle and velocity, *Pattern Recognition*, 2001, vol. 34 (7), pp. 1491-1501. ISSN 0031-3203

[7] EloTouch, *AccuTouch Five-Wire Resistive Touchscreens*, 2011, [online]. Available from http://www.elotouch.com/Products/Touchscreens/ AccuTouch /default.asp (accessed 1st August 2011)

[8] Microchip®, *PIC18F2420/2520/4420/4520 Data Sheet (DS39631E)*, Microchip, 2008, pg. 1

[9] ShreharSha R., *Estimating the ZigBee transmission-range ISM band*, May 4 2007, [online]. Available from http://www.edn.com/ contents/images/6 442439.pdf (accessed 20th August 2011)

[10] Future Technology Devices Ltd, *MM232R USB - Serial UART Development Module Datasheet Doc. Ref. No. FT_000214*, 2010.

[11] Nichols, S., New Interfaces at the Touch of a Fingertip, *Computer*, Aug. 2007, vol.40 (8), pp.12-15. ISSN: 0018-9162