

Effortless Access: Resistive Touch based Wireless Mouse and Keyboard using Arduino

Budhaditya Bhattacharyya
Professor,
School of Electronics Engineering
VIT University
Vellore, India

Vikash Patnaik
Student,
School of Electronics Engineering
VIT University
Vellore, India

Abstract—Wireless devices have become more affordable and popular in this era. Convenience and flexibility are two of the most important advantages of any wireless device over wired ones. In light of this, a prototype for effortless access through portable wireless mouse and keyboard has been conceived in this paper. Using a low cost resistive touch sensor, a mouse with basic operations and a keyboard with conventional functionalities have been implemented over an Arduino platform. The detailed design aspects and algorithmic development of the said prototype along with demonstration of a working model have also been presented.

Keywords— *Mice; Arduino; Xbee; Resistive Touch pad;*

I. INTRODUCTION

Wireless devices are becoming more and more popular today. Convenience and flexibility are two of the most important advantages of any wireless device over wired ones. The direct application of existing wired mouse is not adequately efficient in terms of mobility and disturbances caused by wired connection. In fact this has become the primary reason for the current surge of users, moving towards a viable world of wireless solutions. The main advantages of these wireless devices such as wireless keyboards, headphones and mice are that they can help reduce the clutter of wires around the working area. This can in turn increase the ease of work and fineness of the end result. With a wireless mouse and keyboard, the ease of operation is significantly higher than wired ones. This can even prove very useful while controlling computer from a distance.

The very basis of a “mouse device” involves a pointing device that works on a supporting surface by detecting two-dimensional motion relative to it [1]. Modern day wired mouse involves different features, that results in diverse range of mice such as *Mechanical Mice (wheels)*, *Optical and laser mice*, *Inertial and gyroscopic mice* [2], *3D mice*, *Tactile mice*, *Ergonomic mice*[3], *Gaming mice*[1]. These different mice allow user to perform various system-dependent operations. Even addition of extra buttons and features add more control or dimensional input. The mouse's motion typically translates into the motion of a pointer on a display, which allows for fine control of a graphical user interface [1]. However, each of the different type of mice as mentioned earlier does have respective limitations. In general, *Mechanical Mice (wheels)* suffer heavily due to mechanical wear and tear. Its tracking mechanisms become clogged with lint. The wheels' position is very much dependent on the surface where it is being operated. Even, the bulky size of this mouse makes it difficult to operate, for a longer duration of time. There is a considerable unpredictability of *Optical and*

laser mice in tracking its pointer on glossy and transparent surfaces. Even some mouse-pads, literally cause the cursor to drift unpredictably during operation [4]. In addition to these, there are issues with Mice having less image-processing power. As these mice have problems tracking fast movement compared to some high-quality mice which can track faster than 2 m/s (7km/h). Although *Inertial and gyroscopic mice* as shown in Fig. 1, does not require a surface to operate, inertial mice suffer heavily due to its “first generation” device model.



Fig. 1 A Typical Inertial and gyroscopic mice [2]



Fig. 2 A typical Tactile Mouse [5]

By this it is meant that few major design flaws make it less useful than it could be. While in mid-air, accidental triggering of the laser sensor at the bottom of the mouse, is at times very frustrating. Likewise, the overall shape of this type is a compromise i.e. it's neither the ideal shape for a mouse nor the ideal shape for a remote. Despite being a strong contender of “wireless hands free” device, the *3D mice* was finally discontinued because it did not provide sufficient resolution and precisions which is expected from such a costly device. *Tactile Mice* as shown in Fig. 2, on the other hand can augment user-interfaces such as giving feedback while crossing a window boundary [5]. Unfortunately though, it suffers from constant requirement of information feedback about the location and motions of the cursors on the screen.

For instance, an oblique motion path of the cursor is experienced for horizontal movement of a tactile mouse. Along with this there is a severe issue with “precision of the mouse” when it is lifted and placed at certain other location [5]. *Ergonomic mice* as shown in Fig. 3, provide optimum comfort. In addition avoidance to injuries such as carpal tunnel syndrome, arthritis and other repetitive strain injuries can be avoided by using the same [6]. The only practical problem with this type is the huge cost that one needs to bear compared to the conventional mice. Hence it can't be afforded by many.



Fig. 3 A typical Ergonomic Mouse [6]

Gaming mice have its share of acceptance but limited to pc gamers. With its wide array of controlling buttons, catchy design makes it differ radically from traditional mice. The above comparative discussion on various platforms leads us to the basis of this paper. Clearly it can be observed that all the mentioned mice provide pointing option with reasonable precision. Hence if one can build a prototype where both mouse and keyboard functionalities are indigenously built, it will indeed be a novel concept. Accordingly, in this paper, a successful attempt has been made to design and implement a “wireless hand-held mouse and keyboard” prototype that delivers effortless performance and seamless precisions, compared to that of a conventional mouse. The added benefit of a single touchpad used for keyboard along with mouse makes this work unique from every angle.

Radio Frequencies (RF) predominantly serve as the basis for wireless mouse technology. Similar to other radio technologies, this too requires a transmitter and a receiver [1]. The mouse transmits radio signals to a receiver, which is itself connected to the computer hardware, normally via a wire. This kind of wireless mouse is very reliable and is capable of transmitting the mouse's movements to the receiver from across a room. The flexibility offered by a wireless mouse is of course a huge advantage over wired ones. With the help of the hand held touch screen one can comfortably control mouse pointer and keyboard without worrying about wire being tangled. Following this mode of operation, the prototype in this paper describes the design of a mouse pointer and keyboard device with wireless control and resistive touch sensors. The device has the advantage that it is microcontroller based and thus can be programmed to work as a mouse along with a keyboard. Other advantages include the use it as an ‘air mouse’ as it does not depends on surfaces like conventional optical mouse. The paper further explores the opportunity involving a remote handheld keyboard which works in conjunction to the mouse [1][7]. Instead of a reflective surface [8] for optical mouse, the principle of resistive touch sensor has been taken into cognizance. In addition to it, the wireless function has been implemented by using a Xbee Module [9].

II. SYSTEM DESIGN

A. Implementation Details

The block diagram of the proposed resistive touch based wireless mouse and keyboard has been demonstrated in Fig. 3. In order to take the user input for both mouse as well as keyboard, a resistive touch sensor is used. A 4-wire resistor is connected to the microcontroller Arduino Uno [10], which is based on Atmega 328P. *Uno* receives the data from the touch sensor by means of alternating voltage (5V) and ground across the x-axis and y-axis of the resistive touch sensor [11-12] as shown in Fig. 5. The data is sent to the XBEE S2 which is a 2 mw 2.4 GHz RF device. The XBEE at receiver end receives the data serially and sends it to the Universal Asynchronous Receiver-Transmitter (UART) of Arduino Leonardo [13] *Leonardo* is then programmed to check on the user requirements for mouse or keyboard. The decoded information is then transmitted frame by frame and is extracted using the x and y coordinates. The coordinates control the mouse pointer or keyboard based on user's requirements. The functionality of mouse includes mouse pointing, left and right click, whereas the functionality of keyboard includes all alphabets, numerals and some special keys such as enter,spacebar, backspace and delete.

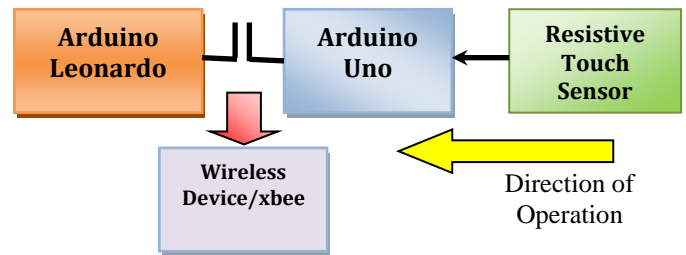


Fig. 4 Block Diagram for wireless mouse and keyboard prototype

B. Working Principle of a Resistive Touch Sensor

a) *Detecting a touch:* To know if the coordinate readings are valid, there must be a way to detect whether the screen is being touched or not. This can be done by applying a positive voltage (VCC) to Y+ through a pullup resistor and applying ground to X-. The pull-up resistor must be significantly larger than the total resistance of the touch screen, which is usually a few hundred ohms. When there is no touch, Y+ is pulled up to

$$Y = \frac{V_{x+}}{V_{Drive}} \times height_{screen} \quad X = \frac{V_{y+}}{V_{Drive}} \times width_{screen}$$

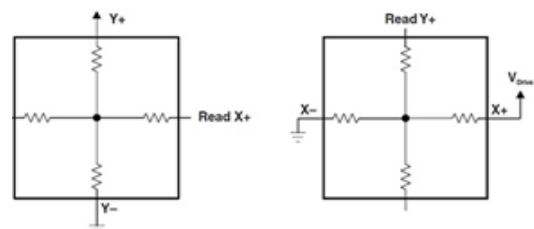


Fig. 5 Voltage Variation

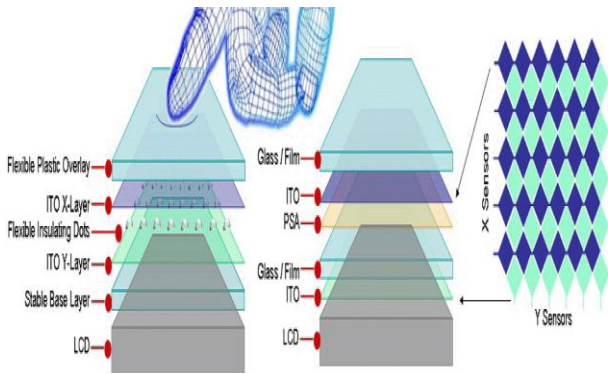


Fig. 6 Touch screen Technology [14]

the positive voltage. When there is a touch, Y+ is pulled down to ground. This voltage-level change can be used to generate a pin-change interrupt. A pictorial representation of touch screen technology has been well presented in Fig. 6

C. Detailed Design Specifications

a) Arduino Board

ArduinoUno [10] is a microcontroller board based on the ATmega 328. It has 14 digital input /output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. It only requires a simple connection to a computer via a USB cable or a power adapter or even a battery to get started. A typical arduino board schematic is represented in Fig. 7

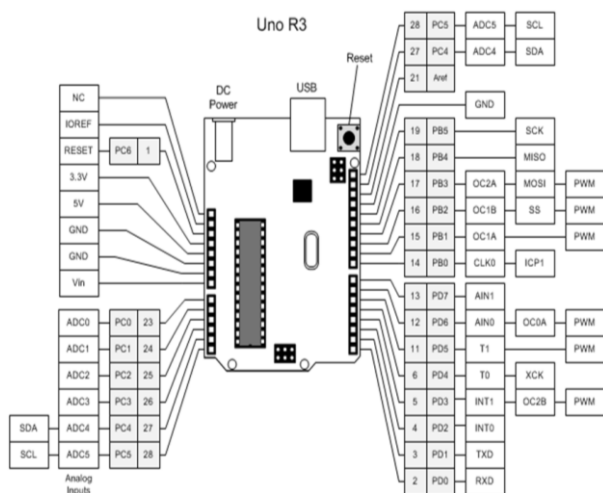


Fig. 7 Arduino schematic diagram [10]

b) Arduino Leonardo:

The Arduino Leonardo[13] is a microcontroller board based on the ATmega32u4. It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button. The Leonardo differs from all preceding boards in that the ATmega32u4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Leonardo to appear to a connected computer as a mouse and keyboard, in addition to a virtual (CDC) serial / COM port. It also has other implications for the behavior of the board.

c) Serial UART

An UART, universal asynchronous receiver / transmitter is responsible for performing the main task in serial communications with computers. The device changes incoming parallel information to serial data which can be sent on a communication line. It takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Serial transmission of digital information (bits) through a single wire or other medium is much more cost effective than parallel transmission through multiple wires. Accordingly, UART is used to convert the transmitted information between its sequential and parallel form at each end of the link. Each UART contains a shift register which is the fundamental method of conversion between serial and parallel forms [10].

d) Arduino Software IDE(Arduino 1.5.8)

The open source Arduino software (IDE) [15] makes it easy to write code and upload to the board. It runs on windows, Mac OS x, Linux. The environment is written in java and based on Processing and other open source software.

e) XBee (XBee S2)

XBee XB24-ZB [9][16] module from Digi. Series 2 improves on the power output and data protocol. Series 2 modules allow you to create complex mesh networks based on the XBee ZB ZigBee mesh firmware. These modules allow a very reliable and simple communication between microcontrollers, computers and connected systems. Point to point and multi-point networks are supported. These are essentially the same hardware as the older Series 2.5, but have updated firmware. A typical XBee XB24-ZB module is shown in Fig. 8, while the schematic is presented in Fig. 9.



Fig. 8 A typical XBee Module [9]

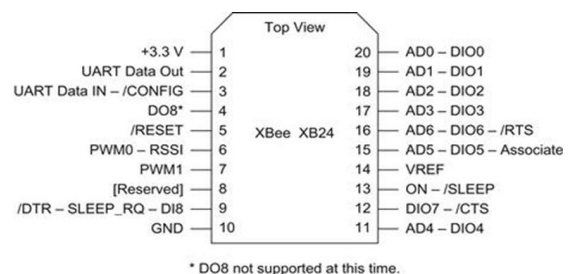


Fig. 9 A schematic of XBee Module [9]

III. IMPLEMENTATION OF THE PROTOTYPE

A. Wireless Mouse Transmitter

The implementation of Wireless Mouse Transmitter (Tx) is as follows:

- i. Four pins of Arduino Uno(A0,A1,A2,A3) are connected to the 4-wire resistive touch screen which are pulled down by 10 kilo ohm resistors so that any floating values are pulled down to ground. This circuit configuration will provide the location of touch on resistive screen.
- ii. To provide a left click and right click, a connection is made to the 5V supply on to a switch. This is then connected to a Arduino Uno pin 1 and pin 2, as both being digital pins. Following this, pulling down of pin 1 and pin 2 takes place using a via 10 kΩ resistors.
- iii. To toggle between mouse and keyboard we will have a flip switch (refer prototype demonstration section)
- iv. Connect the 5V and Ground pin of Arduino to Xbee and DIN(Data Input) of Xbee to Tx pin of Arduino as shown in Fig. 10.

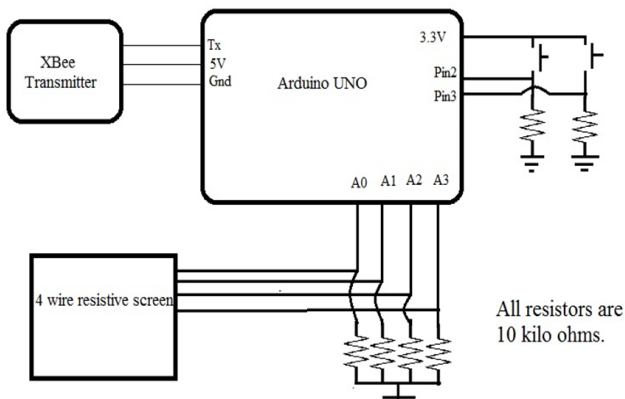


Fig. 10 Wireless mouse Tx

B. Wireless Mouse And Keyboard Transmitter (Arduino Algorithm)

The following algorithm describes the steps for implementing wireless mouse and keyboard transmitter over Arduino platform.

- i. Define four pins as A0,A1,A2,A3 in order to get input from resistive touch sensor.
- ii. Use Pin2, pin3, pin4 as digital pins to input mouse clicks and switch between mouse and keyboard.
- iii. Define four analog integer x1,y1,x2,y2 to get the data from resistive sensor, reading L, reading R for mouse clicks and reading M or K for switch between mouse and keyboard.

- iv. Define functions to get x value and y value which must be alternate as follows:

- First define A0, A2 as input in order to give 5V and ground.
- Define A1,A3 as output and get the point of touch coordinates and store the y1 output and return the value.
- Now define A1, A3 as input in order to give 5V and ground.
- Define A0, A2 as output and get the point of touch coordinate and store the x2 and return the value.

C. Mapping of KEYBOARD in transmitter:

In order to get keyboard using the same resistive touch pad, the following steps are considered:

- i. Define a variable of 'char' type and assign a constant char for a particular x and y range. As for example: 'a' is defined in the range x(94,170) and y(630,730)
- ii. Now check pin1 and pin2 and if any of them are high send a character R or L according to left or right click.
- iii. If pin 3 is high, send M (Mouse) or else send K (Keyboard).
- iv. Now arrange it in a particular format as given by : <L,R,x2,y1,M,alfa> where
 - L:left mouse click or right mouse click or no click
 - x2: x-coordinate
 - y1: y-coordinate
 - M: mouse or keyboard
 - Alfa: alphabet or numbers
- v. Open the serial port at 9600 baud rate and send the above format.

D. Wireless Mouse Receiver

Arduino Leonardo is configured to be used as a receiver. The following description and schematic shown in Fig. 11 demonstrates the implementation of the Mouse Receiver (Rx)

- i. In the receiver side connect the pin2 of Arduino Leonardo to 5V via a flipswitch while pulling down the pin2 via 10kΩ.
- ii. This pin2 is meant as kill switch for the Arduino.
- iii. Connect the 5V and ground to XBee and Rx pin of arduino to DOUT of XBee.

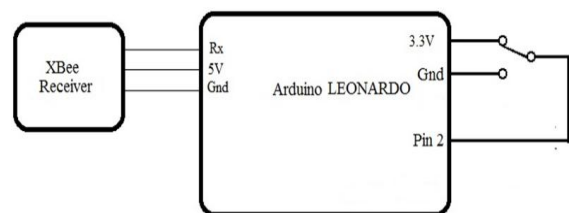


Fig. 11. Wireless Mouse receiver

E. Mouse And Keyboard Receiver(Arduino Algorithm)

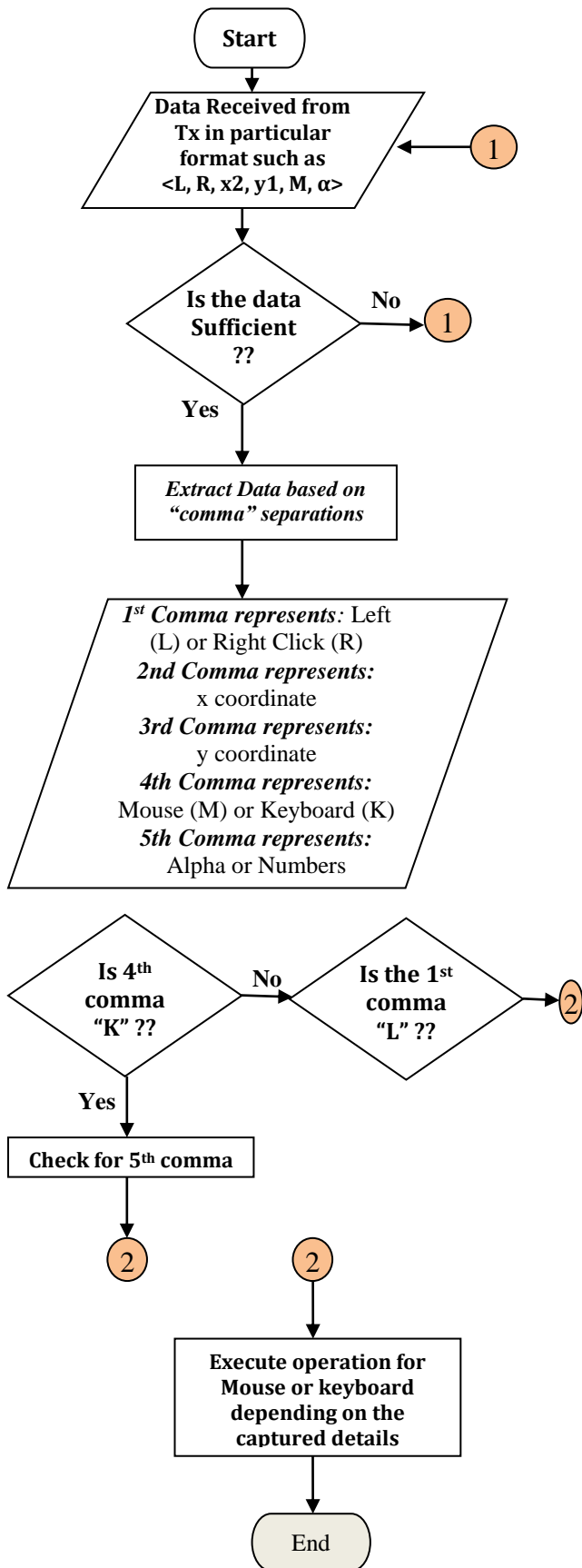


Fig. 12 Algorithm for selecting "Mouse" and Keyboard

IV. PROTOTYPE DEMONSTRATIONS

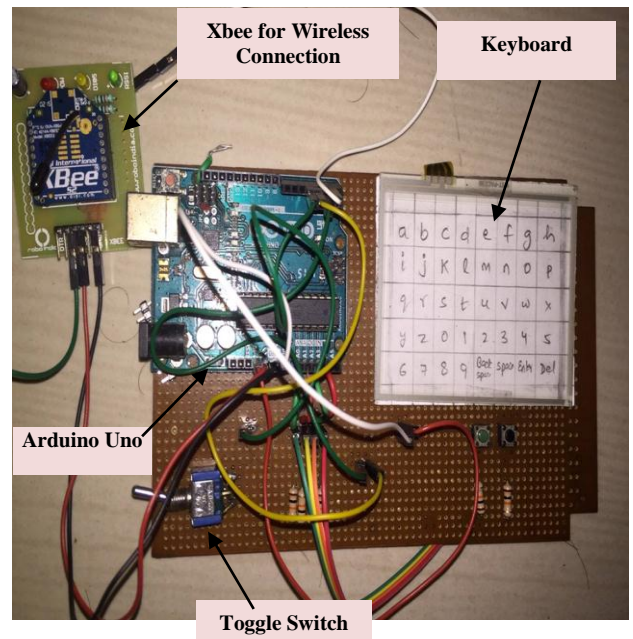


Fig. 13 Project prototype with Xbee Module (Keyboard)

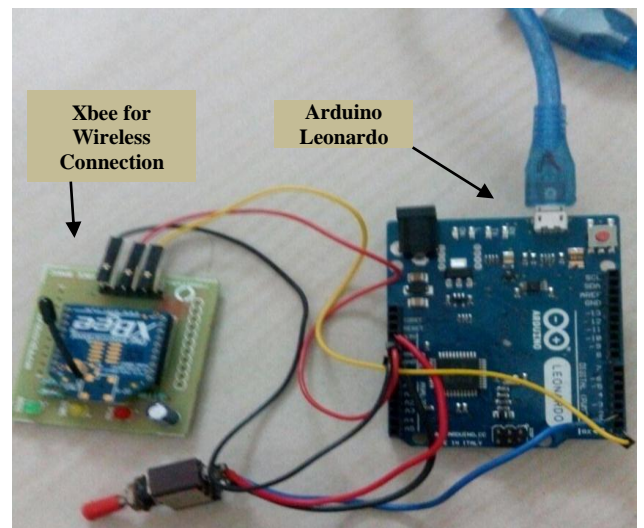


Fig. 14 Project prototype with Arduino Leonardo (receiver)

V. RESULTS AND DISCUSSION

Following the working algorithm of mouse and keyboard receiver in Fig. 12, the prototype demonstration as pointed out in both Fig. 13 and Fig. 14 gives an insight of the implementation details described earlier. By matching both the baud rate of Tx and Rx at 9600 bps, the problem of baud rate synchronization was resolved. It is clear from Fig. 13, that the keyboard (written simply on paper) is placed beneath the restive touch pad. Due to physical constraint of space, the keyboard accommodates all the 26 letter of English alphabet. Along with it accommodates numbers 0-9 and operations such as "space", "Delete", "Backspace" and "Enter". The keyboard is intentionally made as "Non-Qwerty" keyboard to test the efficiency of the prototype. The additional functionality of a conventional keyboard can be easily implemented if one uses a resistive touch pad of higher dimension instead of the one used (258mm x 180mm).

The Xbee module connects with the receiver module to transfer the data wirelessly. Some minor issues arise with Xbee, with the received data being spurts in nature. This was resolved by considering Tx as router and Rx as coordinator. The connection can also be wired like conventional mouse. However, to accommodate the inherent advantages of wireless connection, Xbee module has been integrated. The toggle switch makes the manual change to use the same touch pad to work either as a mouse or a keyboard. Arduino Leonardo, detects whether the received data is for mouse or keyboard. The decision matrix of $\langle L, R, x2, y1, M, \alpha \rangle$ as shown in Fig. 15, is in confirmation to the proposed algorithm. It can be clearly observed that 'M' is selected which indicates mouse operation. The value in x and y indicates the location of the mouse on the screen indicated as the 3rd and 4th entry of Fig.15. Since both L and R indicates null, hence it can be well assumed that the screenshot indicates of mouse is "only" placed on the screen. Neither Left nor Right click has been pressed. If first two entries show either L or R click, then depending on the location of mouse the operation will be executed. For Keyboard also the same diagram can be generated only except the first two entries will be null, the location is not required and K will be displayed along with symbol α .

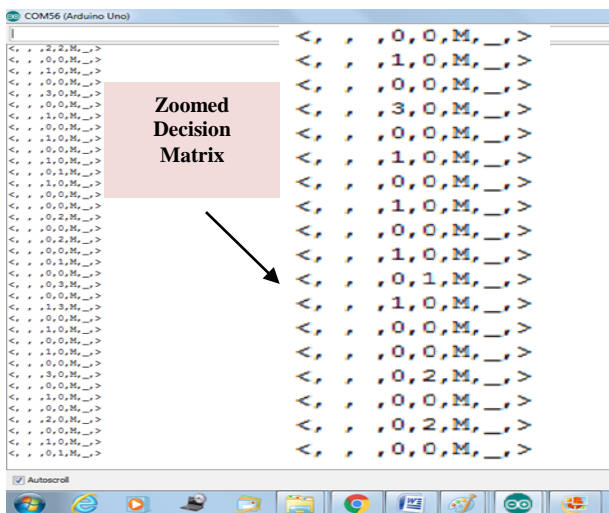


Fig. 15 Decision Matrix for Mouse operation

VI. CONCLUSION

In this paper, a wireless mouse is implemented using a resistive touch sensor pad which eliminates the constraint of a wire. The resistive touch sensor is used as a platform for user interface. A 4-wire resistor is connected to the microcontroller Arduino Uno which receives the data and sends to the XBEE S2 that serves as a RF device. At receiver side XBEE S2 receives the data wirelessly and sends to Leonardo board which controls the mouse or keyboard as per requirement. Hence, in this prototype we have managed to make a wireless mouse that has the following functionality of right and left click and mouse pointer. Along with it, the prototype clearly demonstrated the working of handheld keyboard with features

like 40 keys which include all alphabets, decimal numbers and some keys. The future improvements of this prototype have already been thought of as a fully functioning keyboard with all the features of a conventional keyboard, as well as adding 'scrolls' option in the mouse.

ACKNOWLEDGEMENT

The authors deeply acknowledge the generous support from communication laboratory of School of Electronics Engineering, VIT University.

REFERENCE

- [1] R. Sathyaleena, D. Sobitha, B. Pradeeppraja, "An Overview of Wireless Mouse : History, Challenges and Applications", International Journal of engineering Research and Technology, Vol. 2 Issue 12, December – 2013
- [2] G. akbar, "The knowledge-Inertial and Gyroscopic Mice", Available at <http://theveryknowledge.blogspot.in/2013/04/inertial-and-gyroscopic-mice.html>
- [3] C.J. Snijders, "Evaluation of 8 ergonomic computer mice a regular mouse and the HandShoeMouse by Hippus", Erasmus MC, University Medical Center Rotterdam, 2007.
- [4] "Laser Mouse vs Optical Mouse", Available at https://www.wikivis.com/wiki/Laser_Mouse_vs_Optical_Mouse
- [5] T.Watanabe, T.Tamechika and T.Ifukube, "Exploration of the lines displayed by a tactile mouse", International Conference on Presentation and Blindness, San Marino, May 1998.
- [6] Sculpt Ergonomic Mouse, Available at <https://www.microsoft.com/accessories/en-us/products/mice/sculpt-ergonomic-mouse/16v-00001>
- [7] D.Galloway, "Hack an Old Computer Mouse Into a Wireless Bluetooth Mouse", Available at <http://lifehacker.com/5941600/hack-an-old-computer-mouse-into-a-wireless-bluetooth-mouse>
- [8] "How do optical mice work?". Available at <http://computer.howstuffworks.com/question631.htm>, 08 May 2001.
- [9] XBee 2mW Wire Antenna - Series 2, Available at <https://www.sparkfun.com/products/10414>
- [10] Arduino Uno available at <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- [11] C.D.H. Williams, "Introduction to Sensors" available at <http://newton.ex.ac.uk/teaching/CDHW/Sensors/>
- [12] National Research Council, "Expanding the Vision of Sensor Materials", National Academy Press, Washington, D.C.1995
- [13] Arduino Leonardo available at <https://www.arduino.cc/en/Main/ArduinoBoardLeonardo>
- [14] Planet Analog, "Touchscreens 101: Understanding touchscreen technology and design", June, 2009, Available at http://www.planetanalog.com/document.asp?doc_id=527876
- [15] Arduino 1.5.8 available at <https://www.arduino.cc/en/Main/Software>
- [16] Seong-Ro Lee, R.Piyare, "Performance Analysis of XBee ZB Module Based Wireless Sensor Networks", International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013.