IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 9, Issue 6 Ver. IV (Nov – Dec. 2014), PP 32-39
www.iosrjournals.org

Wireless Gesture Controlled Systems

Rithesh M Nanda[1], Harshini H K[1], Praveen Kuruvadi[1], Ankhit B V[1], C Gururaj[2]
[1]Student, Dept. of Telecommunication, BMS College of Engineering, Bangalore, India
[2]Assistant Professor, Dept. of Telecommunication, BMS College of Engineering, Bangalore, India

Abstract: *Present day scenario is tending towards the elimination of Human Interface device to Computers and other devices. Gaming systems are going towards complete elimination of controllers and moving towards use of cameras and Ultrasound sensors to detect the movements of gamers, which act as inputs rather than the signals sent by controllers. Similar systems have also been used to help the blind and the Paralympic people. This proposed system helps do the same. It translates the gestures and movement of the wrist and fingers to control the mouse of a computer or to interface the same to any device granting wireless control to devices based only on these movements. This can be applied to control wheelchairs for the disabled or a robotic arm for bomb diffusion. The main advantage of this project is that it uses sensors to track all the movements of an arm thereby giving the user freedom to be anywhere within the wireless range to control the system, unlike in systems based on camera to recognize the movements which restrict the user to face the camera at a particular angle. The system currently proposed in the paper is for mouse control and the extent of expansion is limited only by the hardware interfaced to the receiver end.*

Keywords: *generic systems, Gesture Control, keyboard, mouse, sensors*

I. Introduction

Human Interface Devices have always been a limitation for interaction between the human and digital world. The ongoing trend is to ease this, we have been using a Mouse and a Keyboard to interact with Computers. However it is further easier to interact with the computer if natural gestures such as just a tilt of the arm or a point could control the mouse. This has seen a good amount of application in field of gaming eliminating hand held wired controllers such as joysticks transcending to natural hand movements which are involved in the game to gain control over the digital end. These complex systems are handled through the software end middleware which interprets data from sources such as cameras and sensor. This paper aims to provide an embedded low cost solution which does the same in primitive environments without much complexity. The same here is realized through an array of sensors whose data is processed and mapped to a certain parameter that can be controlled in the physical world. The paper discusses the system prototyped to control the mouse of a computer through gestures and tilt of the wrist. The same is realized through the use of Accelerometer, Reed Switch and a Flex sensor.[1 -3]

II. Introduction to Arduino Platform

Arduino is undoubtedly the most used embedded platform for prototyping in the current generation. Powered by 8-Bit Atmel AVR Microcontrollers or a 32-bit Atmel ARM these are highly versatile single-board microcontrollers which have provided an inexpensive and easy way for hobbyists, students, and professionals to create devices that interact with their environment using sensors and actuators.

The current system makes use of the Arduino UNO for the transmitter section which collects all the data and processes them for communication. The board is interfaced with an ADXL335 Accelerometer to detect the tilt or the orientation of the board and a flex sensor to sense finger bends. These two parameters are mapped to the movement of the mouse and scrolling action respectively. Additional interfacing of sensors can be made to make the same system generic, that is, give it capability to drive any other system that can be bridged to the receiver.

The receiver is an Arduino Leonard which has predefined HID Libraries which enables it to act as a mouse or keyboard.

Wireless Communication uses the nRF24l01+ Modules produced by Nordic which are one of the most cost efficient models available in the market.

Moreover the use of Arduino ensures that an open turf is used, with complete freedom to change the design, boot loaders, physical attributes of the system etc.[4]

III. Tracking Hand movements

For the scope of this paper a system was developed to track the motion of the wrist in the 3- axis i.e. the X, Y and Z planes of rotation using an accelerometer mounted on the glove worn by the user. This is similar

DOI: 10.9790/1676-09643239

to tracking the pitch, yaw and roll of an aero plane in the 3-D plane. This accelerometer would in turn give out varying voltages based on the orientation of the wrist. For the mouse control only the X and Y planes are considered to map the movement of the mouse in a 2-D plane on the screen. The left and right clicks of the mouse are done by reed switches placed on the index and the middle finger, which are activated in presence of a magnetic field provided by a neodymium magnet placed on the thumb. The thumb is brought near the particular finger which gives rise to left or right clicks of the mouse respectively. The scroll function of the mouse is controlled by using a flex sensor mounted onto the fingers to capture the folding of the fingers. The variable resistance of the flex sensor which is varied by folding of the fingers can be directly mapped onto the intensity of the scroll up or down function of the mouse. Hence this system which can be easily put together on a glove can track the complete movement of the wrist and hand, which in turn can be used for many applications.[4]

IV. Summary of the sensors used

As mentioned before the movements are tracked using the following sensors. The ADXL335 acting as an accelerometer. Flex sensors for tracking folding of fingers, and reed switches to capture click function.

4.1.1 ADXL 335

ADXL335 is a small, low-power, 3-axis accelerometer which is an electromechanical device that will measure acceleration forces with full-scale sensing range of ±3 g. These forces may be static, like the constant force of gravity, or dynamic - caused by moving or vibrating the accelerometer.

The sensor is a polysilicon surface-micro machined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration.[5]

4.1.2 Noise- Bandwidth trade off of ADXL 335

The ADXL335 has provisions for band limiting the XOUT, YOUT, and ZOUT pins. Capacitors must be added at these pins to implement low-pass filtering for antialiasing and noise reduction. The demodulator output is amplified and brought off-chip through a 32 kΩ resistor. The equation for the 3 dB bandwidth is

$$F-3\ \text{dB} = 1/(2\pi(32\ \text{k}\Omega) \times C(X, Y, Z)) \qquad (4.1)$$

or more simply

$$F\text{–}3\ \text{dB} = 5\ \mu\text{F}/C(X, Y, Z) \qquad (4.2)$$

The tolerance of the internal resistor (RFILT) typically varies as much as ±15% of its nominal value (32 kΩ), and the bandwidth varies accordingly. A minimum capacitance of 0.0047 µF for CX, CY, and CZ is recommended in all cases.

The selected accelerometer bandwidth ultimately determines the measurement resolution (smallest detectable acceleration). Filtering can be used to lower the noise floor to improve the resolution of the accelerometer. Resolution is dependent on the analog filter bandwidth at XOUT, YOUT, and ZOUT. The current system uses a bandwidth of about 500Hz hence the coupling capacitance is about 0.01 µF. The output must be filtered at this point to reduce the noise induced. The noise has the characteristics of white Gaussian noise, which contributes equally at all frequencies and is described in terms of µg/√Hz (the noise is proportional to the square root of the accelerometer bandwidth). The analog bandwidth must be no more than half the analog-to-digital sampling frequency to minimize aliasing. The analog bandwidth can be further decreased to reduce noise and improve resolution. The final tradeoff between noise and bandwidth is given by the equation.

$$RMS_{noise} = \text{Noise Density x } (\sqrt{BW x1.6}) \qquad (4.3)$$

The ADXL335 output is ratiometric therefore the output sensitivity (or scale factor) varies proportionally to the supply voltage.[6]

4.2 Reed Switches

The reed switch is an electrical switch operated by an applied magnetic field. The assembly of the reed switch mainly contains two connection leads in a glass capsule which is hermetically sealed. The reeds have elasticity which help them bend and connect when a magnetic force is applied and restore when it is removed. The magnetic field has to be strong enough so that it crosses the restoring threshold.[7]

If a suitable material like a ferromagnetic piece is kept in this region, it will experience a force in a direction the flux lines point. When it magnetizes the contacts in opposite ways it creates an attractive force between the two reeds and when this attractive force exceeds the restoring elastic force of the reeds, the two connect.

For the present setup the reed switches are used in series with a resistor connected to supply and reed to ground. The voltage at the bridge is measured. This format makes it act like an active low whenever it experiences a magnetic field which we induced through a magnet. This output is read by the digital pin of the Arduino to determine whether it was a left click, right click or to do nothing when it is high.

4.3 Flex Sensors

Flex sensors are passive resistive devices that can be used to detect bending. There is a change in resistance depending on the amount of bend on the sensor. As the amount of bend increases the resistance increases. An un-flexed sensor has a flat resistance of 10 KΩ. As the flex sensor is bent in either direction the resistance gradually changes. When flexed all the way the resistance rises to 20KΩ.

Here the flex sensor is coupled in series with a 10KΩ resistor to form a voltage bridge. The voltage at the middle of the bridge is fed to the analog pin of the Arduino which detects the variation in the voltage value with respect to bending of the sensor and converts it into digital form using it's built in ADC.

4.4 Results and Conclusion of Sensors Used

Following the experimental setup of interfacing the sensors, the following values were set for the software implementation of the sensors to achieve good data read from the sensors as well as reduce the noise.

For the ADXL 335 a coupling capacitance of about 0.01uF was used which gave us a good bandwidth of about 500Hz with minimum noise interference. The variation in bandwidth- capacitance can be seen by the following table:

Table 1. Filter Capacitor Selection, CX, CY, and CZ

Bandwidth (Hz)	Capacitor (µF)
1	4.7
10	0.47
50	0.10
100	0.05
200	0.027
500	0.01

A Vs = 3.3 V, the output sensitivity is typically is 330 mV/g. This is the sensitivity used in the current system. When this output was given to the analog pins of the Arduino the 10-bit ADC mapped the voltage variations of the X and Y planes need between the values of 270 to 400. Depending on this threshold values were set for motion of the mouse in different directions.

The flex sensors were used in form of a voltage bridge in series with a 10 kΩ resistor. This too was fed to the analog pin of the Arduino and converted to digital by the 10-bit ADC. For a supply of 5V the values of the flex sensor ranged from 100 to about 270.[8]

V. Wireless Communication over nRF24L01+ module

The nRF24L01+ is an ultra-low power trans-receiver IC which operates in the 2.4Ghz band. This module can be configured through a Serial Peripheral Interface (SPI) which is a full-duplex bus where data is sent in both ways at once, but on separate lines. nRF24L01+ has internal FIFOs which ensure smooth data flow along with a 6 data pipe multiceiver. The module has a built in state machine which controls the transmission between the chips operating modes and all the configuration registers are accessible in all operational modes. The radio front end uses GFSK modulation. It has user configurable parameters like frequency channel, output power and air data rate. nRF24L01+ supports an air data rate of 250 kbps, 1 Mbps and 2Mbps.

5.1 Communication Operation and Interfacing

These modules have an 8 pin out configuration where the communication happens over the SPI protocol using 4 pins of CSN, SCK, MISO and MOSI. The whole module runs on a supply voltage of 3.3V. The chip also consists of the CE pin which is chip enable pin which is kept active low during operation. The CSN pin refers to the chip select which is also kept active low during transmission and receiving modes. The module communicates with the Arduino over the MISO and MOSI ports in a master-slave format where the SCK gives the clock input to the module. It also consists of an IRQ (interrupt) pin which can be programmed based on user needs. It gives a reliable form of communication using the ACK (acknowledgement) bits to declare whether or not retransmission of a packet is necessary. The also use the CRC to correct bit errors. The data packets are sent in the following format.

Table 2. Data Packet Format

Preamble 1 byte	Address 3-5 bytes	Packet control field 9 bit	Payload 0-32 bytes	CRC 1-2 byte

The SPI protocol can be clearly understood using the following timing diagram for the read and write operations from the register map of the module:

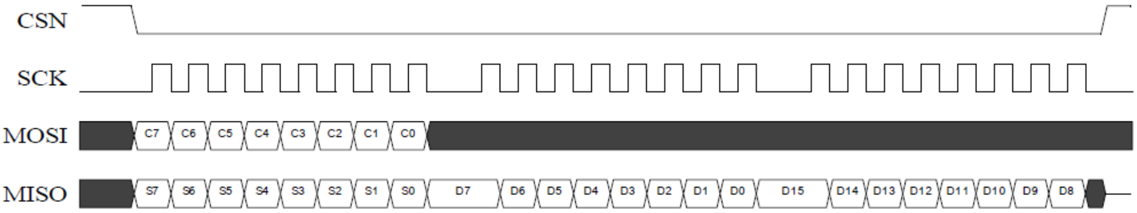
CSN
SCK
MOSI
C7 C6 C5 C4 C3 C2 C1 C0
MISO
S7 S6 S5 S4 S3 S2 S1 S0 D7 D6 D5 D4 D3 D2 D1 D0 D15 D14 D13 D12 D11 D10 D9 D8

Figure 1 SPI Read Operation

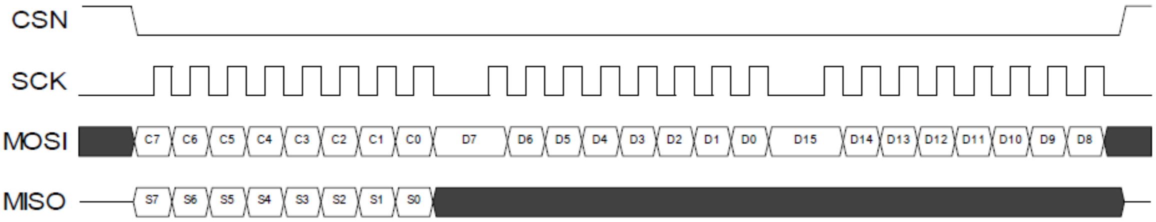
CSN
SCK
MOSI
C7 C6 C5 C4 C3 C2 C1 C0 D7 D6 D5 D4 D3 D2 D1 D0 D15 D14 D13 D12 D11 D10 D9 D8
MISO
S7 S6 S5 S4 S3 S2 S1 S0

Figure 2 SPI Write Operation

5.2 Operating modes

The nRf24L01+ can be configured in 3 different modes - Power Down, Standby, Rx or TX mode. When VDD reaches 1.9V or more the module goes to the Power On reset state and remains in reset until it enters the Power Down mode. In the Power Down mode the module is disabled whereas the register statuses available are maintained and SPI is kept active. Power down mode is entered by setting the PWR_UP bit in the CONFIG register low.

In the Standby mode only a part of the crystal oscillator is active and changes to the active mode happens only if the chip enable (CE) pin is set high. By setting the PWR_UP bit in the CONFIG register to 1, the device enters standby mode.

The TX mode is an active mode for transmitting packets. To enter this mode, the nRF24L01+ must have the PWR_UP bit set high, PRIM_RX bit set low, a payload in the TX FIFO and a high pulse on the CE for more than 10µs. The nRF24L01+ stays in TX mode until it finishes transmitting a packet. If CE = 0, nRF24L01+ returns to standby mode. If CE= 1, the status of the TX FIFO determines the next action. If the TX FIFO is not empty the nRF24L01+ remains in TX mode and transmits the next packet.

NRF24l01+ radio is used as a receiver in the Rx mode. To enter the Rx mode, the PWR_UP bit, PRIM_RX bit and the CE pin are set high. The receiver demodulates the signals from the channel and searches for a valid packet. When a valid packet is found, the payload of the packet is presented in the RX FIFOs. If the RX FIFOs are full, the received packet is discarded.

5.3 Results and Conclusion based on timing and delays of wireless communication

From the analysis of the system it was observed that the switching between the standby modes to the Rx/Tx modes had a delay of about 130uS. Also the CE high should be maintained for a minimum of 10uS for proper operation. The different modes can be controlled by manipulating the bits in the CONFIG register. Communication between the Arduino and the module is maintained at 9600 baud rate. The data pipeline having the register map address of 0xE8E8F0F0E1LL is used. For the receive operation the data from the payload is read from the RX FIFOs and same applies in the transmission operation where the data is written into the TX FIFOs. At the end of each operation the FIFOs are flushed so as to maintain smooth transmission. The data from the sensors on the glove is presented in form of a 5 bit array to the module which relays this array to the receiver at about 25-30 times per second for the smooth operation of the mouse on the receiver end.

VI. Experimental Setup

The following experimental setup was made for the system and is given in form of a block diagram. The transmission system is as follows:

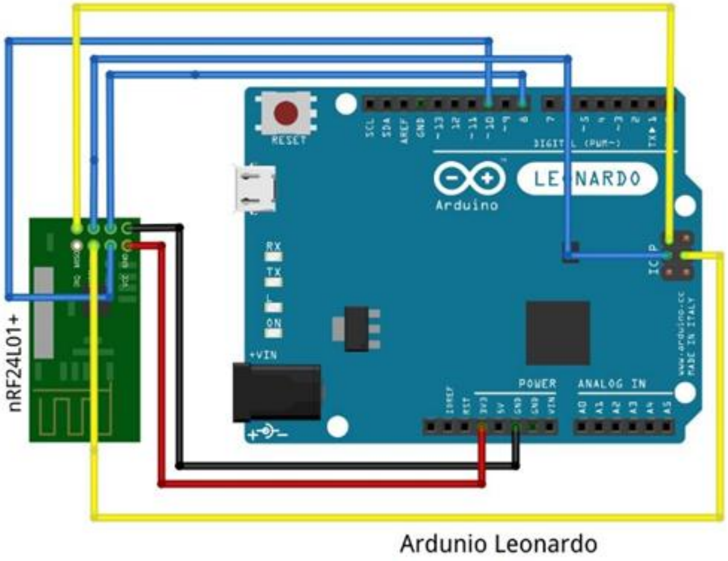

Figure 3. Receiver Circuit

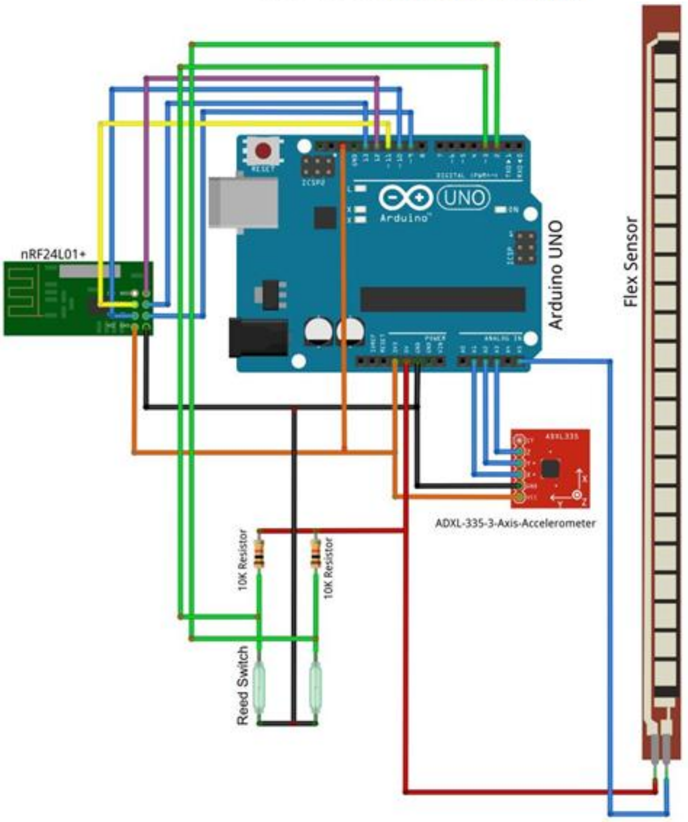

Figure 4. Transmitter Circuit

In the Transmitting Circuit, the three axes of the ADXL-335 are connected to analog pins A1, A2 and A3 on the Arduino Uno and the values from the accelerometer are obtained depending on its tilt. The ADXL-335 runs on a 3.3V supply provided by the Arduino, and the respective VCC and GND pins are connected. The Reed Switches are Active Low devices, which are pulled down to a value of zero when the contacts in the switch meet in the presence of a magnetic field. They use a 5V supply on the Arduino. These are relayed to the digital pins 2 and 3 on the Arduino. The nRF24L01+ wireless transmitter and receiver also run on 3.3V supply. The Chip Enable (CE) and Chip Select Not (CSN) pins on the transmitter are connected to Digital Pins 9 and 10.

Pin 13 provides the Clock signal (SCK) to the module, and pins 11 and 12 perform the function of Master Out Slave In (MOSI) and Master In Slave Out (MISO) respectively. The Flex sensor connected to analog pin A5 can be used to drive other parts of a generic device.

The Receiver Circuit consisting of the nRF24L01+ wireless receiver, interfaced to the Arduino Leonard, obtains the values from the transmitting module. The VCC and GND connections are made, CE and CSN are connected to pins 8 and 10 on the Leonard, and the ICSP pins 3, 4 and 1 are used for SCK, MOSI and MISO respectively. [11-12]

VII. Logic flow and software design

7.1 Transmission

It can be noticed from the diagram the accelerometer values are read from the analog pins of the Arduino. The Arduino has a built in 10-bit ADC which converts the voltage of 0- 5V into values ranging from 0- 1023. The delays in the conversion must be kept well in mind while coding. The analogRead function takes up a 5 milli-second delay. As for the Digital pins which read the values from the reed switches take up 1 millisecond delay. The maximum range of values for the X axis, Y axis and the Flex sensor have been mentioned before. The corresponding values are put in form of an array of size 5 as shown:

Table 3. Array Elements

0 or 1 From Reed switch 1	0 or 1 From Reed switch 2	X axis value	Y axis value	Flex sensor value

This array is then transferred to the nRF24L01+ module for transmission. The module is first initialized and a specific baud rate is set for transmission. It follows the SPI protocol as such the CE and the CSN pins are kept low accordingly during communication. The register maps are flushed and reset initially and a radio is setup. The operation and timing protocols of the module as mentioned before are kept in mind. It is switched to Tx mode and the Tx_PAYLOAD and Tx_PAYLOAD_ACK registers are reset and initialized, and the data packet is sent to Tx FIFO ready for transmission. We use a particular data pipeline of the module as mentioned before. The FIFOs are flushed at the end of the transmission. This goes on continuously at about 25-30 cycles per second. Hence the movements of the hand are sent to the receiver to be translated into the movements of the mouse. The logic flow is as given in Figure. 5

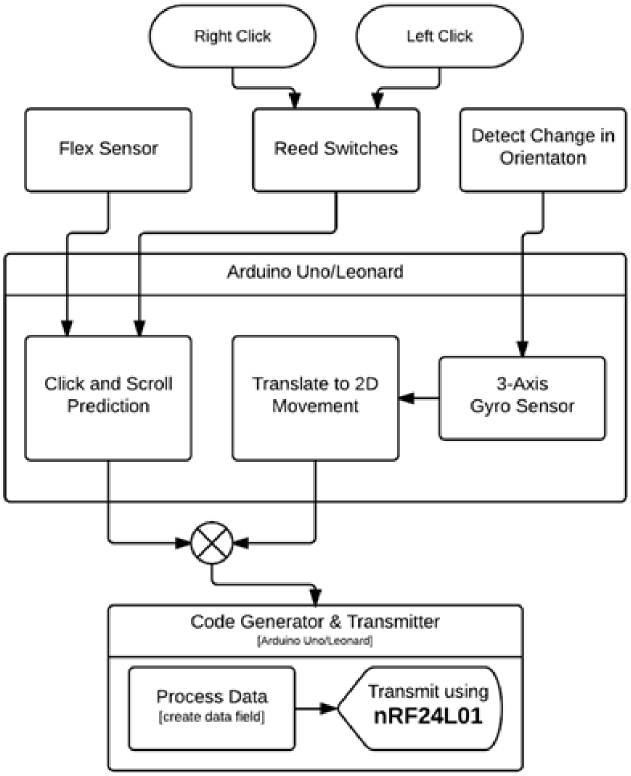

Figure 5. Transmitter Block Diagram

7.2 Receiving

Here the module is again setup with the same specifications as the transmitter, with a baud rate of 9600 and is switched onto the receiving mode. Hence the received packets are in the Rx FIFO which are read using the Rx_PAYLOAD register and this data is then communicated over to the Arduino over the MISO and MOSI ports using the SPI protocol keeping the CE and CSN pins low during communication. Initially the module is initialized as a receiver using the same data pipeline used in the transmitter. The FIFOs are flushed after reception of the information. Upon receiving the array the receiver i.e. in particular the Arduino Leonardo interprets this as mouse movements. The built in Mouse HID driver in the Leonardo makes it easier to map out the values for movements of the mouse. The problem now faced is how to infer movements from the values. The solution presented was simple. Consider a number line representing the range of values from min to max of the X axis in the accelerometer. Now two threshold values are defined at specific points as shown:

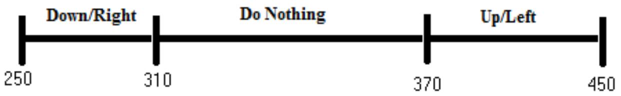

Figure 6. Thresholding Values

The driver can be programmed to move the mouse cursor around 5-10 pixels to the left or right if it crosses the threshold1 or threshold2 respectively. The same logic is applied to the up and down movements of the mouse as well as the scroll up or down function. As for the left and right clicks, the values from the reed switches are active low i.e. a 0 is transmitted when the switch is active hence depending on the particular switch the driver can be programmed to perform the left or right click operation. The logic flow can be observed as follows:

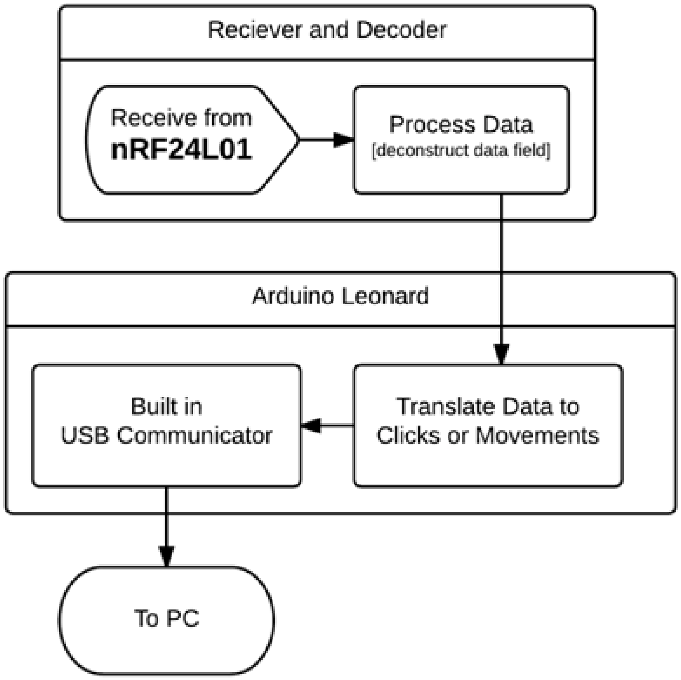
Reciever and Decoder
Receive from nRF24L01
Process Data [deconstruct data field]
Arduino Leonard
Built in USB Communicator
Translate Data to Clicks or Movements
To PC

Figure 7. Receiver Block Diagram

VIII. Results

The Transmitting circuit, consisting of the Arduino Uno along with the ADXL-335, Reed switches, flex sensor and the nRF24L01 wireless transmitter is mounted on a glove and this entire circuit is powered by a 9V Battery connected to the Arduino Uno. The Receiver circuit, which includes the Arduino Leonardo, and the wireless receiver is connected to the target generic device; which in this case is a laptop, to control the mouse. With the inclusion of the specific mouse delays and setting the respective thresholds for each movement obtained from the transmitter, every horizontal and vertical movement of the user's hand can be translated into a smooth movement of the mouse which is detected at the receiving laptop. Thus, the mouse is able to obtain and respond to inputs fairly quickly, without too much of a delay and thus gives a smooth movement. The accuracy of the glove device is also appropriate. 2D motion due to the accelerometer, mouse clicks and scrolling is accomplished effectively. The tilt on the accelerometer effectively controls the speed at which the cursor moves on the screen. The movement of the cursor can is as accurate as up to 2-5 pixels per motion received, hence making it very accurate. The Reed Switches facilitate click operation. When the magnet fitted on the thumb is bought close to one of the switches, the presence of a magnetic field causes the switch to close and the respective click occurs. A delay at the end of each click ensures that only one click is recorded. The Wireless transceivers are observed to work at a distance of 10 feet apart. There is only a very slight delay which is observed when the user performs a gesture and the same is carried out on the computer. This delay is attributed to the transmission and reception between the transceivers, which increases as the modules are set further apart from each other. Overall the entire system is accurate and works with precision.

IX. Conclusion & Future Work

The glove is found to be a very convenient device which can provide a more realistic feel while operating certain generic devices, and it also very comfortable and easy to use. The functionalities of this system extend manifold. The glove can be used by disabled people who have the ability to use their fingers, to operate their wheelchairs or computers. Further, gestures can be programmed onto an autonomous robot, where they can be performed if necessary in unmanned regions. Using the Pitch, Yaw and Roll axes on the accelerometer, the glove can also be used to make music on the go, by varying the levels and intensity on a computer for any particular gesture. The glove can also be used to control a rover remotely, the gestures on the glove controlling the movement of the robot in different directions. A pinch-in-zoom-out and pinch-out-zoom-in gesture can also be implemented using the Z-axis of the accelerometer. The glove can be used in collaboration with a virtual keyboard application eliminating the use of separate inputs. The gloves are hence, very versatile and can be used for many operations. As such this system opens up a world of opportunity for various applications.

References

[1]. Matthias Rehm, Nikolaus Bee, Elisabeth André, Wave Like an Egyptian - Accelerometer Based Gesture Recognition for Culture Specific Interactions, British Computer Society, 2007

[2]. R. Cipolla and A. Pentland, Computer Vision for Human-Machine Interaction, Cambridge University Press, 1998, ISBN 978-0521622530

[3]. Cornell university- glove mouse project by Adam Shih and Hyodong Lee

[4]. Chen, Q., Georganas, N., and Petriu, E. Real-time vision-based hand gesture recognition using haar-like features. In Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE (May 2007), pp. 1–6.

[5]. Perng, J., Fisher, B., Hollar, S., and Pister, K. Acceleration sensing glove (asg). In Wearable Computers, 1999. Digest of Papers. The Third International Symposium on (1999), pp. 178–180.

[6]. Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. A hand gesture interface device. SIGCHI Bull. 17, SI (1987), 189–192.

[7]. J. Kela, et al., "Accelerometer-Based Gesture Control for a Design Environment," Personal and Ubiquitous Com- puting, Vol. 10, No. 5, 2005, pp. 285-299.

[8]. H. Lee, et al., "Select-and-Point: A Novel Interface for Multi-Device Connection and Control Based on Simple Hand Gestures," CHI 2008, Florence, 5-10 April 2008.

[9]. T. Shiratori and J. K. Hodgins, "Accelerometer-based user interfaces for the control of a physically simulated character", ACM Transactions on Graphics, vol. 27, no. 5, (2008), pp. 1–9

[10]. S. P. Priyal, P. K. Bora, "A study on static hand gesture recognition using moments", In Proceedings of International Conference on Signal Processing and Communications (SPCOM), (2010), pp. 1-5

[11]. Hand gesture computing, Gaurav Pradhan, Balakirshnan Prabhakaran, IEEE Conference , 2011

[12]. Juha K., Panu K., Jani M., Sanna K., Giuseppe S., Luca J. and Sergio D. M. Accelerometer-based gesture control for a design environment, Springer, Finland, 2005.