Virtual Mouse

Design and Implementation

Of

An Accelerometer based Wireless Mouse

A Thesis Report

Submitted to:

The School of Electronics and Computer Science

Department of Electrical and Electronics Engineering

By

Arghya Dev Biswas Aryan	09221066
Md. Fahmid Wasif	09221059
Md. Mahmudul Hasan	09221057

Thesis Supervisor

Dr. Mohammed Belal Hossain Bhuian

Department of Electrical and Electronics Engineering

BRAC University

Dhaka, Bangladesh

December 2011

Declaration

We declare that the thesis report is based entirely upon the research conducted by the members of the group under the guidance and supervision of our thesis supervisor. The report focuses on the theoretical aspects of developing an accelerometer based mouse and also illustrates the mechanism of how we have implemented the theoretical aspects of the accelerometer based wireless mouse and it illustrates certain necessary steps required to implement the theories in order to develop an actual prototype of the “Virtual Mouse” in reality. All the reference documents have been stated in the reference section and all the extracted information and interpretation have been done through our own effort. The coding used for the programming of the microcontroller has been cited in the Appendix section of the report. This is our original work and was not submitted elsewhere for the award of any other degree or any other publication.

Date: 15th December 2011

Arghya Dev Biswas Aryan

Md. Fahmid Wasif

Md. Mahmudul Hasan

Dr. Mohammed Belal Hossain Bhuian

Assistant Professor

Department of Electrical and Electronic Engineering

BRAC University

Table of Contents

Thesis Abstract

The concept of Accelerometer Based Hand Action Recognition is a known phenomenon and has its execution in the field of gaming controller. By the use of this concept, we intend to build up a device that we would like to call a “Virtual Mouse”. The main functional aspect of the particular project would be to use an accelerometer to sense the movement done by the hand on which the accelerometer is placed. The cursor will be moved as a result of the movement of the palm. The signal will be transmitted between a user and the computer screen by means of the mouse that we have deigned where the desired instructions would be given by means of the movement of the system via the hands of the user and would subsequently be recognized by the system and the instruction would be executed accordingly. The entire basis of the prototype is to have an interfacing system that would be operating as a mouse that we use in our daily life but it simply is in a total new paradigm keeping the convenience and ease of use in context of the user.

Acknowledgements

We would like to take this opportunity to express our heartiest gratitude to our thesis supervisor Dr. Mohammed Belal Hossain Bhuian who has always provided every possible help and support over this time that we have utilized to develop this project of ours and we would also thank our Co-supervisor Ms. Marzia Alam who has guided us whenever we needed. We have tried to develop a project that can easily be used and can be related to by general people and we had a vision in our mind to implement such a project and our batch mates have also helped us to work as a source of inspiration through ideas about components wherever we struggled but have been able to successfully overcome all obstacles and complete the initiative we took in our hands.

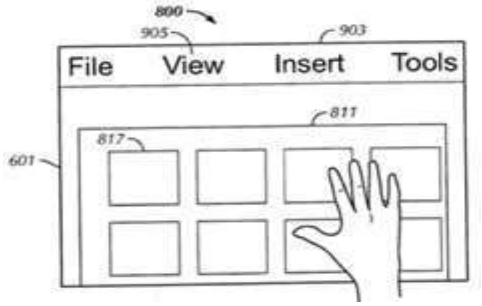
Background

As time is proceeding ahead, technology is improving and evolving every single moment. No one can claim something to be “latest” because the moment someone does so we can see the presence of something newer and better in front of our very own eyes. People endeavor to cope up with the dynamic changes but it is pretty difficult due to lack of availability of adequate and sufficient resources and technology. Two of the basic fundamental intentions of technology are to make things that are not complicated to be understood by the user and makes working of the user more convenient. Things are simple when the interface between human and technology is least complex. It is an outmost effort by scientists to make the interfacing system more and more convenient for the users.

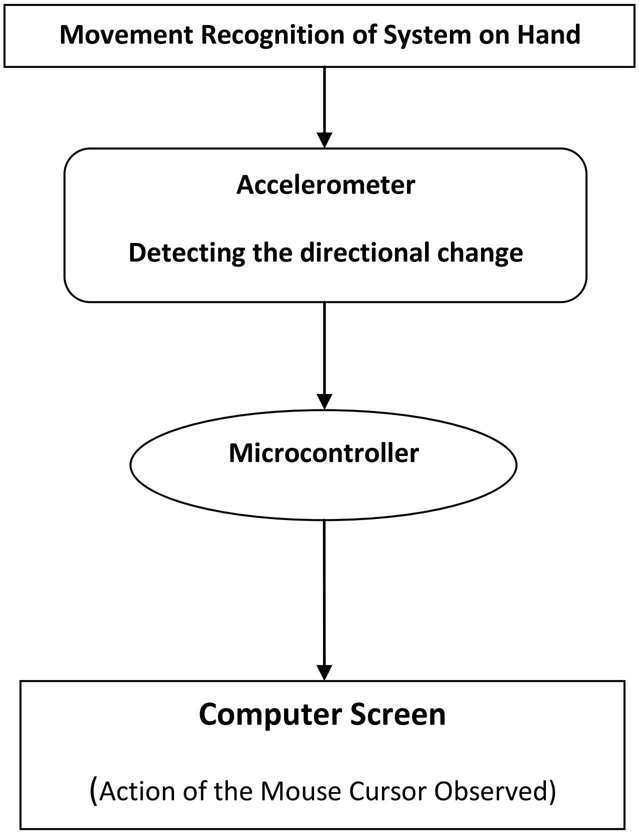
We, for our thesis have designed a wireless interface system which will be able to act as an enhanced version of one of the most common interfacing system which is the mouse. Our interfacing system will be an accelerometer based mouse. The accelerometer based mouse takes into account the instructions of the user as per the movement of the system by the user. The accelerometer processes the movement of the system with respect to the conventional axis in the two dimensions; it recognizes the direction of movement, gets it processed by means of a microcontroller and finally the instructions are interpreted and the cursor of the mouse moves on the screen of the computer accordingly.

We chose this particular topic for our thesis because it is something that can help us develop a real practical execution of the theories and principles that have researching upon and moreover it is the recent trend to develop better interfacing systems thus making this a viable and prospective project.

Figure: How we desire the project would function.



Functional Block Diagram of the System


Movement Recognition of System on Hand
Accelerometer
Detecting the directional change
Microcontroller
Computer Screen
(Action of the Mouse Cursor Observed)

The figure above illustrates the functional block diagram of the system that we wish to design and implement. The entire system requires all the fundamental components to work together.

In the initial stage, the hand gesture of the user is recognized by the accelerometer which reads the movement axis of the hand and gives output as voltages.

The analog hand gesture is then read by a microcontroller which has two built in codes that helps to carry out two tasks. The built-in analog to digital converter helps to convert the analog hand gesture, so that it is understandable by the computer when the signal is fed into the system.

The two coding inside the microcontroller helps to convert the signal and also allow the gesture to be recognized as mouse protocol when fed into the computer system, the other code will help the system to be recognized as an interface between the user and the computer. The Analog to Digital Converter being built-in helps our cause of not having the hassle of having an extra Analog to Digital Converter so it simplifies the system even further.

The signal from the Microcontroller is interpreted as corresponding cursor movement on the computer screen and moves according to the hand gestures of the user which ultimately will help to serve our main purpose of making the mouse that is the interface system independent of any surface.

Working Mechanism of the System as a Whole

The entire system is a simple mechanism and works exactly as a normal mouse currently. In the current status quo, when we see a normal mouse work which is a medium of interface between human user and the computer system. As we see, the mouse helps to carry simple instructions and when we work for executing certain commands, the mouse comes into working action. The “Virtual Mouse”, the prototype that we intend to design would require a mechanism that would execute the same instructions as done by the current mouse but the major difference would be that it would be simpler and would only require the hand gestures of user to execute instructions.

The fundamental components for the desired system that we have designed and tried our best to implement can be segmented in four portions:

I. Accelerometer

II. Microcontroller

III. USB enabled Interface System

IV. Wireless Mouse functionality

I. Working Principle of Accelerometer

We have used a “Small and Thin ±2 g Accelerometer, ADXL322” which works as a transducer; i.e. it recognizes the gesture of the hand and in accordance to the movement towards the axis direction transmits the data to be processed accordingly. The accelerometer measures the acceleration force; the forces may be static for example the constant force of gravity pulling at our feet, or they could be dynamic which is caused due to the movement or vibration of the accelerometer. By measuring the amount of static acceleration due to gravity, we can find out the angle the device is tilted at with respect to the earth. By sensing the amount of dynamic acceleration we would be able to analyze the direction to which the device we are analyzing is moving towards.

The accelerometer that we would use, measures acceleration from a capacitive sensing cell (g-cell) which forms two back-to-back capacitors. As shown in the figure below, when the center plate deflects due to acceleration, the value of capacitors will change and acceleration data can be extracted.

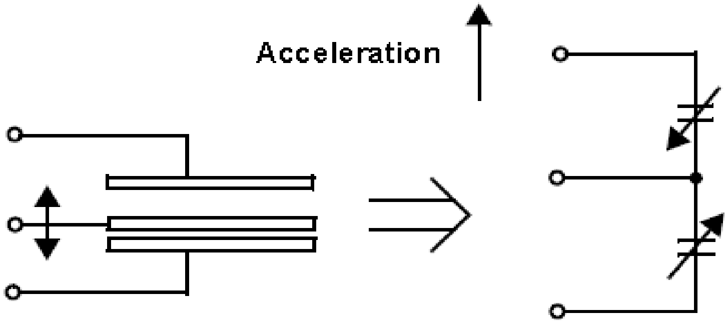


Figure: Physical Model of the Accelerometer

Each accelerometer’s reading consists of dynamic acceleration and static acceleration, namely the gravity. For a z-axis accelerometer placed flat, Figure 2 shows the sign of the reading. The static acceleration is +1g, where as the dynamic acceleration reading is positive if the direction of the acceleration is upward.

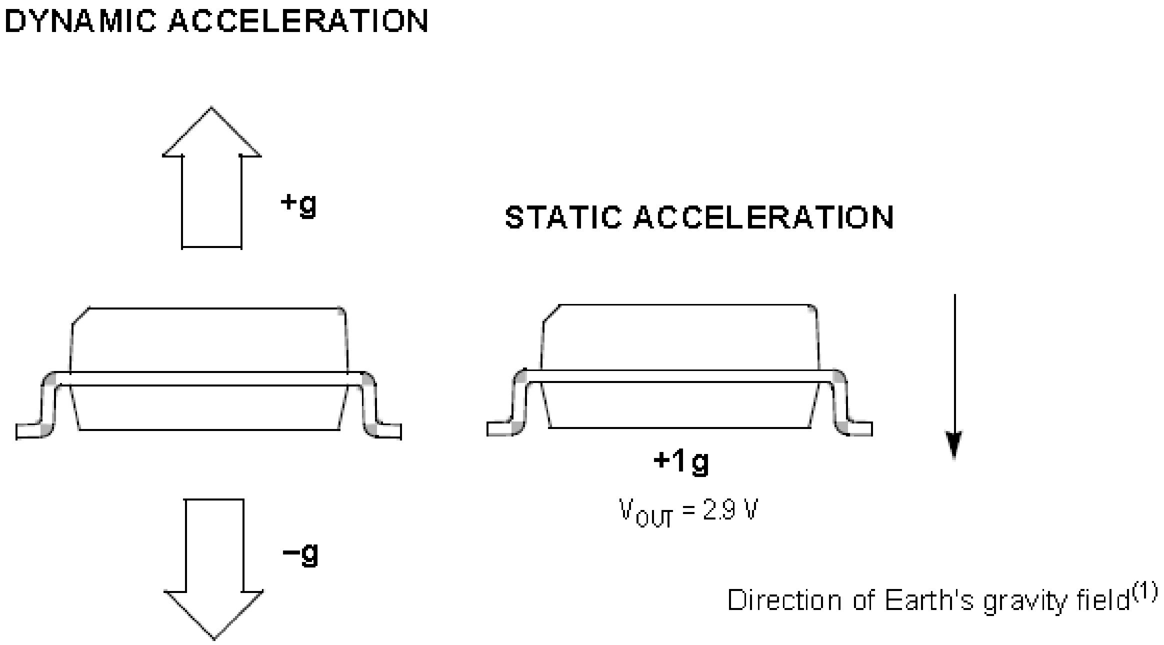


Figure 2: Sign of Measurement of the Accelerometer

By placing three accelerometers on the back of the hand, we set up a coordinate system. If the hand is still, we can measure direction of the hand plane. Moreover, if the finger moves slowly, then at each reading its acceleration are mostly due to gravity.

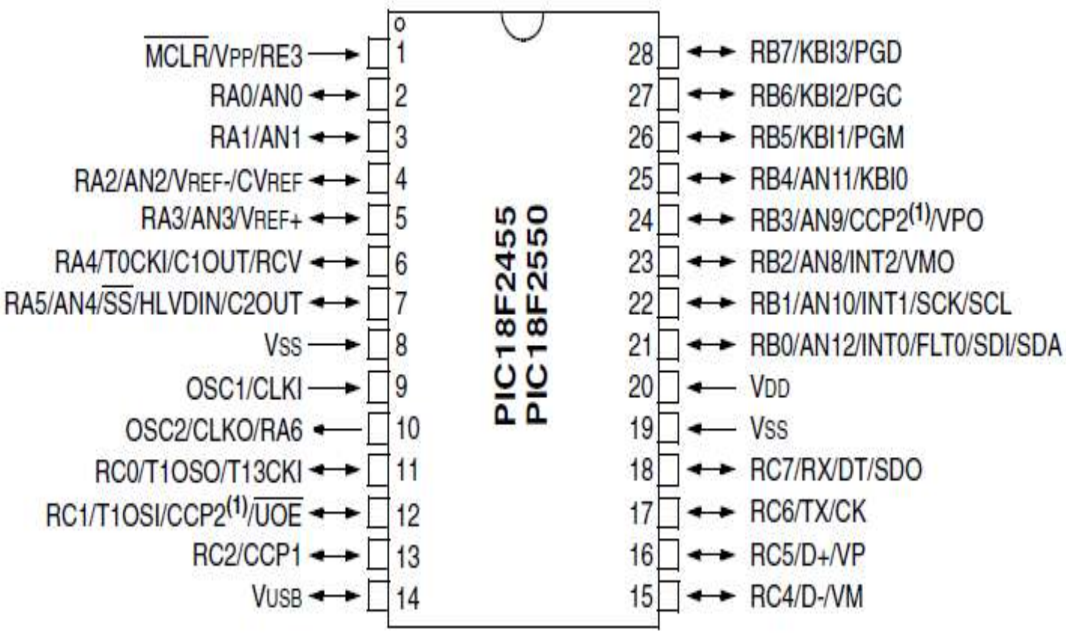
II. Working Principle of a Microcontroller

The Microcontroller acts as the component that helps to put forth the instructions on to the computer for the mouse pointer to abide by the given instructions and the built in program helps to reduce the complications of the coding and allows the interpretation to be done smoothly with ease.

We intend to use **PIC18F2550** as the microcontroller suitable for our project as it will help to carry out all the tasks that we desire to carry out.

The diagram below illustrates the pin configuration of the microcontroller that we have used and it has been selected on the basis of providing us with the ease and liberty of implementing that we have visualized and it has provided us with the sufficient ground of working while bearing in mind the actual intent of the concept being implemented.

28-Pin PDIP, SOIC

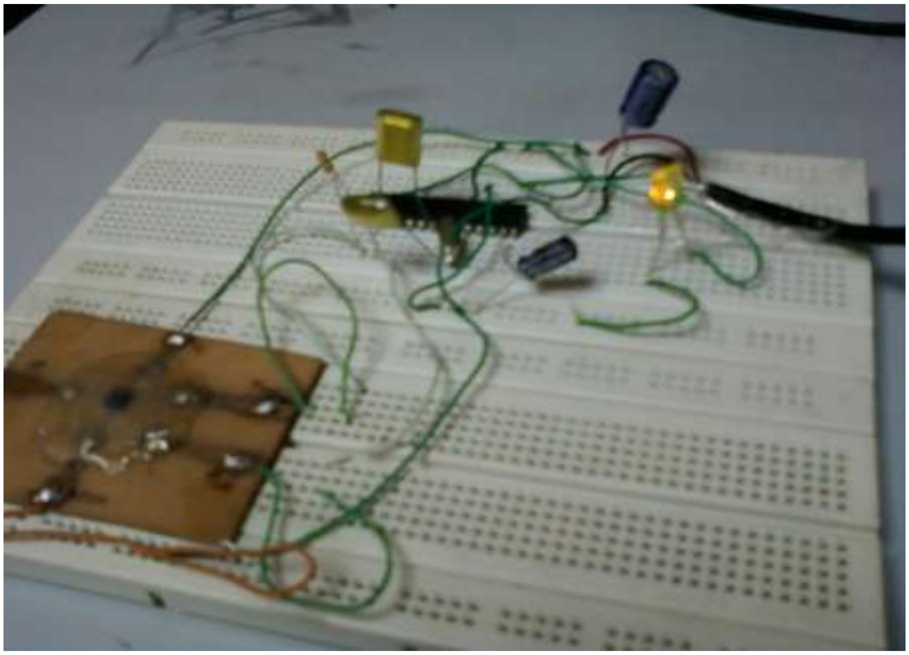


III. Working Principle of USB enabled Interface system

In order to connect the designed system we opted for the USB portal system as it will allow us to have an efficient mechanism of connecting and moreover the USB is one of the prominent port of connecting the interfacing system and can easily be detected by the processing unit of the computer with the least complications.

In order to make our system work we had to initially connect the system with the computer and while we were deciding on what would be the most suitable port to connect the system with the computer we unanimously conceded that bearing in mind the user friendly paradigm and ease in terms of usage, the most convenient method of connecting the system to the computer would be the Universal Serial Bus (USB) portal.

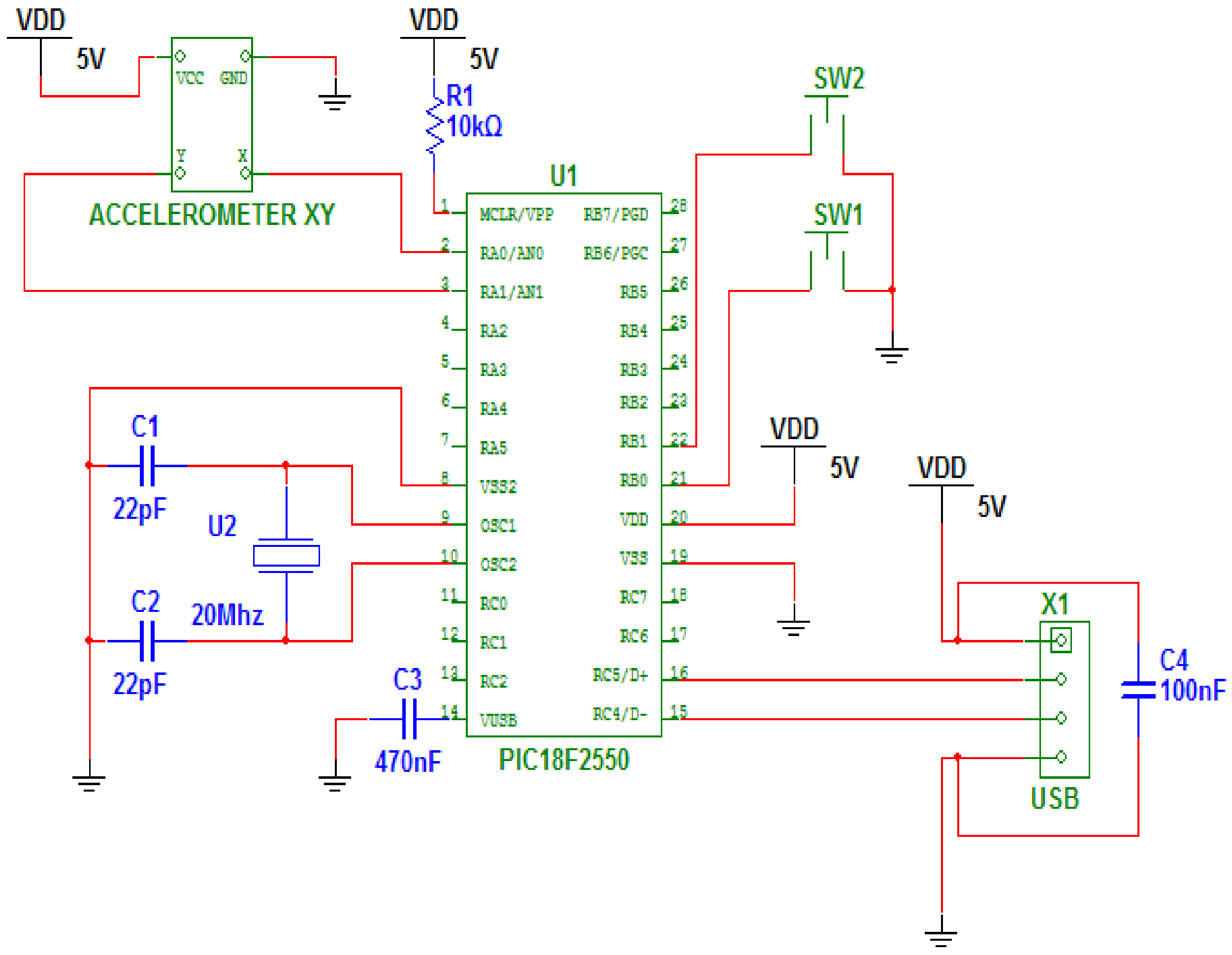
We had to use a microcontroller to synchronize the system with the computer for it to be recognized as an interfacing system as it is done in terms of our conventional mouse and keyboard which happen to be the most common systems used for interfacing between user and computer. The codes had to be used as of the conventional interfacing systems to get the desired output.



The picture shows the circuit that made the mouse suitable to be recognized by the computer via the USB portal.

The Schematic for the PIC USB illustrates that when we had to form a bridge for the computer to recognize the movement in context of the accelerometer having its relative recognition with respect to the conventional x-axis and y-axis. When making the synchronizing for the computer to recognize the movement and for it to be interpreted as cursor movement and basic functioning as a mouse, the microcontroller coding has provided the essentials and necessary instruction interpretation were also done by means of the microcontroller.

Schematic for the PIC USB

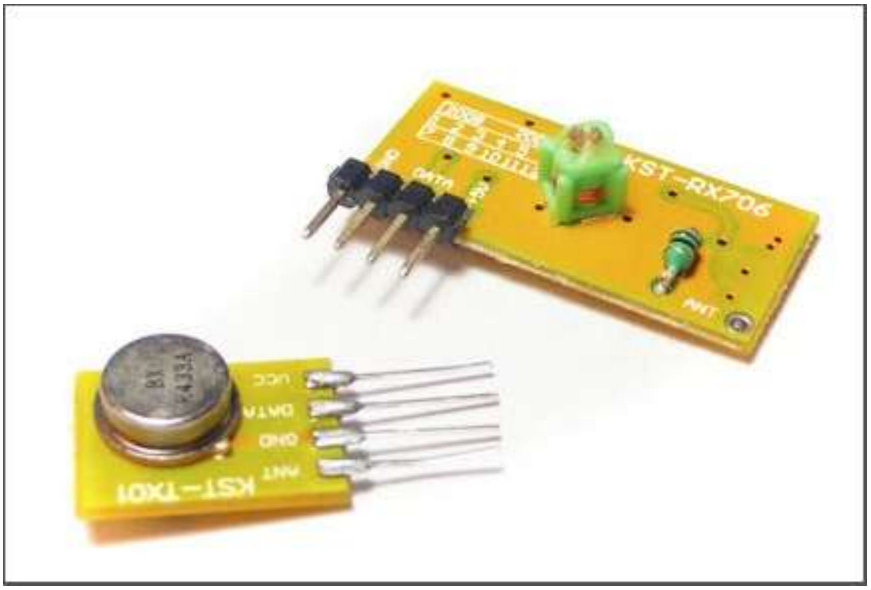


IV. Working Mechanism of the Wireless System in the Mouse

After making the system USB enabled, our intention was to take the next step forward and make the system Wireless to ensure the ease of use, make an enhanced version of the mouse that we had already developed and simply take it to the next level. When we were designing the system to be wireless, we again had to resort to the microcontroller as a mechanism that would allow us to incorporate this function into our “Virtual Mouse”. The wireless system has two crucial elements where one is the transmission part and the other is the receiving part. Both the phase has their own unique functional credential and thus we had to treat them separately. The idea was to ensure that an user was able to work using our system from an adequate distance away from the computer and it would give the user the ease and convenience of work. The intent of making the system Wireless thrived as the different portions had to be dealt with and designed with the difference in mind and we also had to ensure that the function of the mouse as an USB enables system also was persistent so with the cautious effort we did put forth the coding and ended up making the two different and parallel systems using two unique set ups to serve as a transmitting system and receiving system. The two different systems had to function simultaneously so initially we had some difficulty in making the synchronization of the two systems but after a laborious effort and endeavors we ended up in establishing the proper synchronization through use of the coding in the microcontrollers and at the same time making necessary adjustments for the other necessary components to complete the whole system.

In order to make the Wireless system we have used the 433 MHz RF Transmitter.

We felt that to make our desired system the most suitable transmitter would be 433 RF Transmitter because it is ideal for remote control applications where low cost and range is required. The transmitter operates from a 1.5 – 12 V supply, making it ideal for battery-powered applications. The transmitter employs a SAW-stabilized oscillator, ensuring accurate frequency control for best range performance. Output power and harmonic emissions are easy to control, making FCC and ETSI compliance easy. The manufacturing- friendly SIP style package and low cost makes the STT – 433 suitable for high volume applications.



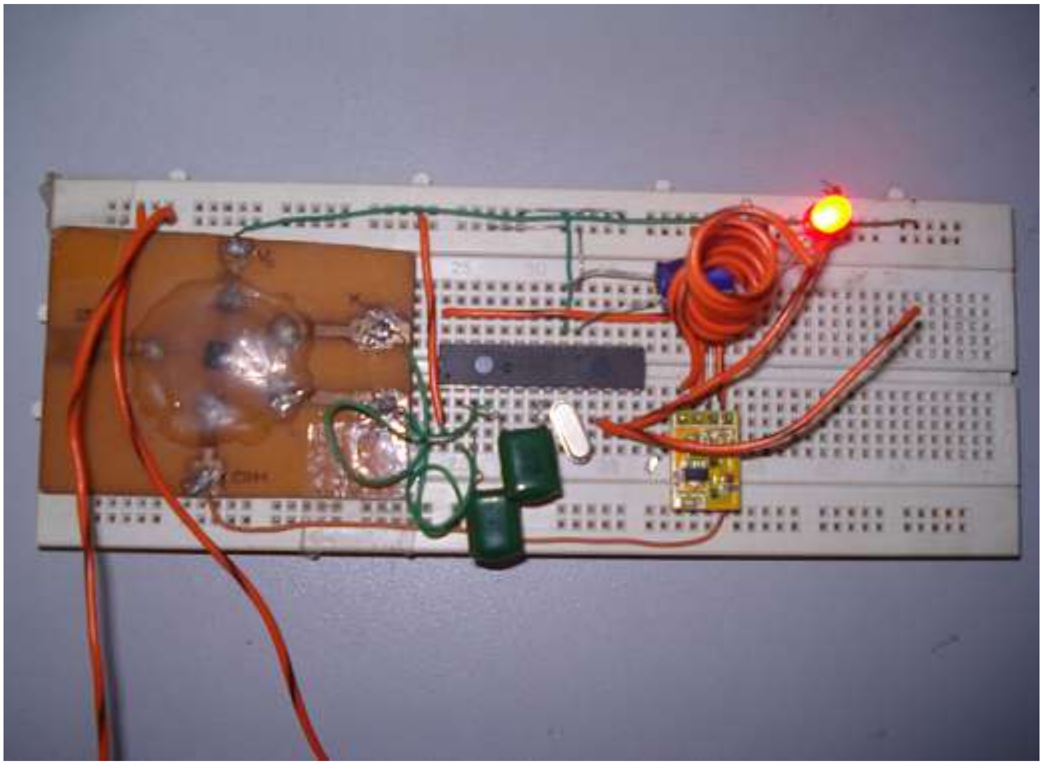
Features of 433 MHz RF Transmitter

- · Low Cost
- · 1.5-12V operation
- · 11mA current consumption at 3V
- · 4 dBm output power at 3V

Applications of 433 RF Transmitter

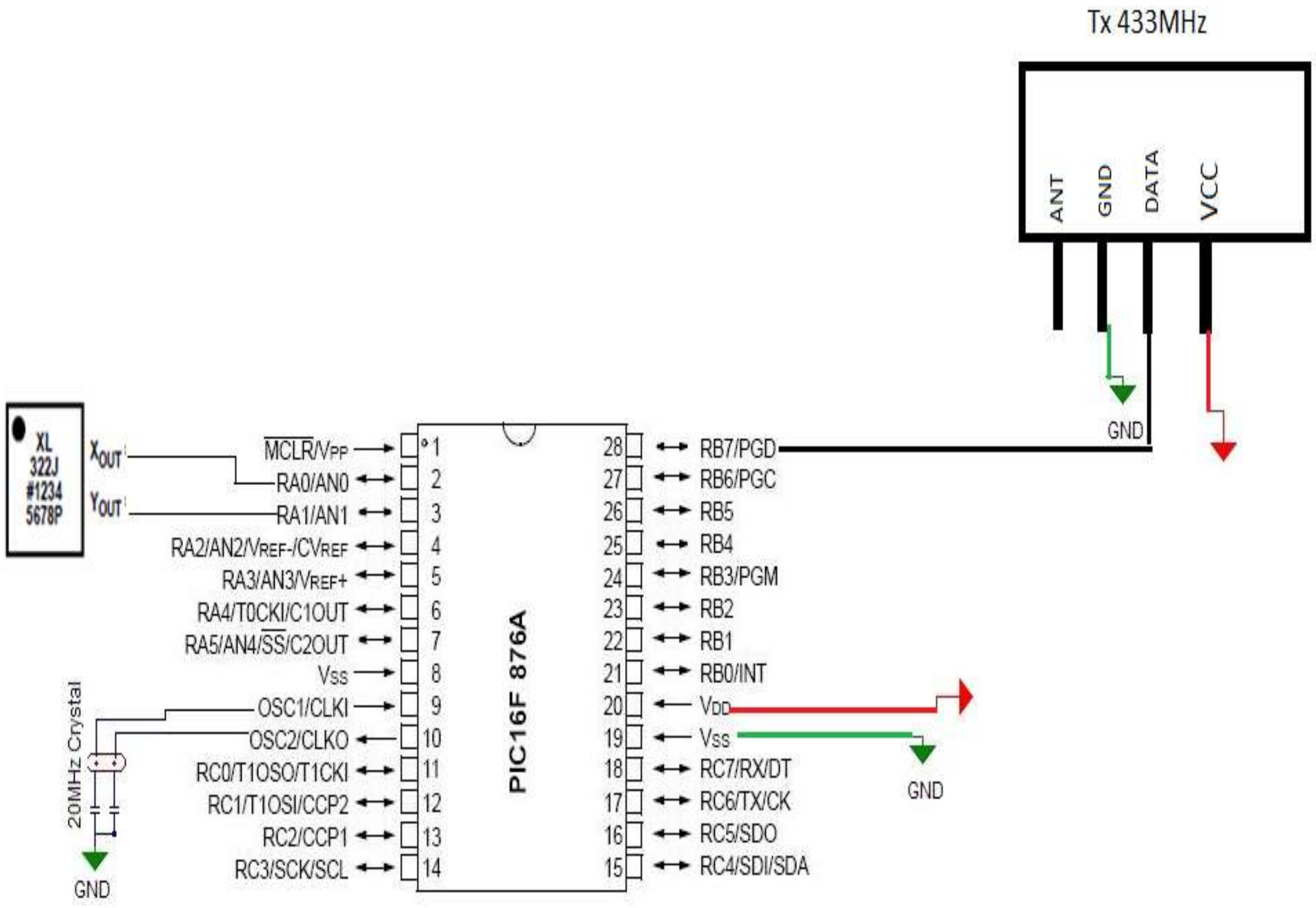
- · Remote Keyless Entry (RKE)
- · Remote Lighting Controls
- · On-Site Paging
- · Asset Tracking
- · Wireless Alarm and Security Systems
- · Long Range RFID
- · Automated Resource Management

a. Transmitting System

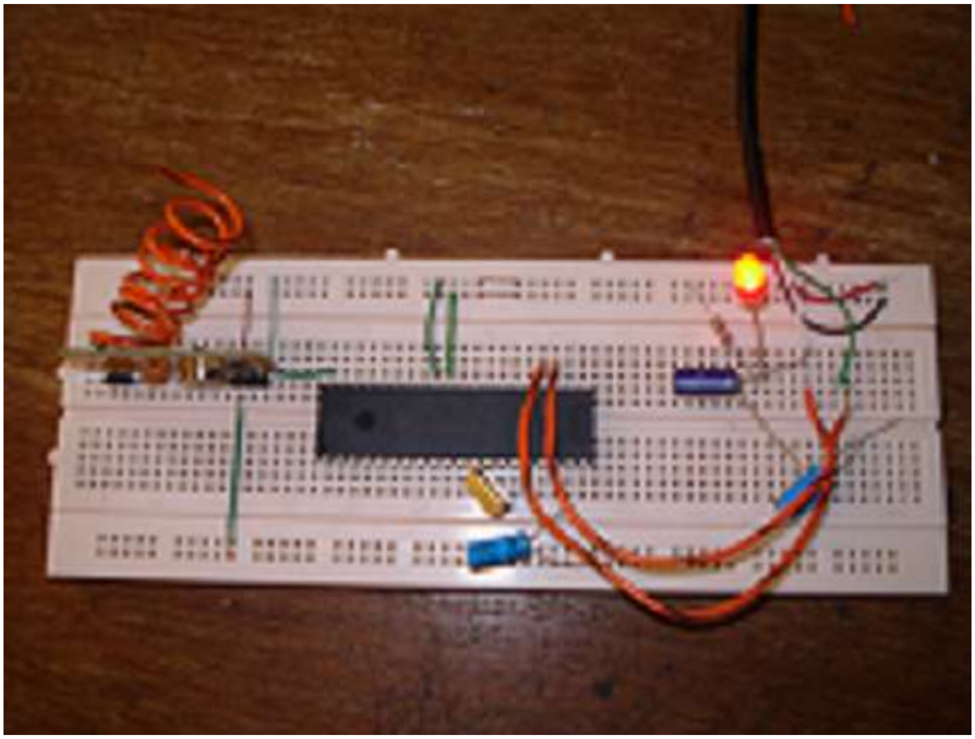


As we can see from the above diagram that the transmission system primarily incorporates the Accelerometer, the microcontroller and the transmitting element. The transmitted signals are sent to be received by the Receiving end. The transmission end through the microcontroller interprets the movement as it is conveyed from the Accelerometer which detects the movement of the system with respect to the conventional x-axis and y-axis and then results in the signal being sent accordingly. The system also sends signals of the clicking that we normally associate with the functioning of a particular interface system.

The diagram below shows the schematic diagram of the microcontroller that showcases the processing of at the various terminals and how the connections have been given to the microcontroller. The codes have been inserted into the microcontroller system using a burner so that it carries out and processes as we have defined according to the coding.

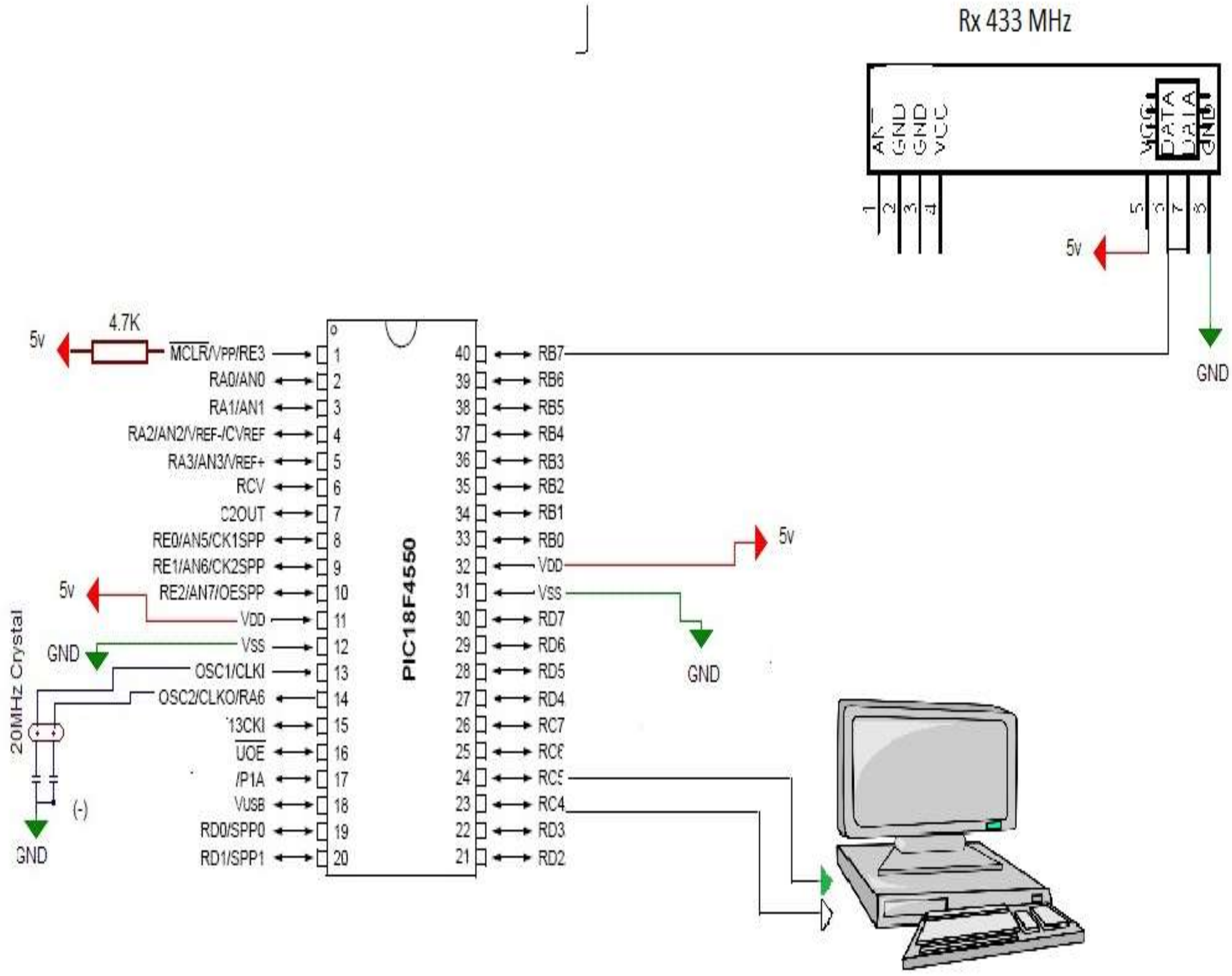


b. Receiving System



As we can observe from the diagram that the receiving end of the system receives the instructions from the transmitter and thus acts as the channel via which the connection is established for the computer to detect the instruction from the transmission system and thus detect the system as an interfacing system to the computer.

As we can see in the schematics of the microcontroller that the receiving system exhibits and it also displays the connection ports. We can observe the various connecting channels that have been used in terms of connecting with the computer. The 20 MHz crystal is used to synchronize the time lapse in context of the microcontroller and thus helps to make the system efficient



Scope and Opportunity of the Project in the Future

In the market, we do come across gaming consoles that allows hand gestures to be recognized but it is still not used as a system of interfacing and can be used to replace the simple mouse that we use normally. In the near future, as we are observing that the interfacing is gradually being taken to the next level of being 3 - dimensional so we intend to make our system being able to incorporate in a system that is compatible to be used in the 3 – dimensional paradigms.

Reference

http://www.dimensionengineering.com/accelerometers.htm

http://alldatasheet.com

http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf

http://nunchuckmouse.blogspot.com/

http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010280

Appendix

a. **Below is the PIC Basic code that works for the USB portal**

```
————————————————————————————————-
Define  OSC     48
'CONFIGURATIONS ARE IN /PBP/18F2550.INC , YOU MIGHT WANT TO EDIT THEM
'ADC
DEFINE ADC_BITS 10        ' Set number of bits in result
DEFINE ADC_CLOCK 6        ' Set clock source  Fosc/64  => TAD => 1.34uS
DEFINE ADC_SAMPLEUS 50     ' Set sampling time in uS
TRISA.0 = 1
TRISA.1 = 1
ADCON1 = %00001101 ' sets AN0,AN1 to analog mode
ADCON2 = %10101110 '
   'bit 0-2: ADCS ,OVERWRITTEN BY ADC_CLOCK,  110: Fosc/64  => TAD => 1.34uS
   'bit 3-5: Aquisition time: 101: 12 TAD =>  16uS
   'bit 6: not used
   'bit 7: Right justify for 10-bit
x    VAR    WORD
y    VAR Word
INTCON2.7=0  'RBPU =0 , TURN ON PORTB PULL-UPS
bt1           VAR PORTB.0
            TRISB.0 = 1
bt2           VAR PORTB.1
            TRISB.1 = 1
bt3           VAR PORTB.2
            TRISB.2 = 1
bt4           VAR PORTB.3
            TRISB.3 = 1
```

```
'USB
buffer    Var    Byte[3]
USBInit
main:
    ADCIN 0,x    'read AN0
    ADCIN 1,y    'read AN1

    'x,y experimental measurments: ~512 @center / ~365 @ -90deg / -655 @ +90deg

    x = (x >> 2) << 2  ' clear last 2 bits
    y = (y >> 2) << 2  ' clear last 2 bits

    'convert x,y from range [384..512..639] to [0..128..255] with edge clipping
    x = ((x MAX 384) MIN 639 ) – 384
    y = ((y MAX 384) MIN 639 ) – 384

    buffer[0] = y    'gamepad's x-axis is accelerometer's y axis
    buffer[1] = x    'gamepad's y-axis is accelerometer's x axis
    buffer[2] = PORTB ^ %00001111 &    %00001111 'separate and reverse first 4 bits
    USBService    ' Must service USB regularly
    USBOut 1, buffer, 3, main  ' Send buffer to endpoint 1
GOTO main    ' Do it forever
```

b. Coding used for the Receiving end of the System (Rx Part) :

```
 #include "RF_Decoder
// Buffers should be in USB RAM, please consult datashe
 unsigned char ReadBuff[16] absolute 0x500;
 unsigned char WriteBuff[16] absolute 0x540;
void main()
 {
   char i, ReceiveData;
  // Enable HID communication
   HID_Enable(&ReadBuff, &WriteBuff);
   while(1)
   {
     // wait until valid data found
     while(GetRfData(&ReceiveData) == 0){}
      WriteBuff[0] = 0;
     // Left/Right movment
     if(ReceiveData.B1 == 1)
     {
       if(ReceiveData.B0 == 0)
       {
         WriteBuff[1] = -4;
       }
       else
       {
         WriteBuff[1] = 4;
       }
     }
     else
     {
       WriteBuff[1] = 0;
 }

 // Up/Down movment
 if(ReceiveData.B3 == 1)
 {
  if(ReceiveData.B2 == 0)
  {
   WriteBuff[2] = 4;
  }
  else
  {
   WriteBuff[2] = -4;
  }
 }
 else
 {
  WriteBuff[2] = 0;
 }
// left/Right click
if(ReceiveData.B5 == 1)
 {
  if(ReceiveData.B4 == 0)
  {
   WriteBuff[0] = 1;
  }
  else
  {
   WriteBuff[0] = 2;
  }
 }
 else
```

```
    {
      WriteBuff[0] = 0;
    }
    HID_Write(&WriteBuff,3);
  }
}
void interrupt()
{
  // USB servicing is done inside the interrupt
  USB_Interrupt_Proc();
}
```

c. Coding used for the Transmitting end of the System (Tx Part) :

```
#  define SW1 RC0_bit
#  define SW2 RC1_bit
#  define SW1_TRIS TRISC0_bit
#  define SW2_TRIS TRISC1_bit
#include "RF_Encoder.h"
void main()
{
  char i, DataToSend;
  ADCON1  = 0x00;            // Configure AN2 pin as analog
  TRISA  = 0xFF;            // PORTA is input
  TRISB  = 0;              // PORTB is output
   SW1_TRIS=1;
  SW2_TRIS=1;
  while(1)
  {
    DataToSend = 0;
    if(ADC_Read(0) < 525)
    {
      // LEFT
      DataToSend |= 0x03;
    }
    if(ADC_Read(0) > 575)
    {
      // RIGHT
      DataToSend |= 0x02;
    }
    if(ADC_Read(1) < 525)
    {
      // DOWN
```

```
     DataToSend |= 0x08;
    }
    if(ADC_Read(1) > 575)
    {
     // UP
     DataToSend |= 0x0C;
    }
      if(SW1==0)
    {
    //Left Click
     DataToSend |= 0x20;
    }
      if(SW2==0)
    {
    //Right Click
     DataToSend |= 0x21;
    }
        SendRfData(DataToSend);
  }
}
```