Wireless Gesture Controlled Pointing Device

Muhammad Asif[1], Rahul Choudhary[2], Sujata Yadav[3], Shanu Sharma[4], Ramona Devi[5], Narinder Singh[6]

[1]Department of Instrumentation and Control Engineering Dr. B. R. Ambedkar National Institute of Technology Jalandhar-144 011 Punjab (India)

[2]Associate Professor, Department of Instrumentation and Control Engineering Dr. B. R. Ambedkar NIT, Jalandhar

Abstract: ***The aim of this project was to create an intuitive glove-based pointing device for multiple applications and to be able to create not just a working project but a fully developed device in terms of intuitive functionality and practical, usable features. It involved working with accelerometers and translates the motion of the hand into various applications in a virtual interface. It is most intuitive for us to use things based on our hand motions, as they form a very basic form of communication, signaling and gesturing. In order to translate these motions into the virtual world we use the accelerometer sensors. The orientation of accelerometers with the ground tells us the orientation our hand is in, because of the components of gravity. Human Computer Interaction is a branch in which developer makes user friendly system. Now days, many persons suffer from repetitive strain injuries due to continuous use of mouse. To avoid such serious problem of injuries and pains, a wearable wireless device is proposed in this project. Due to simple gestures such as left click, right click, and drag operation rotation operation and pointing operation, the proposed device achieves user friendliness and effectively enhances user's interaction with computer. We designed and built a wireless computer pointing device with accelerometer-based movement control. Our implementation allows the user to wear a set of hardware (glove) and control cursor on screen through different hand orientations and finger presses. Users can control their computers with their hands still in mid-air without the hassle of desks surfaces and wires. The proposed system makes the use of accelerometer to sense hand gestures, and according to hand gestures it will perform typical actions such as left click, right click, drag operation, rotation operation and pointing operation. The mouse will be a hand mounted device that maps the movement of the user's hand onto the computer's mouse pointer, having all the standard functionalities as that of a computer mouse. The system is connected to a computer using standard USB and supports Windows, mac and LINUX operating systems. Right movement, Left Movement, right and left clicks are the functionalities implemented.***

Keywords: Human Computer Interaction, Gesture, Accelerometer, Teensy ++ 2.0, Wireless Mouse, RF module-Transmitter and receiver, Arduino UNO (ATMEGA328) microcontroller

1. Introduction

As time is proceeding ahead, technology is improving and evolving every single moment. 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.

We have designed a wireless interface system which will be able to act as an enhanced version of one of the most common interfacing systems 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 three 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.

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 built in code that helps to carry out 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 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 and helps the system to be recognized as an interface between the user and the computer. 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.

1) Logical Structure

At a high level, our design consists of two main parts: a glove and a base station. Operation of our device begins with the glove. A user wearing the glove can use hand tilt orientation and finger presses to operate the glove. The glove senses these user actions via two types of sensors: accelerometers and finger contact pads. After the glove's microcontroller processes the input data, it forwards a message to a transmitter mounted on the glove unit. The transmitter then transmits this message wirelessly to a receiver mounted on the base station. The transmitter forwards the message into a computer HID user friendly format and moves the computer cursor appropriately.

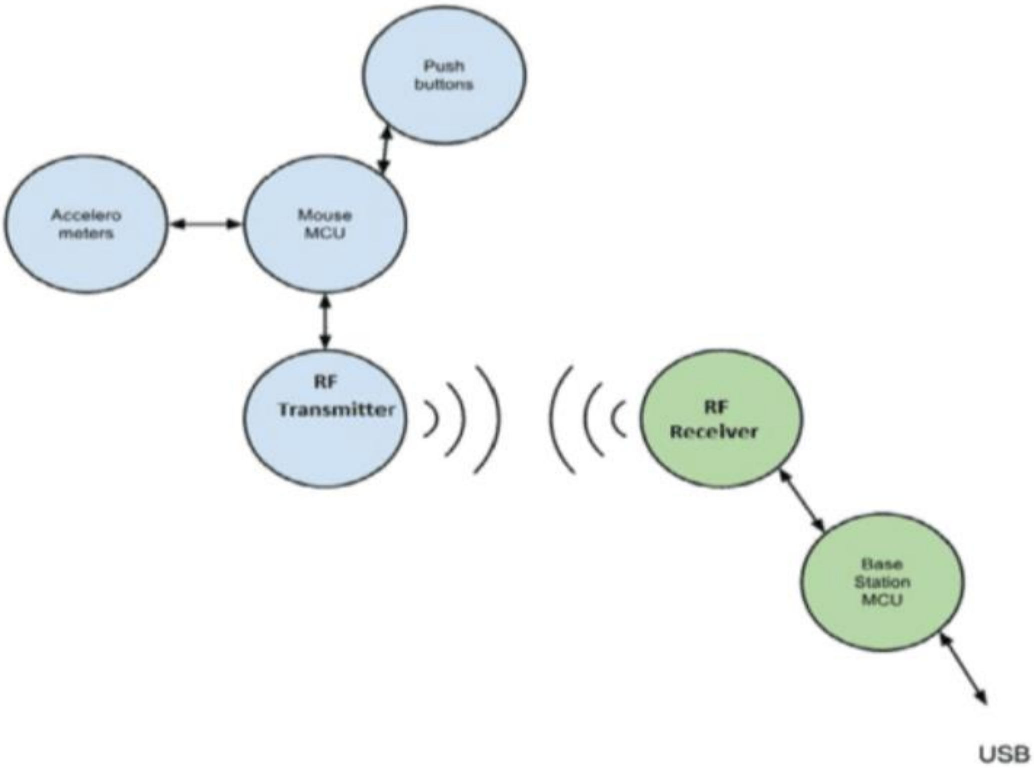


Figure 1: High Level Block diagram of the system.

The block diagram of the system and the communication between glove unit and base station is as follows.

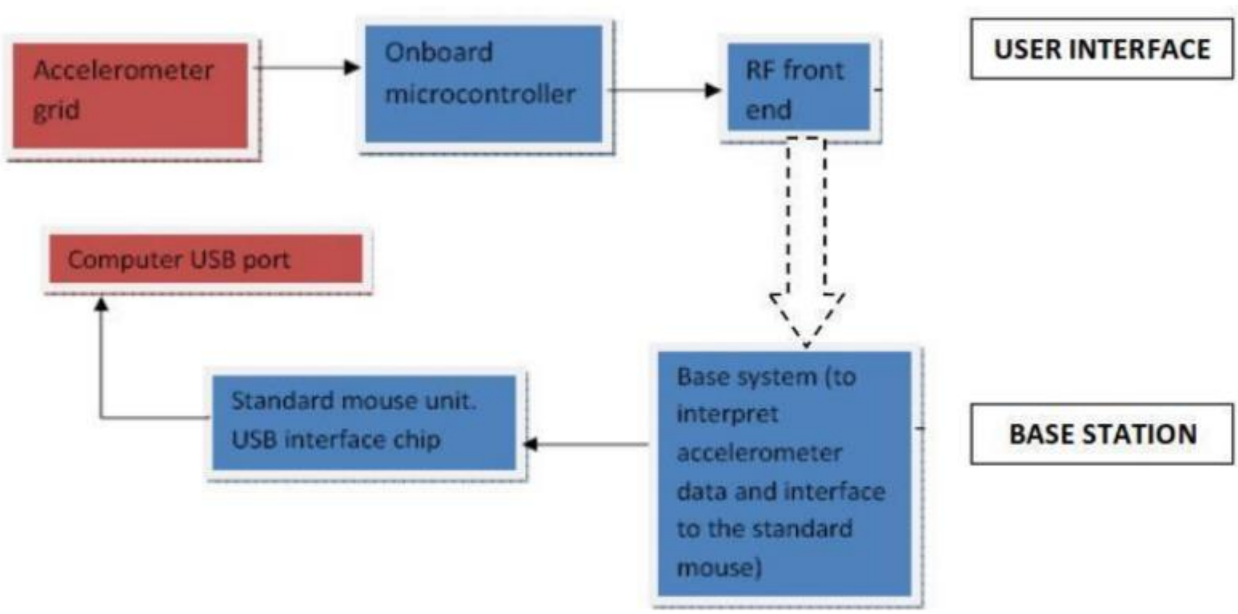


Figure 2: Block Diagram of the system

2) Hardware Design and Implementation

As mentioned above, the physical design of our project includes two main parts: a glove unit and a base station. The glove unit's main purpose is to process information from sensors and transmit the user input to the base station. The base station's purpose is to receive the glove's sent data and forward it to the user's computer in a HID class friendly format. Each unit has its own unique microcontroller and respective peripherals, which are detailed more below.

1.1 Glove Unit

The Glove Unit is the set of hardware that the user physically wears during operation of the Glove Mouse. It carries a Atmega328 microcontroller mounted on a glove. Connected to the glove's microcontroller are 2 switches, a 3-axis accelerometer, a RF Transmitter, a RF Transmitter module with encoder and 8-pin DIP switch.

The user of the Glove Mouse has two primary modes of input as well as one auxiliary mode. The primary modes of input are hand orientation for mouse movement control and buttons for mouse clicks and control over movement enabling. Hand orientation is sensed by the 3-axis accelerometer in terms of tilt, and outputs an analog voltage to pin A.0, A.1, and A.2 of the glove MCU. These pins are connected to the Atmega328's internal analog-to-digital convertors, and are converted to a char value between-128 and 127. Button presses are sensed by 2-pin push buttons placed on the sides of the glove's fingers. The push buttons are connected to pins number 8 and 9 of Atmega328 microcontroller.

In terms of outputs, the Glove Unit's MCU is connected to the input of a RF transmitter module. Specifically, it is connected to pin D.1, D.2, D.3, D.4 of the Atmega328 through encoder HT12E.

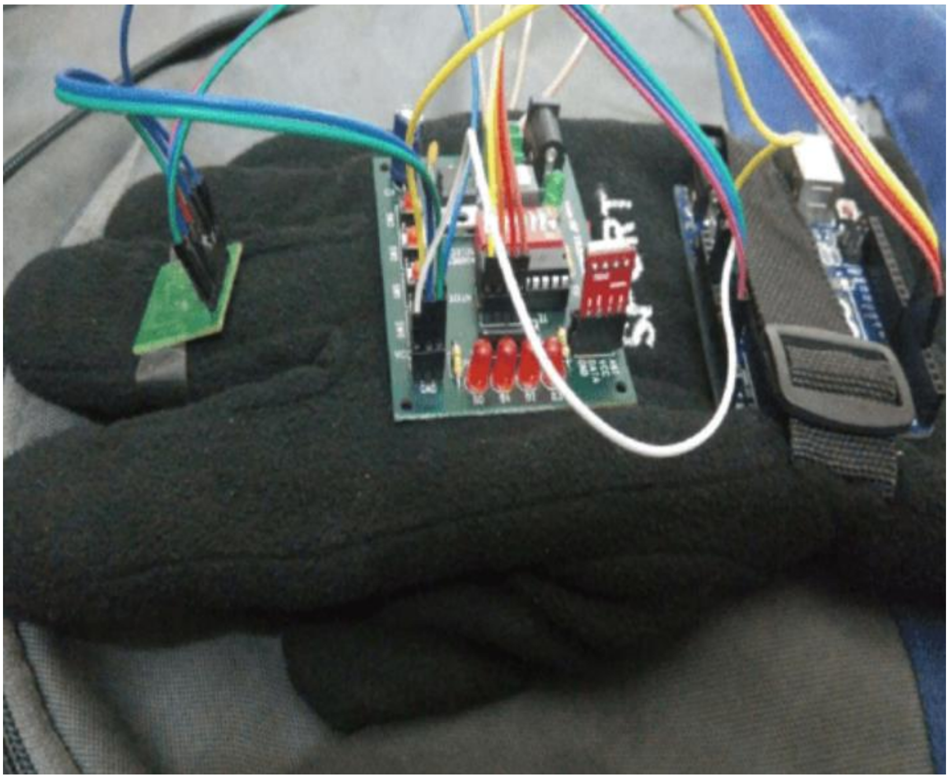


Figure 3: The Glove Unit

A 9V power supply is provided to RF Transmitter module and MCU. This physical design and choice of mounting location by the user is chosen for wearability, comfort, and hardware safety.

1.1.1 Accelerometer

In order to detect and map user hand motions to mouse cursor movement, we chose to use accelerometers to measure hand tilt and orientation. Specifically, we chose to use a Modern Device 3-Axis Accelerometer Module. This chip carries a Freescale MMA7361 three-axis analog accelerometer. The 3-Axis Accelerometer Module additionally features a low-dropout voltage regulator (Microchip MCP1700) to provide 3.3V power to the MMA7361 and 10nF capacitors on the outputs of the accelerometer to minimize clock noise. Because of this, we were able to use the accelerometer module straight out-of-the-box (connecting its power input to Vcc).

1.1.2 Push buttons

We chose to use 2-pin push buttons in order to detect pointing device button commands. We had decided to purchase small push buttons and install them on sides of the glove's finger pads. A push button features two output wires that are shorted together when the button is pressed. Push buttons are connected to glove's design via one lead to ground and the other to a 330Ohm resistor in series with the MCU's input ports.

1.1.3 RF Transmitter

An RF module (radio frequency module) is a small electronic device used to transmit or receive radio signals between two devices. In an embedded system it is often desirable to communicate with another device wirelessly. This wireless communication may be accomplished through optical communication or through radio frequency (RF) communication. For many applications the medium of choice is RF since it does not require line of sight. RF communications incorporate a transmitter and a receiver. They are of various types and ranges. Some can transmit up to 500 feet. RF modules are widely used in electronic design owing to the difficulty of designing radio circuitry. An RF transmitter module is a small PCB subassembly capable of transmitting a radio wave and modulating that wave to carry data. Transmitter modules are usually implemented alongside a micro controller which will provide data to the module which can be transmitted. RF transmitters are usually subject to regulatory requirements which dictate the maximum allowable transmitter power output, harmonics, and band edge requirements. An ISM radio band of 433 MHz frequency RF transmitter is used.

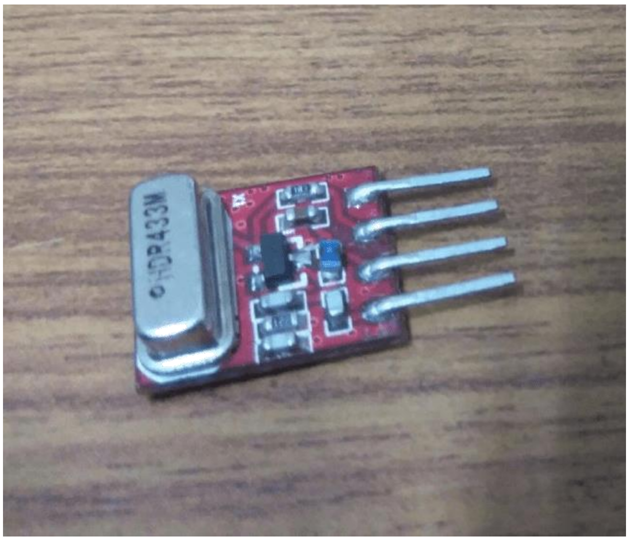


Figure 4: RF transmitter

1.1.4 Transmitter with Encoder module

HT12E is an encoder integrated circuit used in remote control system applications. It is mainly used in interfacing RF and infrared circuits. The chosen pair of encoder/decoder should have same number of addresses and data format.

HT12E converts the parallel inputs into serial output. It encodes the 12 bit parallel data into serial for transmission through an RF transmitter. These 12 bits are divided into 8 address bits and 4 data bits. HT12E has a transmission enable pin which is active low. When a trigger signal is received on TE pin, the programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium. HT12E begins a 4-word transmission cycle upon receipt of a transmission enable. This cycle is repeated as long as TE is kept low. As soon as TE returns to high, the encoder output completes its final cycle and then stops.

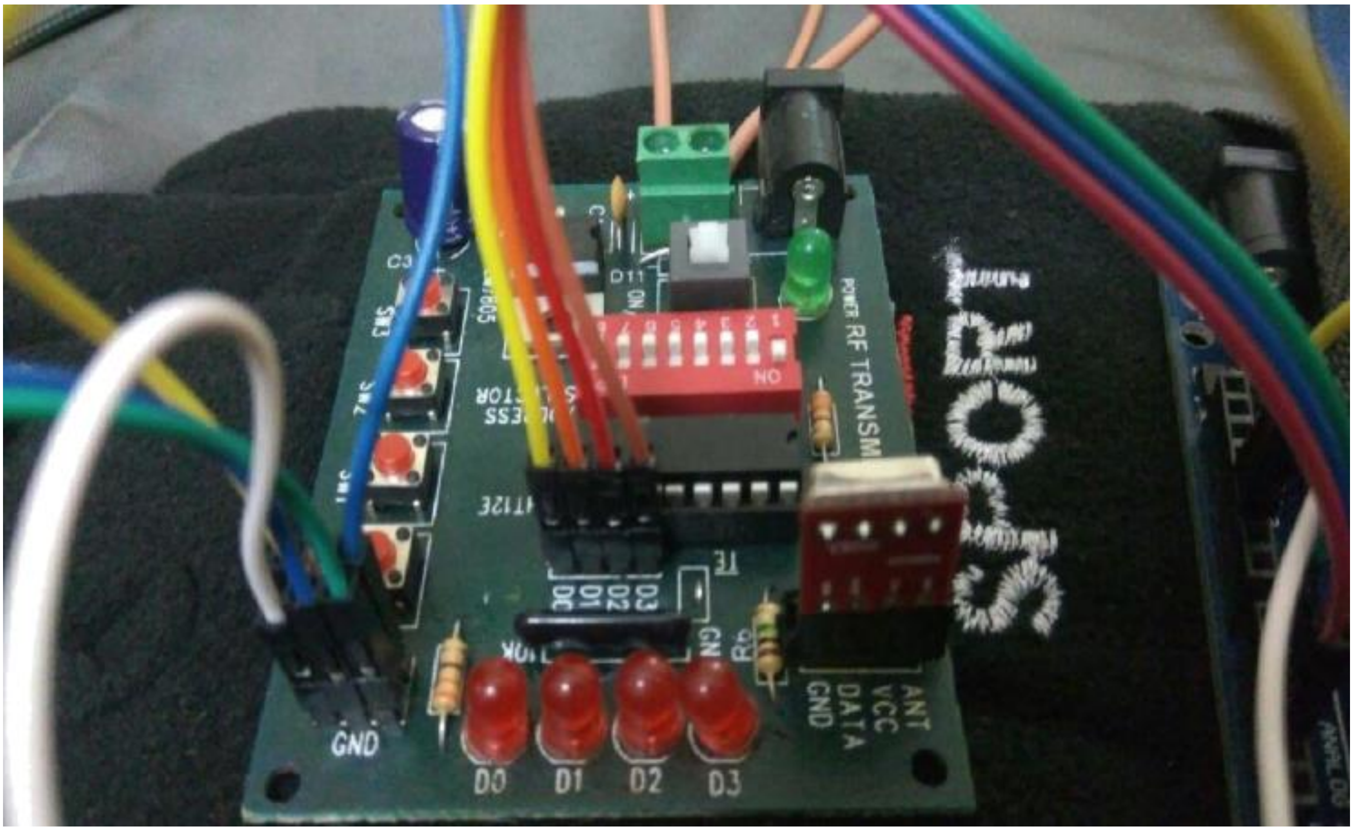


Figure 5: RF transmitter module with encoder

1.1.5 Arduino Uno (ATmega328)

The Arduino UNO is a widely used open-source microcontroller board based on the Atmega328P microcontroller and developed by Arduino. The board is equipped with sets of digital and analog input/output (I/O) pins that can be interfaced to various expansion boards (shields) and other circuits. The board features 14 Digital pins and 6 Analog pins. It is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts. Layout and production files for some versions of the hardware are also available. The Atmega328 on the Arduino Uno comes preprogrammed with a bootloader that allows to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Arduino UNO is generally considered the most user-friendly and popular board or the Arduino board series.

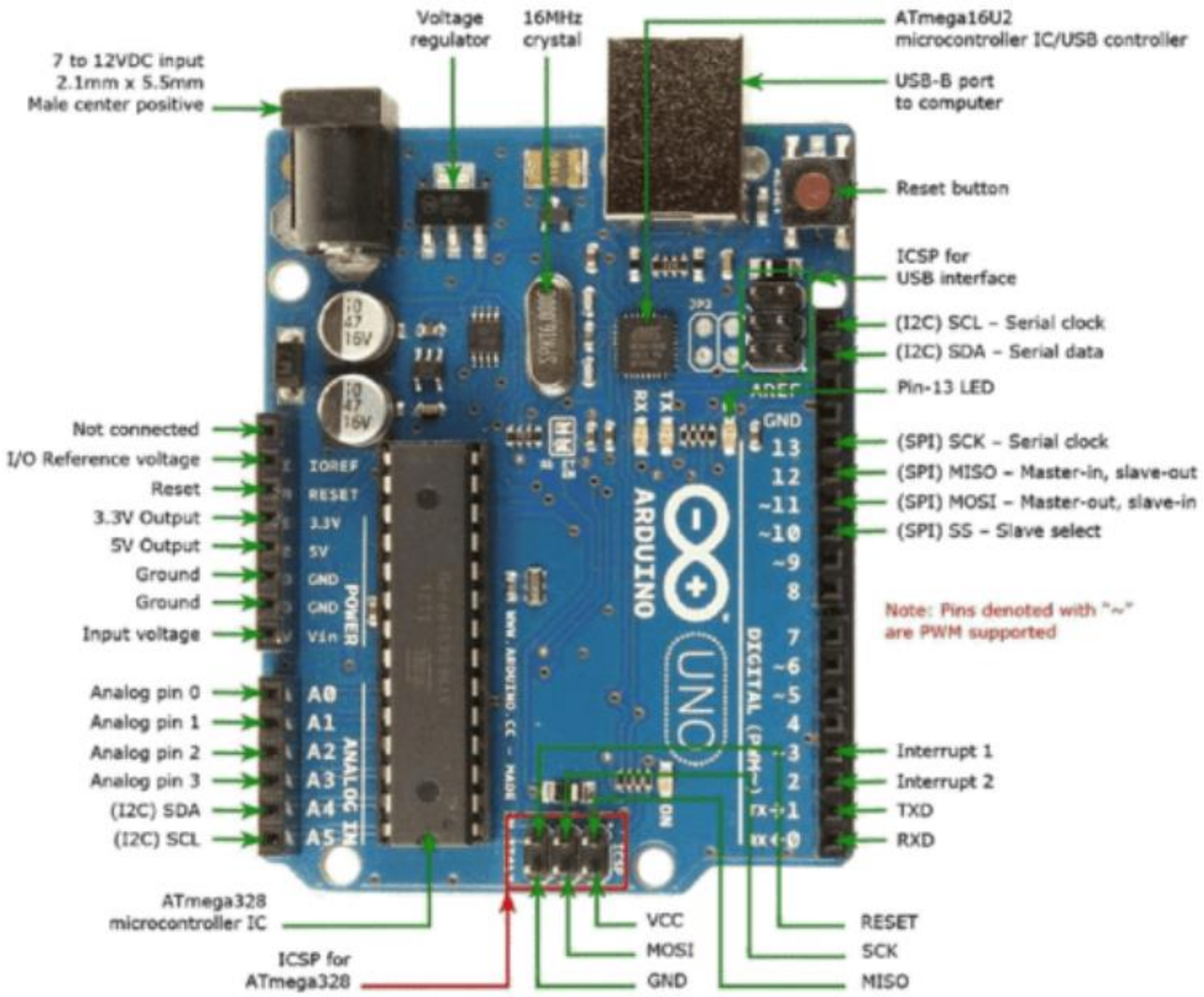


Figure 6: Pin diagram of Arduino UNO

1.2 Base Station

The base station serves one important purpose: to receive packets sent from the Glove Unit, wrap them in a proper HID format, and forward mouse commands to the user's computer. It carries a AT90USB3284 microcontroller mounted on a Teensy++ 2.0 board. Connected to that microcontroller are an RF Receiver and RF receiver module with decoder.

Wired connections on the base station solder board are fairly elementary. The base station microcontroller receives input to RXD1 (pin D.2) from the wireless transceiver's TXD0 output. The wireless transceiver Vcc and ground are connected to MCU Vcc and ground. The MCU is powered via the Teensy's USB mini B plug. For debugging and status updates, one LED is connected to MCU Vcc (so it turns on when the base station is powered) and another is connected to the transceiver-to-MCU wire, so it flickers during message receives.

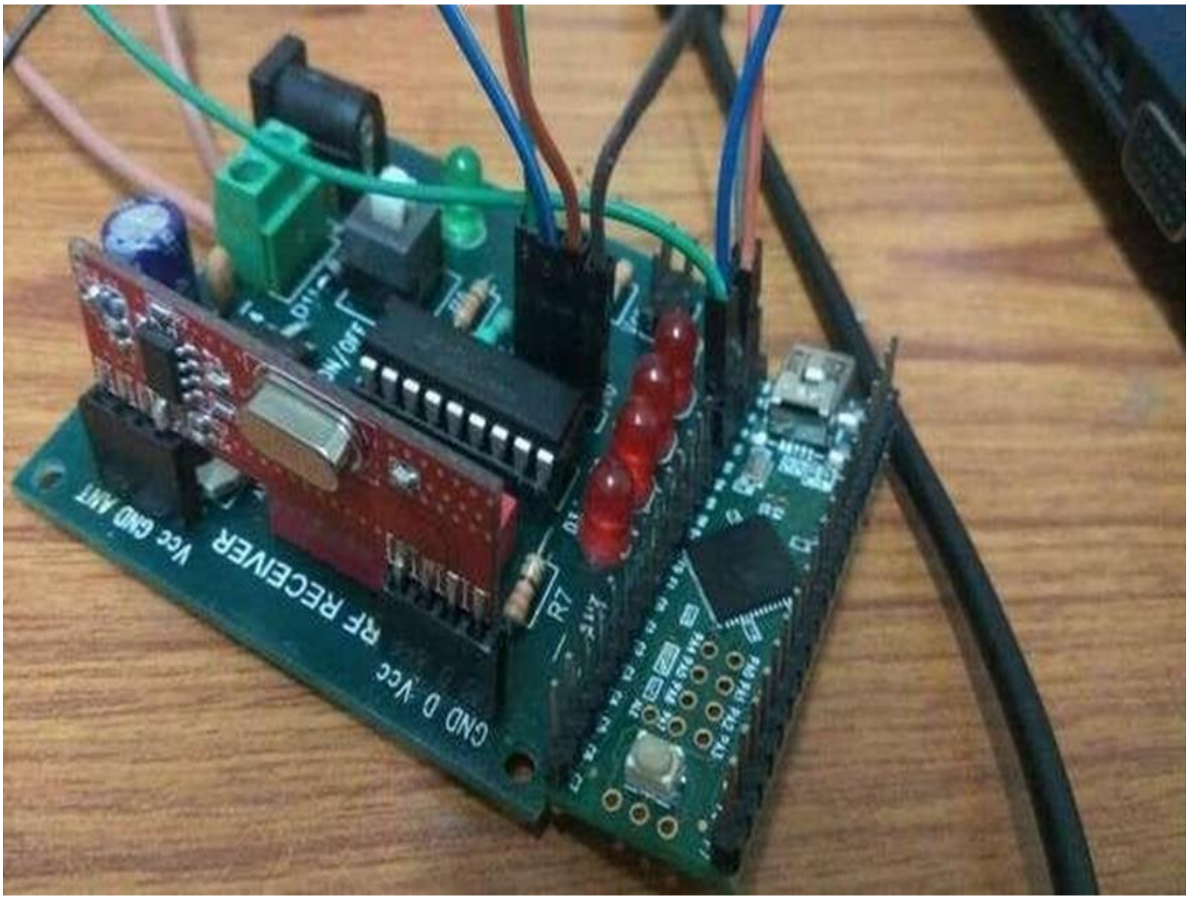


Figure 7: The base station

1.2.1 Teensy++ 2.0

The Teensy++ 2.0 is microcontroller development board supporting USB-based projects. It features a AT90USB1286 microcontroller. We chose to use a Teensy over the duplicate of the glove MCU for several reasons. The main criteria for selecting this particular device were its support for USB communication. This way, we wouldn't have to develop/use additional software for the required USB HID class. Additionally, the Teensy carried a Atmel microcontroller that used very similar specifications to what we were already used to from the Atmega328. Lastly, the Teensy was available as a very small, user friendly package with several supporting example code libraries.

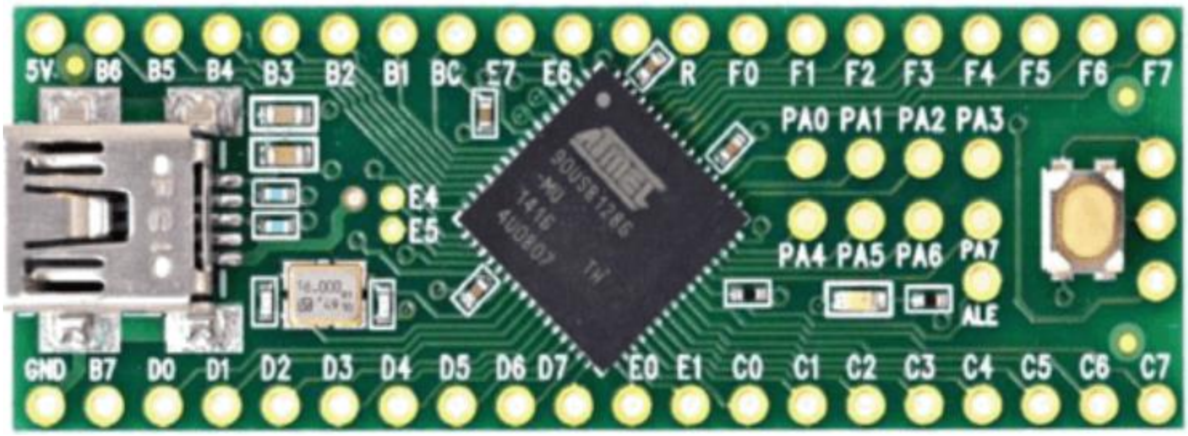


Figure 8: Teensy++2.0

1.2.2 Receiver Module

An RF receiver module receives the modulated RF signal, and demodulates it. There are two types of RF receiver modules: superheterodyne receivers and super-regenerative receivers. Super-regenerative modules are usually low cost and low power designs using a series of amplifiers to extract modulated data from a carrier wave. Super-regenerative modules are generally imprecise as their frequency of operation varies considerably with temperature and power supply voltage. Superheterodyne receivers have a performance advantage over super-regenerative; they offer increased accuracy and stability over a large voltage and temperature range. This stability comes from a fixed crystal design which in the past tended to mean a comparatively more expensive product. However, advances in receiver chip design now mean that currently there is little price difference between superheterodyne and super-regenerative receiver modules. An ISM radio band of 433 MHz frequency RF receiver is used.

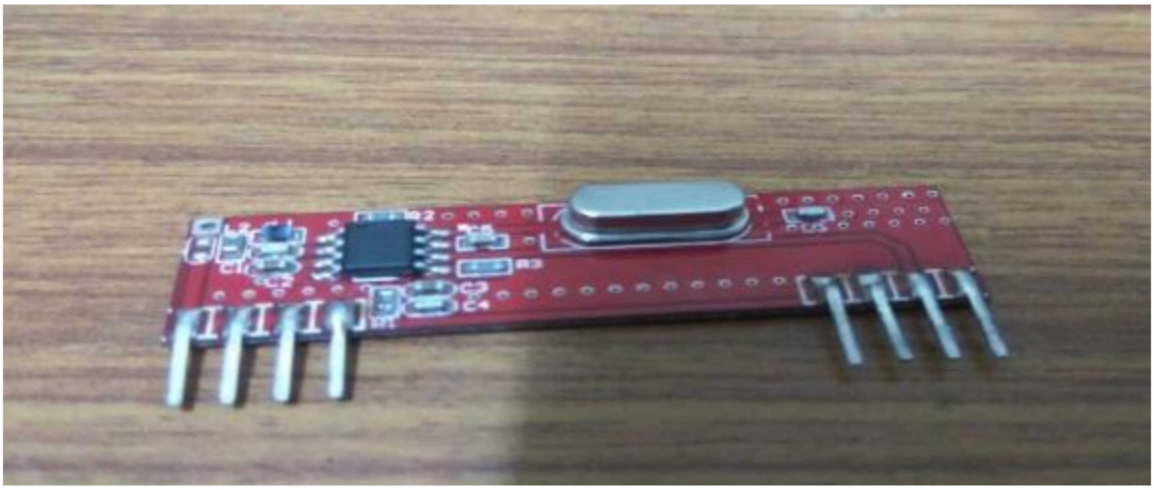


Figure 9: RF receiver

1.2.3 RF Receiver module with decoder

The HT 12D ICs are series of CMOS LSIs for remote control system applications. This ICs are paired with each other. For proper operation a pair of encoder/decoder with the same number of address and data format should be selected. The Decoder receive the serial address and data from its corresponding decoder, transmitted by a carrier using an RF transmission medium and gives output to the output pins after processing the data.

Figure 10: RF Receiver module with decoder

Software Design and Implementation

1.3 Mouse Movement

In this project, a 3-axis accelerometer is used to control cursor movement. The accelerometer reads the tilt of each axis and outputs each as an analog voltage. The three ports of MCU are connected to the tilt outputs of the accelerometer respectively. Then, each of the voltage value of tilt is converted to digital numbers. These numbers are converted to final mouse movement values by noise detection and scaling. In order to provide accurate mouse-motion and a user-friendly interface, several different types of post-processing are applied to the converted ADC values

1.4 Analog-to-Digital Conversion

The Arduino Uno microcontroller contains several single-channel internal analog-to-digital converters (ADC), which each convert an analog input voltage signal to a digital value. The x, y, and z-axis tilt outputs of the accelerometer are connected to the ADC channel 2, 1, and 0. Once the ADC converts the tilt voltage value into digital bits, the digital numbers are stored in ADC data registers. The ADC channel modes are set up such that only the high byte of ADC data register is read. We set the voltage reference to Vcc, and ADC prescalar to 128. The timer 0 compare match ISR (interrupt service routine) is used for regular ADC sampling. The timer 0 prescalar is set to 256 and the compare register is set to 250 time ticks. This allows the program to execute the ISR every 4 milliseconds. In the ISR, acceltime (the time counter for ADC sampling) in decreased.

1.5 Initial Axes Calibration

When the move or scroll is enabled, the calibration function is used to read the current tilt values. This function averages 1024 samples to get reliable offset values, which are used as the starting reference point. By subtracting those offsets from every ADC sample, the user can start at any position and use that hand orientation as the current reference axes. Once calibration is finished, the ADC samples are read in the ISR. The corresponding offset value is subtracted from each sample. Then, each sample is zeroed if its absolute value is smaller than the threshold. We chose the threshold experimentally such that natural user hand quivering is treated as a noise instead of valid mouse movement. Because of the fact calibration should be executed before the program starts to read the ADC samples.

1.6 Hand positioning

Although any starting tilt position is allowed (theoretically), we wanted to consider a special case: handshake position. The hand position is more ergonomic and allows the user easier hand rotation in both directions. In this case, instead of x and y axes, the y and z axes are read out of the accelerometer. This allows the user an up-down motion to change the y cursor position, which is consistent with that of flat position

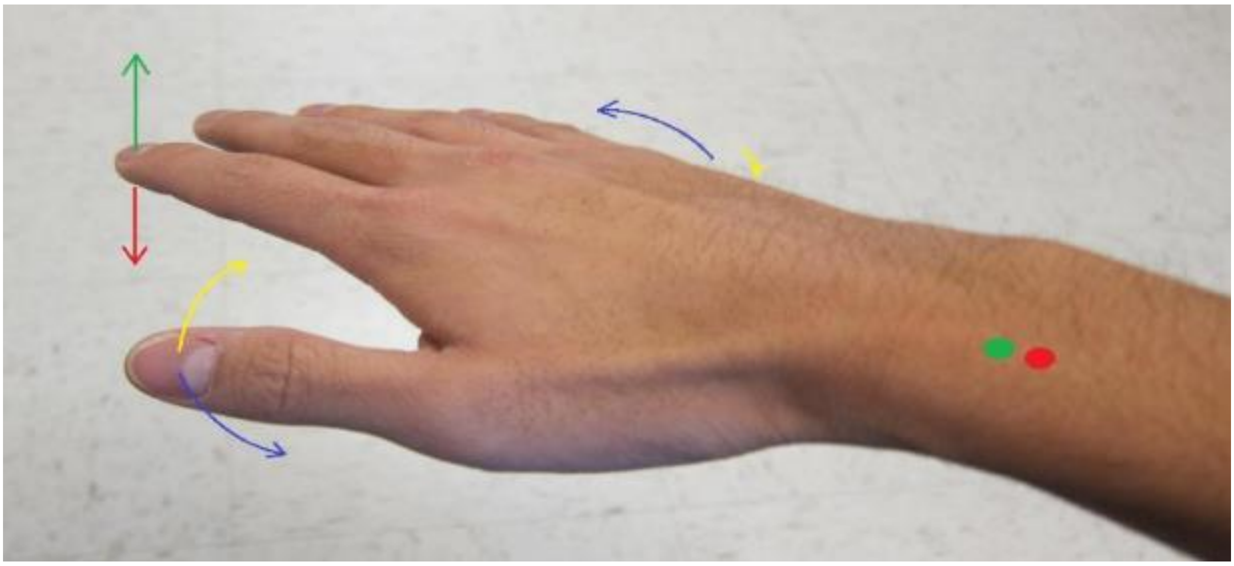


Figure 11: Hand positioning for flat mode. Blue rotation represents left movement, yellow is right, green is up, and red is down (the dot represents pivoting above that point).

1.7 Button Clicking and Switching

Push buttons are connected to the digital pins 8 and 9 for left click and right click. When each button is pressed or released, a packet with the corresponding click information is transmitted to the base station to perform mouse click. When these buttons are pressed and then released, the corresponding flag is toggled. If move enable flag is on, the ADC is sampled and the cursor speed is transmitted to the base station.

Along with the normal button clicks, a rapid fire mode was added as a user option. When the switch for rapid fire mode is on, the mouse is clicked repeated approximately every 60ms while left click is depressed. When left click is pressed and the rapid fire mode switch is on, the flag for sending rapid fire is set. In the ISR, if the rapid-fire flag is on, the rapid fire message is transmitted every 30ms.

Every button presses and release is debounced using a state machine with four states. The state is first initialized as NoPush, where it waits for a button press. The port C pins were set as inputs with respective pull-up resistors turned on. If any button is pressed, the corresponding bit in PINC will be set to low. If a press is detected, the state changes to MaybePush, where any change of PINC value is checked. If current PINC value has changed, then it may not be a valid press and so the state returns to NoPush. If PINC value remained the same, it is a valid press and so the state proceeds to Push. On this transition to the Push state, button press messages are transmitted. The release is debounced similarly, and the button release messages are transmitted or the move/scroll enable flags are toggled on the transition from Push to NoPush. The detailed state diagram is shown below:

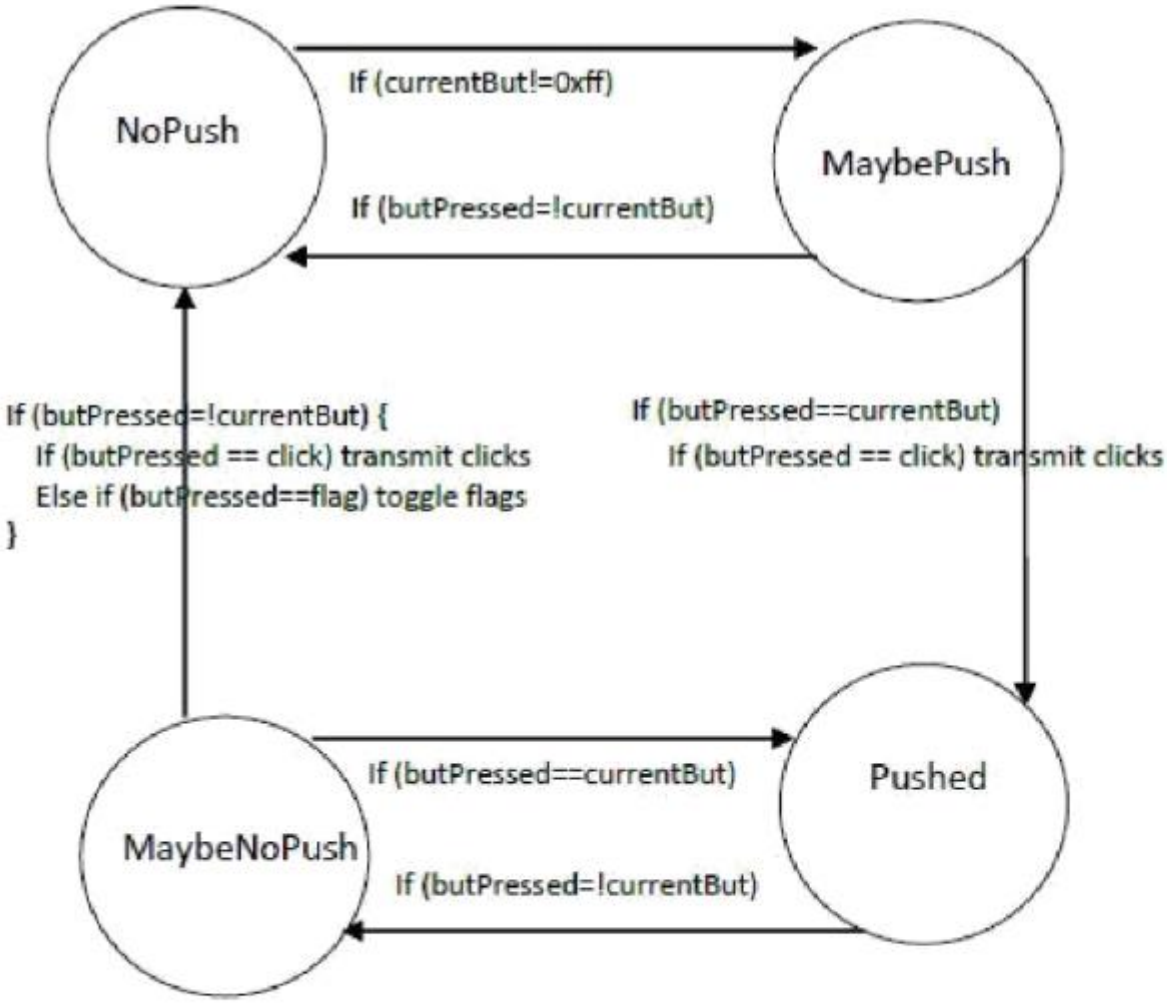


Figure 12: Packet Format between Transmitter and Receiver

1.8 Wireless Communication

For wireless communication between the glove (Atmega 328 microcontroller) and the base station (Teensy 2.0++), RF module-Transmitter and receiver was used. Each RF module and microcontroller communicates through wired connection between USART ports. The glove side MCU converts the user's glove control into a packet. The packet is then loaded onto the RF transmitter and is transmitted to the base station. The RF receiver on the base station receives this packet and passes it to the base station MCU (Teensy 2.0++). The base station MCU finally decodes the packet information and converts it into a real mouse cursor movement or click.

All four units in our project are programmed with a baud rate of 9600 bits/second. The glove MCU and transmitting transceiver pass data to the base station transceiver and MCU respectively. Thus, glove MCU is initialized as transmitter enabled while the base station MCU is initialized as receiver enabled for USART setting. The USART is operated in SPI mode with odd parity check enabled, 2-bit stop bits, and 9-bit data size in a frame.

The glove transmitter sends packets that correspond to the action performed by the user on the glove. The different actions are encoded by the glove MCU as below:

Table 1: Action and Packet Type Encoded by the Transmitter

Action	Packet Type
Tilt	Only if move is enabled by the move enable button and scroll is not enabled, transmit a packet with x and y axis movement every 8 ms.
Scroll	If scroll is enabled, transmit a packet with 1 or -1 as the scroll data every 720ms, depending on the sign of the y tilt position.
Button Click	Transmit a packet with click packet set to corresponding click value. This contains left click, middle click, and right click.
Left Click with Rapid Fire	Only if rapid fire mode is enabled and the left click button is pressed down, transmit a packet with rapid fire mode value every 30 ms.

Table 2: Action and Packet Type Decoded by the Base Station

Packet Type	Action
Tilt or Scroll	Move the cursor or scroll by calling *usb_mouse_move*
Button Click	Press or release corresponding button by calling *usb_mouse_buttons*
Rapid Fire	Click the left click button every 60ms by calling *usb_mouse_buttons* repeatedly whenever the rapid fire packet arrives

1.9 Software Development

1.9.1 Glove Station

```
Setup:
Baudrate (9600)

Loop:
        X ← xpin.read
        Y ← ypin.read
        Z ← zpin.read
        If(x<320):
                Moveright;
        Else if(x<395):
                Moveleft;
        Else if(y>400):
                Moveforward;
        Else if(y<360):
                Movebackward;
      Else:
                Stop;
```

1.9.2 Transmit

```
 Include Virtualwire library
 Initialize:
        Transmission pin ← value
```

```
        Data array for transmission ← data
  Setup:
        Baudrate(9600)
        Virtualwire tx ← true
        Virtualwire setup speed ← 4000
        Virtualwire pin ← pin select
  Loop:
        Data read ← data from pin
        Modifydata()
        Virtualwire send data ← to pin
```

1.9.3 Receive

```
Include virtualwire library
Initialization:
        Variables to hold serial data
Setup:
        Baudrate(9600)
        Virtualwire rx ← true
        Virtualwire setup speed ← 4000
        Virtualwire pin ← pin select
        vw.receiving start()
loop:
        data array ← received
        data; modifyfunction();
```

1.9.4 Base Station

```
Include bounce library (teensydunio)
Initialization:
        Cursor_distance ← 5000
Setup:
        Baudrate(9600)
Loop:
        moveCursor_Right();
        moveCursor_left();
        moveCursor_Up();
    moveCursor_Down();
```

2. Literature Review

The design of wearable computers and accompanying peripherals is an important field of investigations. Despite this, very little work has been published concerning design guidelines for wearable computers that are grounded on experimental evidence. A notable contribution to the field is the work of Gemperle et al. (1998) examining dynamic wearability, that explores the comfortable and unobtrusive placement of wearable forms on the body in a way that does not interfere with movement. The design guidelines presented by Gemperle et al. (1998) are derived from their exploration of the physical shape of the wearable computing devices and pertain to the creation of wearable forms in relation to the human body. The work of Gemperle et al. (1998) produced design guidelines for the on body placement of wearable forms, in contrast to the guidelines derived from this body of work in order to inform the design of wearable pointing devices and serve to assist users in selecting a pointing device suitable for use in a wearable context.

2.1 Input devices for pointing interaction

Pointing is fundamental to most human-computer interactions (Oakley et al., 2008, Andres et al., 2013). A mechanism for continuous pointing suitable for use with wearable computing is vital (Oakley et al., 2008, Chen et al., 2013) as various forms of wearable computing systems are adopted. In addition to the aforementioned commercially available pointing devices, researchers have also explored different means by which to perform command entry and cursor control in a wearable system. Manipulating a two degree of freedom device to control an on-screen cursor is an important design issue for wearable computing input devices. Rosenberg (1998) explores the use of bioelectric signals generated by muscles in order to control an onscreen pointer. Using electromyogram (EMG) data of four forearm muscles responsible for moving the wrist, the Biofeedback Pointer allows the control of a pointer via hand movements (up, down, left, right). The intensity of the wrist movement and therefore generated muscle activity is responsible for controlling the velocity of the pointer. In a user study where three participants carried out point and select tasks, the Biofeedback Pointer was found to perform an average of 14% as well as the standard mouse it was being compared to. An acceleration sensing glove (ASG) was developed by Perng et al. (1999) using six, 2-axis accelerometers with a wrist-mounted controller, RF transmitter and battery pack. Five accelerometers were placed on the fingertips and one on the back of the hand. The accelerometer placed on the back of the hand operated as a tilt-motion detector moving an on-screen pointer. The thumb, index and middle fingers were used to perform mouse clicks by curling the fingers. The glove could also be used to recognize gestures.

2.2 Evaluation of wearable pointing devices

A review of wearable literature reveals that there have been a limited number of empirical studies dedicated exclusively to investigating the usability of pointing devices for wearable computers. Section 2.2 presented novel devices by which to perform command entry and cursor control in a wearable system using specialized hardware and/or software. Typically, the usability of those novel devices was evaluated via formal or informal user evaluations and in some cases the devices were also compared against commercially available devices. In addition to the description of the novel devices, Section 2.2 also presented the results of their evaluation where reported by the researchers. The studies presented in this section however describe the research undertaken that focuses on the evaluation of commercial pointing devices for use with wearable computers. Chamberlain and Kalawsky (2004) evaluate participants performing selection tasks with two pointing systems (touch screen with stylus and off-table mouse) while wearing a wearable computer. Participants were required to don a wearable computer system (Xybernaut MA IV) housed in a vest with a fold down touch screen and had a five minute training session to familiarize themselves with the input devices. Participants did not wear a head-worn display, however were evaluated while wearing a wearable computer in stationary and walking conditions. For the walking condition, participants were required to walk around six cones placed at one meter intervals apart in a figure eight motion. Interestingly, the authors have chosen to evaluate two different interaction methods, where by the stylus does not have an onscreen cursor however, the off-table mouse has an onscreen cursor. In light of this, the results are not surprising. The authors conclude that the stylus was the fastest device (918.3 ms faster than the off-table mouse in the standing condition and 963.7 ms faster than the off-table mouse in the walking condition) and had the lowest cognitive workload (evaluated via NASA-TLX). The mouse took 145.9 ms longer to use while walking than while standing and the 483 stylus took 101 ms longer to use while walking than while standing however, these findings were not statistically significant. The off-table mouse reported the lower error rate (the stylus resulting in 24.2 and 19.7 more errors in the standing and walking conditions respectively).

Witt and Kluge (2008) evaluated three wearable input devices used to perform menu selection tasks in aircraft maintenance. The study evaluated both domain experts and laymen performing short and long menu navigation tasks. The input devices consisted of a pointing device (handheld trackball), a data glove for gesture recognition and speech recognition. For short menu navigation tasks; laymen performed fastest with the trackball, followed by speech then data glove (statistical difference Provisional Design guidelines for wearable pointing devices between the trackball and data glove only)

2.3 Ranking of pointing devices:

The series of studies undertaken (Zucco et al., 2005, Zucco et al., 2006, Zucco et al., 2009) reveal that there is a measurable difference between the pointing devices and that a ranking can be established pertaining to the use of the devices in a wearable context. Table 3 provide a visual summary of pointing device performance for each of the three interaction tasks evaluated. The table is broken up into standing and walking mobility conditions and the solid black lines placed between the devices are to indicate their separate groupings in order to communicate ranking.

Table 3: Summary of device ranking for each usability study (error)

	Error	Target selection	Drag and drop	Menu selection two-level
Standing	Lower	Trackball	Gyroscopic	Trackball
		Twiddler2	Touchpad	Twiddler2
		Gyroscopic	Twiddler2	Touchpad
	Higher		Trackball	Gyroscopic
Walking	Lower		Touchpad	Trackball
			Twiddler2	Touchpad
			Trackball	Twiddler2
	Higher		Gyroscopic	Gyroscopic

2.4 Scope

Scope and Benefits

2.4.1 Design Solutions Comparison

After considering the requirements, some alternative possible design solutions listed as followed: Hotkey, mouse functions and combination trigger

Electric conductive material on critical position: We understand that for a certain gesture, we have an agreement on position of each finger, thus, through the contact of conductive material on different position, we can produce a different voltage potential that representing different gestures. This solution involves lower precision and lower cost as well as lower durability.

Buttons: By pressing one button or a combination of buttons located on different area on the glove, we can also trigger different signal thus the launch of hotkeys. This solution will bring better user experience since the buttons will not be stationary and easily pressed due to the soft nature of our hand and glove. However, the cost and challenge of the design will be lowest.

Flexible potential meter: The use of flexible potential meter will measure the curvature of each finger and return a signal representation. By gathering 5 of these signals, we can recognize what gesture the hand currently has, thus trigger the corresponding hotkey or function. This solution involves high precision and high cost.

2.4.2 Benefits

The new generation control system that remotely controls screen cursors without the demand of a flat and stable surface or physical contact with the screen. It changes the fact that the user's activity is bonded by the location of mouse and keyboard or the location of the screen. As long as that the user is able to see the screen and connect the system through RF module-transmitter and receiver, the user can control the system with single hand and fingers' movement.

The second benefit will be that, as a wearable device, and a device that uses gesture to trigger functions, this will bring more intuitive and attractive gaming experience to gamers. With future support to game designs, we can map the realistic user gestures to in-game operation. For example, pulling a trigger in shooting games will be launched by pulling a trigger gesture with the glove.

For the possible future expansion, we are planning to add compatibility to Google Cardboard VR technology. Nowadays, Virtual Reality is a high focused market that all the giant companies are trying to take a share. Although there is complete independent VR system such as Oculus Rift, the extremely high cost of the system is blocking lots of potential VR customers from entering VR world, not to mention that to support the system, we need an even more costly PC. In contrast, Google's

Cardboard which takes advantage of smartphones' popularity and processing power is a much more acceptable solution. However, when the smartphone is loaded into the Cardboard, there is no way to interact with the phone using the touch screen. Although Google has built an NFC button on the Cardboard that can support basic enter function, we think there is far more to explore and create with the unlimited potential of future VR games and applications.

3. Result

In this proposed system, based on the processed inputs of sensors given by the 3 aixs accelerometer via RF module transmitter and receiver the program at the computer shows respective operation for that particular gesture

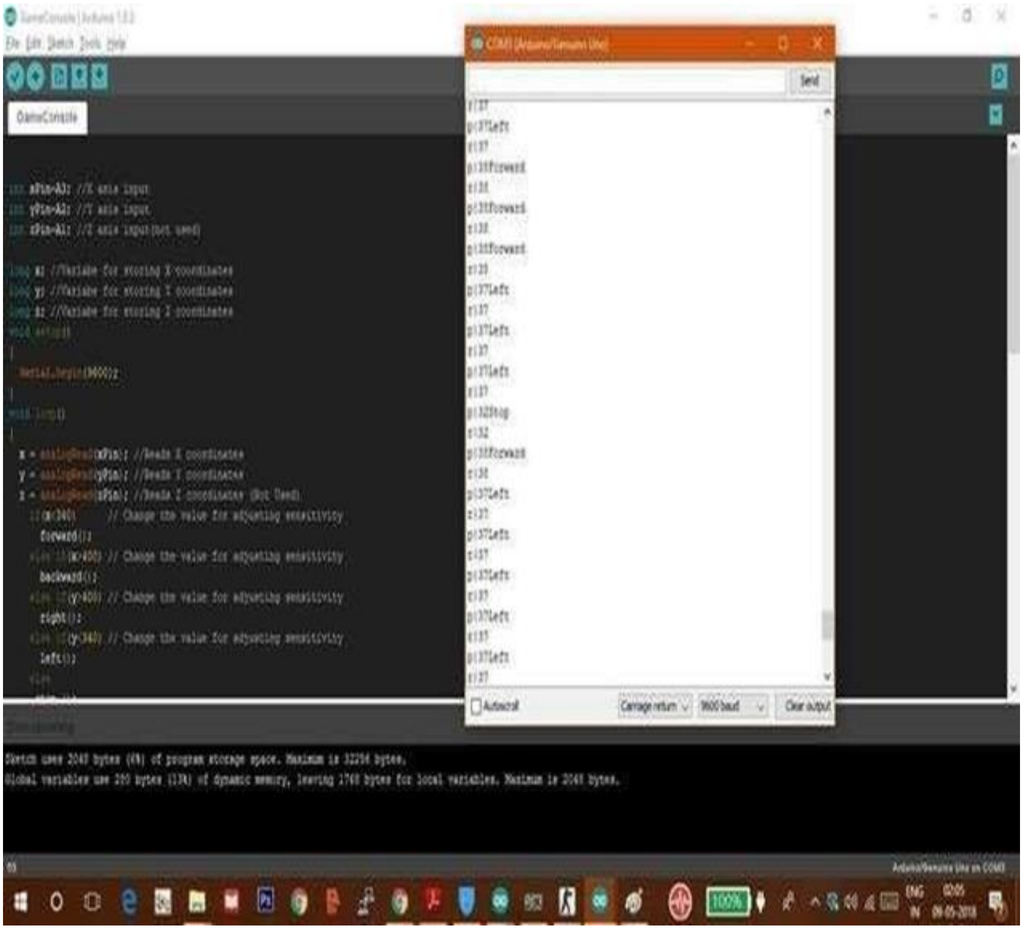


Figure 13: Glove unit Code execution

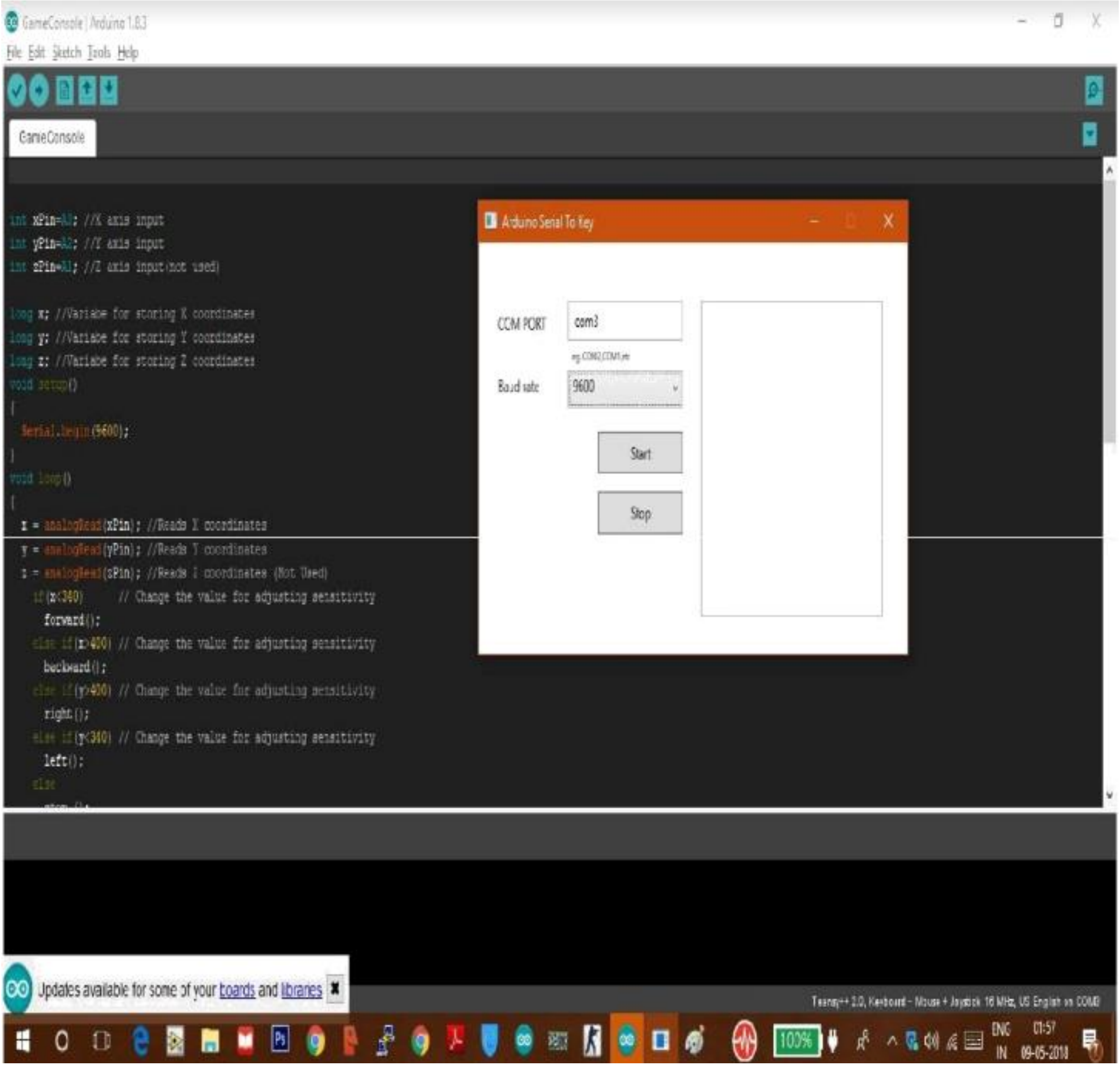


Figure 14: Accelerometer testing

4. Conclusion

Human Computer Interaction Device is key area in modern electronics. The proposed wearable device can be treated as the new age input device.

There are many primary input devices, but these devices are still application specific. The proposed system is easy to use for computer applications that are mouse specific.

Due to simple hand gestures, anyone can use the system conveniently. Therefore, system achieves user friendliness. As this device provides wireless interface, it provides mobility to user.

The proposed device is wearable hand glove unit due to which user does not requires keeping or putting a hand over a mouse and wrist on a mouse pad. Therefore, this wearable hand glove system avoids aches, pains and injuries.

References

[1] 2018. *Lifehacker. Com*. https://lifehacker. com/5973902/hacker-challenge-winner-build-a-wireless-wearable-pointing-device.
[2] 2018. *Lifehacker. Com*. https://lifehacker. com/5973902/hacker-challenge-winner-build-a-wireless-wearable-pointing-

device.
[3] Douglas, Sarah A, Peter J Thomas, and Ray J Paul.1997. *Ergonomics Of Computer Pointing Devices*. London: Springer London.
[4] Allan, Alasdair.2011. *Basic Sensors In Ios*. Sebastopol, CA: O'Reilly.
[5] M. Helander, T. Landauer and P. Prabhu, *Handbook of human-computer interaction*. Amsterdam: Elsevier, 1997.
[6] F. Landragin, *Man-machine dialogue*. London: Iste, 2013.
[7] J. LaViola, *3D user interfaces*.
[8] Allan, *Basic sensors in iOS*. Sebastopol, CA: O'Reilly, 2011.
[9] T. LC, U. Foot, T.3.1, S. Foot and T.2.0, "Getting Started with the Teensy- learn. sparkfun. com", *Learn. sparkfun. com*, 2018. [Online]. Available: https://learn. sparkfun. com/tutorials/getting-started-with-the-teensy. [Accessed: 06-Jun-2018].
[10] Y. D. Kataware and U. L. Bombale, "A Wearable Wireless Device for Effective Human Computer Interaction", *International Journal of Computer Applications*, vol.99, no.9, pp.9-14, 2014.
[11] C. Baber, "Wearable Computers: A Human Factors Review", *International Journal of Human-Computer Interaction*, vol.13, no.2, pp.123-145, 2001.
[12] Aaras and O. Ro, "Workload When Using a Mouse as an Input Device", *International Journal of Human-Computer Interaction*, vol.9, no.2, pp.105-118, 1997.
[13] L. Zeng, "Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Edition) by B. Shneiderman and C. Plaisant", *International Journal of Human-Computer Interaction*, vol.25, no.7, pp.707-708, 2009.
[14] M Lind, "A De-Construction of Wireless Device Usage", *International Journal of Technology and Human Interaction*, vol.3, no.2, pp.34-44, 2007.
[15] J. Zucco and B. Thomas, "Design Guidelines for Wearable Pointing Devices", *Frontiers in ICT*, vol.3, 2016.
[16] "Exploring the Connection between Wearable Computing and Affective Computing-Design Guidelines for Affective Wearable Systems-", *Journal of Digital Design*, vol.9, no.4, pp.411-420, 2009.

Appendix

Teensy++ 2.0
Features

a) High Performance, Low Power AVR® 8-Bit Microcontroller
b) Advanced RISC Architecture - 135 Powerful Instructions - Most Single Clock Cycle Execution - 32 x 8 General Purpose Working Registers - Fully Static Operation - Up to 16 MIPS Throughput at 16 MHz - On-Chip 2-cycle Multiplier
c) Non-volatile Program and Data Memories - 64/128K Bytes of In-System Self-Programmable Flash
d) Endurance: 100, 000 Write/Erase Cycles - 4K/8K (64K/128K Flash version) Bytes Internal SRAM - Up to 64K Bytes Optional External Memory Space - Programming Lock for Software Security
e) USB 2.0 Full-speed/Low-speed Device and On-The-Go Module - Complies fully with: - Universal Serial Bus Specification REV 2.0 - On-The-Go Supplement to the USB 2.0 Specification Rev 1.0 - Supports data transfer rates up to 12 Mbit/s and 1.5 Mbit/s
f) USB Full-speed/Low Speed Device Module with Interrupt on Transfer Completion - Fully independent 832 bytes USB DPRAM for endpoint memory allocation - Suspend/Resume Interrupts
   - Power-on Reset and USB Bus Reset - 48 MHz PLL for Full-speed Bus Operation
g) Peripheral Features - Two 8-bit Timer/Counters with Separate Rescaled and Compare Mode
   - Real Time Counter with Separate Oscillator - Four 8-bit PWM Channels - Six PWM Channels with Programmable Resolution from 2 to 16 Bits - Output Compare Modulator - 8-channels, 10-bit ADC - Programmable Serial USART - Master/Slave SPI Serial Interface- Byte Oriented 2-wire Serial Interface - Programmable Watchdog Timer with Separate On-chip Oscillator - On-chip Analog Comparator- Interrupt and Wake-up on Pin Change
h) Special Microcontroller Features - Power-on Reset and Programmable Brown-out Detection
   - Internal Calibrated Oscillator - External and Internal Interrupt Sources - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
i) I/O and Packages - 48 Programmable I/O Lines - 64-lead TQFP and 64-lead QFN
   - Operating Voltages - 2.7-5.5V
   - Operating temperature - Industrial (-40°C to +85°C)
   - Maximum Frequency - 8 MHz at 2.7V-Industrial range - 16 MHz at 4.5V-Industrial range

Arduino Uno Features

Microcontroller ATmega328
Operating Voltage 5V Input Voltage (recommended) 7-12V
Input Voltage (limits) 6-20V
Digital I/O Pins 14 (of which 6 provide PWM output)
Analog Input Pins 6
DC Current per I/O Pin 40 mA
DC Current for 3.3V Pin 50 mA
Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM 2 KB (ATmega328)
EEPROM 1 KB (ATmega328)
Clock Speed 16 MHz

433 MHz RF Module - Transmitter and receiver

Specifications:

j) Wireless (RF) Simplex Transmitter and Receiver
k) Receiver Operating Voltage: 3V to 12V
l) Receiver Operating current: 5.5mA
m) Operating frequency: 433 MHz
n) Transmission Distance: 3 meters (without antenna) to 100 meters (maximum)
o) Modulating Technique: ASK (Amplitude shift keying)
p) Data Transmission speed: 10Kbps
q) Circuit type: Saw resonator
r) Low cost and small package

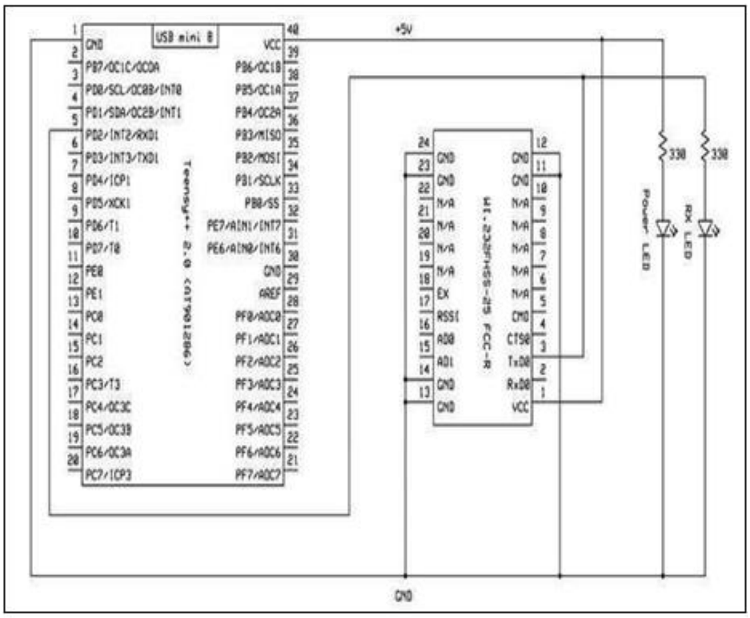
Encoder-HT12E

s) 18 PIN DIP
t) Operating Voltage: 2.4V ~ 12V
u) Low Power and High Noice Immunity CMOS Technology
v) Low Standby Current and Minimum Transmission Word
w) Built-in Oscillator needs only 5% Resistor
x) Easy Interface with and RF or an Infrared transmission medium
y) Minimal External Components

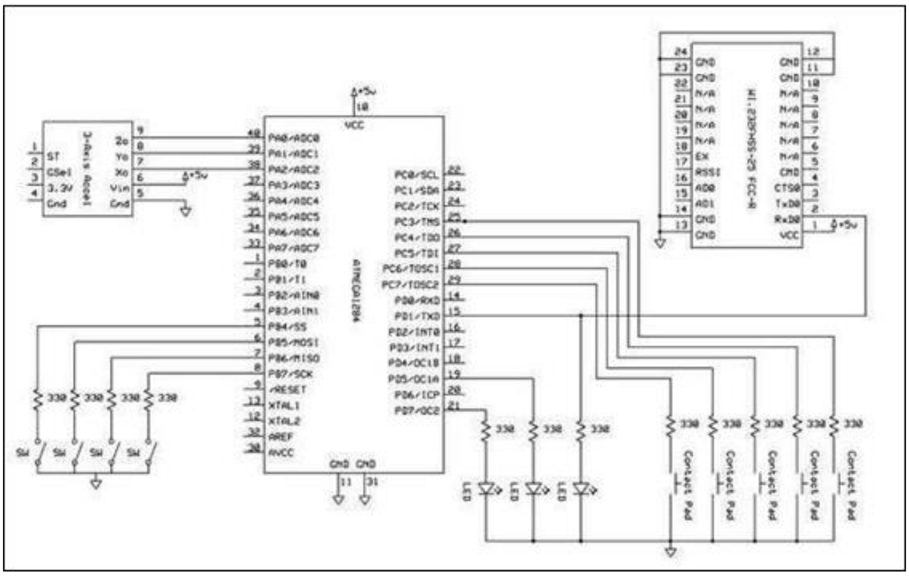
Decoder-HT12D

z) 18 PIN DIP, Operating Voltage: 2.4V ~ 12.0V
aa) Low Power and High Noice Immunity, CMOS Technology
bb) Low Stand by Current, Trinary address setting
cc) Capable of Decoding 12 bits of Information
dd) ~ 12 Address Pins and 0 ~ 4 Data Pins
ee) Received Data are checked 2 times, Built in Oscillator needs only 5% resistor
ff) VT goes high during a valid transmission
gg) Easy Interface with an RF of IR transmission medium
hh) Minimal External Components

Referred circuit: BASE STATION



Referred Glove Unit


3-Axis Accel
ST
CSel
3.3V
Gnd
Zo
Yo
Xo
Vin
Gnd
+5v
VCC
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0
PB1/T1
PB2/AIN0
PB3/AIN1
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
/RESET
XTAL1
XTAL2
AREF
AVCC
ATMEGA1284
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP
PD7/OC2
GND
GND
330
SW
LED
Contact Pad
EX
RSSI
AD0
AD1
CTS0
TxD0
RxD0