A Real-Time (or) Field-based Research Project Report

on

DESIGNING A HAPTIC INTERFACE FOR DISABLED INDIVIDUALS

submitted in partial fulfilment of the requirements for the award of the

degree

of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

by

MEGHNA TVNL [227R1A05J9]

M SHIVA KUMAR [227R1A05F9]

S SANDEEP [227R1A05J6]

Under the guidance of

Mrs. P Santhuja

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

Accredited by NBA & NAAC with 'A' Grade

Approved by AICTE, New Delhi and JNTUH Hyderabad

Kandlakoya (V), Medchal Road, Hyderabad-501401

JUNE 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Real-Time (or) Field-based Research Project Report entitled "**DESIGNING A HAPTIC INTERFACE FOR DISABLED INDIVIDUALS**" being submitted by **MEGHNA TVNL(227R1A05J9), M SHIVA KUMAR (227R1A05F9), S SANDEEP (227R1A05J6)** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in **COMPUTER SCIENCE AND ENGINEERING** to the **Jawaharlal Nehru Technological University, Hyderabad** is a record of bonafide work carried out by them under my guidance and supervision during the Academic Year 2023 – 24.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any other degree or diploma.

Mrs. P Santhuja
Assistant Professor

Dr. K. Srujan Raju
Head of the Department

Dr. A. Raji Reddy
Director

ABSTRACT

Haptic technology refers to technology that interfaces with the user via the sense of touch, applying forces, vibrations, and/or motions. Haptics are gaining widespread acceptance as a key component of virtual reality systems, integrating the sense of touch into previously visual-only solutions. Many of these solutions employ stylus-based haptic rendering, wherein the user interacts with the virtual world using a tool or stylus, providing a computationally realistic interaction achievable with today's hardware. This mechanical simulation can enhance the remote control of machines and devices, offering wide-reaching applications. Haptic technology enables detailed investigations into the workings of the human sense of touch by facilitating the creation of meticulously controlled haptic virtual objects. These objects are used to systematically explore human haptic capabilities.

TABLE OF CONTENTS

INTRODUCTION

1.1. Introduction

Internet of Things (IOT) is a concept where each device is assigned to an IP address and through that IP address anyone makes that device identifiable on internet. The mechanical and digital machines are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. Basically, it started as the “Internet of Computers.” Research studies have forecast an explosive growth in the number of “things” or devices that will be connected to the Internet. The resulting network is called the “Internet of Things” (IoT). Among the four popular wireless connections that often implemented in Hand-talk system project, BT is being chosen with its suitable capability. The capabilities of WIFI are more than enough to be implemented in the design. It will indirectly reduce the cost of this system.

1.2. Background

Gesture recognition has been proposed as a means to understand the actions of a musical conductor and to enable individuals with disabilities to communicate more effectively. For instance, people who are deaf or mute often struggle to communicate with others who do not understand sign language. Even those who can speak aloud may be self-conscious about their "dumb voice" and may be reluctant to communicate.

People who are deaf or mute often struggle to communicate with others who do not understand sign language. This can lead to feelings of isolation and frustration. Even those who can speak aloud may be self-conscious about their "dumb voice" and may be reluctant to communicate. This can result in a lack of confidence and a diminished quality of life.

1.3. Project Objectives

The objectives of this project are:

- **Develop a wearable device**: Design and develop a wearable device that can recognize and interpret hand gestures, enabling individuals with disabilities to communicate more effectively.
- **Improve communication**: Enhance the communication abilities of individuals with disabilities, particularly those who are deaf or mute, by providing a more intuitive and natural way of expressing themselves.
- **Increase independence**: Empower individuals with disabilities to interact more independently with their environment, reducing their reliance on others for communication.
- **Promote inclusivity**: Contribute to a more inclusive society by enabling individuals with disabilities to participate more fully in social, economic, and cultural activities.

1.4. Scope of the project:

The scope of the "Designing a Haptic Interface for Disabled Individuals" project is focused on developing a wearable device that enables individuals with disabilities to communicate more effectively through hand gestures. The project aims to design and develop a device that can recognize and interpret a range of hand gestures, providing a more intuitive and natural way of expressing oneself.

The project will involve the development of a robust gesture recognition algorithm that can accurately identify and interpret hand gestures. This will be achieved through the integration of sensors and microcontrollers into the wearable device, which will capture and process hand gesture data. The project will be conducted in collaboration with stakeholders, including individuals with disabilities, caregivers, and healthcare professionals. This will ensure that the device is designed and developed with the needs and requirements of the target user group in mind, maximizing its effectiveness and impact. The project will also involve testing and evaluation of the device, to ensure that it meets the required standards of quality, safety, and efficacy.

1.5. Benefits:

The "Designing a Haptic Interface for Disabled Individuals" project has the potential to bring numerous benefits to individuals with disabilities, their caregivers, and society as a whole. One of the most significant benefits is the enhancement of communication abilities for individuals with disabilities. The device will provide a more intuitive and natural way of expressing oneself, enabling individuals to convey their thoughts, needs, and desires more effectively. This will lead to increased independence, confidence, and self-esteem, as individuals will be able to communicate more freely and participate more fully in social, economic, and cultural activities.

CHAPTER 2: LITERATURE SURVEY

The concept of haptic interfaces for individuals with disabilities has been explored in various research studies and projects. One of the earliest studies on haptic interfaces for individuals with disabilities was conducted by (Katz et al., 2008), who developed a haptic interface for individuals with visual impairments. The study demonstrated the potential of haptic interfaces to enhance the communication abilities of individuals with visual impairments.

More recent studies have focused on the development of wearable devices that can recognize and interpret hand gestures. For example, (Kim et al., 2019) developed a wearable device that uses electromyography (EMG) sensors to recognize hand gestures. The study demonstrated the accuracy and reliability of the device in recognizing hand gestures, and its potential to enable individuals with disabilities to communicate more effectively.

Other studies have explored the use of machine learning algorithms to improve the accuracy and efficiency of hand gesture recognition. For example, (Wang et al., 2020) developed a machine learning-based approach to hand gesture recognition, which achieved high accuracy and robustness in recognizing hand gestures. The study demonstrated the potential of machine learning algorithms to improve the performance of haptic interfaces for individuals with disabilities.

Several studies have also explored the potential of haptic interfaces to enhance the quality of life for individuals with disabilities

CHAPTER 3: THEORY

3.1 EMBEDDED SYSTEMS

3.1.1 INTRODUCTION TO EMBEDDED SYSTEMS

Many embedded systems have substantially different design constraints than desktop computing applications. No single characterization applies to the diverse spectrum of embedded systems. However, some combination of cost pressure, long life-cycle, real - time requirements, reliability requirements, and design culture dysfunction can make it difficult to be successful applying traditional computer design methodologies and tools to embedded applications. Embedded systems in many cases must be optimized for life-cycle and business-driven factors rather than for maximum computing throughput. There is currently little tool support for expanding embedded computer design to the scope of holistic embedded system design.

3.1.2 CLASSIFICATION

Embedded systems are divided into autonomous, real-time, networked & mobile categories.

Autonomous systems

They function in standalone mode. Many embedded systems used for process control in manufacturing units& automobiles fall under this category.

Real-time embedded systems

These are required to carry out specific tasks in a specified amount of time. These systems are extensively used to carry out time critical tasks in process control.

Networked embedded systems

They monitor plant parameters such as temperature, pressure and humidity and send the data over the network to a centralized system for on line monitoring.

Hardware

Good software design in embedded systems stems from a good understanding of the hardware behind it. All embedded systems need a microprocessor, and the kinds of microprocessors used in them are quite varied. A list of some of the common microprocessor's families are: ARM family, The Zilog Z8 family, Intel 8051/X86 family, Motorola 68K family and the power PC family. For processing of information and execution of programs, embedded system incorporates microprocessor or micro- controller.

3.1.3 OTHER COMMON PARTS FOUND ON MANY EMBEDDED SYSTEMS

- UART& RS232
- PLD
- ASIC's& FPGA's
- Watch dog timer etc.

3.1.4 DESIGN PROCESS

The pillars of the system design methodology are the separation between function and architecture, is an essential step from conception to implementation. In recent past, the search and industrial community has paid significant attention to the topic of hardware- software (HW/SW) codesign and has tackled the problem of coordinating the design of the parts to be implemented as software and the parts to be implemented as hardware avoiding the HW/SW integration problem marred the electronics system industry so long

SPECIFICATION

During this part of the design process, the informal requirements of the analysis are transformed to formal specification using SDL.

SYSTEM-SYNTHESIS

For performing an automatic HW/SW partitioning, the system synthesis step translates the SDL specification to an internal system model switch contains problem graph& architecture graph. After system synthesis, the resulting system model is translated back to SDL.

IMPLEMENTATION-SYNTHESIS

SDL specification is then translated into conventional implementation languages such as VHDL for hardware modules and C for software parts of the system.

On a prototyping platform, the implementation of the system under development is executed with the software parts running on multiprocessor unit and the hardware part running on a FPGA board known as phoenix, prototype hardware for Embedded Network Interconnect Accelerators.

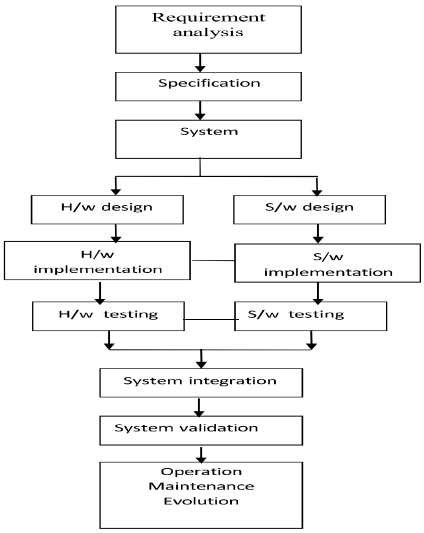
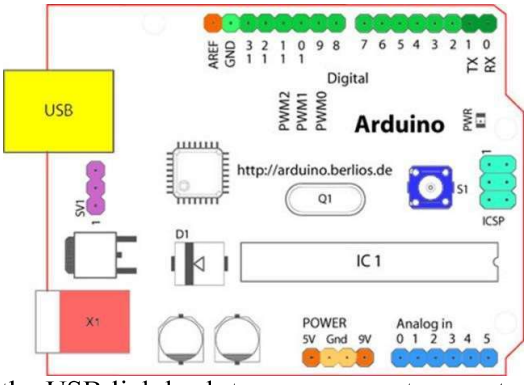


Fig: Embedded Development Life Cycle

3.2 ARUDINO

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an AT mega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output pins, all on a single chip. Unlike, say, a Raspberry Pi, it's designed to attach all kinds of sensors, LEDs, small motors and speakers, servos, etc. directly to these pins, which can read in or output digital or analog voltages between 0 and 5 volts. The Arduino connects to your computer via

USB, where you program it in a simple language (C/C++, similar to Java) from inside the free Arduino IDE by uploading your compiled code to the board. Once programmed, the



Arduino can run with the USB link back to your computer, or stand-alone without it — no keyboard or screen needed, just power.

DIGITAL PINS

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pin Mode (), Digital Read (), and Digital Write () commands. Each pin has an internal pull-up resistor which can be turned on and off using digital Write () (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40mA.

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and Lilypad Arduino,

they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).

- **External Interrupts: 2 and 3**. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt () function for details.

POWER PINS

- **VIN (sometimes labelled "9V"):** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Also note that the Lily Pad has no VIN pin and accepts only a regulated input.
- **5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3 (Diecimila-only):** A 3.3-volt supply generated by the on-board FTDI chip.
- **GND:** Ground pins.

OTHER PINS

- **AREF:** Reference voltage for the analog inputs. Used with analog Reference ().
- **Reset:** (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

VCC: Digital supply voltage.

GND: Ground.

Reset (Reset Input):

A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

XTAL1:

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2:

Output from the inverting Oscillator amplifier.

AVCC:

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

AREF:

AREF is the analog reference pin for the A/D Converter.

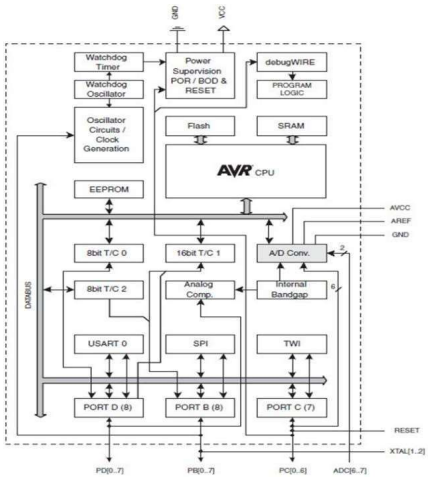
FEATURES

- 1.8-5.5V operating range
- Up to 20MHz
- Part: ATMEGA328P-AU

- 32kB Flash program memory
- 1kB EEPROM
- 2kB Internal SRAM
- 2 8-bit Timer/Counters
- 16-bit Timer/Counter
- RTC with separate oscillator
- 6 PWM Channels
- 8 Channel 10-bit ADC
- Serial USART
- Master/Slave SPI interface
- 2-wire (I2C) interface
- Watchdog timer
- Analog comparator
- 23 IO lines
- Data retention: 20 years at 85C/ 100 years at 25C
- Digital I/O Pins are 14 (out of which 6 provide PWM output)
- Analog Input Pins are 6.
- DC Current per I/O is 40 mA
- DC Current for 3.3V Pin is 5

AVR CPU CORE

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing



two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

OVERVIEW

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code. The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

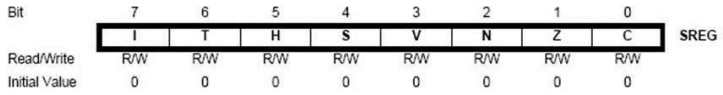


Fig: AVR status register

STACK POINTER

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. Note that the Stack is implemented as growing from higher to lower memory locations. The Stack Pointer Register always points to the top of the Stack. The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. A Stack PUSH command will decrease the Stack Pointer.

Instruction	Stack pointer	Description
PUSH	Decremented by 1	Data is pushed onto the stack
CALL ICALL RCALL	Decremented by 2	Return address is pushed onto the stack with a subroutine call or interrupt
POP	Incremented by 1	Data is popped from the stack
RET RETI	Incremented by 2	Return address is popped from the stack with return from subroutine or return from interrupt

www.vvshare.com

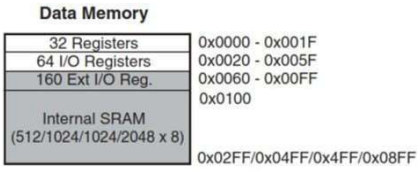
The AVR ATmega128A Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be presents and SPL - Stack Pointer High and Low Register.

INTERRUPT RESPONSE TIME

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed. During this four-clock cycle period, the Program Counter is pushed onto the Stack.

SRAM Data Memory:

ATmega328 is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in the Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.



When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.

Fig: Data Memory Map

Interrupts

This section describes the specifics of the interrupt handling as performed in the

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 0
4	0x0006	PCINTO	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter 2 Overflow

Atmega328. In Atmega328Each Interrupt Vector occupies two instruction words and the Reset Vector is affected by the BOOTRST fuse, and the Interrupt Vector start address is affected by the IVSEL bit in MCUCR.Table 2.2 Reset and Interrupt Vectors in A.

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x000	0x002
1	1	0x000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x002
0	1	Boot Reset Address	Boot Reset Address + 0x0002

Table 2.3 Reset and Interrupt Vectors Placement in ATmega328 and ATmega328P

- **Pin out**: Added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino. Due that operates with 3.3V. The second one is a not connected pin that is reserved for future purposes.

- **Stronger RESET circuit.**
- **At mega 16U2 replace the 8U2.**

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Arduino Characteristics

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery.

CHAPTER 4: ANALYSIS AND DESIGN

4.1 HARDWARE COMPONENTS

4.1.1 LCD (Liquid Cristal Display)

Introduction:

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other. Without the liquid crystals between them, light passing through one would be

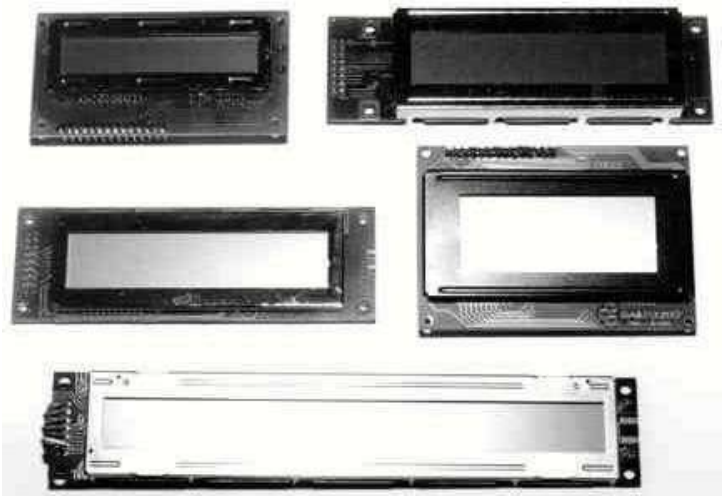
blocked by the other. The liquid crystal twists the polarization of light entering one filter to allow it to pass through the other.

Features:

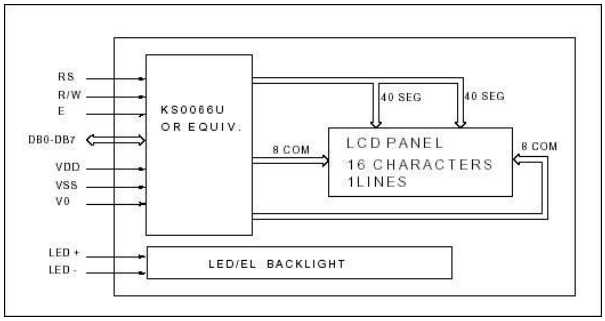
1. Interface with either 4-bit or 8-bit microprocessor.
2. Display data RAM
3. 80x8 bits (80 characters).
4. Character generator ROM.
5. 160 different 5 x7 dot-matrix character patterns.
6. Character generator RAM.
7. 8 different users programmed 5 x7 dot-matrix patterns.
8. Built-in reset circuit is triggered at power ON.
9. Built-in oscillator.

Data can be placed at any location on the LCD. For 16×1 LCD, the address locations are

Shapes and sizes:



Even limited to character-based modules, there is still a wide variety of shapes and sizes available. Line lengths of 8,16,20,24,32 and 40 characters are all standard, in one-, two- and four-line versions.



Electrical block diagram

CONTROL LINES:

EN:

Line is called "Enable." This control line is used to tell the LCD that you are sending it

PIN	SYMBOL	FUNCTION
1	Vss	Power Supply(GND)
2	Vdd	Power Supply(+5V)
3	Vo	Contrast Adjust
4	RS	Instruction/Data Register Select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power Supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

data. To send data to the LCD, your program should make sure this line is low (0) and then

set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

RS:

Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

RW:

Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are writing commands, so RW will almost always be low.

Logic status on control lines:

• E - 0 Access to LCD disabled

- 1 Access to LCD enabled

• R/W - 0 Writing data to LCD

- 1 Reading data from LCD

• RS - 0 Instructions

-1 Character

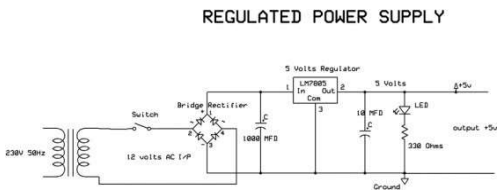
Entering Text:

First, a little tip: it is manually a lot easier to enter characters and commands in hexadecimal rather than binary (although, of course, you will need to translate commands from binary couple of sub-miniature hexadecimal rotary switches is a simple matter, although a little bit into hex so that you know which bits you are setting). Replacing the D.I.L. switch pack with a of re-wiring is necessary.

The switches must be the type where On = 0, so that when they are turned to the zero position, all four outputs are shorted to the common pin, and in position “F”, all four outputs are open circuit.

All the available characters that are built into the module are shown in Table 3. Studying the table, you will see that codes associated with the characters are quoted in binary and hexadecimal, most significant bits (“left-hand” four bits) across the top, and least significant bits (“right-hand” four bits) down the left.

REGULATED POWER SUPPLY:

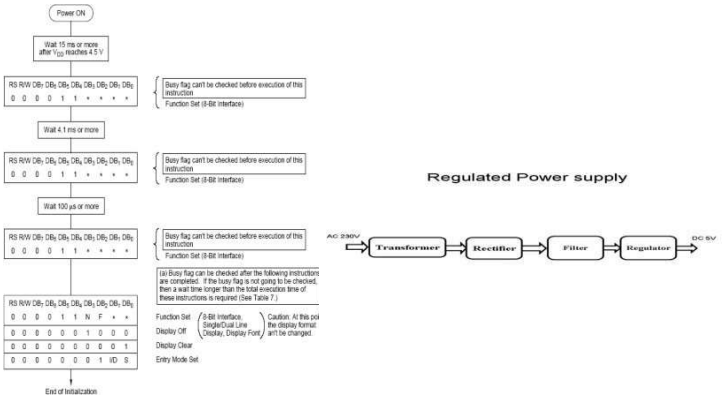


Introduction:

. Power supply is a supply of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others.

A power supply may include a power distribution system as well as primary or secondary sources of energy such as

- Batteries.
- Chemical fuel cells and other forms of energy storage systems.
- Solar power.
- Generators or alternators.



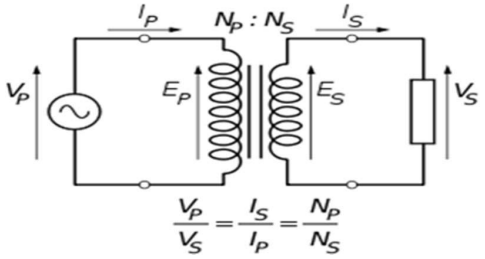
Block Diagram

Fig: Regulated Power Supply

The basic circuit diagram of a regulated power supply (DC O/P) with led connected as load is shown in fig:

Fig: Circuit diagram of Regulated Power Supply with Led connection

The components mainly used in above figure are



- 230V AC MAINS
- TRANSFORMER
- BRIDGE RECTIFIER(DIODES)
- CAPACITOR
- VOLTAGE REGULATOR (IC 7805)
- RESISTOR
- LED (LIGHT EMITTING DIODE)

The detailed explanation of each and every component mentioned above is as follows:

Step 1: Transformation: The process of transforming energy from one device to another is called transformation. For transforming energy, we use transformers.

Transformers:

The input coil is called the PRIMARY WINDING; the output coil is the SECONDARY WINDING. Fig: 3.3.4 shows step-down transformer.

Fig: Step-Down Transformer

The voltage induced in the secondary is determined by the TURNS RATIO.

$$\frac{\text{primary voltage}}{\text{secondary voltage}} = \frac{\text{number of primary turns}}{\text{number of secondary turns}}$$

For example, if the secondary has half the primary turns; the secondary will have half the primary voltage.

Another example is if the primary has 5000 turns and the secondary has 500 turns, then the turn's ratio is 10:1.

If the primary voltage is 240 volts, then the secondary voltage will be x 10 smaller = 24 volts. Assuming a perfect transformer, the power provided by the primary must equal the power taken by a load on the secondary. If a 24-watt lamp is connected across a 24-volt secondary, then the primary must supply 24 watts.

Some transformers have an electrostatic screen between primary and secondary. This is to prevent some types of interference being fed from the equipment down into the mains supply, or in the other direction. Transformers are sometimes used for IMPEDANCE MATCHING.

We can use the transformers as step up or step down.

Step Up transformer:

In case of step-up transformer, primary windings are every less compared to secondary winding.

Because of having more turns secondary winding accepts more energy, and it releases more voltage at the output side.

Step down transformer:

In case of step-down transformer, Primary winding induces more flux than the secondary winding, and secondary winding is having a smaller number of turns because of that it accepts a smaller number of fluxes, and releases less amount of voltage.

Battery power supply: A battery is a type of linear power supply that offers benefits that traditional line-operated power supplies lack: mobility, portability and reliability. A battery consists of multiple electrochemical cells connected to provide the voltage desired.

The lead-acid storage battery may be used. This battery is rechargeable; it consists of lead and lead/dioxide electrodes which are immersed in sulfuric acid. When fully charged, this type of battery has a 2.06-2.14 V potential (A 12-volt car battery uses 6 cells in series). During discharge, the lead is converted to lead sulphate and the sulfuric acid is converted to water

Step 2: Rectification

The process of converting an alternating current to a pulsating direct current is called as rectification. For rectification purpose we use rectifiers.

Rectifiers:

A rectifier is an electrical device that converts alternating current (AC) to direct current (DC), a process known as rectification. Rectifiers have many uses including as components

of power supplies and as detectors of radio signals. Rectifiers may be made of solid-state diodes, vacuum tube diodes, mercury arc valves, and other components.

A device that it can perform the opposite function (converting DC to AC) is known as an inverter.

Bridge full wave rectifier:

The Bridge rectifier circuit is shown in figure, which converts an ac voltage to dc voltage using both half cycles of the input ac voltage. The Bridge rectifier circuit is shown in the figure. The circuit has four diodes connected to form a bridge. The ac input voltage is applied to the diagonally opposite ends of the bridge. The load resistance is connected between the other two ends of the bridge.

For the negative half cycle of the input ac voltage, diodes D2 and D4 conduct whereas, D1 and D3 remain OFF. The conducting diodes D2 and D4 will be in series with the load resistance RL and hence the current flows through RL in the same direction as in the previous half cycle. Thus, a bi-directional wave is converted into a unidirectional wave.

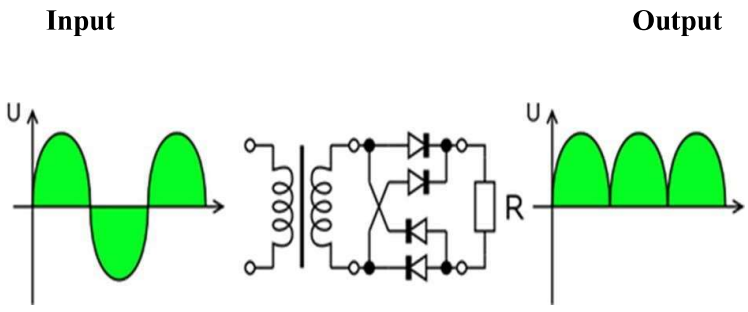
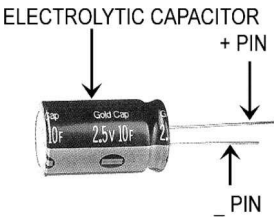


Fig: Bridge rectifier: a full-wave rectifier using 4 diodes

Now -a -days Bridge rectifier is available in IC with a number of DB107. In our project we are using an IC in place of bridge rectifier.



Features:

- Good for automation insertion
- Surge overload rating - 30 amperes peak
- Ideal for printed circuit board
- Reliable low-cost construction utilizing moulded

- Glass passivated device
- Polarity symbols moulded on body
- Mounting position: Any
- Weight: 1.0 gram

Step 3: Filtration:

The process of converting a pulsating direct current to a pure direct current using filters is called as filtration.

Filters:

Electronic filters are electronic circuits, which perform signal-processing functions, specifically to remove unwanted frequency components from the signal, to enhance wanted ones.

Introduction to Capacitors:

The Capacitor or sometimes referred to as a Condenser is a passive device, and one which stores energy in the form of an electrostatic field which produces a potential (static voltage) across its plates. In its basic form a capacitor consists of two parallel conductive plates that are not connected but are electrically separated either by air or by an insulating material called the Dielectric.

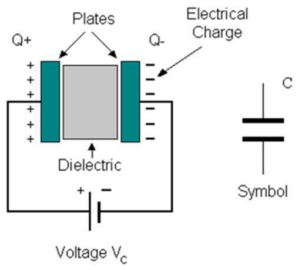


Fig: Construction Of a Capacitor

Units of Capacitance:

Microfarad (μF) 1μF = 1/1,000,000 = 0.000001 = 10-6 F

Nano-farad (nF) 1nF = 1/1,000,000,000 = 0.000000001 = 10-9 F

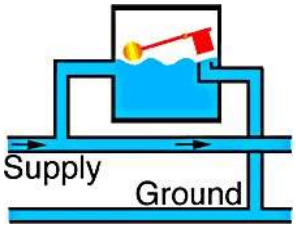
Pico farad (pF) 1pF = 1/1,000,000,000,000 = 0.000000000001 = 10-12 F

Operation of Capacitor:

Think of water flowing through a pipe. If we imagine a capacitor as being a storage tank with an inlet and an outlet pipe, it is possible to show approximately how an electronic capacitor works.

First, let's consider the case of a "coupling capacitor" where the capacitor is used to connect a signal from one part of a circuit to another but without allowing any direct current to flow.

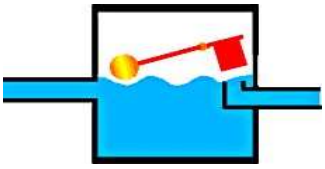
If the current flow is alternating between zero and a maximum, our "storage tank" capacitor will allow the current waves to pass through.



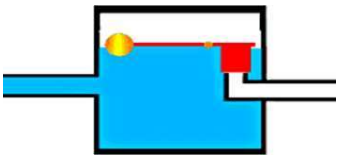
However, if there is a steady current, only the initial short burst will flow until the "floating ball valve" closes and stops further flow.

So, a coupling capacitor allows "alternating current" to pass through because the ball valve doesn't get a chance to close as the waves go up and down. However, a steady current quickly fills the tank so that all flow stops.

A capacitor will pass alternating current but (apart from an initial surge) it will not pass D.C.



Where a capacitor is used to decouple a circuit, the effect is to "smooth out ripples". Any



ripples, waves or pulses of current are passed to ground while D.C. Flows smoothly.

Step 4: Regulation

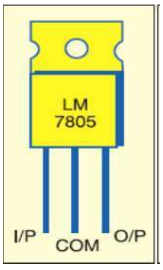
The process of converting a varying voltage to a constant regulated voltage is called as regulation. For the process of regulation, we use voltage regulators.

Voltage Regulator:

A voltage regulator (also called a 'regulator') with only three terminals appears to be a simple device, but it is in fact a very complex integrated circuit. It converts a varying input voltage into a constant 'regulated' output voltage.

It is not possible to obtain a voltage lower than the stated rating. You cannot use a 12V regulator to make a 5V power supply. Voltage regulators are very robust. These can withstand over-current draw due to short circuits and also over-heating. In both cases, the regulator will cut off before any damage occurs. The only way to destroy a regulator is to apply reverse voltage to its input. Reverse polarity destroys the regulator almost instantly.

Resistors:



A resistor is a two-terminal electronic component that produces a voltage across its terminals that is proportional to the electric current passing through it in accordance with Ohm's law:

V = IR

Resistors can be made to control the flow of current, to work as Voltage dividers, to dissipate power and it can shape electrical waves when used in combination of other components. Basic unit is ohms.

Theory of operation:

Ohm's law:

The behaviour of an ideal resistor is dictated by the relationship specified in Ohm's law:

V = IR

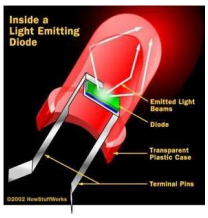
Ohm's law states that the voltage (V) across a resistor is proportional to the current (I) through it where the constant of proportionality is the resistance (R).

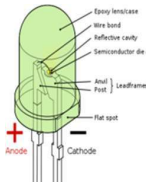
Power dissipation:

The power dissipated by a resistor (or the equivalent resistance of a resistor network) is calculated using the following:

Fig: Resistor

Fig: Colour Bands in Resistor

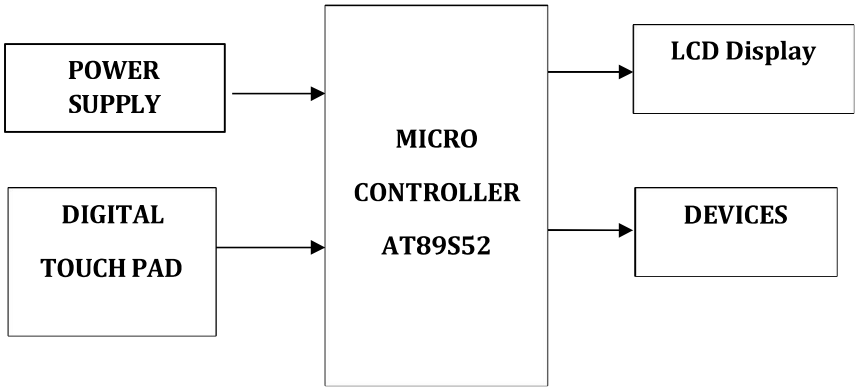




LED:

A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices, and are increasingly used for lighting. Introduced as a practical electronic component in 1962, early LEDs emitted low-intensity red light, but modern versions are available across the visible, ultraviolet and infrared wavelengths, with very high brightness. The internal structure and parts of a led are shown below.

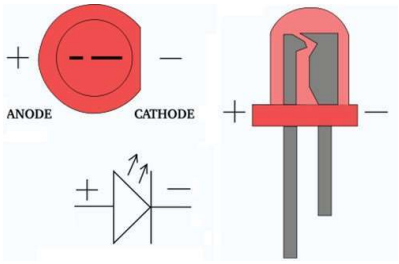
BLOCK DIAGRAM



Working:

The structure of the LED light is completely different than that of the light bulb. Amazingly, the LED has a simple and strong structure. The light-emitting semiconductor material is what determines the LED's colour. The LED is based on the semiconductor diode.

When a diode is forward biased (switched on), electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence and the colour of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor. An LED is usually small in area



(less than 1 mm2), and integrated optical components are used to shape its radiation pattern and assist in reflection. LEDs present many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved robustness, smaller size, faster switching, and greater durability and reliability. The electrical symbol and polarities of led are shown in fig:

Fig: Electrical Symbol & Polarities of LED

LED lights have a variety of advantages over other light sources:

- High-levels of brightness and intensity
- High-efficiency

- Low-voltage and current requirements
- Low radiated heat
- Can be easily controlled and programmed

Applications of LED fall into three major categories:

- Visual signal application where the light goes more or less directly from the LED to the human eye, to convey a message or meaning.
- Illumination where LED light is reflected from object to give visual response of these objects.
- Generate light for measuring and interacting with processes that do not involve the human visual system.

SOFTWARE DESCRIPTION

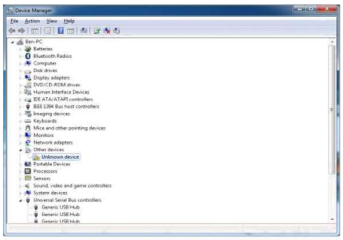
ARDUINO SOFTWARE:

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals.

What you will need:

- A computer (Windows, Mac, or Linux)
- An Arduino-compatible microcontroller (anything from this guide should work).

- A USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer (check out this USB buying guide if you're not sure which cable to get).
- An Arduino Uno
- Windows 7, Vista, and XP



- Installing the Drivers for the Arduino Uno (from Arduino.cc)
- Plug in your board and wait for Windows to begin its driver installation process After a few moments, the process will fail, despite its best efforts.
- Finally, navigate to and select the Uno's driver file, named "ArduinoUNO.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you cannot see the .inf file, it is probably just hidden. You can select the 'drivers' folder with the 'search sub-folders' option selected instead. Windows will finish up the driver installation

LAUNCH AND BLINK!

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

- Launch the Arduino application.
- If you disconnected your board, plug it back in.
- Open the Blink example sketch by going to: File > Examples > 1. Basics > Blink.
- Select the type of Arduino board you're using: Tools > Board > your board type.

- Select the serial/COM port that your Arduino is attached to: Tools > Port > COMxx

If you're not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

With your Arduino board connected, and the Blink sketch open, press the 'Upload' button

CHAPTER 5: IMPLEMENTATION

```
#include <LiquidCrystal.h>

#include <stdio.h>

LiquidCrystal lcd(6, 7, 5, 4, 3, 2); int

button1  = 10;

int button2  = 11;

int light    = 9;

int fan      =    8;

int buzzer   = 13;

int tempc=0;

char pastnumber[11];

unsigned char rcv,count,gchr='x',gchr1='x',robos='s';
```

```
char gpsval[50];

// char dataread[100] = "";

// char lt[15],ln[15];

int    i=0,k=0,lop=0;

int      gps_status=0;

float       latitude=0;

float      logitude=0;

String      Speed="";

String gpsString="";

char *test="$GPRMC";

//int hbtc=0,hbtc1=0,rtrl=0;

unsigned  char  gv=0,msg1[10],msg2[11];

float lati=0,longi=0;

unsigned       int       lati1=0,longi1=0;

unsigned       char       flat[5],flong[5];

unsigned  char  finallat[8],finallong[9];

int ii=0,rchkr=0;

String inputString = "";         // a string to hold incoming data

boolean stringComplete = false;  // whether the string is complete
```

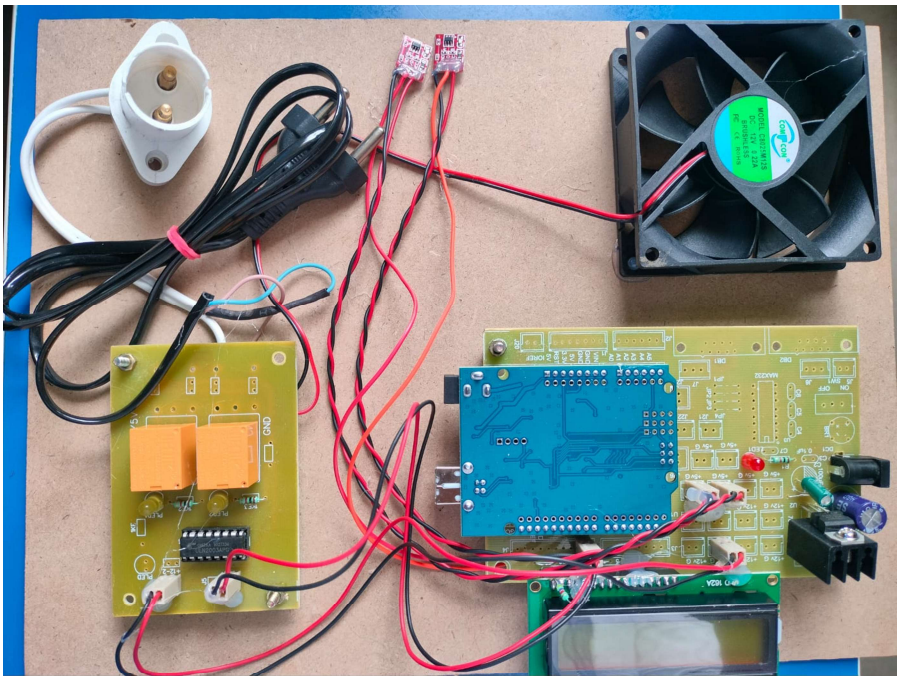
```
void okcheck()
{
 unsigned  char  rcr;
 do{
    rcr = Serial.read();
  }while(rcr == 'K');
}
void sound()
{
 digitalWrite(buzzer,LOW);delay(1500);digitalWrite(buzzer,HIGH);
}
void setup()
{
 Serial.begin(9600);

 lcd.begin(16,   2);lcd.cursor();
 lcd.print("Designing Haptic");
 lcd.setCursor(0,1);
 lcd.print("Interface System");
```

```
    delay(1500);


    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Light:");     //6,0
    lcd.setCursor(0,1);
    lcd.print("Fan:");       //6,1
}


int sts1=0,sts2=0;
void loop()
{
  if(digitalRead(button1) == HIGH)
    {delay(700);
     while(digitalRead(button1) == HIGH);
     sts1++;
     if(sts1 == 1)
       {
        lcd.setCursor(6,0);lcd.print("ON  ");
```

```
    digitalWrite(light, HIGH);
   }
  if(sts1 == 2)
   {                                    sts1=0;
    lcd.setCursor(6,0);lcd.print("OFF ");
    digitalWrite(light, LOW);
   }
 }

if(digitalRead(button2) == HIGH)
 {delay(700);
  while(digitalRead(button2) == HIGH);
  sts2++;
  if(sts2 == 1)
   {
    lcd.setCursor(6,1);lcd.print("ON   ");
    digitalWrite(fan, HIGH);
   }
  if(sts2 == 2)
```

```
{

Sts2=0;

Lcd.cursor(6,1); lcd.print("OFF   ");

}}

Delay(500);

}
```

CHAPTER 6 : TESTING/DEBUGGING RESULTS




LED1= ON
LED2= ON


TOUCH

CHAPTER 7 : CONCLUSION

Designing a haptic interface for disabled individuals within an IoT project framework presents a significant advancement in assistive technology, fostering inclusivity and enhancing the quality of life. The integration of haptic feedback mechanisms allows users to interact with their environment through tactile sensations, providing a sense of touch that can compensate for visual or auditory impairments. This innovation leverages IoT's connectivity, enabling real-time communication between devices and users, which is particularly beneficial for those with mobility or sensory challenges.The project's success hinges on meticulous attention to user needs, ensuring the interface is intuitive and accessible.

CHAPTER 8 : REFERENCES

Alghamdi, A., & Moore, P. (2019). A Review of Haptic Feedback Systems and Their Applications. Journal of Computer and System Sciences, 85(2), 263-275. doi:10.1016/j.jcss.2018.09.005.

Bach-y-Rita, P., & Kercel, S. W. (2003). Sensory Substitution and the Human–Machine Interface. Trends in Cognitive Sciences, 7(12), 541-546. doi:10.1016/j.tics.2003.10.013.

Feenstra, P., Fiedler, S., & Bos, J. E. (2011). Haptic Feedback Improves Task Performance in a Virtual Reality Environment. Journal of Ergonomics, 54(10), 867-880. doi:10.1080/00140139.2011.605991.

CHAPTER 9 : APPENDICES

Appendix A: User Research and Surveys

1. **Survey Instrument**:
    - Questions used to gather information about user needs, preferences, and experiences with current assistive technologies.
2. **Demographics**:
    - Summary of participants' demographic information, including age, gender, type of disability, and technology proficiency.
3. **Findings**:
    - Key insights and patterns identified from survey responses.
    - User feedback on desired features and improvements for the haptic interface.

Appendix B: Technical Specifications

1. **Hardware Components**:
    - List and specifications of the hardware used in the haptic interface, including sensors, actuators, and connectivity modules.
2. **Software Architecture**:
    - Detailed description of the software architecture, including data flow diagrams, communication protocols, and integration with IoT platforms.