VIRTUAL KEYBOARD

A PROJECT REPORT

Submitted by

ABHISHEK. K

ARJHUN. M

ARULLANANDH. S

VASIGARAN. S

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2018

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“VIRTUAL KEYBOARD”** is the bonafide work of **ABHISHEK.K (715514106002), ARJHUN.M (715514106007), ARULLANANDH.S (715514106008) and VASIGARAN.S (715514106057)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. E. Malar M.E, Ph.D,	Dr. G. Santhanamari M.E, Ph.D,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Assistant Professor
ECE Department	ECE Department
PSG Institute of Technology	PSG Institute of Technology
and Applied Research	and Applied Research
Coimbatore-641 062.	Coimbatore-641 062.

Submitted for Anna University Viva-Voce examination held on 17-04-2018

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We would like to thank the **Management** of PSG Institute of Technology and Applied Research for providing us with excellent facilities for the completion of this project.

We are grateful to **Dr. P. V. Mohanram**, our beloved Principal for his support and providing all the facilities in our institution.

We would like to thank **Dr. G. Chandramohan**, our beloved vice principal for his encouragement and support.

We are thankful to **Dr. E. Malar,** Professor and Head, Department of Electronics and Communication Engineering, for her continued encouragement for the successful completion of this project.

It is our pleasure to thank our Project Guide **Dr. G. Santhanamari,** Assistant Professor, Department of Electronics and Communication Engineering, for her valuable guidance and helping in all possible ways to complete the project.

We also would like to thank all the faculty members and staffs of Electronics and Communication Engineering Department for their kind co-operation and encouragement during the course of this work.

ABSTRACT

The keyboard acts as the key input device of computer and other smart electronic devices. The next generation of keyboards is virtual keyboards which uses virtual keys and has the same functions as a mechanical keyboard. The aim of this project is to design and implement an effective virtual keyboard. The virtual keyboard requires air typing for entering the input rather than the typical mechanical key presses. The text entering module is an empty frame with an array of lasers aligned exactly opposite to Light Dependent Resistors in the form of a matrix. The key press is recognized by the interruption of laser light falling on the Light Dependent Resistors. This virtual keyboard uses NI myRIO as a processor and is programmed in LabVIEW. NI myRIO will identify the character that is pressed and send the data to the console on the lab view for display.

Keywords: Lasers, Light Dependent Resistors, NI myRIO, Virtual Keyboard.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

SYMBOL	EXPANSION
A	Ampere
MHz	Mega Hertz
V	Volt
kV	Kilo Volt
mm	Millimeter
cm	Centimeter
Ω	Ohm
db	Decibel
W	Watt
◦C	Degree Celsius

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
LASER	Light Amplification by Stimulated Emission of Radiation
myRIO	my Reconfigurable Input Output
VI	Virtual Instrument
PLA	Poly Lactic Acid
LDR	Light dependent resistor
3-D	Three-Dimension

CHAPTER 1
INTRODUCTION

Technology is scaling at a very faster rate. From time to time, Human-Computer interaction is improved for the purpose of simplifying human lives. Hand held devices have become a part of our day to day life. Keyboard remains a major kind of input device for computer and other smart electronic devices for human-computer interactions. As the key input device of computer and other handheld devices, keyboard plays an irreplaceable role in many areas for data input so that we can look at the output screen to interpret the output data. However, traditional keyboards are too bulky and are inconvenient to carry. In addition to that the traditional keyboards offer very little scope in terms of enhancement. The size of laptops and desktops are becoming smaller in this age of miniaturization. Also, the traditional keyboards act as a hindrance to further miniaturization. Moreover, the traditional ones have a major drawback in the form of wear and tear, when used for a very long time.

Hence instead of using conventional keyboards, the demand for an intuitive, immersive and cost-efficient interaction device has arisen. The Virtual Keyboard is a perfect solution. Virtual Keyboards may be used in some cases to reduce the risk of keystroke logging. For example, Westpac's online banking service uses a Virtual Keyboard for the password entry, as does Treasury Direct. It is more difficult for malware to monitor the data entered via the Virtual Keyboard, than it is to monitor real keystrokes. Thus, it is difficult to spy on the data entered by users. In this project, a Virtual Keyboard implementation of that kind is presented. Virtual Keyboard could become an important entity of augmented reality in the future.

The approach towards the development of the Virtual Keyboard, abandons the concept of the physical keyboard in its traditional way. On the other hand, in pursuit of a new conceptual keyboard, the full functionalities of its counterparts should be preserved and processing capabilities should not be compromised for the sake of compactness.

In this project work, the Virtual Keyboard system combines Light Amplification by Stimulated Emission of Radiation (LASER), Light Dependent Resistor (LDR) with my Reconfigurable Input Output (NI myRIO) processor. Different from brain-computer interface, special gloves and other methods, visual-based Virtual Keyboards do not require redundant equipment to wear, and keep small in size, thus it has broader future.

The proposed Virtual Keyboard is a LASER keyboard and it has no physical buttons. Placing the finger in the paths of LASER light implies the pressing of keys. This device can be programmed to display letters of any language, by mapping those characters to the key presses. Thus, the Virtual Keyboard acts a multiple language insertion device. It includes numbers, letters, modifying keys and special characters. The virtual text entry system is actually air typing to engage effortlessly during typing.

CHAPTER 2
LITERATURE REVIEW

Various new techniques have been adopted on the design schemes of Virtual Keyboard such as computer interaction based on LASER and image processing by Xiaolin Su et al (2015), gesture recognition by Mantyjarvi. J et al (2002) and brain-computer interface by Hari Singh Dhillon et al (2009).

Xiaolin Su et al (2015) developed a Virtual Keyboard, where the keystroke can be detected accurately by image processing, including morphology principle and ellipse fitting is discussed. The Keyboard pattern is projected on the horizontal plane by the projector, forming Virtual Keyboard layout that allows users to "press key". In normal indoor lighting conditions, the projection technology can clearly project full-size keyboard pattern on any plane, allowing users to operate and type as easy as a traditional keyboard.

Mantyjarvi. J et al (2002) proposed the realization of Virtual Keyboard utilizing IR proximity sensing system and keystroke recognition with k-NN and MLP neural networks is demonstrated. Unfortunately, in the experiments of Mantyjarvi. J et al (2002), the finger had to be held perpendicular to the surface of the keyboard. This is obviously an unwanted feature.

An arbitrary quadrangle-shaped panel e.g., an ordinary piece of paper and a tip pointer e.g., fingertip as an intuitive, wireless and mobile input device are employed by Zhang. Z et al (2001). The tracking of panel determines the projective mapping between the panel at the current position and the display, which in turn maps the tip position to the corresponding position on the display. By detecting the clicking and dragging actions, the system can fulfill many tasks such as controlling a remote large display and simulating a physical keyboard.

Scherer. R et al (2004) developed an asynchronously controlled three-class brain-computer interface-based spelling device Virtual Keyboard, operated by spontaneous electroencephalogram and modulated by motor imagery. At the same time, multiple classes and asynchronous control can limit the usability of the system. Users do require more training and the cognitive load is higher.

The system designed by Du. H et al (2005), consists of a pattern projector and a true-3-D Three-Dimensional range camera for detecting the typing events. It exploits depth information acquired with the 3-D range camera and detects the hand region using a pre-computed reference frame. The fingertips are found by analyzing the hands' contour and fitting the depth curve with different feature models. It is accurate and robust. However, it requires a 3-D optical imaging system.

Huan Du et al (2008) developed a Multi-Level Feature Matching method for 3-D hand posture reconstruction of a Virtual Keyboard system. The human hand is modeled with a mixture of different levels of detail, from skeletal to polygonal surface representation. Different types of features are extracted and paired with the corresponding model. The matching is performed in a bottom-up order by SCG optimization with respect to the state vector of motion parameters. The low level of matching provides initial guess to the high level of matching, refining the precise position of the hand hierarchically.

Hubert Cecotti et al (2016) proposed a new portable and noninvasive eye-trackers allow the creation of robust Virtual Keyboards that aim to improve the life of disabled people who are unable to communicate. The Virtual Keyboard is based on a menu selection with eight main commands that allow us to spell 30 different characters and correct errors with a delete button.

The systems developed by Katz I et al (2007) and Echtler. F et al (2008) provide special features such as hand gestures and multi–touch but Katz I et al (2007) requires multiple cameras.

The touch detection system created by Adajania. Y et al (2010) is designed to detect a touch by comparing the ratio of black pixels to the number of white ones. It is understood that a ratio is acquired by searching small regions around the fingertips and comparing the number of white pixels to black ones, where black pixels represent the shadow. If the ratio of white to black pixels exceeds a certain threshold, a touch has occurred. However, these methods are sensitive to the direction of lighting, where in many cases only a thin portion of the shadow is captured by the camera. Therefore, when the finger seems close to touching the surface, it is still far away from it, since the pixel difference is small.

In the systems proposed by Kenkichi Yamamoto et al (2006) and Hirobe et al (2009), high speed camera is used, and a special in-air movement should be made. Babic. R. V et al (2002) employs the principle of particular correspondence between sensor state and symbol, where each independently movable finger joint activates its own sensor, with a substrate in the form of glove to associate all these numerous sensors. In the system proposed by Caslav Livada et al (2017), digital image processing methods are used, such as binarization, morphological operation and filtering. Real time image acquisition is performed by a mid-range quality web camera in order to identify the method of simultaneous fingertip and keyboard character recognition in order to produce a fully functional keyboard for user input.

Ming-Wei Chang et al (2014) proposed the design, implementation and evaluation of a text input system for head mounted devices called air typing, which requires only a standard camera and is shown to be comparable in effectiveness to single-hand text input on tablet computers in a lab setting. Air typing features a novel two-level virtual keyword layout, which substantially improves the typing speed by cutting down unnecessary hand movements during typing and greatly simplifies the associated image processing task by doing away with fine-grained matching between fingertips and keys.

CHAPTER 3
METHODOLOGY

The functioning of Virtual Keyboard combines the photoresistance of LDR and the intensity of laser Light. The functioning of this proposed Virtual Keyboard is discussed in this chapter.

3.1 STIMULATED EMISSION OF LASER

The REES52 Red LASER diodes, with a 650nm red wavelength are used. They can be driven from 5V. The spectrum can be seen as shown in the Figure 3.1.

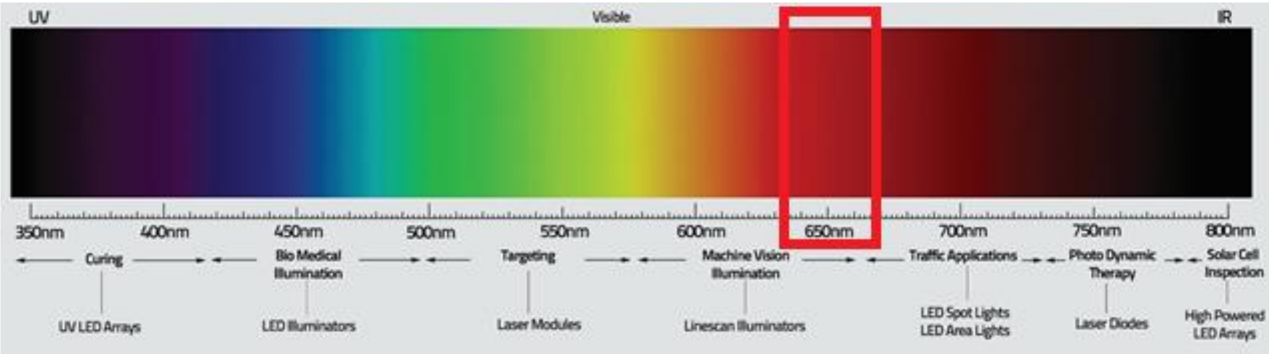

Figure 3.1 Electromagnetic Spectrum

A LASER is a device that emits light through a process of optical amplification based on the stimulated emission of electromagnetic radiation. A LASER consists of a gain medium, a mechanism to energize it to provide optical feedback. The gain medium is a material with properties that allow it to amplify light by way of stimulated emission. Light of a specific wavelength that passes through the gain medium is amplified i.e. increases in power. For the gain medium to amplify light, it needs to be supplied with energy in a process called pumping. The energy is typically supplied as an electric current or as light at a different wavelength. Pump light may be provided by a flash lamp or by another LASER.

The most common type of LASER uses feedback from an optical cavity, which is a pair of mirrors on either end of the gain medium. Light bounces back and forth between the mirrors, passing through the gain medium and being amplified each time. Typically, one of the two mirrors, the output coupler, is partially transparent and some of the light escapes through this mirror. Depending on the design of the cavity i.e., whether the mirrors are flat or curved, the light coming out of the LASER may spread out or form a narrow beam. In analogy to electronic oscillators, this device is sometimes called a LASER oscillator.

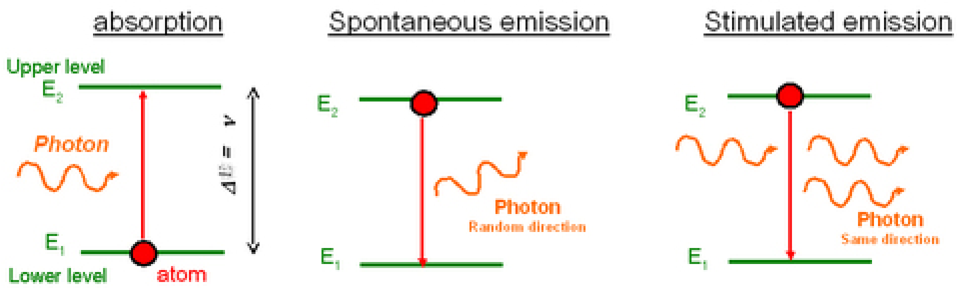

Figure 3.2 Spontaneous and Stimulated Emission

When an electron absorbs energy either from photons or phonons, it receives that incident quantum of energy. But transitions are only allowed in between discrete energy levels. This leads to emission lines and absorption lines. This can be seen in Figure 3.2.

When an electron is excited from a lower to a higher energy level, it will not stay that way forever. An electron in an excited state may decay to a lower energy state which is not occupied, according to a particular time constant characterizing that transition. When such an electron decays without any external influence emitting a photon, it is called as spontaneous emission. The phase associated with the photon that is emitted is random. A material with many atoms in such an excited state may thus result in radiation which is very spectrally limited, but the individual photons would have no common phase relationship and would emanate in random directions. This is the mechanism of fluorescence and thermal emission.

An external electromagnetic field at a frequency associated with a transition can affect the quantum mechanical state of the atom. As the electron in the atom makes a transition between two stationary states, it enters a transition state which does have a dipole field, and which acts like a small electric dipole, and this dipole oscillates at a characteristic frequency. In response to the external electric field at this frequency, the probability of the atom entering this transition state is greatly increased. Thus, the rate of transitions between two stationary states is enhanced beyond that due to spontaneous emission. Such a transition to the higher state is called absorption, and it destroys an incident photon. A transition from the higher to a lower energy state, however, produces an additional photon. The characteristics of the LASER used can be seen in Table 3.1.

TABLE 3.1 CHARACTERISTICS OF LASER

S.No	Parameter	Value
1	Wavelength of light	650 nm
2	Optical power output	5 Mw
3	Operating voltage	5 V
4	Operating temperature	-10 °C to 40 °C
5	Operating current	~30 mA
6	Spot	10 mm to 15 mm at 15 m
7	Life span	>1000 hours

3.2 LASER MATRIX

LASER diodes are placed in a matrix fashion and the light from the LASER's are made to intersect at their respective rows and columns. The end point of these LASER will be the LDR's. So, when the light falls on the LDR the value obtained across the LDR will be higher than predetermined threshold 3 V, hence the value is treated as digital high and when the LASER is blocked by any object, then the value at the LDR will be less than 3 volts, so the value is treated as digital low. When the human hands are intercepted between any intersections of the LASER matrix, a respective key will be pressed and, in this way, multiple keys can be mapped to the multiple intersections of the LASER matrix.

Since LASER have very high frequency around 4 x 10^{14} Hz, it can be said that there will only be minute interference or no interference at all. So, it is assured that the components will work at all times without any faults.

3.3 SYSTEM ARCHITECTURE

Virtual Keyboard system mainly consists of two modules: an input module with LASER and LDR and a processing module containing NI myRIO processor. Fig 3.3 shows Virtual Keyboard system architecture. Asynchronously the data from the LDR gets updated in myRIO. When a key is pressed inside the body of the keyboard, the values is sent to myRIO will validate and find the position pressed and send the character mapped to that particular location. These values are mapped according to the standard “qwerty” keyboard. The keyboard can also be changed according to the user specification.

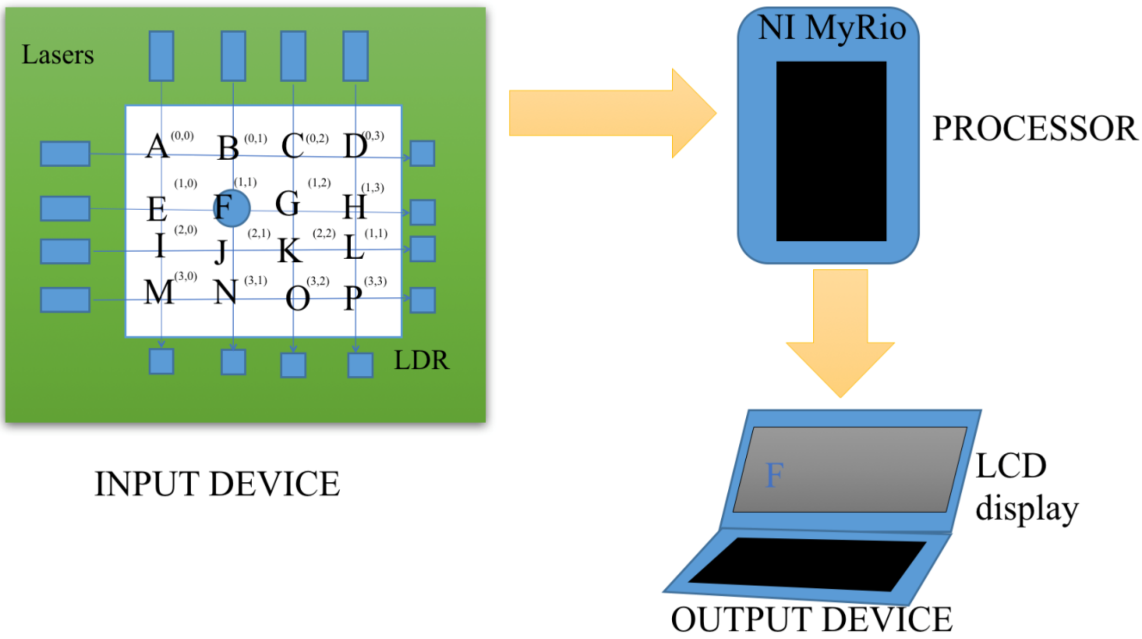

Figure 3.3 System Architecture

3.4 INPUT MODULE

The core components used in this input module includes LASER diodes and LDR's. The keyboard module was built by 3-D printing. 3-D printing refers to processes in which material is joined or solidified under computer control to create a 3-D object, with material being added. 3-D printing is used in both rapid prototyping and additive manufacturing. Objects can be of almost any shape or geometry and typically are produced using digital model data from a 3-D model or another electronic data source such as an Additive Manufacturing File. Stereo lithography is one of the most common file types that is used for 3-D printing. Unlike material removed from a stock in the conventional machining process, 3-D printing or Additive Manufacturing builds a 3-D object from SOLIDWORKS model or Additive Manufacturing File, usually by successively adding material layer by layer. Before printing a 3-D model from a Stereo lithography file, it must first be examined for errors. Most Computer Aided Design applications produce errors in output Stereo lithography files of the following types:

1. Holes
2. Faces normals
3. Self-intersections
4. Noise shells
5. Manifold errors

The necessity for 3-D printing arises for the sake of better accuracy and finishing. Moreover, 3-D printing model replicas don't show discrepancies in the form of dimensions. The modeling is done with the help of cube pro software. The modeling material is red Poly Lactic Acid (PLA), organic thermoplastics. PLA will give the product better structural integrity and will be more suited to mechanical use given the material can better withstand the elements. It also offers a shinier and smoother appearance.

The model is cuboid in shape with one of the faces left open and the rest are covered. There are 14 tiny slots on each of the two longer faces of the cuboid. There are 5 similar slots on the each of the smaller faces of the cuboid. The slots at the faces of the modeled material are to aid the insertion of LASER's and LDR's. The positions of the LASER and its opposite LDR slots are precisely aligned in a straight line.

A photoresistor is a light-controlled variable resistor. The resistance of a photoresistor decreases with increasing incident light intensity; in other words, it exhibits photoconductivity. A photoresistor can be applied in light-sensitive detector circuits, and light-activated and dark-activated switching circuits. A photoresistor is made of a high resistance semiconductor. In the dark, a photoresistor can have a resistance as high as several MΩ, while in the light, a photoresistor can have a resistance as low as a few hundred ohms. If incident light on a photoresistor exceeds a certain frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electrons conduct electricity, thereby lowering resistance. The resistance range and sensitivity of a photoresistor can substantially differ among dissimilar devices. Moreover, unique photoresistors may react substantially differently to photons within certain wavelength bands.

The fourteen holes on one of the longer faces of the 3-D printed material are inserted with LASER diodes. The LASER Diode is shown in Figure 3.4. Similarly, the five holes on one of the smaller faces of the 3-D material are inserted with LASER diodes. The holes on the faces parallel to the faces with LASER diodes are inserted with LDR's. The LDR is shown in Figure 3.5. The LASER diodes are given a voltage of 5 V from any external supply or more efficiently from one of output ports of the NI myRIO processor.

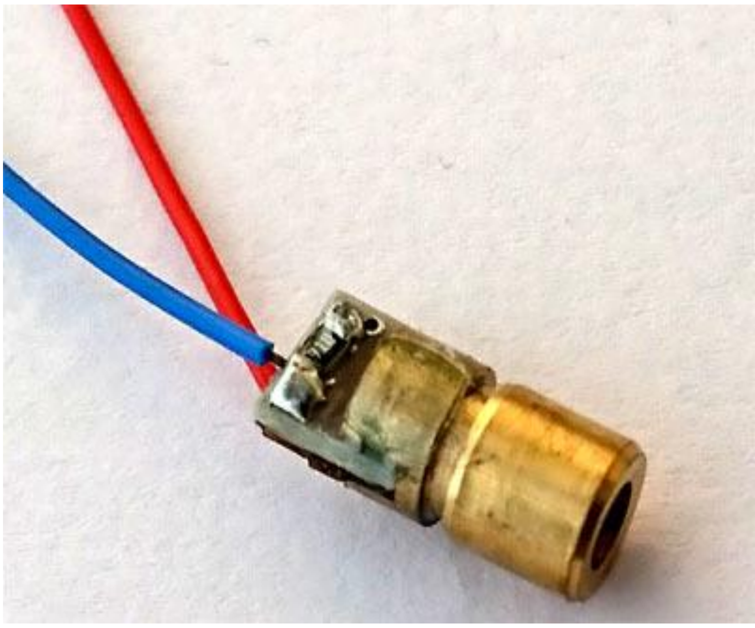

Figure 3.4 LASER Diode

Figure 3.5 LDR

3.5 3-D MODEL IN SOLIDWORKS

The design for the 3-D model is done in SOLIDWORKS. It uses 3-D printing for realization of the model. The design for the 3-D model is pictorially represented in this section.

3.5.1 Isometric View

The isometric view of Virtual Keyboard is shown in the Figure 3.6.

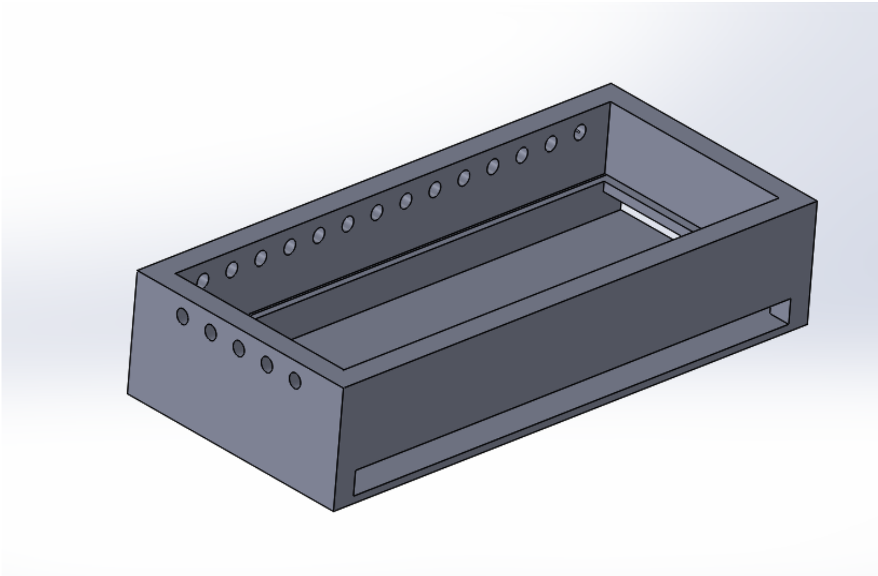

Figure 3.6 Isometric View of Virtual Keyboard

3.5.2 Top View

The top view of Virtual Keyboard is shown in the Figure 3.7.

Figure 3.7 Top View of Virtual Keyboard

3.5.3 Front View

The front view of Virtual Keyboard is shown in the Figure 3.8.

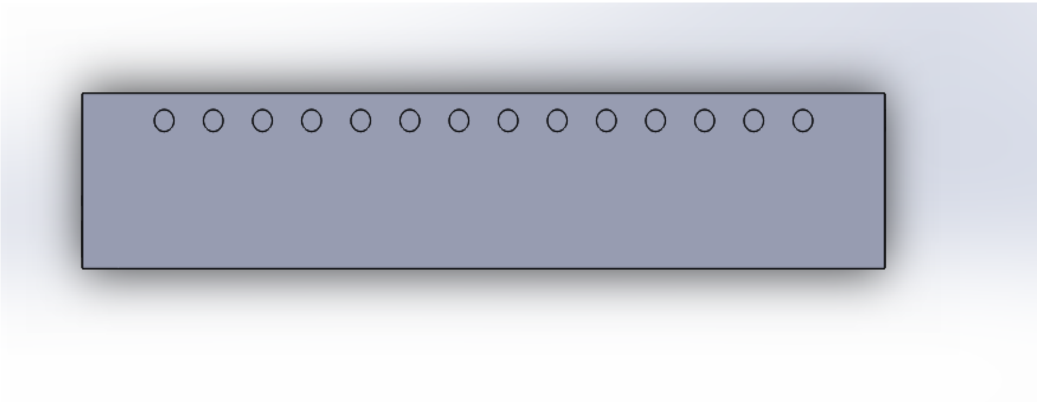

Figure 3.8 Front View of Virtual Keyboard

The dimensions of the material created in SOLIDWORKS is mentioned in the Table 3.2.

TABLE 3.2 DIMENSIONS OF THE PRINTED MATERIAL

S. No	Parameters	Values
1	Length	245 mm
2	Breadth	110 mm
3	Height	48 mm
4	Color	Red
5	Material	PLA
6	Dimension of the hole	6 mm

3.6 PROCESSING MODULE

The processor used is NI myRIO. The National Instruments MyRIO-1900 is a portable reconfigurable I/O device used to design control robotics and mechatronics system. The hardware is based on Xilinx Zynq-7010 with a dual-core ARM Cortex-A9 processor and an FPGA with 28,000 programmable logic cells, and features 10 analog inputs, 6 analog outputs, audio I/O channels, and up to 40 lines of digital input/output (DIO).

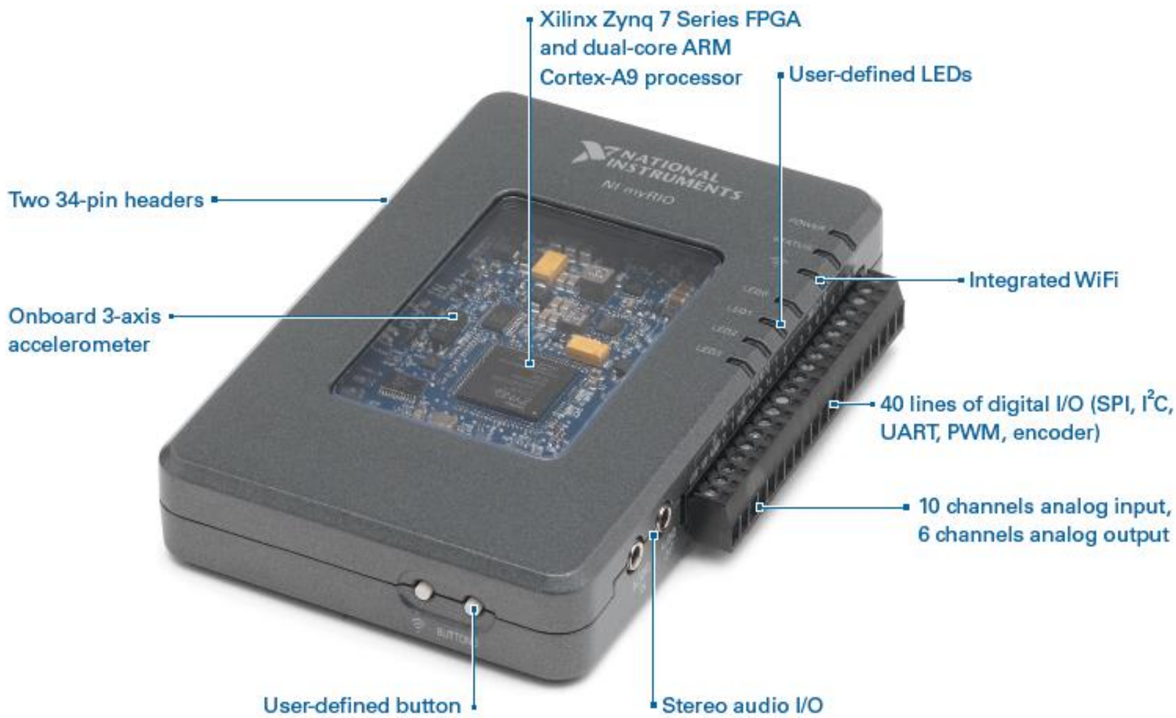

Figure 3.9 NI MyRIO-1900

The NI myRIO-1900 provides analog input, analog output, digital input and output, audio, and power output in a compact embedded device. The NI myRIO is shown in Figure 3.9. The NI myRIO-1900 connects to a host computer over USB and wireless 802.11 b, g, n. NI myRIO is based on the same LabVIEW RIO architecture as NI's industrially used NI CompactRIO and NI Single-BoardRIO products.

These products combine a processor, FPGA, and I/O, and are fully programmable with LabVIEW. In fact, NI myRIO uses the same Xilinx Zynq All-Programmable Silicon on Chip technology found in NI's newest CompactRIO, the cRIO-9068.

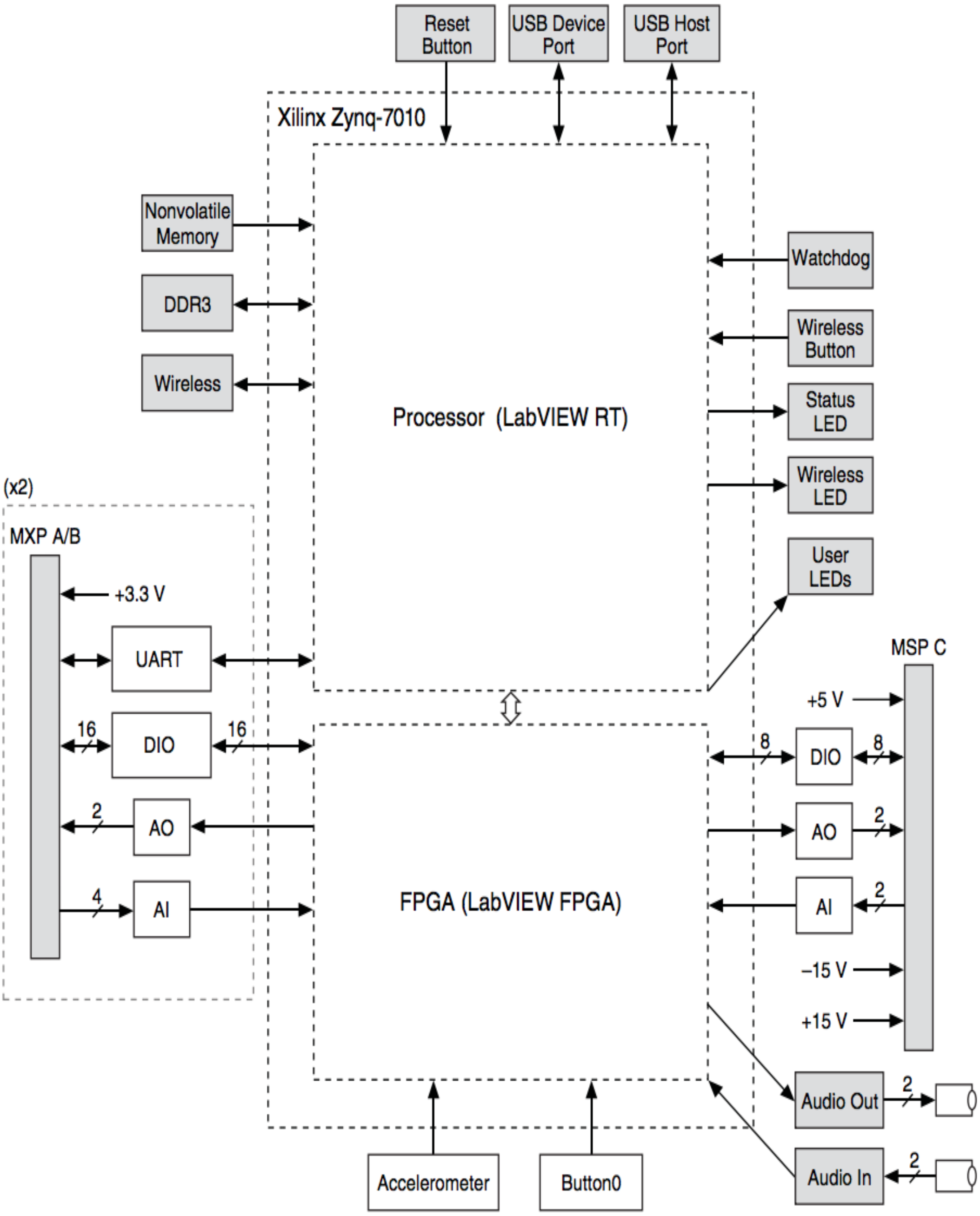

Figure 3.10 NI myRIO-1900 Hardware Block Diagram

NI ships the FPGA of myRIO pre-defined with AI, AO, PWMs, Quad Encoder inputs, UART, SPI. The Hardware architecture of NI myRIO is shown in Figure 3.10. NI myRIO-1900 Expansion Port Connectors A and B carry identical sets of signals. The signals are distinguished in software by the connector name, as in Connector A / DIO1 and Connector B / DIO1. The pin configuration of MXP connectors A and B is shown in Figure 3.11 and the characteristics are shown in Table 3.3.

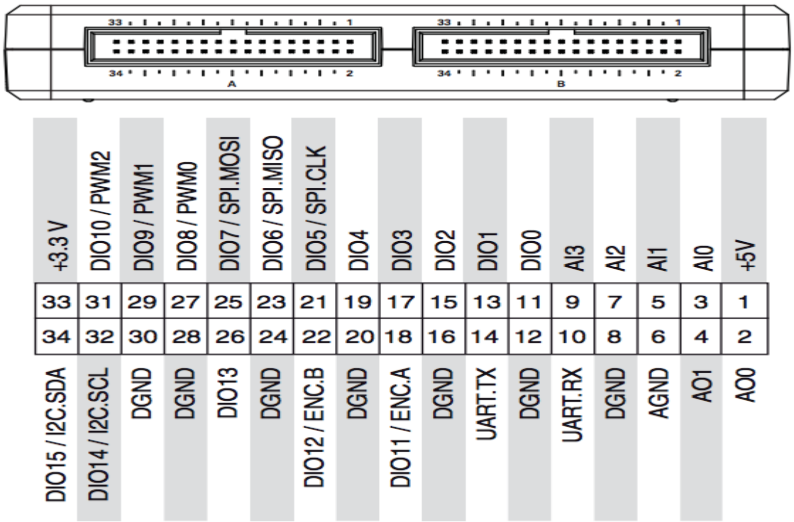

Figure 3.11 Primary/Secondary Signals on MXP Connectors A and B

The pin configuration of MSP connector C is shown in Figure 3.12 and the characteristics are shown in Table 3.4.

TABLE 3.3 CHARACTERISTICS OF MXP CONNECTORS A AND B

S. No	Signal Name	Reference	Direction	Description
1	+5V	DGND	Output	+5 V power output.
2	AI <0-3>	AGND	Input	0-5 V, referenced, single-ended analog input channels.
3	AO <0-1>	AGND	Output	0-5 V referenced, single-ended analog output.
4	AGND	N/A	N/A	Reference for analog input and output.
5	+3.3V	DGND	Output	+3.3 V power output.
6	DIO <0-10>	DGND	Input or Output	General-purpose digital lines with 3.3 V output, 3.3 V/5 compatible input.
7	UART.RX	DGND	Input	UART receive input. UART lines are electrically identical to DIO lines.
8	UART.TX	DGND	Output	UART transmit output. UART lines are electrically identical to DIO lines.
9	DGND	N/A	N/A	Reference for digital signals, +5 V, and +3.3 V.

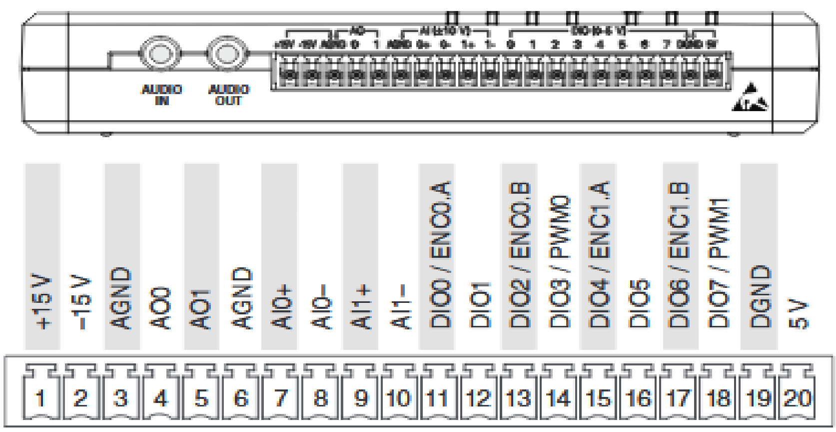

Figure 3.12 Primary/Secondary Signals on MSP Connector C

TABLE 3.4 CHARACTERISTICS OF PRIMARY / SECONDARY SIGNALS ON MSP CONNECTOR C

S. No	Signal Name	Reference	Direction	Description
1	+15V/-15V	AGND	Output	+15 V/-15 V power
2	AI0+/AI0-; AI1+/AI1-	AGND	Input	±10 V, differential analog input channels
3	AO <0..1>	AGND	Output	±10 V referenced, single-ended analog output channels
4	AGND	N/A	N/A	Reference for analog input and output and +15 V/-15 V power output.
5	+5V	DGND	Output	+5 V power output.
6	DIO <0..7>	DGND	Input or Output	General-purpose digital lines with 3.3 V output, 3.3 V/5 compatible input.
7	DGND	N/A	N/A	Reference for digital signals, +5 V power

The NI myRIO microcontroller is programmed with a lab view code to differentiate the position of key presses. LabVIEW is different from most other general-purpose programming languages in two major ways. First, G programming is performed by wiring together graphical icons on a diagram, which is then compiled directly to machine code so the computer processors can execute it. G code developed with LabVIEW executes according to the rules of data flow instead of the more traditional procedural approach. It promotes data as the main concept behind any program. Dataflow execution is data-driven, or data-dependent. The flow of data between nodes in the program, not sequential lines of text, determines the execution order. There are 14 LASER's on one of the longer side no of the cubical input module and 5 LASER's on one of the shorter ones. They are inserted through the holes. On the other two sides which are parallel where LASER's are kept, LDR's are placed. Thus, forming a two-dimensional matrix 5 x 14 which is shown in Figure 3.13.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	~ `	1	2	3	4	5	6	7	8	9	0	_ -	+ =	Back-space
2	Tab	q	w	e	r	t	y	u	i	o	p	[	]	\ |
3	Caps-lock	a	s	d	f	g	h	j	k	l	: ;	" '	Enter	
4			z	x	c	v	b	n	m	< ,	> .	? /		
5	SPACE BAR													

Figure 3.13 Proposed Virtual Keyboard Layout

CHAPTER 4
SOFTWARE DESCRIPTION

The SOLIDWORKS is the software for designing the 3-D model and LabVIEW is used for programming the NI myRIO. The description of these softwares are discussed in this chapter.

4.1 SOLIDWORKS

SolidWorks is a solid modeling computer-aided design and computer-aided engineering computer program that runs on Microsoft Windows. The software used is shown in Figure 4.1.

Figure 4.1 Software for creating 3-D model

SolidWorks is a solid modeler and utilizes a parametric feature-based approach to create models and assemblies. The software uses Parasolid-kernel.

Parameters refer to constraints whose values determine the shape or geometry of the model or assembly. Parameters can be either numeric parameters, such as line lengths or circle diameters, or geometric parameters, such as tangent, parallel, concentric, horizontal or vertical, etc. Numeric parameters can be associated with each other through the use of relations, which allows them to capture design intent.

Design intent is how the creator of the part wants it to respond to changes and updates. For example, the hole at the top of a beverage can has to stay at the top surface, regardless of the height or size of the can. SolidWorks allows the user to specify that the hole is a feature on the top surface and will then honor their design intent no matter what height they later assign to the can.

Features refer to the building blocks of the part. They are the shapes and operations that construct the part. Shape-based features typically begin with a 2-D or 3-D sketch of shapes such as bosses, holes, slots, etc. This shape is then extruded or cut to add or remove material from the part. Operation-based features are not sketch-based, and include features such as fillets, chamfers, shells, applying draft to the faces of a part, etc.

Building a model in SolidWorks usually starts with a 2-D sketch. The sketch consists of geometry such as points, lines, arcs, conics, and splines. Dimensions are added to the sketch to define the size and location of the geometry. Relations are used to define attributes such as tangency, parallelism, perpendicularity, and concentricity. The parametric nature of SolidWorks means that the dimensions and relations drive the geometry, not the other way around. The dimensions in the sketch can be controlled independently, or by relationships to other parameters inside or outside of the sketch.

In an assembly, the analog to sketch relations are mates. Just as sketch relations define conditions such as tangency, parallelism, and concentricity with respect to sketch geometry, assembly mates define equivalent relations with respect to the individual parts or components, allowing the easy construction of assemblies. SolidWorks also includes additional advanced mating features such as gear and cam follower mates, which allow modeled gear assemblies to accurately reproduce the rotational movement of an actual gear train.

Finally, drawings can be created either from parts or assemblies. Views are automatically generated from the solid model, and notes, dimensions and tolerances can then be easily added to the drawing as needed. The drawing module includes most paper sizes and standards.

4.2 LabVIEW

The LabVIEW environment is the ecosystem in which LabVIEW exists. It comprises the different elements that are needed to develop programs within LabVIEW.

Essentially, the LabVIEW environment consists of many tools to aid the development of LabVIEW applications. These include LabVIEW Virtual Instrument (VI) manager, the programming tools, debugging features, templates and ready built sample examples, and an easy interface to the hardware drivers. The explorer environment is shown in Figure 4.2.

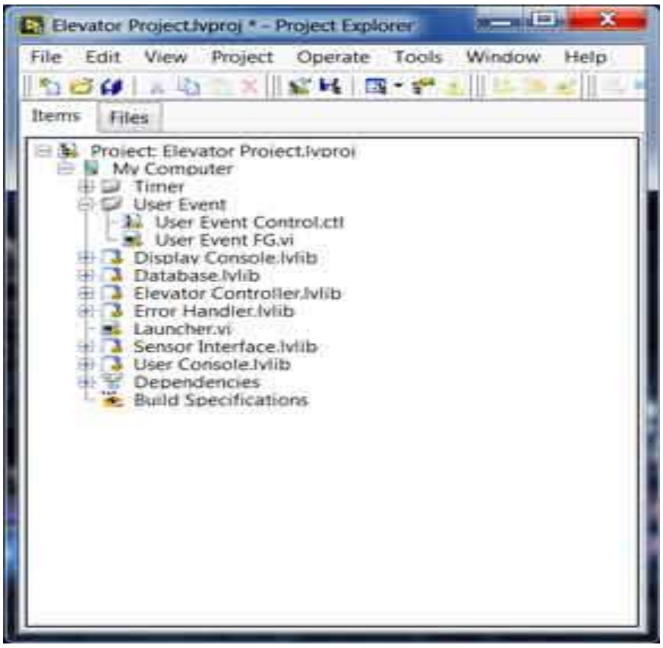

Figure 4.2 Project Explorer Window in LabVIEW

The project explorer is the starting point for all applications. It is where all code modules, libraries and data files are stored for easy management and use. The style is familiar for both Windows and Apple operating system users. It is also where any programming to deploy onto hardware would be stored.

4.2.1 LabVIEW VI

The Virtual Instrument or VI is an integral element of the LabVIEW environment. VI's are individual code modules that make up a complete application.
An application could be as simple as a single VI, but normally many more are included and some applications may utilize hundreds of VI's or possibly more dependent upon the particular application. The virtual instrument front panel in labVIEW is shown in Figure 4.3.

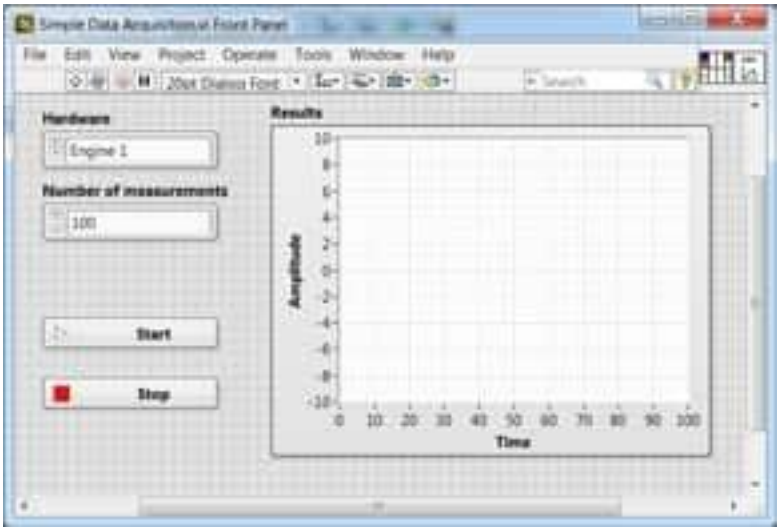

Figure 4.3 Virtual Instrument performing Data Acquisition

4.2.2 VI Front Panel

The LabVIEW front panel is what the user of the completed application will see. It enables them to interact with the VI, inputting controls and also seeing results. It can be likened to the font panel of a test instrument or other piece of equipment.

The LabVIEW VI front panel can be built up from scratch using the palette of different controls, indicators and data types.

4.2.3 VI Block Diagram

The LabVIEW VI block diagram is where the functionality of the VI is programmed in G. The block diagram defines the functionality, and also providing a visual representation of it. In this way the block diagram is similar to a flow diagram within a program. The block diagram is shown in Figure 4.4.

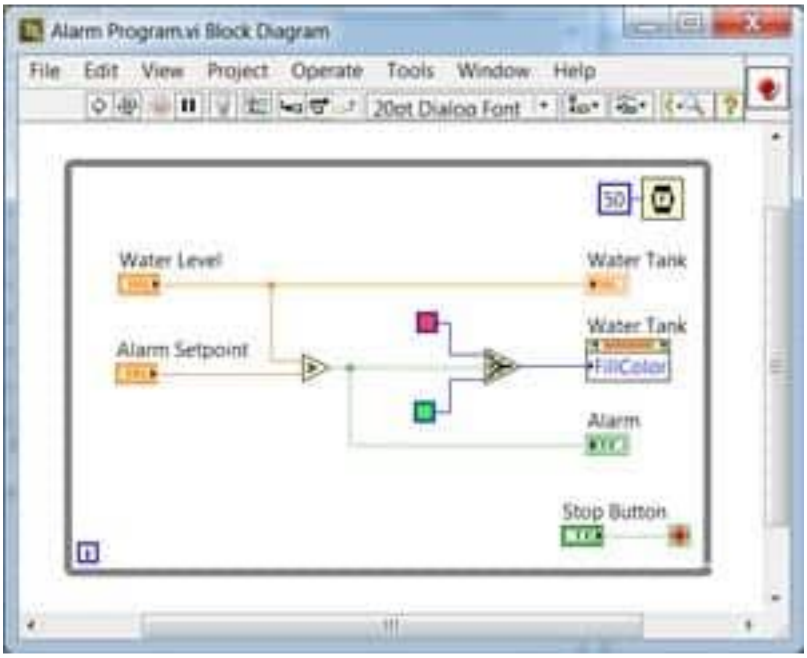

Figure 4.4 Block Diagram in LabVIEW

4.2.4 LabVIEW programming elements

LabVIEW programming is undertaken on the block diagram of the Virtual Instrument. G programming is a technical name for the LabVIEW programming language but nowadays the term is largely unused and the name LabVIEW has become to mean the language as well as the software itself.

As with all programming languages, there are inputs, actions, and outputs. In LabVIEW programming these are known as Controls, Functions and Indicators.

4.2.4.1 Controls

Any LabVIEW control on the front panel will have a corresponding element on the block diagram. The user can input data into the control for use within the overall program. It is possible to connect the controls to a function to perform a particular action. Controls can come in different data types like single, double, string.

The Controls palette can be accessed from the front panel window by selecting View Controls Palette. Alternatively, it can be accessed by right clicking on any empty space in the front panel window.

The Controls palette for LabVIEW programming is broken into various categories these can be exposed as required to show some or all of these categories to suit the requirements for the application.

4.2.4.2 Functions

LabVIEW functions are taken from the Functions palette on the block diagram and are given inputs and they perform an action on this. LabVIEW has a huge range of different functions ranging from simple mathematics to video processing, spectral analysis and the like.

4.2.4.3 Indicators

LabVIEW indicators are similar to controls, having a Front Panel Counterpart in which they display the output of the block diagram to the user. Within the block diagram all the LabVIEW programming elements, i.e. controls, indicators and functions, are connected together. This is achieved using “wires.” The data can be considered to flow along these wires. There are different wire types which are indicated by the color and style of representation. The different controls, functions and indicators are shown in Figure 4.5.

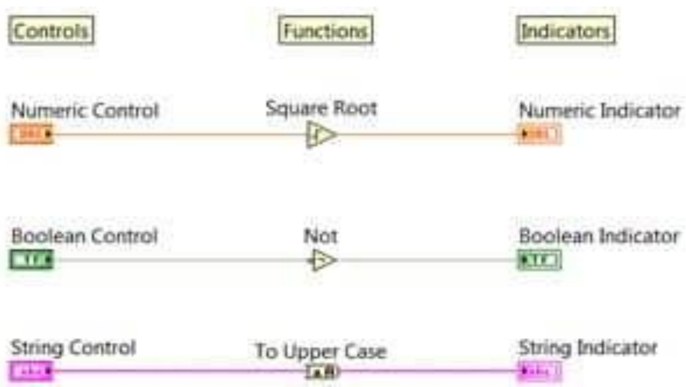

Figure 4.5 Control, Function and Indicator in LabVIEW

Each wire has a single data source, but it is possible to wire it to many VIs and functions that read the data. Wires are different colors, styles, and thicknesses, depending on their data types as shown above. They may be:

- Numeric integer (Blue)
- Numeric floating point (Orange)
- Boolean (Green)
- String (Pink)

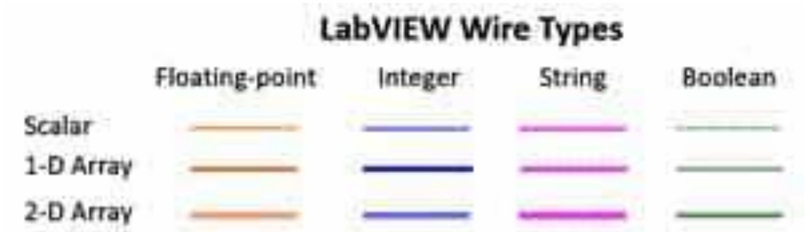

Figure 4.6 Different Types of Wire representation in LabVIEW

The appearance of the wire indicates whether it is scalar, a 1-D array, or a 2-D array. On a LabVIEW screen, a broken wire appears as a dashed black line with a red X in the middle. Broken wires occur for a variety of reasons. One common reason is when wiring two objects with incompatible data types. The different wire types are shown in Figure 4.6.

4.2.5 LabVIEW Dataflow and Programming

With the text-based programming, the order of execution is set up by the order of the lines that is the execution is in a sequential manner. Examples of these languages include Visual Basic, C++, Java.

With the graphical programming, it is set by the dataflow within the diagram. Within this concept, a function is not able to execute until it has received all its inputs. Once it has all of its inputs, it executes its functions and passes on its output to the next node.

In the diagram below data flows from left to right and this means that the multiplication function cannot execute until the divide function has completed. Therefore, the order of execution has been set. It should be noted that the execution follows the actual dataflow and not the position within the window. The dataflow input is shown in Figure 4.7.

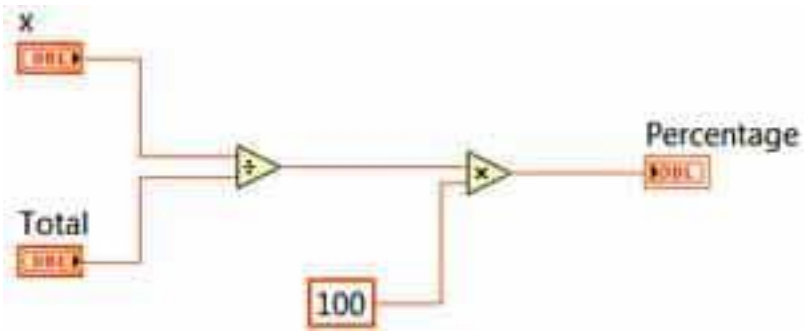

Figure 4.7 Dataflow from Input to Output in LabVIEW

An instrument driver provides the interface between the core LabVIEW software and the instrument or device that it needs to control or receive data from.

The LabVIEW instrument driver is essentially a set of software routines that enable data to be sent to an instrument or device and data received from it. Within the overall driver there are individual sub-routines that enable various tasks to be performed. These include: configuring the device, reading from, writing to, and triggering the device or instrument. Instrument drivers simplify instrument control and reduce LabVIEW program development time by eliminating the need to learn the programming protocol for each instrument.

The LabVIEW driver takes the commands from LabVIEW and then converts them into the instructions required for that device, sending them over the relevant interface whether it be USB, serial, Wi-Fi, Ethernet, GPIB or any other interface applicable for that device.

LabVIEW drivers are developed for a host of different instruments. Some are developed by National Instruments, the company that develops and owns LabVIEW, but other drivers may be developed by third parties – possibly the manufacturers of the devices. Some LabVIEW drivers may incorporate all the remote controls applicable to the device and over a variety of interfaces applicable to the device. Others may have a limited set of instructions or capabilities. Additionally, many are available free of charge, but for others there may be costs associated with them. It depends upon the devices, the manufacturers and the developers of the code.

CHAPTER 5

WORKFLOW AND TESTING

The program for Virtual Keyboard and the testing process employed to determine accuracy and key response time are done in G language. The code flow, the connection diagram and test results are discussed in this chapter.

5.1 CODE FLOW

First the VI specific to myRIO is used to get the data at that instant. The whole workflow is placed in a while loop so that data will be collected repeatedly at every instant. The code for input data collection is shown in Figure 5.1.

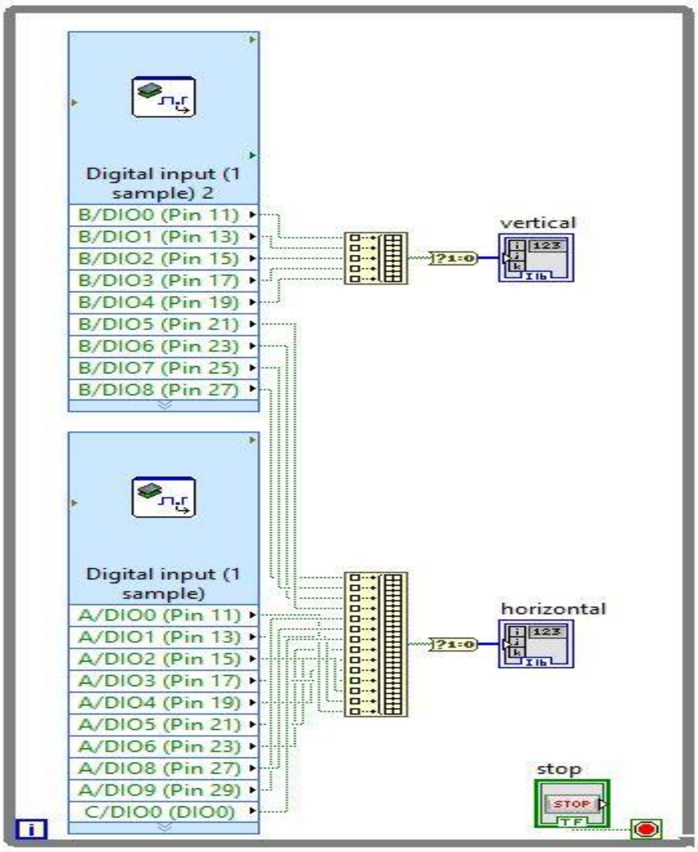

Figure 5.1 Input Data Collection in LabVIEW

If this data input is taken individually and processed repeatedly, it consumes lot of time and is less efficient. Hence the data is taken into arrays to save memory and improve processing of data. Two arrays, one for horizontal rows and another for vertical columns are used to get data in rows and columns. The program thus takes the input as the corresponding coordinate on the matrix that identifies a key press. Pins in ports A, B and C are used as pull up transistors to carry the signal to the microcontroller from the LDR's. A threshold is fixed to differentiate between key press and idle state. During the idle state i.e. when the light falls on the LDR, the voltage across LDR is more than 3 volts which is considered as logic '1' and key detection is not identified. During key press, the light energy from LASER diode is blocked then the voltage across LDR will be less than 3 volts and the corresponding array coordinate for the horizontal row array and vertical column array changes to logic "0". The array coordinates are taken as variables as "x" and "y". The code for processing the input data is shown in Figure 5.2.

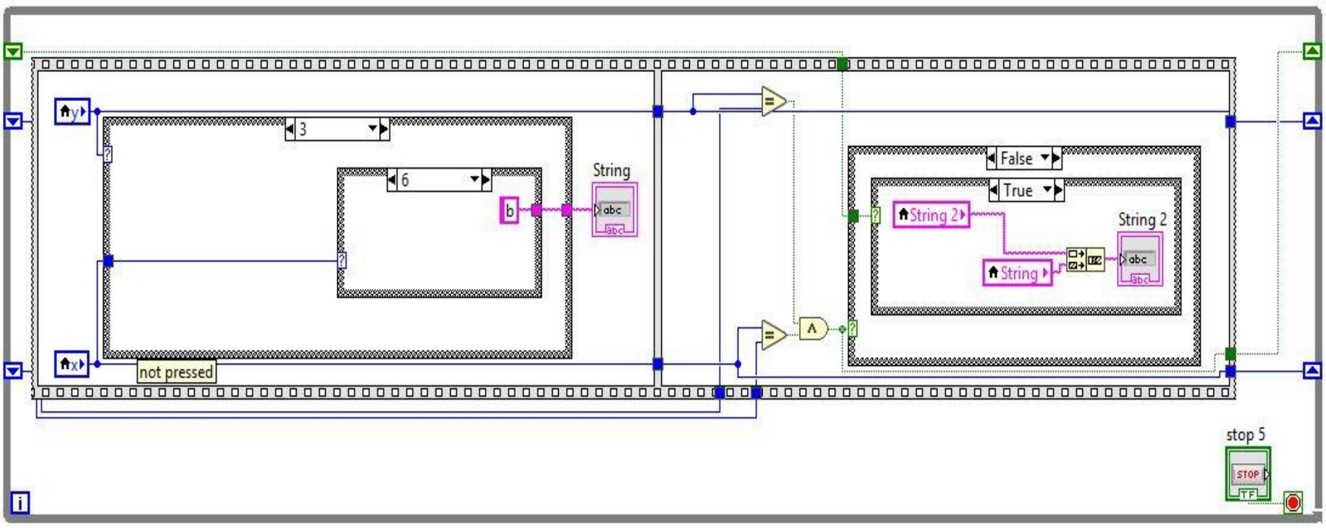

Figure 5.2 Processing of Input Data in LabVIEW

Every key in the keyboard is mapped to all the values of x and y with the help of nested switch statements. The same logic is used to concatenate all the key press to form a single string. The code for conversion of input data into a matrix is shown in Figure 5.3.

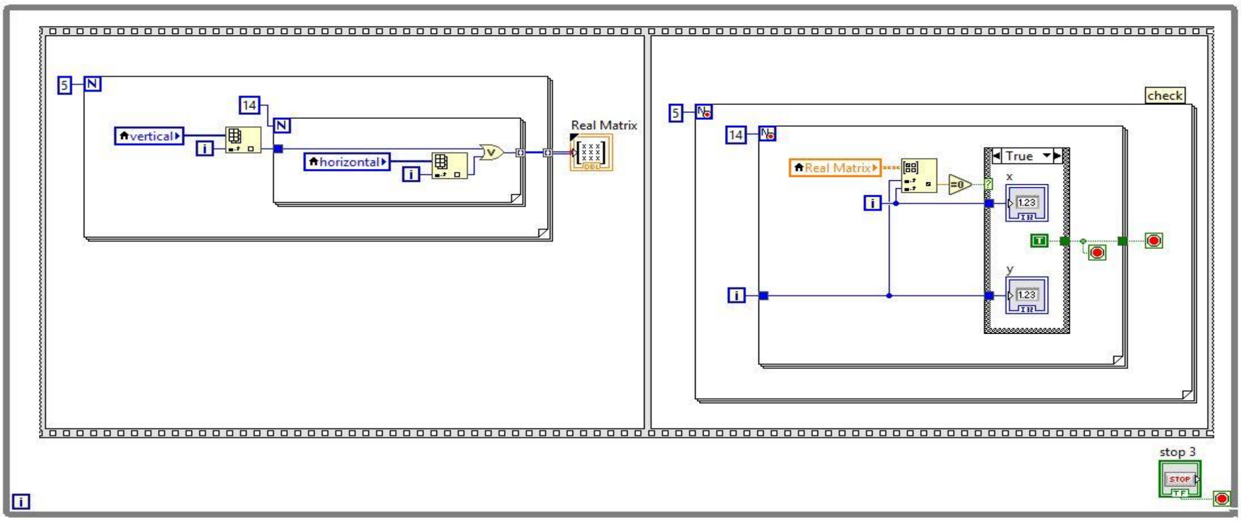

Figure 5.3 Conversion of Data to two-dimensional matrix in LabVIEW

Here, a function named index array is used to get the individual elements from the two arrays and place them in the matrix zeroth index of the two arrays are placed in (0,0) position of the matrix. To take all the elements one at a time, nested for loops are used to get the elements and fill the matrix completely. The next set of nested for loops is used to take the x and y values to map the individual keys.

5.2 CONNECTION DIAGRAM

The connection diagram of this project between NI myRIO and LDR for recognition of a key press is shown in Figure 5.4.

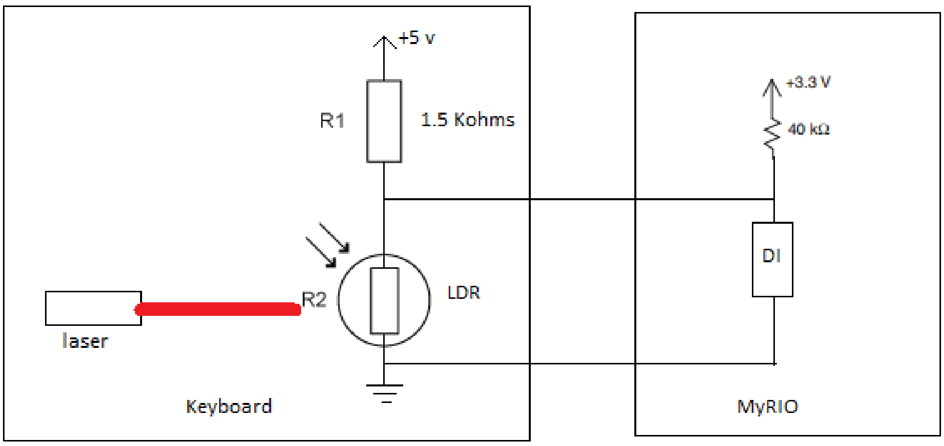

Figure 5.4 Overall Connection of NI myRIO and LDR

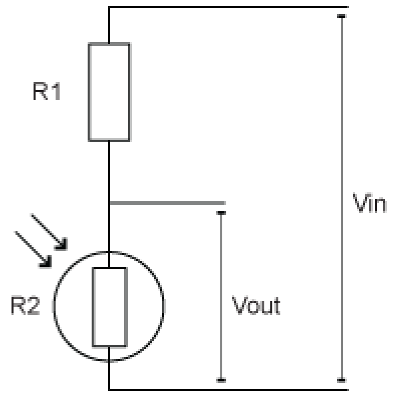

Figure 5.5 Diagram representing Output and Input voltage of LDR

The voltage changes in the LDR is measured by using the voltage dividing circuit as shown in the Figure 5.5 and the formula used to calculate the constant resistor is given below.

R1: 1.5 K ohms

R2: 20M ohm in dark and 300 ohms in Power greater than 11 watt

Vin: 5 Volt

Vout: Given to digital input of NI myRIO with reference to ground

Formula

$$Vout = Vin\ (R2/(R1+R2))$$

The internal connection of the DI pin in myRIO is shown in Figure 5.6.

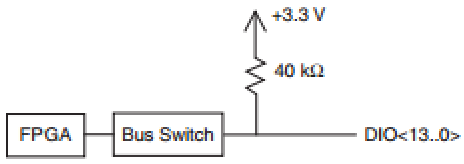

Figure 5.6 Internal Connection of DI pin of myRIO

5.3 VIRTUAL KEYBOARD TEST SUITE

The efficiency in the functioning of the keyboard is tested by identifying the detail of the factors which contribute to it. These factors are keyboard response time and key accuracy.

5.3.1 Keyboard Response Time

Keyboard response time represents the time requirement in milliseconds by the keyboard to capture the key press and display the corresponding character on the screen. In case of normal keyboard, the response time is a factor of the key debounce response, the scan rate, the internal CPU translation, the signaling speed, the interrupt latency of the main computer and the driver code. If the keyboard responds too slowly, it can turn the simplest of typing tasks into a misery. When tapping away at a keyboard the words on the screen should appear at a steady speed. But it’s not uncommon for keyboard response times to lag behind the typing skills of faster writers, or to outpace those needing more time to adjust to learning a new keyboard. Thus, keyboard response time plays a crucial role in determining the better functioning of any keyboard.

Here a testing process is employed to identify the keyboard response time. The testing process involves subtracting the timestamp values at the position of the code, where the input is taken and another time stamp at the position of the code where the key is displayed. Since it is a Virtual Keyboard the debouncing criteria is not applicable but additionally the lab view code pipeline execution time itself will add up to the keyboard response time. The results of the response time are shown in Table 5.1 and the response time is shown in Figure 5.7.

TABLE 5.1 TEST RESULTS OF KEY RESPONSE TIME

S.No	Trials	Keyboard response time (in milliseconds)
1	Trial 1	6
2	Trial 2	8
3	Trial 3	12
4	Trial 4	9
5	Trial 5	8

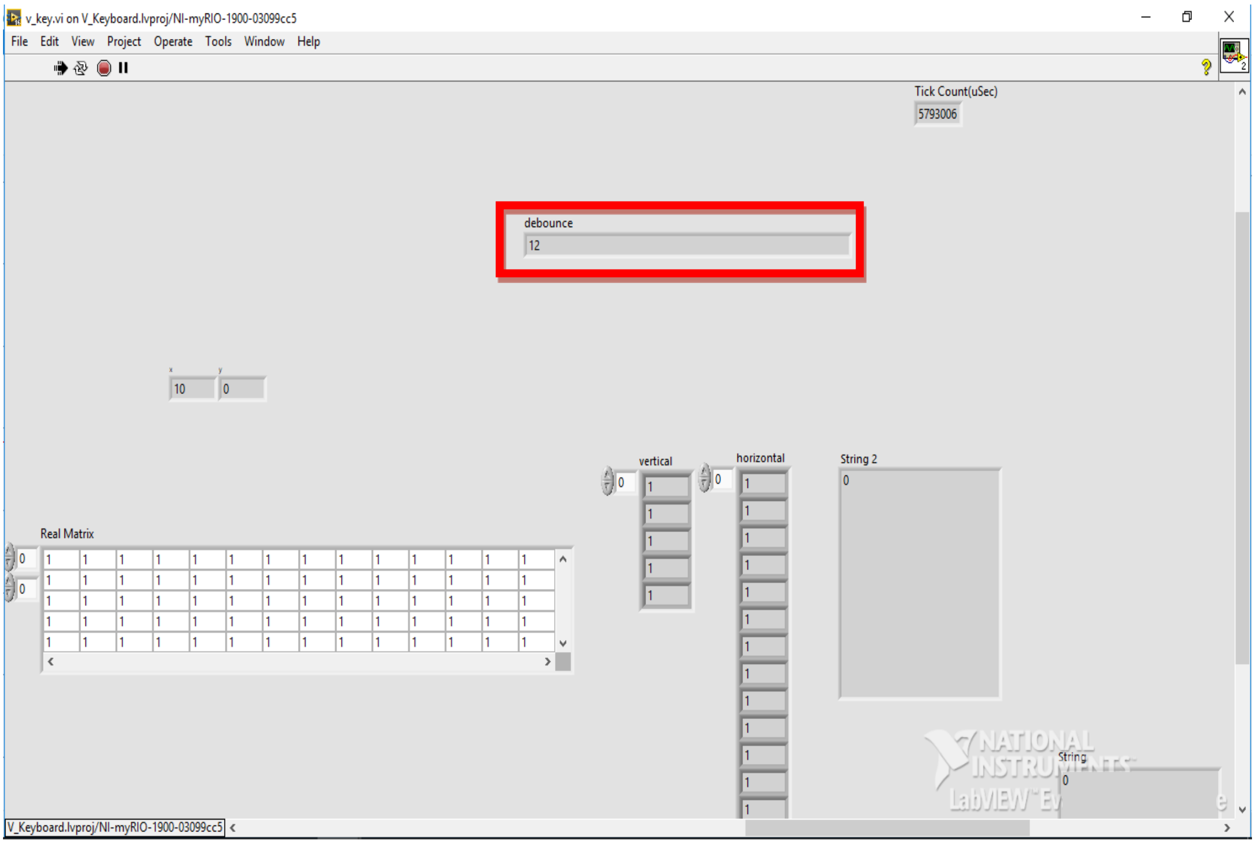

Figure 5.7 Keyboard Response Time Test

The proposed Virtual Keyboard achieves a maximum keyboard response time of 12 ms and on average response time of 8.6 ms.

5.3.2 Key accuracy

Virtual Keyboards are vulnerable to “key stroke deviations”. There are different ways keystroke deviation may occur. When keys are pressed at a faster rate, the keystroke is easy to deviate. Repeated striking of the same key results in unexpected recognition due to deviations. Thus, users cannot get the feedback just as striking a real keyboard. If the fingertip almost falls in the expected key stroke position, the result turns out to be more accurate. When the fingertip falls in the position between the two keys, it might result in erroneous key recognition. Another testing process is employed to arbitrarily identify the accuracy of key recognition. Keys are selected at random and are typed repeatedly and then the key recognition statistics is collected. This is repeated for single characters and multiple characters. The results of the accuracy trial are shown in Table 5.2.

Formula

Accuracy = (Number of success trials / total number of trials) * 100%.

TABLE 5.2 TEST RESULTS OF KEY ACCURACY TRIAL

S.No	Character	No of trials	Accuracy
1	“a”	20	100%
2	“y”	20	100%
3	“r”	20	100%
4	“8”	20	100%
5	“1”	20	100%
6	“pqr”	20	90%
7	“501”	20	95%
8	Space bar	20	100%

The final hardware setup of Virtual Keyboard comprising of the input module and the processor module is shown in Figure 5.8.

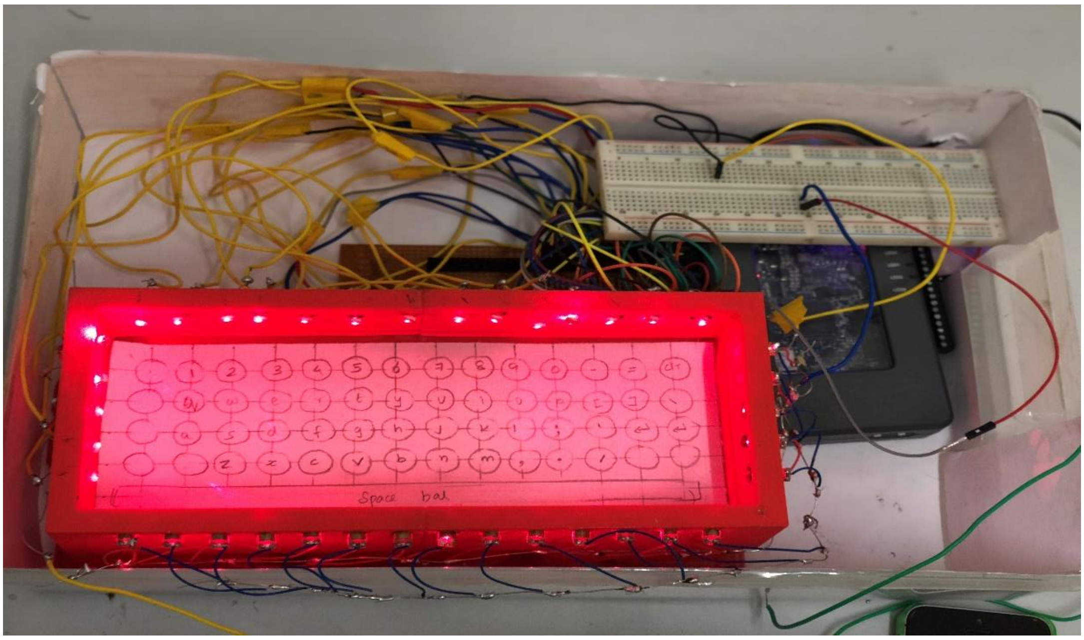

Figure 5.8 Hardware Setup

CHAPTER 6
CONCLUSION

6.1 THESIS RESULT

Nowadays, an increasing number of devices in various streams have been developed to dramatically improve our daily life. Human-computer interaction is a field that expands upon both software and hardware design. In that way, Virtual Keyboards are the most primitive alternative text entry mechanism. Virtual Keyboards are useful tools, which ease effortless entry of textual data. Offering better advantages from the traditional keyboards in the form of portability, security from key logging, resistance to wear and tear, it proves to be a device that anyone would like to own it.

6.2 FUTURE SCOPE

Application Specific Integrated Circuit can replace NI myRIO for improving portability and performance. Also, from the test results, it is observed that the accuracy of the key recognition and robustness can be increased by optimizing the alignment of LASER's and LDR's. Test results show that this Virtual Keyboard can meet the practical needs with good results and replace the traditional mechanical keyboards.

APPENDIX

NI myRIO characteristics

Parameter	Value
Processor type	Xilinx Z-7010
Processor speed	667 MHz
Processor cores	2
Nonvolatile memory	256 MB
DDR3 memory	512 MB
FPGA type	20 MHz
Radio mode	IEEE 802.11 b,g,n
Frequency band	ISM 2.4 GHz
Channel width	20 MHz
TX power	+10 dBm max (10 mW)
Outdoor range	Up to 150m (line of sight)
Antenna directivity	Omnidirectional
Security	WPA, WPA2, WPA2-Enterprise
USB host port	USB 2.0 Hi-Speed
USB device port	USB 2.0 Hi-Speed
Analog Output Resolution	12 bits
Analog Output Overload	±16 V
MSP connector Configuration	Two single-ended channels
MSP connector range	±10 V
MSP connector Absolute accuracy	±200 mV
MSP connector Current drive	2 mA
MSP connector Slew rate	2 V/μs
MXP connectorsConfiguration	Two single-ended channels per connector
MXP connectors Range	0 V to +5 V
MXP connectors Absolute	50 mV
MXP connectors Current drive	3 mA
MXP connectors Slew rate	0.3 V/μs
Audio output Configuration	One stereo output consisting of
Audio output impedance	100Ω in series with 22μF
Audio output Bandwidth	70 Hz to >50 kHz into 32Ω load;2 Hz to
Number of Digital I/O lines MXP	2 ports of 16 DIO lines (one port per

Number of Digital I/O lines MSP	1 port of 8 DIO lines
Direction control	Each DIO line individually
Logic level	5 V compatible LVTTL input; 3.3 V
Minimum pulse width	20 ns
Maximum frequencies for SPI	4 MHz
Maximum frequencies for PWM	100 kHz
Maximum frequencies for	100 kHz
Maximum frequencies for I^2C	400 kHz
UART lines Maximum baud rate	230,400 bps
UART lines Data bits	5, 6, 7, 8
UART lines Stop bits	1, 2
UART lines Parity	Odd, Even, Mark, Space
UART lines Flow control	XON / XOFF
Number of axes in Accelerometer	3
Accelerometer Range	±8 g
Accelerometer Resolution	12 bits
Accelerometer Sample rate	800 S/s
Accelerometer Noise	3.9 mg rms typical at 25 °C
+5 V power output-Output	4.75 V to 5.25 V
+5 V power output -Maximum	100 mA
+3.3 V power output-Output	3.0 V to 3.6 V
+3.3 V power output -Maximum	150 mA

REFERENCES

1. Andrew D. Wilson, (2005) ‘A compact interactive tabletop projection-vision system’, published in the 18th annual ACM symposium on User interface software and technology.

2. Babic,. R.V, (2002) ‘Sensorglove-a new solution for kinematic Virtual Keyboard concept’, Published in First International IEEE Symposium Intelligent Systems.

3. Caslav Livada, Miro Proleta, Kresimir Romic, Hrvoje Leventic, (2017) ‘Beyond the touch: a web camera based Virtual Keyboard’, Published in ELMAR International Symposium.

4. Cihan Topal , Burak Benligiray , Cuneyt Akinlar, (2012) ‘On the efficiency issues of Virtual Keyboard design’, Published in IEEE International Conference Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS).

5. Dikovita. C. R, D.P. Abeysooriya ,(2013) ‘V-Touch: Markerless LASER-Based Interactive Surface’, Published in International Conference on Advances in ICT for Emerging Regions (ICTer).

6. Du. H, T. Oggier, F. Lustenberger et al., (2005) ‘A Virtual Keyboard based on true-3D optical ranging’, published in British Machine Vision Conference.

7. Echtler. F, M. Huber, G. Klinker, (2008) ‘Shadow Tracking on Multi- Touch Tables, Germany’.

8. Hari Singh Dhillon, Rajesh Singla, Navleen Singh Rekhi, Rameshwar Jha, (2009) ‘EOG and EMG based Virtual Keyboard: A brain-computer interface’, Published in IEEE International Conference on Computer Science and Information Technology.
9. Hirobe. Y, T. Niikura, Y. Watanabe, T. Komuro, M. Ishikawa, (2009) ‘Vision-based Input Interface for Mobile Devices with High-speed Fingertip Tracking’, Published in ACM Symposium on User Interface Software and Technology, Canada.
10. Huan Du, Edoardo Charbon, (2008) ‘A Virtual Keyboard system based on Multi-Level Feature Matching’, Published in Conference on Human System Interactions.
11. Hubert Cecotti, (2016) ‘A Multimodal Gaze-Controlled Virtual Keyboard’, Published in IEEE Transactions on Human-Machine Systems.
12. Katz I, Gabayan K, Aghajan H, (2007) ‘A Multi-Touch Surface Using Multiple Cameras, Dept. of Electrical Engineering’, Stanford University.
13. Kenkichi Yamamoto, Satoshi Ikeda, Tokuo Tsuji, and Idaku Ishii, (2006) ‘A Real-time Finger-tapping Interface Using High-speed Vision System’, Published in IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan.
14. Mantyjarvi. J, J. Koivumaki, P. Vuori, (2002) ‘Keystroke recognition for Virtual Keyboard’, Published in IEEE International Conference on Multimedia and Expo.
15. Ming-Wei Chang , Tzi-cker Chiueh , Chia-Ming Chang, (2014) ‘Virtual Keyboard for head mounted display-based wearable devices’, Published in 20th IEEE International Conference on Parallel and Distributed System.

16. Scherer. R, G. R. Muller, C. Neuper et al., (2004) ‘An asynchronously controlled EEG-based Virtual Keyboard: improvement of the spelling rate’, Published in IEEE Transactions on Biomedical Engineering.
17. Soumalya Ghosh1, Sayan Sarcar, Manoj Kumar Sharma Debasis Samanta, (2010) ‘Effective Virtual Keyboard Design with Size and Space Adaptation’, Published in IEEE Students' Technology Symposium (TechSym).
18. Wooguil Pak , Youngrok Cha , Sunki Yeo, (2016) ‘High accessible Virtual Keyboards for preventing key-logging’, Published in Eighth International Conference on Ubiquitous and Future Networks (ICUFN).
19. Xiaolin Su,Yunzhou Zhang, Qingyang Zhao, Liang Gao, (2015) ‘Virtual Keyboard: A human-computer interaction device based on LASER and image processing’ ,Published in IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER).
20. Y. Adajania, J. Gosalia, A. Kanade, H. Mehta, N. Shekokar, (2010) ‘Virtual Keyboard Using Shadow Analysis’, Published in IEEE Conference on Emerging Trends in Engineering and Technology (ICETET).
21. Zhang. Y, Y. Wu, Y. Shan et al., (2001) ‘Visual panel: virtual mouse keyboard and 3D controller with an ordinary piece of paper’, Published in workshop on Perceptive user interfaces.
22. Zhiyuan Zheng, Kun Yang, Ji Pei, (2011) ‘Design and Implement of a Kind of Virtual Keyboard Based on Microcomputer and CMOS Camera’, Published in IEEE 13th International Conference on Communication Technology (ICCT).