Video Interfaced Gaming Device

Project Report submitted in partial fulfillment of the
requirement for the degree of
Bachelor of Technology
in

Electronics and Communication Engineering

under the Supervision of

Prof. D. C. Kulshreshtha

By

Vidit Arora (*091035*)
Sachin Chopra (*091039*)
Pranay Mehrotra (*091040*)

to

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Jaypee University of Information and Technology
Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled **"Video Interfaced Gaming Device"**, submitted by **Vidit Arora** *(091035)*, **Sachin Chopra** *(091039)* and **Pranay Mehrotra** *(091040)* in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 29/5/13

Prof. D. C. Kulshreshtha

Professor

Acknowledgement

Working on this project has provided a brisk and viable experience to us. Our experience while working in this project ranges from the moment of anxiety, when we could not solve for several days, to moments of ecstasy when after struggling for several days we were ultimately able to find solution to our problems. It would not have been possible for us to work on this project without the help of many people. We would like to express our gratitude to all of them but some omissions are inevitable.

It is said that behind the accomplishment of any effort there are dedicated individuals who put in their best to get the job done. Firstly, we would like to express our appreciation and gratitude to our esteemed project guide **Professor D.C. KULSHRESHTHA** for the endless hours of help, suggestions, ideas and advice during the development of this project. Their technical know-how in this subject is only surpassed by their experience. Secondly, we would like to thank **Mrs. MEENAKSHI SOOD** for the practical knowledge she gave us both in terms of Electronics as well as otherwise.

We are also grateful to all the faculty members of the **Electronics and Communication Engineering** department of **Jaypee University of Information Technology** for their sincere remarks and suggestions about our project and grateful for their time to time useful and productive suggestions. We are also expressing our gratitude towards the laboratory technician **Mr. Manoj Pandey** for his valuable efforts in the laboratory that has provided our project to get on to a platform.

Table of Contents

Lists of Figures and Tables

Abstract

The world was first introduced to Gaming Consoles in 1972 when Magnavox released the Odyssey for home use. Though it had limited success, other gaming magnates like Atari and Fairchild followed it up by releasing some hugely successful gaming consoles. Many generations and up-gradations later, presently the gaming console stands revolutionized as a device present in most homes. Currently, the biggest names in the gaming console industry are Sony with the PlayStation® series, Microsoft's Xbox series and the entire range of consoles from Nintendo.

With our project, we aim to accomplish something extraordinary. The aim of the project is to conceptualize one unit out of two extremely popular units used in the modern world. We aim to create a standalone prototype that has a Television screen with an interfaced Gaming Console along with it. This has been done commercially only on a very small scale and only basic games have been attempted to be made. With the advent of High Definition Televisions and Gaming Consoles, we aim to come up with a Gaming Console interfaced Television screen.

Our aim is to implement a small scale model of the gaming console interfaced television. To do so, we will be using the 'Eagle' design tool and use it to design two Printed Circuit Boards (PCBs). One will be used as a signal transmitter and the other as the signal receiver.

The implementation is done using two LCD boards and a sensor. We have made use of optical sensors and an accelerometer. A switch is used to move between TV view and Motion Gaming mode on the circuits.

The programming and the codes are written in BASCOM.

CHAPTER 1

INTRODUCTION

1.1 Conceptualizing the Project

Industry estimates show the global entertainment industry to be worth $2 trillion in 2011. The gaming industry forms a sizeable chunk of this total. While handheld gaming consoles are extremely popular, with the recent changes that have taken **place in the technological world, High Definition gaming is the new state of** normal. With HD gaming consoles now being associated with televisions, virtual reality has never been so close to reality.

However, most gaming consoles are sold independently and can be integrated with almost all television sets, by most different manufacturers. This is very convenient for electronics manufacturers as they can make money by profiting off the sale of both commodities independently.

Our project is planned and making an individual, standalone unit that has a television screen as well as a gaming console interfaced. The hardware implementation for the project involves using two micro-controlled boards and implementing programming codes for two different circuits, a transmitter and a receiver.

We will be implementing the project idea using two Circuit Boards and the Micro Controller ATmega-16. The displays of both the boards will be synchronized. The sensors used to show the switching between the two boards will be an accelerometer and optical sensors.

The further sections of this report aim to elucidate the different stages of our project work which is primarily divided into two parts:

1.1.1 Phase One:

In this phase of our project, our aim was to study the working circuit of an LCD television along with a SONY PlayStation® 2 in order to interface both into a single unit. This aimed at giving us an idea of the complexity of both the circuits involved and whether it will be possible for us, at this level to carry out their interfacing.

Unfortunately, due to complexity involved we realized that it was beyond our scope at this time to undertake this project.

1.1.2 Phase Two

After submitting our report on the study of how the project works, our focus shifted to creating a prototype that could satisfy our objective. To create this prototype, we used two Printed Circuit Boards, the ATmega 16 Microcontroller, the 7805 Voltage Regulator, Liquid Crystal Displays, Light Emitting Diodes, basic electrical components and an accelerometer to mimic a gaming device.

This report primarily deals with *'Phase Two'* of our study.

1.2 Literature Study

The initial phase of the project has primarily been a literature analysis followed by evaluations of the compatibility of the two individual devices. Since this has not been attempted before and currently, the market does not have any similar product being sold. While we have not conducted a cost-effectiveness based study on this, if that is successfully implemented, this project goes beyond the scope of our current abilities.

This caused us to rethink our plan and in turn, decide to carry out only a small part of the study of this project as the first phase of our project.

Obtaining the Data Sheets of the components used was not an easy task. Due to discrepancies between the models of the devices sold in different countries, the service manuals we obtained also belong to another country. This has meant that we are unsure whether we will be able to successfully obtain the hardware through Indian suppliers.

The cost of the units is also very high. The SONY® PlayStation series has now reached the 3rd stage. The PlayStation® 2 was discontinued and the PlayStation® 3 series starts at Rs. 21,000 in the market. We have faced difficulties while trying to obtain a working (preferably second-hand) model of the PlayStation® 2. The cost of the television set is also northwards of the Rs. 15,000 mark.

Apart from the monetary issues, another major issue we have faced is the viability of the project. All the components we have studied are copyrighted and patented by SONY®. We do not want to violate any laws and due to this reason meddling with their circuits may turn out to be a bad idea.

To implement the circuital interfacing of a Television unit with a Gaming Console, an initial study of the Television Circuit was carried out. Also, the inside circuitry of the SONY PlayStation® was also carried out.

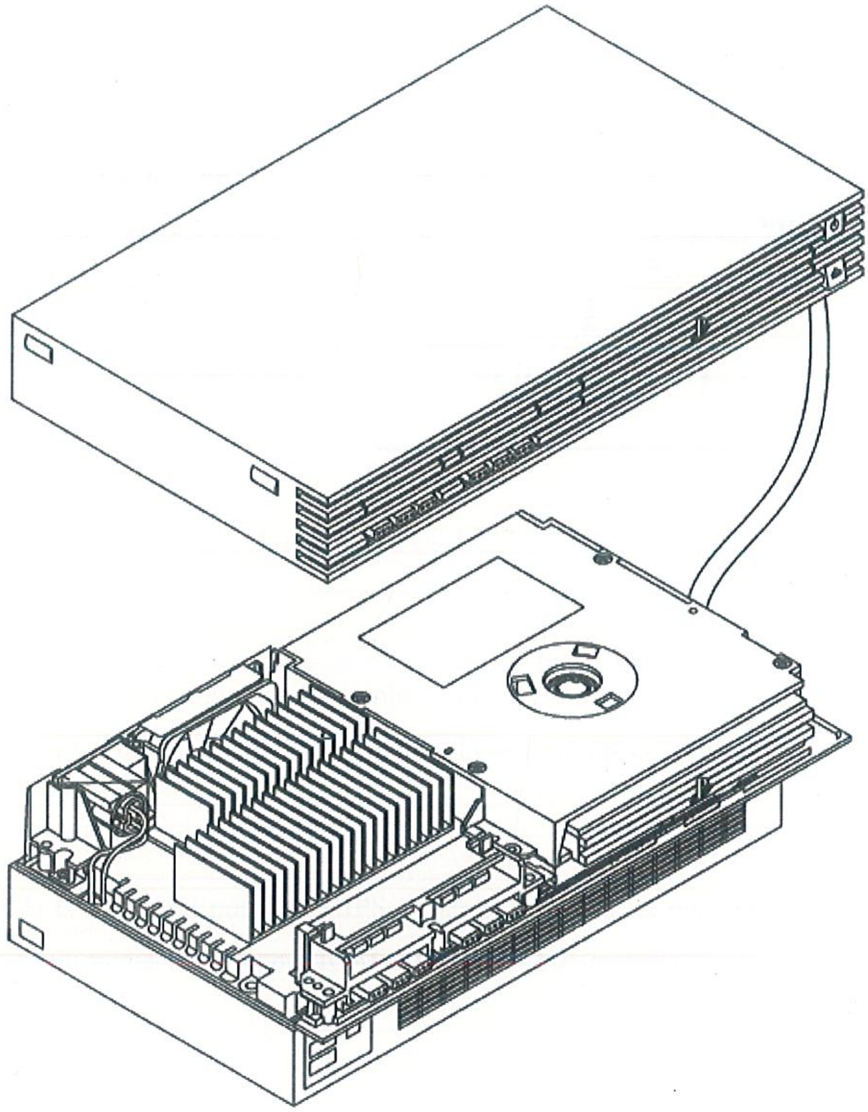

Figure 1: PlayStation 2

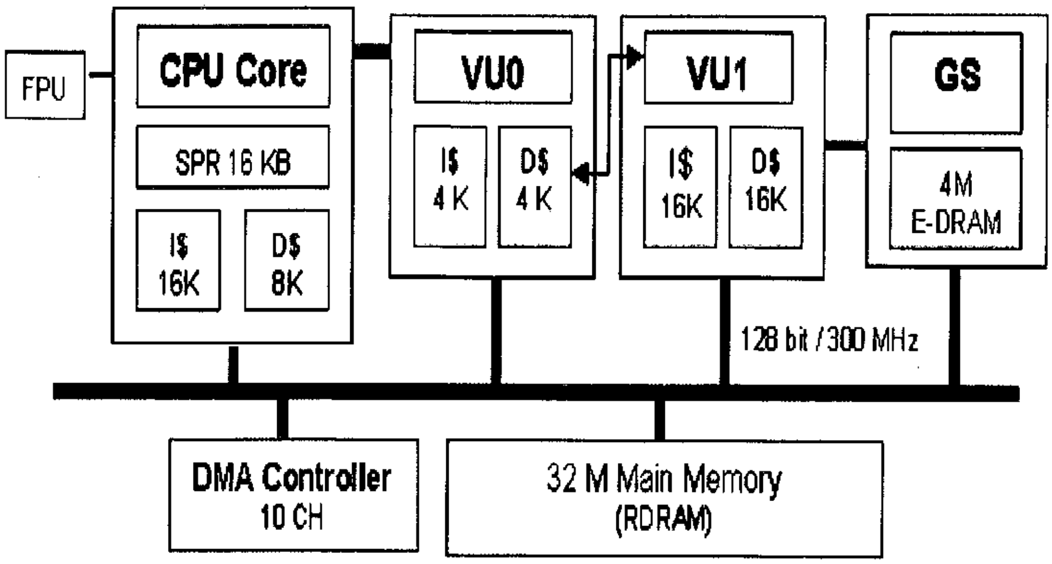

Figure 2: Schematic PlayStation 2

The above diagram shows the basic block diagram of the internal motherboard of the PlayStation® 2 motherboard. The individual components are defined as follows:

The CPU is a general purpose MIPS variant CPU with its own FPU, 128 bit SIMD integer multimedia extensions, ICACHE, DCACHE and a special on-chip "Scratch pad" memory of 16K.

Vector Units

These units are responsible for integral and float logic operations. These take place simultaneously with a single cycle throughput.

- **Vector Unit 0 (VU 0)** – It has a 4k instruction RAM and another 4k data RAM. This is connected to the CPU and is capable of being a standalone unit.

- **Vector Unit 1 (VU 1)** – It has a 16k instruction RAM and another 16k data RAM. This is connected to the Graphics Synthesizer. It has no impact on CPU processing.

Graphics Synthesizer (GS)

It has 4 MB of embedded DRAM making the GS extremely quick at calculations pertaining to polygon setup and fill-rate operations. This unit can support dots, triangles, stripes, fans, lines and sprites.

DMA Controller (DMAC)

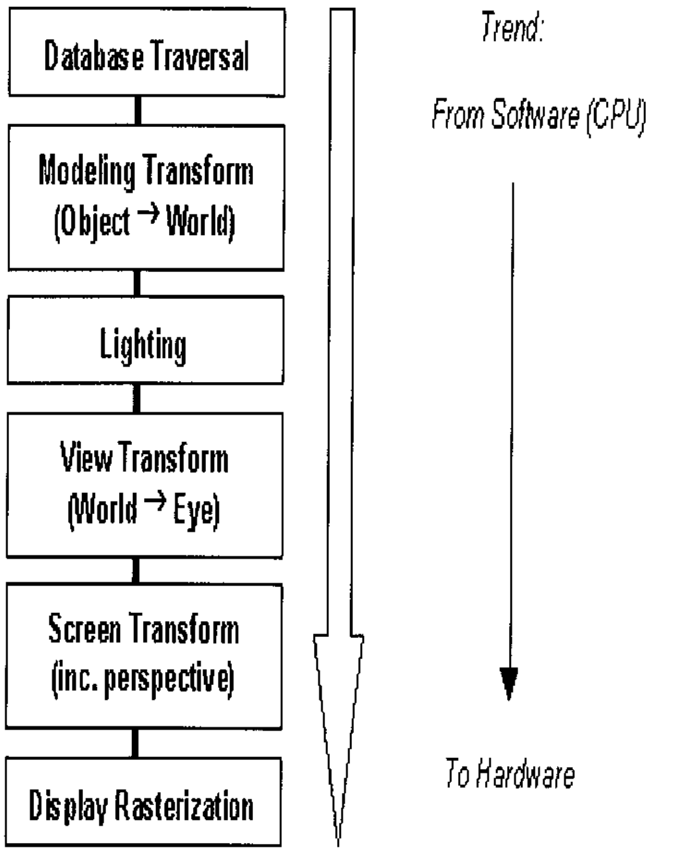

Figure 3: DMAC

It acts as a bus and is responsible for all the data transition between the CPU, vector units and all other processing units.

When the VU1 accepts data from the DMA, it has another parallel unit which can perform data unpacking and re-formatting operations so that the input stream is in the perfect format for VU1 microcode operation. This unit also allows for VU1 data memory to be double buffered so that data can be loaded into the VU1 via DMA at the same time as the VU1 is processing data and sending primitives to the GS.

1.3 Voltage Regulators Used

A voltage regulator is needed to obtain a fixed DC supply voltage to be supplied to the Microcontroller. The 78xx are positive voltage regulators where "xx" indicates the output voltage level. 7805 is a +5V DC voltage regulator.

7805 is an integrated three-terminal positive fixed linear voltage regulator. It supports an input voltage of 7 volts to 35 volts and output voltage of 5 volts. It typically has a current rating of 1 amp although both higher and lower current models are available. Its output voltage is fixed at 5.0V. The 7805 also has a built-in current limiter as a safety feature. The 7805 will automatically reduce output current if it gets too hot. The 7805 is one of the most common and well-known of the 78xx series regulators, as its small component count and medium-power regulated 5V make it useful for powering TTL devices.

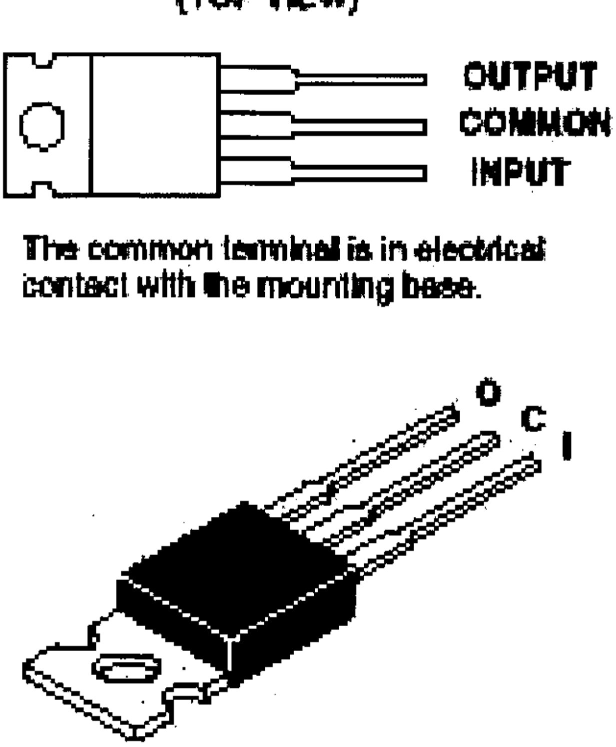

Figure 2: 7805 Voltage Regulator

1.4 Liquid Crystal Displays

LCD is Liquid Crystal Display that uses the light modulating properties of liquid crystals. LCD displays utilize two sheets of polarizing material with a liquid crystal solution between them. An electric current passed through the liquid causes the crystals to align so that light cannot pass through them. Each crystal, therefore, is like a shutter, either allowing light to pass through or blocking the light.

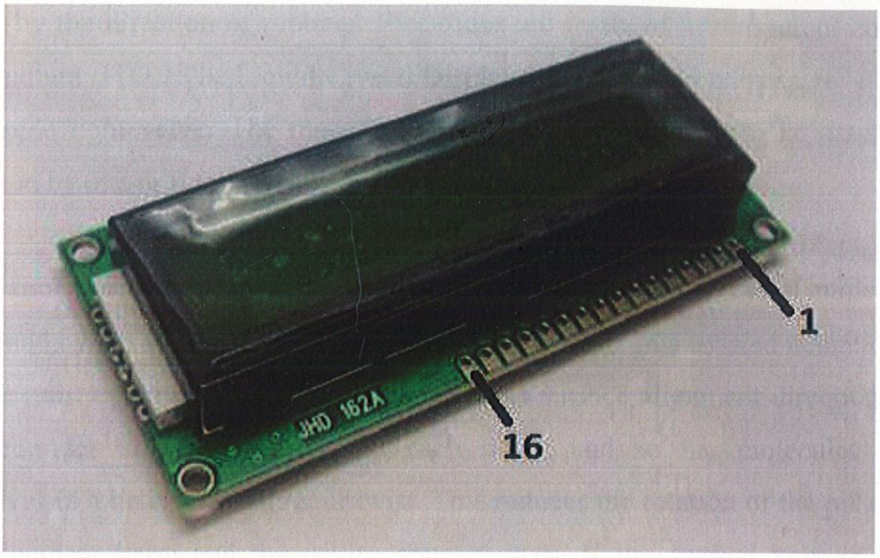

Figure 3: Liquid Crystal Display

An LCD monitor consists of five layers: a backlight, a sheet of polarized glass, a "mask" of pixels, a layer of liquid crystal solution responsive to a wired grid of x, y coordinates, and a second polarized sheet of glass. By manipulating the orientations of crystals through precise electrical charges of varying degrees and voltages, the crystals act like tiny shutters, opening orclosing in response to the stimulus, thereby allowing degrees of light that have passed through specific colored pixels to illuminate the screen, creating a picture.

Each pixel of an LCD typically consists of a layer of molecules aligned between two transparent electrodes, and two polarizing filters, the axes of transmission of which are (in most of the cases) perpendicular to each other. With no actual liquid crystal between the polarizing filters, light passing through the first filter would be blocked by the second (crossed) polarizer.

The surfaces of the electrodes that are in contact with the liquid crystal material are treated so as to align the liquid crystal molecules in a particular direction. This treatment typically consists of a thin polymer layer that is unidirectionally rubbed using, for example, a cloth. The direction of the liquid crystal alignment is then defined by the direction of rubbing. Electrodes are made of a transparent conductor called Indium (ITO). The Liquid Crystal Display is intrinsically a "passive" device; it is a simple light valve. The managing and control of the data to be displayed is performed by one or more circuits commonly denoted as LCD drivers.

Before applying an electric field, the orientation of the liquid crystal molecules is determined by the alignment at the surfaces of electrodes. In a twisted nematic device (still the most common liquid crystal device), the surface alignment directions at the two electrodes are perpendicular to each other, and so the molecules arrange themselves in a helical structure, or twist. This reduces the rotation of the polarization of the incident light, and the device appears grey. If the applied voltage is large enough, the liquid crystal molecules in the center of the layer are almost completely untwisted and the polarization of the incident light is not rotated as it passes through the liquid crystal layer. This light will then be mainly polarized perpendicular to the second filter, and thus be blocked and the pixel will appear black. By controlling the voltage applied across the liquid crystal layer in each pixel, light can be allowed to pass through in varying amounts thus constituting different levels of gray.

The optical effect of a twisted nematic device in the voltage-on state is far less dependent on variations in the device thickness than that in the voltage-off state. Because of this, these devices are usually operated between crossed polarizers such that they appear bright with no voltage (the eye is much more sensitive to variations in the dark state than the bright state). These devices can also be operated between parallel polarizers, in which case the bright and dark states are reversed. The voltage-off dark state in this configuration appears blotchy, however, because of small variations of thickness across the device.

Both the liquid crystal material and the alignment layer material contain ionic compounds. If an electric field of one particular polarity is applied for a long period of time, this ionic material is attracted to the surfaces and degrades the device performance. This is avoided either by applying an alternating current or by reversing the polarity of the electric field as the device is addressed (the response of the liquid crystal layer is identical, regardless of the polarity of the applied field).

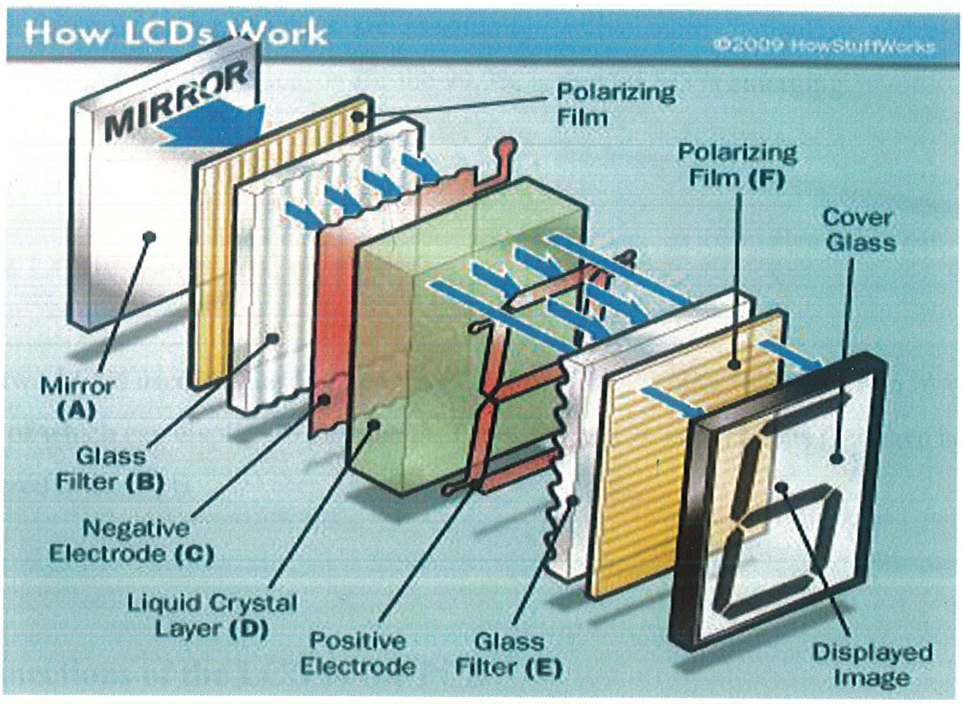

Figure 4: How the LCD works

1.4.1 Types of LCDs:

There are mainly two types of LCD

- **Passive Display**

Passive displays are widely used with segmented digits and characters for small readouts in devices such as calculators, fax machines and remote controls, most of which are monochrome or have only a few colors.

• **Active Display**

Used in all LCD TVs and desktop computer monitors and 99.9% of all laptops, active displays are essentially "active matrix" displays and almost always color. The reason for the 99.9% is that OLED is emerging.

1.4.2 LCDs used

The two LCDs used are the 16x2 passive displays. This implies they have two rows, both of which can display 16 characters. This LCD has 16 connections that have been soldered to the PCB.

Connections of the LCD to the PCB

- Pin 1 of LCD to ground
- Pin 2 of LCD to Vcc (5V)
- Pin 3 of LCD to ground
- Pin 4 of LCD to PortB.2 (2nd pin of port B)
- Pin 5 of LCD to Ground
- Pin 6 of LCD to PortB.3
- Pin 7-10 are left floating (open)
- Pin 11 of LCD to PortB.4
- Pin 12 of LCD to PortB.5
- Pin 13 of LCD to PortB.6
- Pin 14 of LCD to PortB.7
- Pin 15 of LCD to Vcc (5V)
- Pin 16 of LCD to Ground

1.5 Sensors

A sensor is a device that measures a particular characteristic of an object or system. Some sensors are purely mechanical, but most sensors are electronic, returning a voltage signal that can be converted into a useful engineering unit. Sensors take advantage of the mechanical or electrical response of its components to relate the response to a relevant quantity. Engineers use sensors in test and monitoring applications, but homeowners interact with sensors every day. Automobiles are filled with sensors, from the engine to the airbag.

Electrical sensors examine the change in electrical or magnetic signals based on an environmental input. Examples of electrical sensors are metal detectors, RADAR systems and even simple electrical meters such as voltmeters and ohmmeters. Magnetometers are widely used in traffic intersections to detect the presence of a vehicle.

For the prototype, sensors have been employed to sense and display change. When the prototype circuit is switched to 'gaming mode', there are two sensors that are activated. These sensors are:

a) Accelerometer
b) Light Emitting Diodes

These have been discussed in further detail.

1.5.1 ADXL335 Accelerometer

An accelerometer is an electromechanical device that will measure acceleration forces. These forces may be static, like the constant force of gravity pulling at your feet, or they could be dynamic - caused by moving or vibrating the accelerometer.

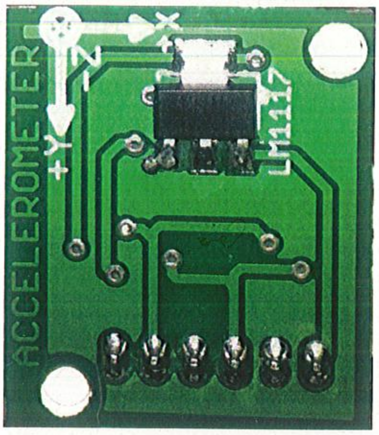

Figure 5: Accelerometer Circuit

By measuring the amount of static acceleration due to gravity, you can find out the angle the device is tilted at with respect to the earth. By sensing the amount of dynamic acceleration, you can analyze the way the device is moving. At first, measuring tilt and acceleration doesn't seem all that exciting. However, engineers have come up with many ways to make really useful products with them.

In the computing world, IBM and Apple have recently started using accelerometers in their laptops to protect hard drives from damage. If you accidentally drop the laptop, the accelerometer detects the sudden freefall, and switches the hard drive off so the heads don't crash on the platters. In a similar fashion, high g accelerometers are the industry standard way of detecting car crashes and deploying airbags at just the right time.

The accelerometer sensor used by us is the ADXL335, 3-axis accelerometer. However, for the purpose of this project only the x-axis has been employed. The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

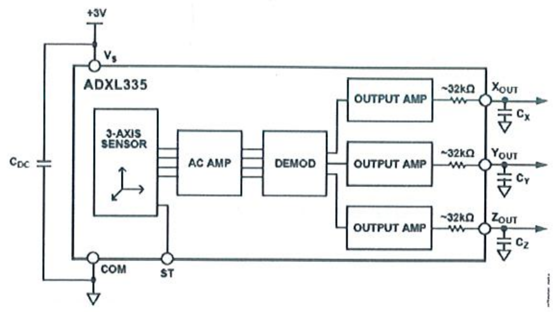

Figure 6: Block Diagram of ADXL335

The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of ±3 g mini-mum. It contains a polysilicon surface-micromachined sensor and signal conditioning circuitry to implement an open-loop acceleration measurement architecture. The output signals are analog voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

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

Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

The demodulator output is amplified and brought off-chip through a 32 kΩ resistor. The user then sets the signal bandwidth of the device by adding a capacitor. This filtering improves measurement resolution and helps prevent aliasing.

Mechanical Sensor

The ADXL335 uses a single structure for sensing the X, Y, and Z axes. As a result, the three axes' sense directions are highly orthogonal and have little cross-axis sensitivity. Mechanical misalignment of the sensor die to the package is the chief source of cross-axis sensitivity. Mechanical misalignment can, of course, be calibrated out at the system level.

Performance

Rather than using additional temperature compensation circuitry, innovative design techniques ensure that high performance is built in to the ADXL335. As a result, there is no quantization error or non-monotonic behavior, and temperature hysteresis is very low (typically less than 3 mg over the −25°C to +70°C temperature range).

1.5.2 Light Emitting Diode (LED)

A light-emitting diode (LED) is a semiconductor light source. When a light forward biased (switched on), electrons are able to recombine with electron holes within the device, releasing energy in the form of photons.

A diode is the simplest sort of semiconductor device. Broadly speaking, a semiconductor is a material with a varying ability to conduct electrical current. Most semiconductors are made of a poor conductor that has had impurities (atoms of another material) added to it. The process of adding impurities is called doping.

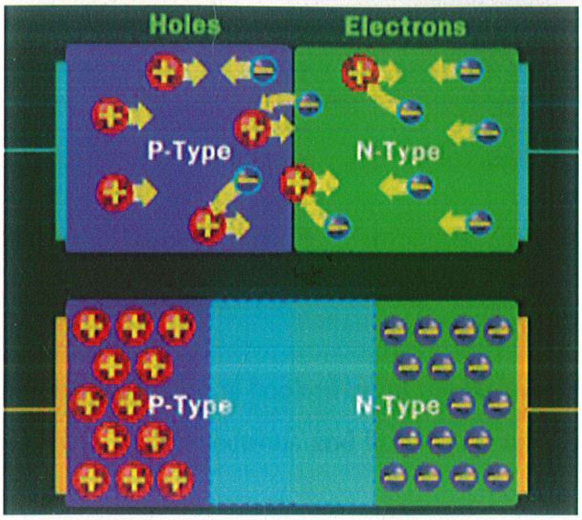

Figure 7: Working of a Diode

At the junction, free electrons from the N-type material fill holes from the P-type material. This creates an insulating layer in the middle of the diode called the depletion zone.

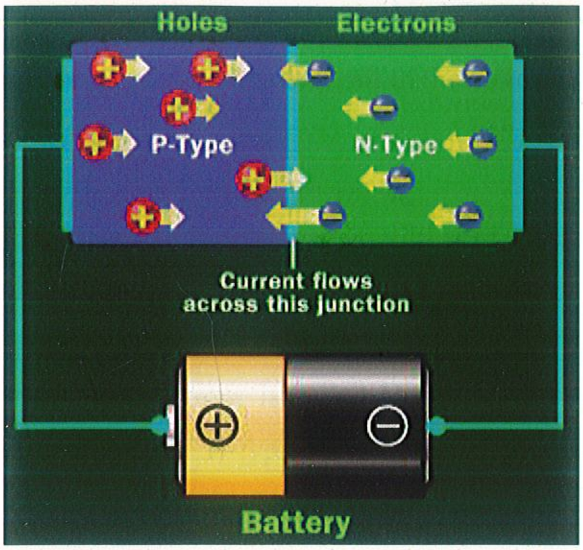

Figure 8: Diode in Forward Bias

When the negative end of the circuit is hooked up to the N-type layer and the positive end is hooked up to P-type layer, electrons and holes start moving and the depletion zone disappears.

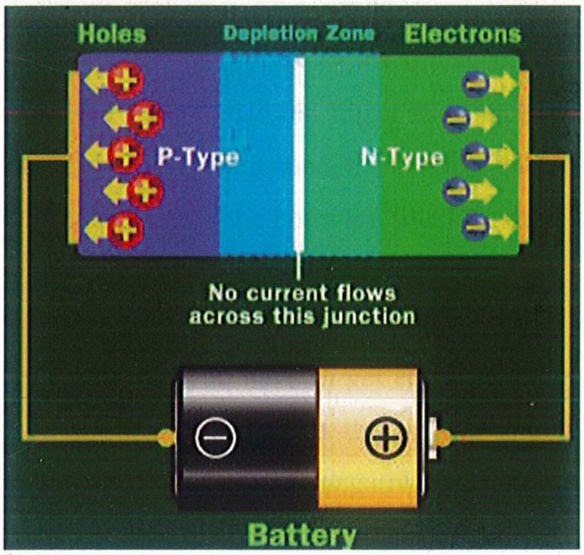

Figure 9: Diode in Reverse Bias

In the case of LEDs, the conductor material is typically aluminum-gallium-arsenide (AlGaAs). In pure aluminum-gallium-arsenide, all of the atoms bond perfectly to their neighbors, leaving no free electrons (negatively charged particles) to conduct electric current. In doped material, additional atoms change the balance, either adding free electrons or creating holes where electrons can go. Either of these alterations make the material more conductive.

A semiconductor with extra electrons is called N-type material, since it has extra negatively charged particles. In N-type material, free electrons move from a negatively charged area to a positively charged area.

A semiconductor with extra holes is called P-type material, since it effectively has extra positively charged particles. Electrons can jump from hole to hole, moving from a negatively charged area to a positively charged area. As a result, the holes themselves appear to move from a positively charged area to a negatively charged area.

A diode consists of a section of N-type material bonded to a section of P-type material, with electrodes on each end. This arrangement conducts electricity in only one direction. When no voltage is applied to the diode, electrons from the N-type material fill holes from the P-type material along the junction between the layers, forming a depletion zone. In a depletion zone, the semiconductor material is returned to its original insulating state -- all of the holes are filled, so there are no free electrons or empty spaces for electrons, and charge can't flow.

To get rid of the depletion zone, you have to get electrons moving from the N-type area to the P-type area and holes moving in the reverse direction. To do this, you connect the N-type side of the diode to the negative end of a circuit and the P-type side to the positive end. The free electrons in the N-type material are repelled by the negative electrode and drawn to the positive electrode. The holes in the P-type material move the other way. When the voltage difference between the electrodes is high enough, the electrons in the depletion zone are boosted out of their holes and begin moving freely again. The depletion zone disappears, and charge moves across the diode.

If you try to run current the other way, with the P-type side connected to the negative end of the circuit and the N-type side connected to the positive end, current will not flow. The negative electrons in the N-type material are attracted to the positive electrode. The positive holes in the P-type material are attracted to the negative electrode. No current flows across the junction because the holes and the electrons are each moving in the wrong direction. The depletion zone increases.

1.5.2.1 How Can a Diode Produce Light?

Light is a form of energy that can be released by an atom. It is made up of many small particle-like packets that have energy and momentum but no mass. These particles, called photons, are the most basic units of light.

Photons are released as a result of moving electrons. In an atom, electrons move in orbitals around the nucleus. Electrons in different orbitals have different amounts of energy. Generally speaking, electrons with greater energy move in orbitals farther away from the nucleus.

For an electron to jump from a lower orbital to a higher orbital, something has to boost its energy level. Conversely, an electron releases energy when it drops from a higher orbital to a lower one. This energy is released in the form of a photon. A greater energy drop releases a higher-energy photon, which is characterized by a higher frequency.

As we saw in the last section, free electrons moving across a diode can fall into empty holes from the P-type layer. This involves a drop from the conduction band to a lower orbital, so the electrons release energy in the form of photons. This happens in any diode, but you can only see the photons when the diode is composed of certain material. The atoms in a standard silicon diode, for example, are arranged in such a way that the electron drops a relatively short distance. As a result, the photon's frequency is so low that it is invisible to the human eye -- it is in the infrared portion of the light spectrum. This isn't necessarily a bad thing, of course: Infrared LEDs are ideal for remote controls, among other things.

Visible light-emitting diodes (VLEDs), such as the ones that light up numbers in a digital clock, are made of materials characterized by a wider gap between the conduction band and the lower orbitals. The size of the gap determines the frequency of the photon -- in other words, it determines the color of the light. While LEDs are used in everything from remote controls to the digital displays on electronics, visible LEDs are growing in popularity and use thanks to their long lifetimes and miniature size. Depending on the materials used in LEDs, they can be built to shine in infrared, ultraviolet, and all the colors of the visible spectrum in between.

CHAPTER 2

Microcontroller

A microcontroller (sometimes abbreviated μC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Some common microcontrollers and vendors are:

- ARM core processors (from many vendors)
- Atmel AVR (8-bit), AVR32 (32-bit), and AT91SAM (32-bit)
- Cypress Semiconductor PSoC (Programmable System-on-Chip)
- Freescale Cold Fire (32-bit) and S08 (8-bit)
- Freescale 68HC11 (8-bit)
- Intel 8051

2.1 Advantages of using a Microcontroller

A Microcontroller can be defined as a "Computer-on-Chip". Just as a Personal Computer has input devices like keyboards, mouse, etc, output devices like monitor, printer, etc and the Central Processing Unit, a Microcontroller also has Input/Output ports and a Processor embedded into a single chip. A Microcontroller is a complete self-sufficient system usually requiring no external components.

Often people misuse the terms "Microprocessor" and "Microcontroller", but both of them are different. A Microprocessor is just a processor similar to a CPU of a PC. A Microprocessor has no Input/Output ports or Memory on chip. A lot of peripheral components are required for the functioning of the Microprocessor. While a Microcontroller is a complete self-sufficient system with on chip I/O ports, Memory and various other features.

Due to this reason, the ATmega 16 was the microcontroller selected.

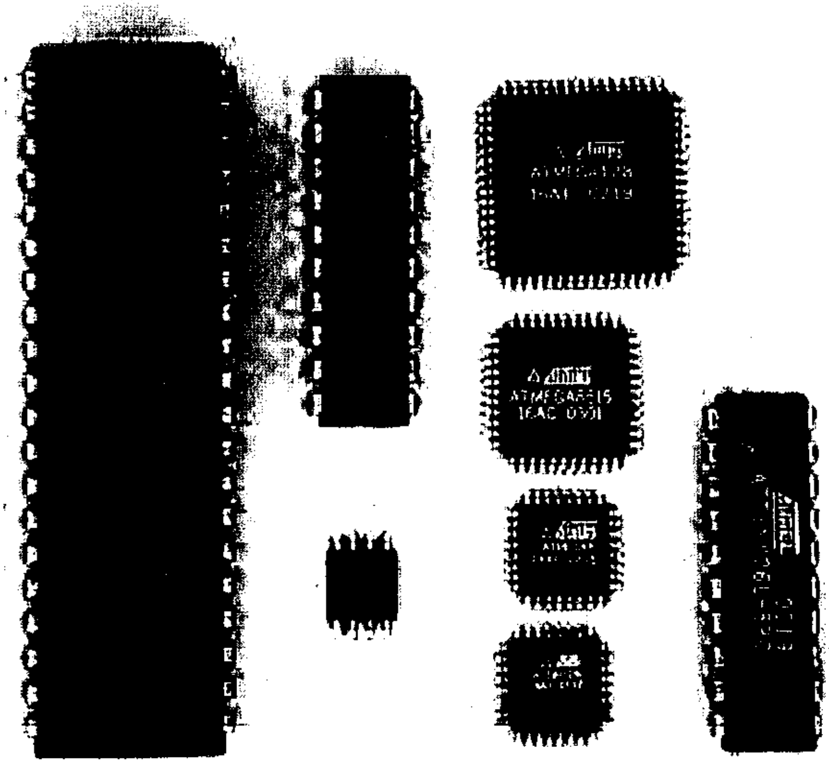

Figure 10: ATMEL Family of Microcontrollers

2.1.1 AVR Family

The AVR is a modified Harvard architecture 8-bit RISC single chip microcontroller which was developed by Atmel in 1996. The AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

2.2 Atmega 16 Microcontroller

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

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.

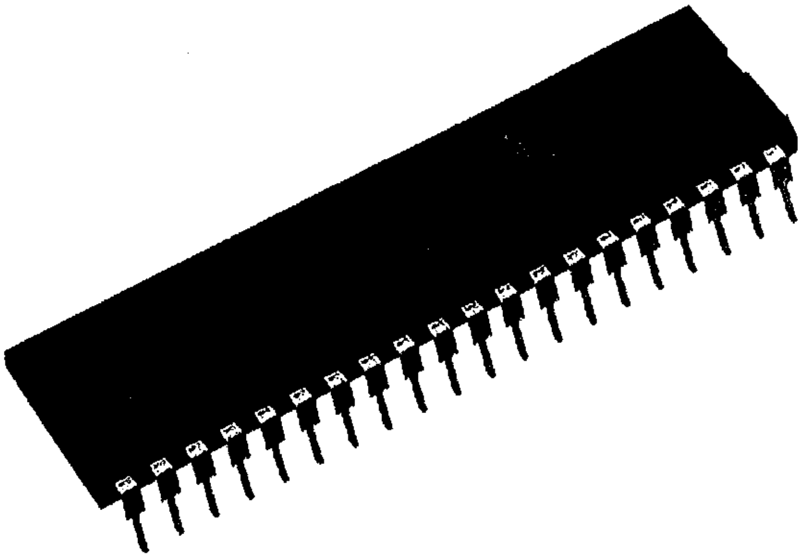

Figure 11: ATmega 16 Microcontroller

The ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous

Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

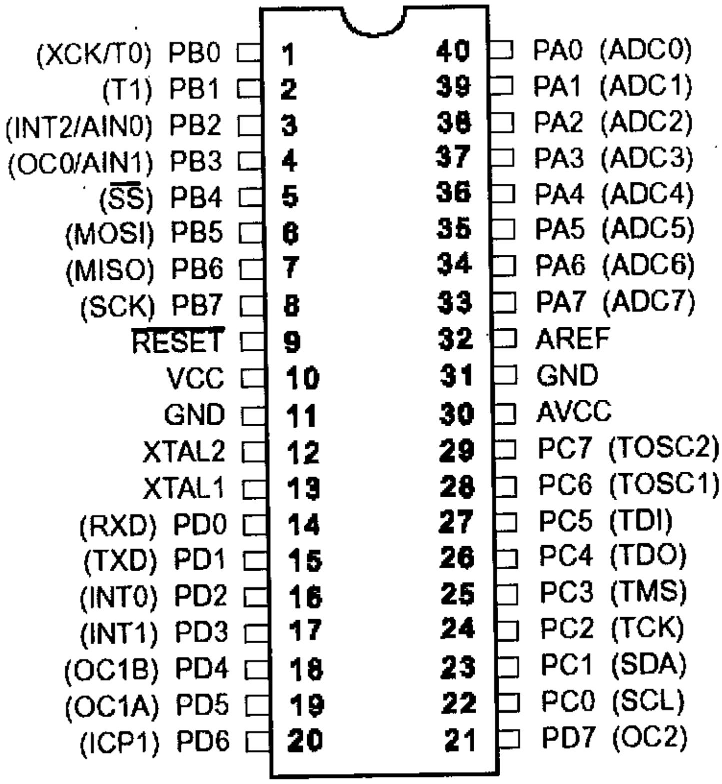

Figure 12: Pin-out of the ATmega 16 Microcontroller

2.2.1 Pin Descriptions of the ATmega 16 Microcontroller

Table 1

Pin No.	Pin name	Description	Alternate Function
1	(XCK/T0) PB0	I/O PORTB, Pin 0	T0: Timer0 External Counter Input. XCK : USART External Clock I/O
2	(T1) PB1	I/O PORTB, Pin 1	T1:Timer1 External Counter Input
3	(INT2/AIN0) PB2	I/O PORTB, Pin 2	AIN0: Analog Comparator Positive I/P INT2: External Interrupt 2 Input
4	(OC0/AIN1) PB3	I/O PORTB, Pin 3	AIN1: Analog Comparator Negative I/P OC0 : Timer0 Output Compare Match
5	(SS) PB4	I/O PORTB, Pin 4	In System Programmer (ISP) Serial Peripheral Interface (SPI)
6	(MOSI) PB5	I/O PORTB, Pin 5	
7	(MISO) PB6	I/O PORTB, Pin 6	
8	(SCK) PB7	I/O PORTB, Pin 7	
9	RESET	Reset Pin, Active Low Reset	
10	Vcc	Vcc = +5V	
11	.GND	GROUND	
12	XTAL2	Output to Inverting Oscillator Amplifier	
13	XTAL1	Input to Inverting Oscillator Amplifier	
14	(RXD) PD0	I/O PORTD, Pin 0	USART Serial Communication Interface
15	(TXD) PD1	I/O PORTD, Pin 1	
16	(INT0) PD2	I/O PORTD, Pin 2	External Interrupt INT0
17	(INT1) PD3	I/O PORTD, Pin 3	External Interrupt INT1
18	(OC1B) PD4	I/O PORTD, Pin 4	PWM Channel Outputs
19	(OC1A) PD5	I/O PORTD, Pin 5	

20	(ICP) PD6	I/O PORTD, Pin 6	Timer/Counter1 Input Capture Pin
21	PD7 (OC2)	I/O PORTD, Pin 7	Timer/Counter2 Output Compare Match Output
22	PC0 (SCL)	I/O PORTC, Pin 0	TWI Interface
23	PC1 (SDA)	I/O PORTC, Pin 1	
24	PC2 (TCK)	I/O PORTC, Pin 2	JTAG Interface
25	PC3 (TMS)	I/O PORTC, Pin 3	
26	PC4 (TDO)	I/O PORTC, Pin 4	
27	PC5 (TDI)	I/O PORTC, Pin 5	
28	PC6 (TOSC1)	I/O PORTC, Pin 6	Timer Oscillator Pin 1
29	PC7 (TOSC2)	I/O PORTC, Pin 7	Timer Oscillator Pin 2
30	AVcc	Voltage Supply = Vcc for ADC	
31	GND	GROUND	
32	AREF	Analog Reference Pin for ADC	
33	PA7 (ADC7)	I/O PORTA, Pin 7	ADC Channel 7
34	PA6 (ADC6)	I/O PORTA, Pin 6	ADC Channel 6
35	PA5 (ADC5)	I/O PORTA, Pin 5	ADC Channel 5
36	PA4 (ADC4)	I/O PORTA, Pin 4	ADC Channel 4
37	PA3 (ADC3)	I/O PORTA, Pin 3	ADC Channel 3
38	PA2 (ADC2)	I/O PORTA, Pin 2	ADC Channel 2
39	PA1 (ADC1)	I/O PORTA, Pin 1	ADC Channel 1
40	PA0 (ADC0)	I/O PORTA, Pin 0	ADC Channel 0

Chapter 3

BASCOM Programming Codes

3.1 BASCOM Syntax

$regfile = "m16def.dat"

(It refers to the name of register file. The register files are stored in the BASCOM-AVR application directory with .DAT extension. The register file holds information about the chip such as the internal registers and interrupts addresses. Since we are using Atmega16 Microcontroller, we will define $regfile= "m16def.dat")

$crystal = 1000000

(1000000 is 1 MHz frequency that a user can set freely for the microcontroller. It defines the clock speed at which you want to run your microcontroller.)

Config LCD = 16 * 2

(It is the type of LCD you want to configure. It can be: 40 * 4, 16 * 1, 16 * 2, 16 * 4, 16 * 4, 20 *2 or 20 * 4 or 16 * 1a or 20*4A. 16*2 is the LCD type which means this LCD prints 16 characters per two lines.)

Config Lcdpin = Pin, Db4 = PortB.4, Db5 = PortB.5, Db6 = PortB.6, Db7 = PortB.7, E = PortB.3, Rs = PortB.2

(This line gives a description about the LCD pin connections with the Microcontroller Ports.)

Config ADC = Single, Prescaler = Auto, Reference = AVcc

(ADC – It defines the Running mode. Its value is SINGLE.

PRESCALER - A numeric constant for the clock divider. Use AUTO to let the compiler generate the best value depending on the XTAL.

REFERENCE - Some chips like the M163 have additional reference options. Its value may be OFF , AVCC or INTERNAL.

Single means instructing the ADC to fetch the value only when its asked to. "Auto" means that the ADC can automatically set its frequency in regard with the microcontroller frequency in the program.
Reference is set as AVcc because Aref voltage is referred from the voltage supply to the ADC's i.e. AVcc.)

Start ADC
(StartAdc id=s the command given to initialize the ADCs.)

Config Timer1 = Pwm, Pwm = 8 , Prescale = 1 , Compare A Pwm = Clear Down , Compare B
Pwm = Clear Down
(We use PWM to control the motor speed which is of 8 bit that is why we set PWM= 8, PWM works on the same frequency as the Microcontroller, and the Channels A & B are set as cleardown to vary the speeds from 0 to 255 in increasing order.)

StartTimer1

(StartTimer1 is used to start the PWM channels.)

• DEFINING VARIABLES

SYNTAX:

DIM (var) as type Var- Name of Variable

Type - Bit, Byte, Word, Integer, Long, Single, Double or String

Example

Dim A as Integer

Dim B as String * 8

First statement is defining A variable as integer and second one is defining B variable as String of 8 characters long. Other than Integer and String there are many data types available in BASCOM.

• START & CLEAR COMMANDS

Start Command: This command is use to start the specified device.

Syntax

START device

Device - TIMER0, TIMER1, COUNTER0 or COUNTER1, WATCHDOG, AC (Analog comparator power) or ADC (A/D converter power)

Example –

Start ADC

CLS Command: Clear the LCD display and set the cursor to home.

Syntax/ Example – Cls

• LOOPS

If – Else statement, Loops and Select – case statement

BASCOM allows using all types of loops in the program like do, while and for. Concept of using these loops is same as using them in other languages like C. Given below are syntaxes of all loops you can use in BASCOM –

1. Do Loop

Do

<statements>

Loop

2. If – else statement

If (condition) then

<statements>

Else

<statements>

Endif

• GETADC COMMAND

This command is used to take input from the analog sensor connected to the development board. This command retrieves the analog value from channel 0-7 of port A. The range of analog value is from 0 to 1023.

Syntax

var = GETADC (channel [, offset])

Var- The variable in which the value will be stored.

Channel – It is the pin no of port A to which analog sensor is connected.

Offset – It is an optional numeric variable that specifies gain or mode.

Example

L = Getadc (2)

Here, in above example, the analog value of the input provided by the sensor connected to pin 2 of port A is stored in variable L.

• LCD COMMAND

It is used to display a constant or variable on LCD screen.

SYNTAX:

LCD x

X - Variable or constant to be displayed on LCD

For displaying string / text, Use LCD "text"

For displaying variable, Use LCD A (A refers to the variable)

For displaying text/variable in next line, we use command LOWERLINE

Example

LCD A; "hello"

Lowerline

LCD "Gaming Embedded Television"

Output on LCD will be:

"Value of Variable A", Hello

Gaming Embedded Television

• WAITMS & PWMXX COMMAND

Waitms command: Suspends program execution for a given time in mS.

Syntax

WAITMS mS

Ms- The number of milliseconds to wait. (1-65535)

Example: Waitms 200

PWMXX command: It is used to set the speed of motor

Syntax

PwmXX = value

XX- it is the channel of a motor

Value – any integer value ranging from 0 to maximum speed .

Example

Pwm1a = 180

• PORTX.Y COMMAND

PORTX.y command: it is used to set the direction of the motor

Syntax

PORTX.y = value

X.y - 'X' as port number and 'y' as pin number

Value - 0 for clock rotation and 1 for anti clock rotation

Example:

PortD.3 = 1

3.2 BASCOM Codes Used

This section show the codes used to program the microcontrollers on the transmitter and receiver circuit boards.

3.2.1 Transmitter Code

```
$regfile = "m16def.dat"
$crystal = 1000000
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6 , Db7 = Portb.7 ,
E = Portb.3 , Rs = Portb.2
Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc
Config Portc.1 = Output
Config Portc.2 = Output
Config Portc.3 = Output
Config Portc.4 = Output


Dim A As Integer , B As Integer , C As Bit
Do
```

```
Cls
C = Pina.0
If C = 1 Then
Portc.1 = 0
Portc.2 = 1
Portc.3 = 1
Portc.4 = 1

Elseif C = 0 Then
A = Getadc(1)
If A < 320 Then
Lcd "left"
Portc.1 = 1
Portc.2 = 0
Portc.3 = 1
Portc.4 = 1
Elseif A > 350 Then
Lcd "right"
Portc.1 = 1
Portc.2 = 1
Portc.3 = 0
Portc.4 = 1
Else
Lcd "default"
Portc.1 = 1
Portc.2 = 1
Portc.3 = 1
Portc.4 = 0
End If
End If
Loop
End
```

3.2.2 Receiver Code

```
$regfile = "m16def.dat"
$crystal = 1000000
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Portb.4 , Db5 = Portb.5 , Db6 = Portb.6 , Db7 = Portb.7 ,
E = Portb.3 , Rs = Portb.2
Dim A As Bit , B As Bit , C As Bit , D As Bit , E As Integer


Do
Cls
A = Pinc.1
B = Pinc.2
C = Pinc.3
D = Pinc.4


If A = 0 And B = 1 And C = 1 And D = 1 Then
Lcd "sony"
Waitms 300
Elseif B = 0 And A = 1 And C = 1 And D = 1 Then
Lcd "hello"
For E = 1 To 4
Shiftlcd Right
Wait 1
Next
Elseif C = 0 And A = 1 And B = 1 And D = 1 Then
Lcd "     hello"
For E = 1 To 4
Shiftlcd Left
Wait 1
Next
```

```
Elseif D = 0 And A = 1 And B = 1 And C = 1 Then
Lcd "default"
End If
Loop
End
```

CHAPTER 4

RESULTS

This section describes the output of the implemented system. Several testing has been performed to ensure its execution and produce the intended result. The prototype system is designed to show a gaming console interfaced television. Using ATmega 16 microcontrollers on the transmission and receiver boards we have tried to show that upon transmission of signal the circuit switches from television mode to gaming mode.

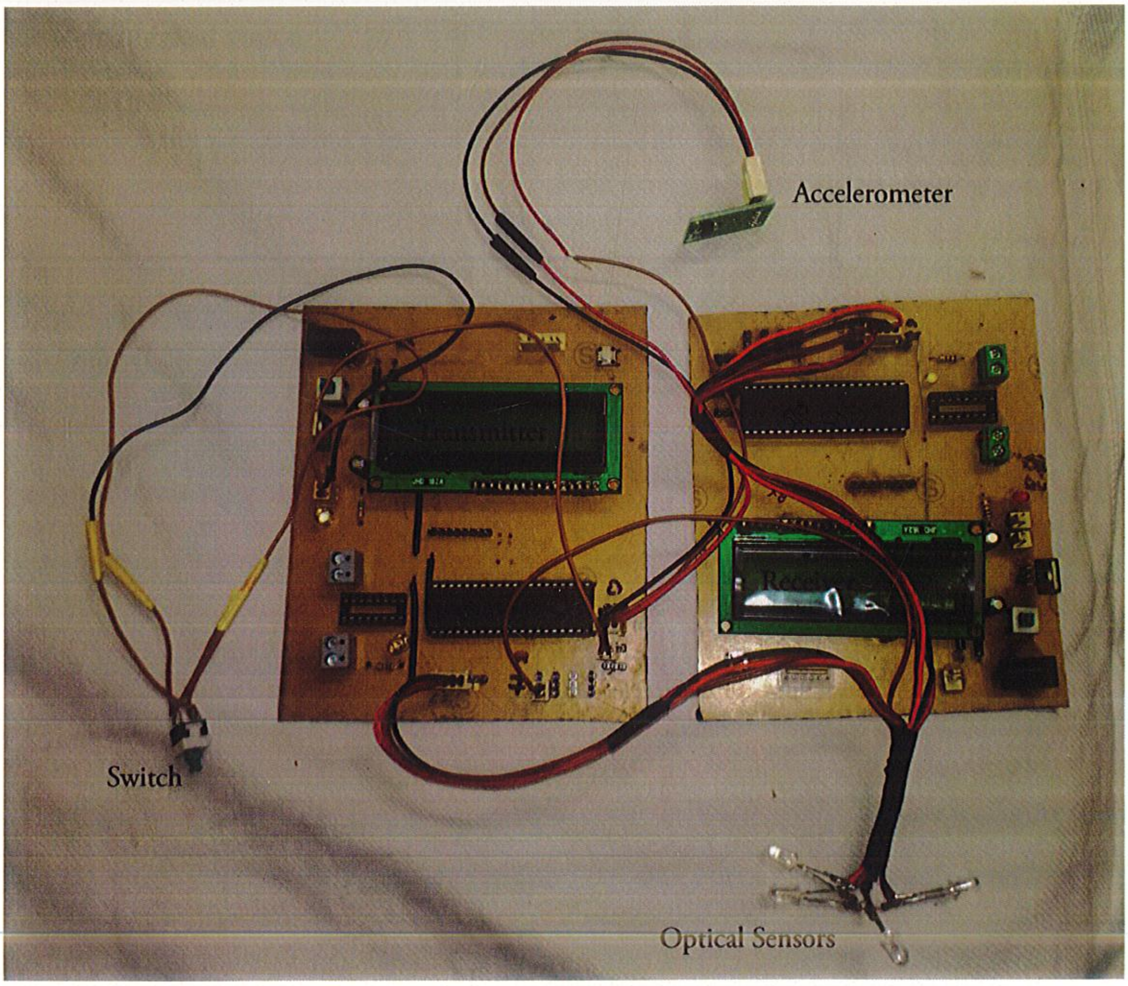

Figure 13: The Prototype

The Prototype operates in the following way:

Originally, the LCD displays a message "Hello". After that, using the switch, the 'Gaming Mode' in the circuit is enabled. When this mode is switched on, the Accelerometer can be used to play the game on the LCD screen. The Optical Sensors switch ON and OFF based on the mode and direction of the motion sensor.

4.1 Future Scope

As has already been mentioned, such a standalone model has not been implemented in the commercial sector. If this prototype is turned into a commercially viable product, then a gaming console with an interfaced television screen can be very successful in the technological sector.

References

This section highlights the references and links used in the project.

http://www.atmel.com/Images/doc2466.pdf

http://xa.yimg.com/kq/groups/19936382/1268050060/name/BoardManual_Atmega16.pdf

http://www.fairchildsemi.com/ds/LM/LM7805.pdf

https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSA-FBS-3.3v.pdf

http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf

http://freedatasheets.com/datasheet-download/444885a22e13792c7a3fea9a983b13ea/ADXL335

http://store.sony.com/wcsstore/SonyStyleStorefrontAssetStore/pdf/warranty/SEL-asset-346771.pdf