A PROJECT REPORT

On

Accelerometer Controlled Advanced Maze Game Box

Submitted in partial fulfillment of the
Requirement for the Award of the Degree
Of

BACHELOR OF TECHNOLOGY

In

ELECTRICAL AND ELECTRONICS ENGINEERING

Submitted By
Jitendra Sharma
Meghna Goyal
Arun Panwar

Submitted To
(Dr.) Akhilendra Yadav
Electrical Deptt.

DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING ROORKEE
ROORKEE – 247667

May, 2014

Candidate's Deceleration

We hereby, certify that the work which is being presented in the report entitled, **"ACCELEROMETER CONTROLLED ADVANCED MAZE GAME BOX"** , in the partial fulfillment of the requirements for the award of the degree of **B-Tech** in **ELECTRICAL AND ELECTRONICS ENGINEERING**, submitted in the Department of Electrical Engineering, College of Engineering Roorkee, affiliated to Uttarakhand Technical University, Dehradun (India), is an authentic record of our own work carried out during session 2013-14, under the supervision of **(Dr.) Akhilendra Yadav,** Department of Electrical Engineering of College of Engineering Roorkee, Roorkee (India).

The matter embodied in this project report has not been submitted by us for the award of any other degree or diploma.

Dated: May 20, 2014

Meghna Goyal	Arun Panwar	Jitendra Sharma
(10060108011)	(10060108063)	(10060108101)

Certificate

This is to certify that the project work **"ACCELEROMETER CONTROLLED ADVANCED MAZE GAME BOX"** is a bonafide record of work done by **Arun Panwar, Meghna Goyal, Jitendra Sharma** under our guidance in partial fulfillment of the requirement for the B-Tech project as per the record of **Uttarakhand Technical University, Dehradun** in academic session of 2013-14 at **College Of Engineering Roorkee, Uttarakhand , India -247667.**

Mrs. Anuradha
(Project Coordinator)
Electrical Department

(Dr.) Akhilendra Yadav
(Project Guide)
(Head of the Department)
Electrical Department

Acknowledgement

We would like to express our profound and sincere gratitude to our project guides, **(Dr.) Akhilendra Yadav, Head of Electrical Department,** College of Engineering Roorkee, for their constant and valuable guidance, inspiration, forbearance and keen interest during this project work.

We take this opportunity to thank **Mrs. Anuradha,** Assistant Professor, Department of Electrical Engineering for providing the departmental facilities for carrying out this work.

Finally, we would like to acknowledge the immeasurable contribution of our **parents**, without their constant inspiration and comradeship this work would never have been over. After completion of this project we can confidently say that this experience has not only enriched us with technical knowledge but also enhanced the maturity of thoughts and vision.

Name of students

Arun Panwar

Meghna Goyal

Jitendra Sharma

Dated: May 20, 2014

TABLE OF CONTENT

List of Figures

ABSTRACT

Accelerometer Controlled Advanced Maze Game Box basically is the collection of different types of mazes games, which includes simple, complex and dynamic mazes, a ball which is controlled by the tilt of the accelerometer and the time constraint game play.

The main objective is to design a simplest and efficient approach in which the player has to move ball through the maze to reach another end by using traditional way of tilting to control the ball based on the classic find-your-way-in-maze puzzle and Designing of Dynamic mazes.

This game box is implemented using 128x64 Graphic LCD as a display unit, ATMega 32 as a controller and an Accelerometer as a sensor. A maze, a small shiny ball and we used to apply our elemental understanding of gravity to bring the ball out of the maze by integrating this most intriguing 'motion sensing' phenomenon in the electronic framework.

Keywords: Graphic LCD, ATMega 32 Microcontroller, Algorithm, Maze game

CHAPTER-1

INTRODUCTION

A maze is a tour puzzle in the form of a complex branching passage through which the solver must find a route. The pathways and walls in a maze are fixed, and puzzles in which the walls and paths can change during the game are categorised as tour puzzle. The Labyrinth is the oldest known maze. Technically the maze is distinguished from the labyrinth, which has a single through-route with twists and turns but without branches, and is not designed to be as difficult to navigate. In everyday speech, both maze and labyrinth denote a complex and confusing series of pathways.

Ball-in-a-maze puzzles are dexterity Puzzle which involve manipulating either a maze or Labyrinth) or one or several balls so that the ball or balls are manoeuvred towards a goal. Toys like this have been popular since Pigs in Clover was invented by Charles Martin Crandall in the 1880s, with the patent being applied for in 1889. In some versions a wooden labyrinth is tilted using two knobs and the ball has to be navigated past a series of holes and obstacles. A magnet is used in other versions where the balls have to be manipulated rather than the maze.

As the thing should be changed with change in time. This classic Ball-in-a-maze Puzzle is also reinvented. This game contain three different levels i.e, simple, complex and dynamic maze and played by controlling a small shiny ball which works on the principle of gravity and controlled by tilling the game box in different direction(s). So the game is converted into a modern day indulgence! So here we have implemented this game design using accelerometer ICs and integrate this most intriguing 'motion sensing' phenomenon in the electronic framework.

Accelerometer Controlled Advanced Maze Game Box basically is a collection of Different type of maze games, which includes simple, complex and dynamic mazes. The main objective is to design the classic find-your-way-in-the-maze puzzle in which a player has to find the way with the help of ball using the traditional way of tilting to control the ball. The game is implemented on a 128*64 GLCD screen (on which the game is displayed), using microcontroller Atmega32 (the controlled of

game i.e, brain of the game box) and have to design maze layout. It is required to implement the maze along with a ball display. The ball's pixels needs to be controlled by the accelerometer using which the ball needs to be taken out of the maze. It includes some features which are as follows:-

1. Scoreboard according to the time taken and number of level completed to get the ball out of the maze.
2. Interactive Welcome Screen at the beginning.

3. Multiple levels - A player will move to the next level on clearing the current level. The level may differ in maze design, number of balls, no. of dynamic elements and Hurdles etc.

4. Counter in the right corner of the GLCD. As the player starts the game the value of timer goes on decreasing. When it reaches to zero the game gets over.

5. Level Indicator in the left corner of the GLCD. It shows the level which is being played.

6. Menu for Selecting Different Channels.

7. Switches to control the game play which includes mainly 5 switches which are as follows:

i. Restart feature enables the player to restart the game.
ii. Pause switch pauses the game. As we press this switch the time will stop changing.
iii. Resume enables the player to resume the game from where it has been paused and the counter will start flowing again from its previous value.
iv. Reset resumes the particular level.
v. High Score of game has been recorded which can be viewed by pressing the high score switch. It also stores the top three scores of the games played.

8. Battery to make setup Portable.

CHAPTER-2

COMPONENTS AND SOFTWARES

2.1 HARDWARE USED

Three hardware's are used in the project and a handmade PCB board on which these Graphic LCD and microcontroller has been mounted. The name of the hardware devices followed by their brief description is given below:

1. Graphic LCD
2. Accelerometer
3. Atmega32

2.1.1 Graphic LCD

Introduction: 128x64 segment graphic lcd with green coloured backlight is used in the project. This graphic lcd is having inbuilt KS0108 controller and a product of Samsung electronics. It works on 5 Volt Supply.It has 20 Pin Interface in which there are 8 data pins and LED+, LED-, Vss, Vdd, v0,D/I, R/W, E, CS1, CS2, RST, Vee pins are available where each pin is having some special features. Like LED+, LED- are used for providing supply for backlight of lcd, CS1, CS2 are the chip select pin as lcd is divided into two segments so to enable it these two pins are used. Vss, is for giving ground to the lcd. Vdd is for giving the supply. It is also having 8 data pins. The figure given below shows the outer look of the graphic lcd which is being used in the project.

Fig 1: Graphic LCD (128*64)

Block Diagram: The block diagram of the graphic lcd is given below. The diagram shows that the lcd is divided into two segments each of one is having 64 segments.CS1 is the chip select pin far enabling the first part of the segment and CS2 is used for enabling the second part. DB0-DB7 are the 8 data pins which are connected to both of the segments. There are four pins which are used for regulating the power circuit of the graphic lcd these are named as Vdd is for giving supply to the lcd, Vss is for providing ground,Vo is set for adjusting the screen contrast, Vout is the output voltage for lcd driving. The supply for backlight is provided by the pins named LEDA, LEDK.

Features:

Display Format: 128 X 64 Dots

LCD Mode: STN – Yellow Green – Positive – Transmissive

Driving Method: 1/64 Duty, 1/9 Bias

Backlight: LED, yellow green color

Outline Dimension: 93.0 (W) x 70.0 (H) x 13.5 (T) mm

Viewing Area: 70.0 (W) x 39.0 (H) mm

Dot Size: 0.48 x 0.48 mm

Weight: 75 g

Controller: S6B0108 (KS0108B)

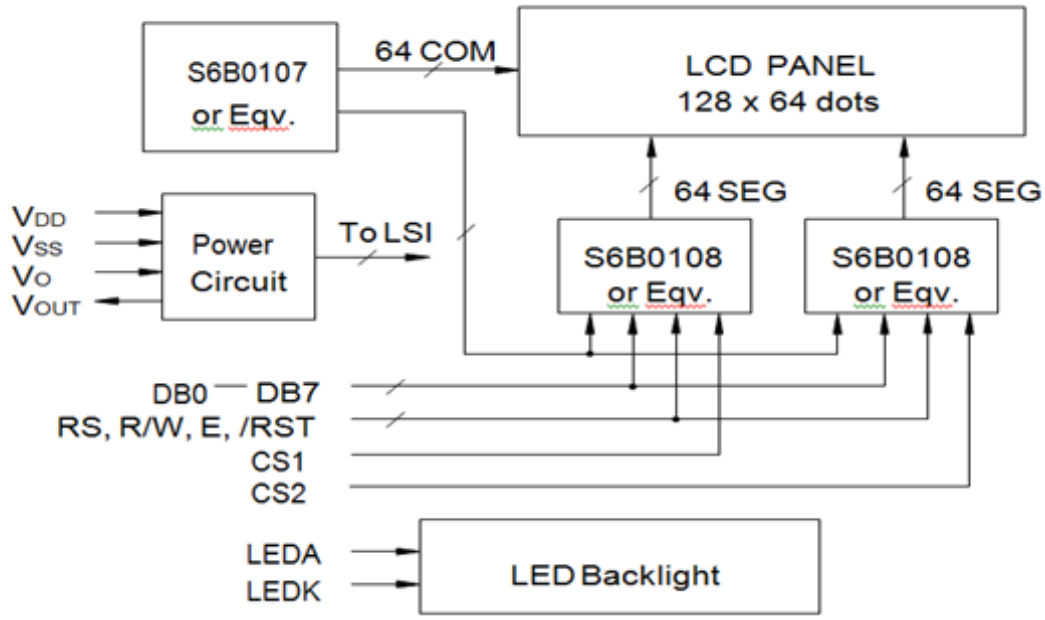


Fig 2: Block Diagram Graphic LCD

Power Supply: 5V voltage for the LED backlight should be supplied to Pin19 (LEDA) and Pin20 (LEDK) terminal of the interface, it should not be supplied to the Anode/Cathode terminal of the LED backlight directly. In the pin connection diagram we can see that the positive terminal of +5V is connected to LEDA and its negative terminal to LEDK. The supply of +5V is given to vdd pin. This is how the whole power supply box of the lcd is controlled.

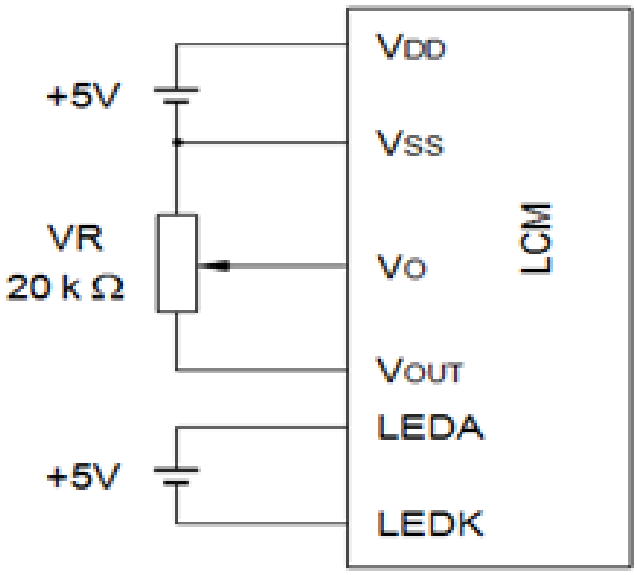


Fig 3: Pin Connection for Power Supply

5V voltage for LED backlight should be supplied to pin 19(LEDA) and Pin 20(LED K) terminal to interface, it should not be supplied to anode cathode terminal of LED backlight directly.

Pin Description: The table given below shows the proper description of every pin of the graphic lcd. The table mainly consist of four columns namely pin no, symbol, level, function. Pin no. Column is having the number of pins, symbol column consists of the name pins correspondingly, moreover level keeps status of pins to which is required to make them enabled, function column describes the purpose or functioning of the pins. 20 Pin Interface in which there are 8 data pins and LED+, LED-, Vss, Vdd, v0,D/I, R/W, E, CS1, CS2, RST, Vee pins are available where each pin is having some special features. Like LED+, LED- are used for providing supply for backlight of lcd, CS1, CS2 are the chip select pin as lcd is divided into two

segments so to enable it these two pins are used. There are 20 pins in the lcd n their description is shown in the table in detail which is as follows:

Pin No.	Symbol	Level	Function
1	VSS	0V	Ground
2	VDD	+5V	Power supply for logic
3	VO	-	Operating voltage for LCD (contrast adjusting)
4	RS	H/L	Register selection H:Displaydata L: Instruction code
5	R/W	H/L	Read/Write selection
6	E	H,H→L	Enable signal. Read data when E is "H", write data at the falling edge of E
7	DB0	H/L	Databit0
8	DB1	H/L	Databit1
9	DB2	H/L	Databit2
10	DB3	H/L	Databit3
11	DB4	H/L	Databit4
12	DB5	H/L	Databit5
13	DB6	H/L	Databit6
14	DB7	H/L	Databit7
15	CS1	H	Chip selection for IC1,active"H"
16	CS2	H	Chip selection for IC2,active"H"
17	RST	L	Reset signal, active "L"
18	VOUT	-10V	Output voltage for LCD driving
19	LEDA	+5V	Power supply for LED backlight
20	LEDK	0V	Power supply for LED backlight

Display Control Instruction: The display control instructions control the internal state of the KS0108B. Instruction is received from MPU to KS0108B for the display control. There are mainly 7 instructions which come into scene while working with the lcd, for each instruction the status of each pin is being shown in the table. If the pin has to be high then it is set by the value 1 and for low status the value of pin is 0. The table also gives the description of functioning of each instruction. The following table shows various instructions:

Instruction	RS	R/W	DB	DB	DB	DB	DB	DB	DB1	DB0	Function
Display ON/OFF	0	0	0	0	1	1	1	1	1	0/1	Controls the display on or off. Internal status and display RAM data
Set Address(Y	0	0	0	1	Y address(0-63)						Sets the Y address at the Y address counter.
Se t Page(X address)	0	0	1	0	1	1	1	Page(0-7)			Sets the X address at the X address register.
Display Start Line (Z address)	0	0	1	1	Display start line(0-63)						Indicates the display data RAM displayed at the top of the screen.
Status Read	0	1	BUSY	0	ON/OFF	RESET	0	0	0	0	Read status. BUSY 0:Ready 1:Ininternaloperation ON/OFF 0:DisplayON 1:DisplayOFF RESET 0:Normal
Write Display Data	1	0	Display Data								Writes data(DB0~DB7)into display data RAM. After writing
Read Display Data	1	1	Display Data								Reads data(DB0~DB7)from display data RAM to

Display ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1	1	1	1	D

The display that appears when D is 1 and disappears when D is 0. Though the data is not on the screen with D=0, it remains in the display data RAM. Therefore, you can make it appear by changing D=0 into D=1.

Set Address(Y Address)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Y address (AC0~AC5) of the display data RAM is set in the Y address counter. An address is set by instruction and increased by 1automatically by read or write operations of display data.

Set Page(X Address)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	0	1	1	1	AC2	AC1	AC0

X address (AC0-AC2) of the display data RAM is set in the X address register.

Writing or reading to perform MPU is executed in this specified page until the next page is set. X address (AC0-AC2) of the display data RAM is set in the X address register. Writing or reading to perform MPU is executed in this specified page until the next page is set.

Display Start Line (Z Address)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	1	AC5	AC4	AC3	AC2	AC1	AC0

Z address (AC0~AC5) of the display data RAM is set in the display start line register and displayed at the top of the screen.

Status Read

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BUSY	0	ON/OFF	RESET	0	0	0	0

There are three operations whose values are set in order to get the desired function done. These are BUSY, ON/OFF,RESET. The description of these is given as.When BUSY is 1, the chip is executing internal operation and no instructions are accepted. When BUSY is 0, the chip is ready to accept any instructions. When ON/OFF is 1, the display is off. When ON/OFF is 0, the display is on. When RESET is 1, the system is being initialized. In this condition, no instructions except status read can be accepted. When RESET is 0, initializing has finished and the system is in the normal operation condition.

BUSY: When BUSY is 1, the chip is executing internal operation and no instructions are accepted. When BUSY is 0, the chip is ready to accept any instructions.

ON/OFF: When ON/OFF is 1, the display is off.

When ON/OFF is 0, the display is on.

RESET: When RESET is 1, the system is being initialized. In this condition, no instructions except status read can be accepted. When RESET is 0, initializing has finished and the system is in the normal operation condition.

Write Display Data

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write data (D0~D7) into the display data RAM. After writing instruction, Y address is increased by 1 automatically.

Read Display Data

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

Read data (D0~D7) from the display data RAM. After reading instruction, Y address is increased by 1automatically. One time of dummy read must be required after column address setting.

2.1.2 ACCELEROMETER

Introduction: The ADXL335 is a small, thin, low power (Low power - 350 µA (typical), complete 3-axis accelerometer, 10,000 *g* shock survival with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 *g*. It has Single-supply operation of 1.8V to 3.6V with excellent temperature stability and BW adjustment with a single capacitor per axis. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. It does not need external components and consume low current (500 µA). Accelerometer is highly sensitive for small movements and fast turn on time.

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. This Accelerometer has Five Pin Interface i.evcc, gnd, x-out, y-out, z-out. It operates at 5 Volt. It is purchased from Robokits.co.in and product code is RKI-1028.

Fig 4: Accelerometer

Block Diagram:

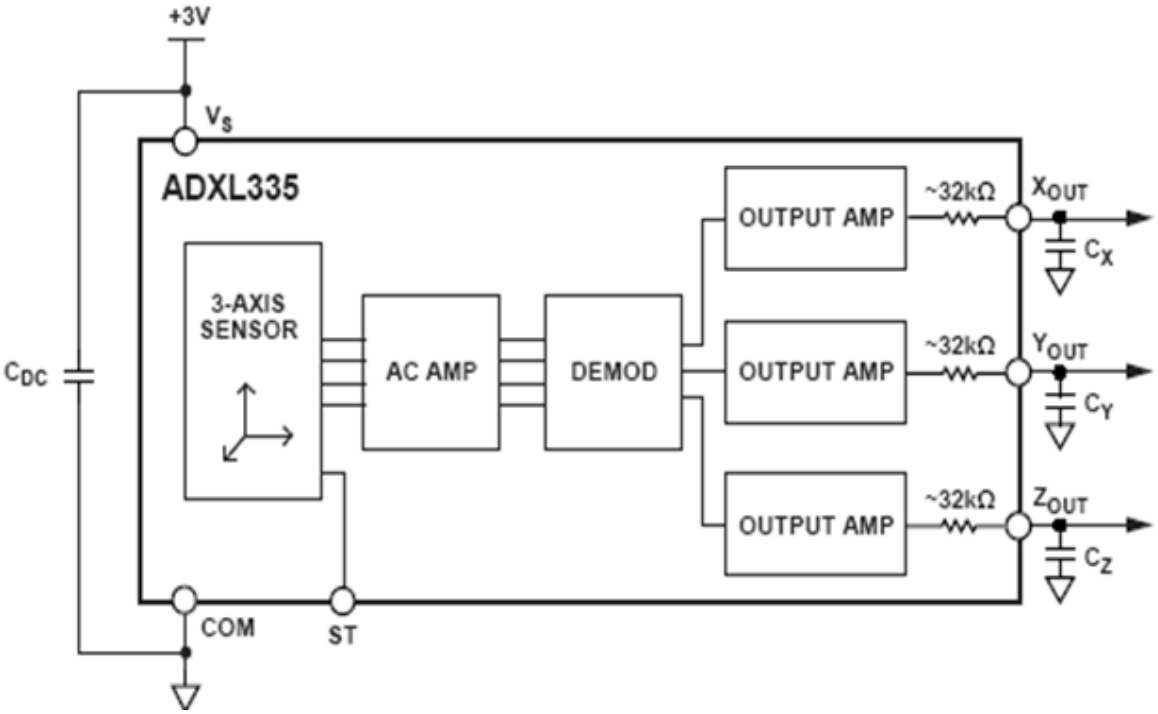


Fig 5: Block Diagram of Accelerometer

Pin Configuration:

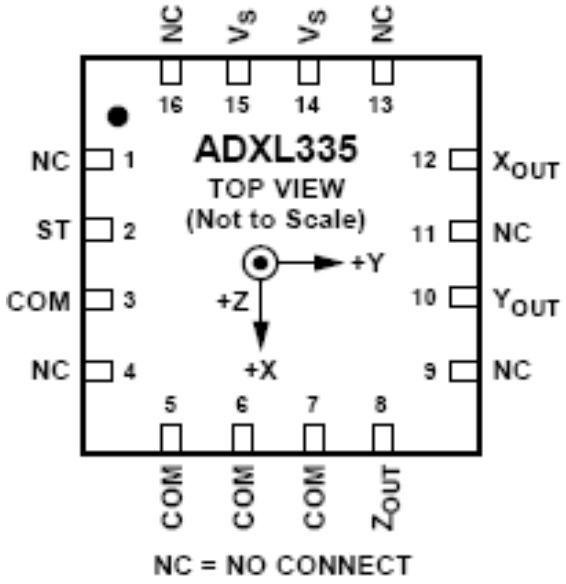


Fig 6: Pin Configuration of Accelerometer

Pin Description

Pin No.	Mnemonic	Description
1	NC	No Connect[1].
2	ST	Self-Test.
3	COM	Common.
4	NC	No Connect[1].
5	COM	Common.
6	COM	Common.
7	COM	Common.
8	Z_{OUT}	Z Channel Output.
9	NC	No Connect[1].
10	Y_{OUT}	Y Channel Output.
11	NC	No Connect[1].
12	X_{OUT}	X Channel Output.
13	NC	No Connect[1].
14	V_S	Supply Voltage (1.8 V to 3.6 V).
15	V_S	Supply Voltage (1.8 V to 3.6 V).
16	NC	No Connect[1].
EP	Exposed Pad	Not internally connected. Solder for mechanical integrity.

[1]NC pins are not internally connected and can be tied to COM pins, unless otherwise noted.

Theory of Operations: The ADXL335 is a complete 3-axis acceleration measurement system. The ADXL335 has a measurement range of ±3 *g* mini-mum. It contains a poly silicon surface-micro machined 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 poly silicon surface-micro machined structure built on top of a silicon wafer. Poly silicon 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 Sensors: 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 m*g* over the −25°C to +70°C temperature range.

2.1.3 ATMEGA 32

Introduction: The Atmel AVR ATmega32 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, theATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed.

The Atmel 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.

The ATmega32 provides the following features: 32Kbytes of In-System Programmable Flash program memory with Read-While-Write capabilities, 1024bytes EEPROM, 2Kbyte 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 ATmega32 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

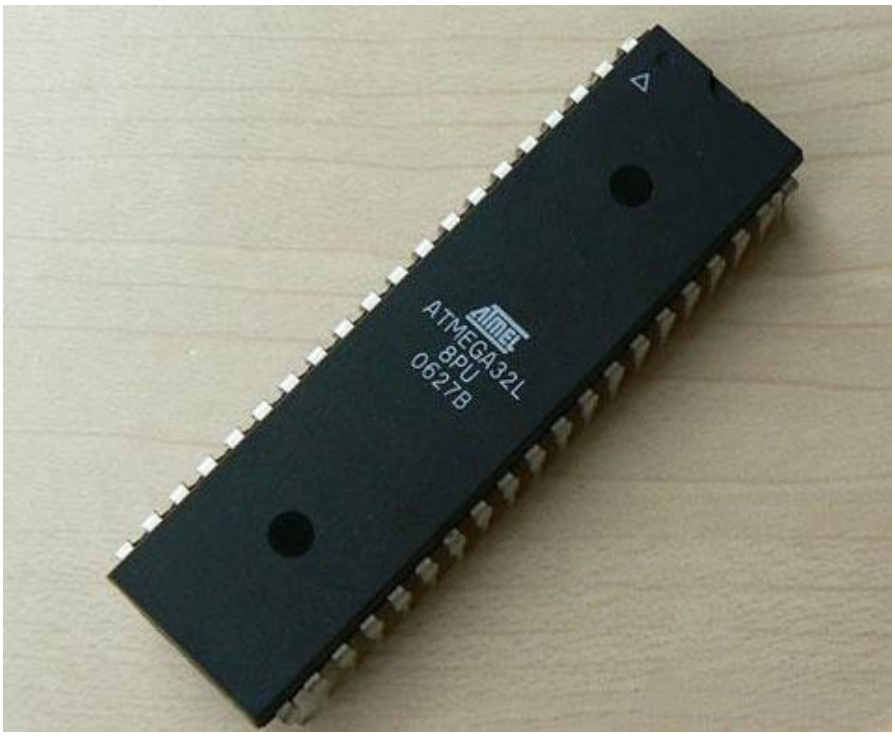


Fig 7: Atmega 32

Pin Configuration and Description: In this MAZE GAME BOX Project:

a) Data Pins (GLCD) are connected at PORT D.
b) Control Pins (GLCD) are connected at PORT C.
c) Accelerometer o/p Pins are connected at PORT A.

Description of all of pins of microcontroller is as follows:

- Vcc **:** Digital supply voltage.
- GND**:** Ground.
- Port A (PA7..PA0): Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when are set condition becomes active, even if the clock is not running.
- Port B (PB7..PB0)**:** Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.
- Port C (PC7..PC0)**:** Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5 (TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs. The TD0 pin is tri-stated unless TAP states that shift out data are entered. Port C also serves the functions of the JTAG interface.
- Port D (PD7..PD0)**:** Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

- RESET**:** Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.
- XTAL1**:** Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
- XTAL2 : Output from the inverting Oscillator amplifier.
- AVCC: AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.
- AREF**:** AREF is the analog reference pin for the A/D Converter.

Pin Diagram:

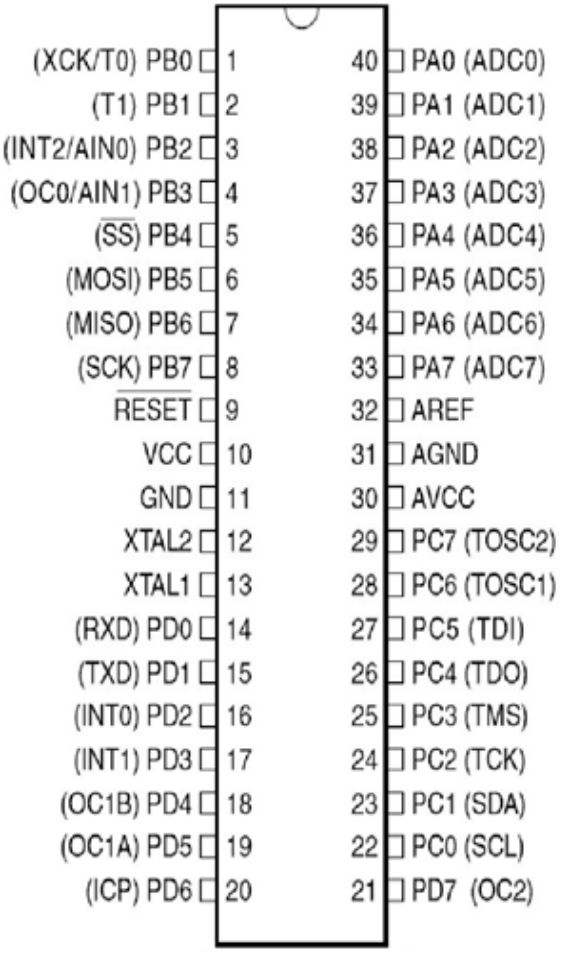


Fig 8: Pin Diagram Atmega 32

2.2 SOFTWARE USED

Basically there are three software's used in the project which are as follows:

1. Atmel AVR Studio 4 has been used for coding of the project.
2. Proteus is used for simulation.
3. Sinaprog is used for programming microcontroller.

2.2.1 AVR Studio 4: AVR Studio was created by Atmel in order to help developers to create applications for AVR microcontrollers using C/C++ programming languages. This piece of software comes with a large number of tutorials, which allow the users to get familiar with the application. The program stands as a complete pack for programmers that use C++ and other programming languages. It provides the users with access to the tools for writing, building and debugging their codes.

The menu of this application is easy-to-use and offers access to powerful tools for both beginners and experienced developers, making it easy for the users to find their way through C/C++ programming. Some of the key features are: "cycle correct" simulator with advanced debugging functionality, rich SDK that enables tight integration of customer plug-ins and compatibility with many Microsoft Visual Studio plug-ins. Also the tool provides a "split window" button that allows the users to work on more than one project at a time.

All in all AVR Studio is a complete tool for programmers which develop, test and debug C/C++ applications; you should give this tool a try as it comes in handy for programming AVR microcontrollers.

Advantages: The main advantages of the AVR Studio are:-

1. It support both beginners as well as advance users
2. It has wide variety of tools

In the whole project we used Atmel® AVR Studio® 4. It is the Integrated Development Environment (IDE) for developing and debugging embedded Atmel AVR® applications, and gives you a seamless and easy-to-use environment to write, build, and debug your C/C++ and assembler code. It makes editing and debugging source code easier by seamlessly bringing together an intelligent editor with assisted code writing, a wizard for quickly creating new projects, the AVR Software Framework source code library, a GNU C/C++ Compiler, a powerful simulator, and the front-end visualize for all of the Atmel AVR programmers and in-circuit debuggers.

A screen short of Atmel AVR Studio 4 is as follows:-

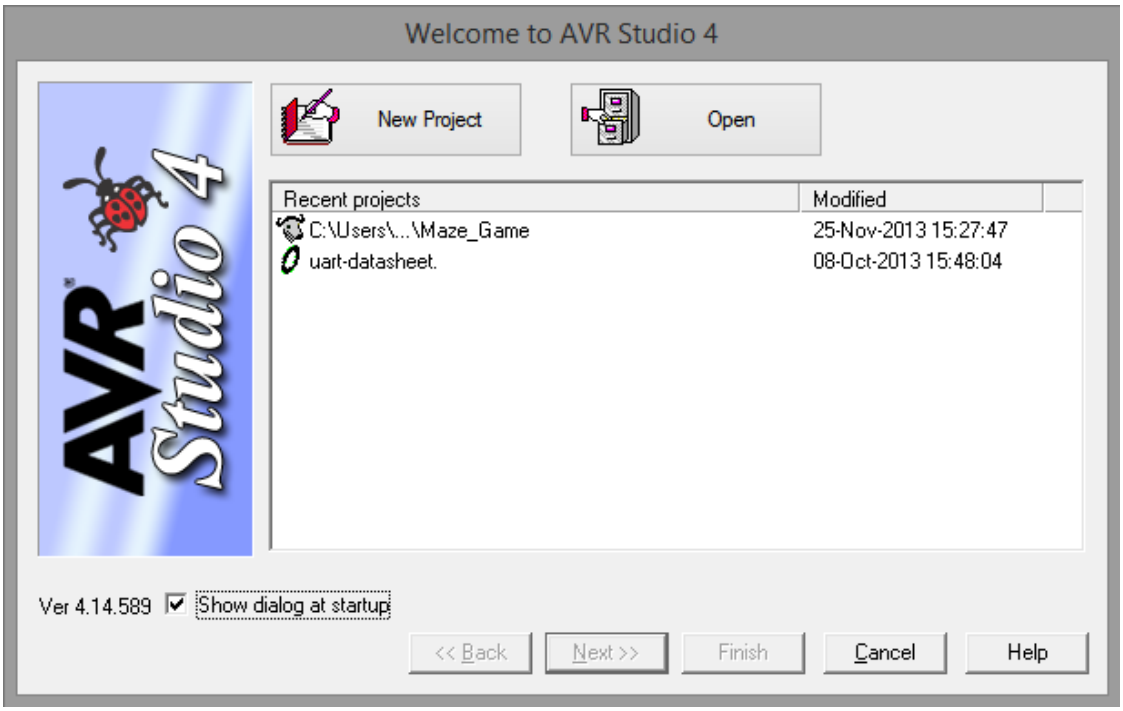


Fig 9: Screen Shot Of AVR Studio

2.2.2 Proteus (ISIS): PROTUES combines advanced schematic capture, mixed mode SPICE simulation, PCB layout and auto routing to make a complete electronic design system. The PROTUES product range also includes our revolutionary VSM(virtual system modelling) technology, which allow you to simulate micro-controller based design, complete with all the surrounding electronic.

Intelligent Schematic Input System: ISIS lies right at the heart of the PROTUES system and is far more than just another schematic package. It has powerful environment to control most aspects of the drawing appearance. Whether your requirement is the rapid entry of complex design for simulation & PCB layout, Or the creation of attractive Schematic for publication ISIS is the right tool for the job Product.

Features:

1. Produces publication quality schematic.
2. Style templates allow customization of supplied library.
3. Mouse driven context sensitive user interface.
4. Automatic wire routing and junction dot placement.
5. Full support for buses including sub- circuit ports and bus pins.

6. Large and growing component library of over 8000 parts.

Complete setup of the graphic LCD with all the connections tested on the simulation software "PROTEUS".

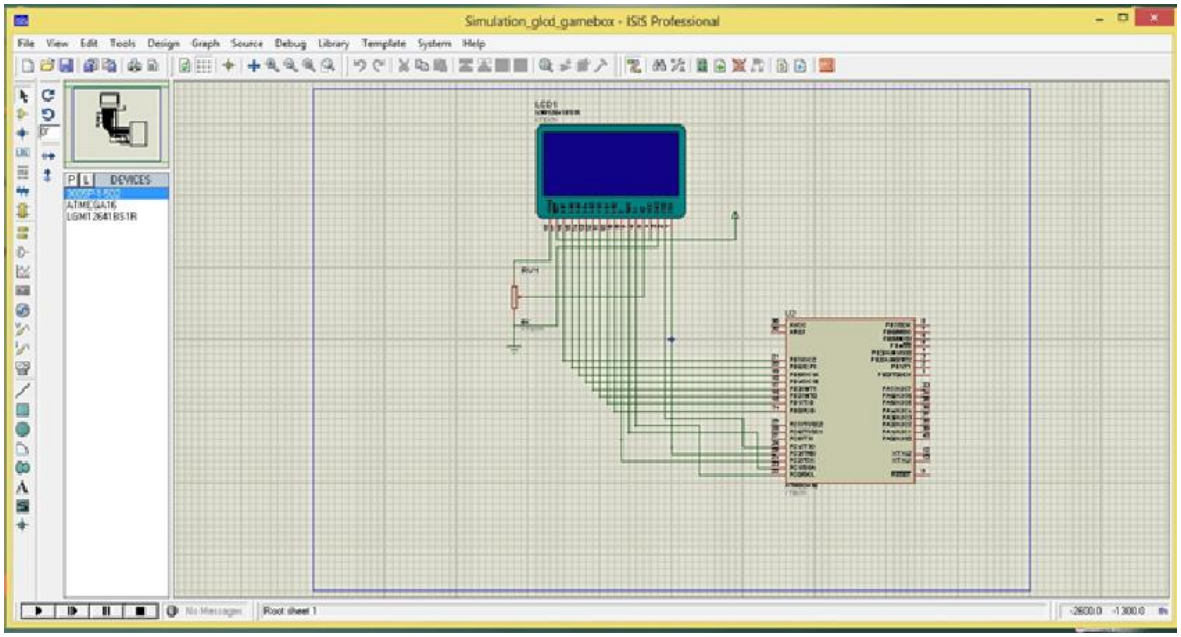


Fig 10: Screen shot of Proteus ISIS

2.2.3 SinaProg: SinaProg is a hex downloader application with AVR Dude and Fuse Bit Calculator. This is used to download code/program and to set fuse bits of a AVR based microcontroller. In this project the USBasp programmer has been used with a frequency setting of 1 MHz of device ATMega 32. JTAG is disabled in fuse bits.

Fig 11: Screen Shot of SinaProg

2.3 ADC (ANALOG TO DIGITAL CONVERTER)

For interfacing of accelerometer with microcontroller ADC has been used.ADC Channel is mounted on PORT A of the ATMega 32, so the Pins of the accelerometer connected to the PORT A.

The ATmega32 features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows 8 single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND). The device also supports 16 differential voltage input combinations.

Features

1. 10-bit Resolution
2. 0.5 LSB Integral Non-linearity
3. ±2 LSB Absolute Accuracy
4. 13 - 260 µs Conversion Time
5. Up to 15 kSPS at Maximum Resolution
6. 8 Multiplexed Single Ended Input Channels
7. 7 Differential Input Channels
8. 2 Differential Input Channels with Optional Gain of 10x and 200x
9. Optional Left adjustment for ADC Result Readout
10. 0 - VCCADC Input Voltage Range
11. Selectable 2.56V ADC Reference Voltage
12. Free Running or Single Conversion Mode
13. ADC Start Conversion by Auto Triggering on Interrupt Sources
14. Interrupt on ADC Conversion Complete
15. Sleep Mode Noise Canceler

Operations: The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB. Optionally, AVCC or an internal 2.56V reference voltage may be connected to the AREF pin by writing to the REFS n bits in the ADMUX Register. The internal voltage reference may thus be decoupled by an external capacitor at the AREF pin to improve Noise immunity. The analog input channel and differential gain are selected by writing to the MUX bits in ADMUX. Any of the ADC input

pins, as well as GND and a fixed band gap voltage reference, can be selected as single ended inputs to the ADC. A selection of ADC input pins can be selected as positive and negative inputs to the differential gain amplifier. If differential channels are selected, the differential gain stage amplifies the voltage difference between the selected input channel pair by the selected gain factor. This amplified value then becomes the analog input to the ADC. If single ended channels are used, the gain amplifier is bypassed altogether.

The ADC is enabled by setting the ADC Enable bit, ADEN in ADCSRA. Voltage reference and input channel selections will not go into effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power saving sleep modes.

The ADC generates a 10-bit result which is presented in the ADC Data Registers, ADCH and ADCL. By default, the result is presented right adjusted, but can optionally be presented left adjusted by setting the ADLAR bit in ADMUX.

If the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH, to ensure that the content of the Data Registers belongs to the same conversion. Once ADCL is read, ADC access to Data Registers is blocked. This means that if ADCL has been read, and a conversion completes before ADCH is read, neither register is updated and the result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL Registers is re-enabled. The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the Data Registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

ADC Input Channel: In Free Running mode, always select the channel before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the first conversion to complete, and then change the channel selection. Since the next conversion has already started automatically, the next result will reflect the previous channel selection. Subsequent conversions will reflect the new channel selection.

ADC Voltage Reference: The reference voltage for the ADC (VREF) indicates the conversion range for the ADC. Single ended channels that exceed VREF will result in codes close to 0x3FF. VREF can be selected as either AVCC, internal 2.56V reference, or external AREF pin.

AVCC is connected to the ADC through a passive switch. The internal 2.56V reference is generated from the internal band gap reference (VBG) through an internal amplifier. In either case, the external AREF pin is directly connected to the ADC, and the reference voltage can be made more immune to noise by connecting a capacitor between the AREF pin and ground. VREF can also be measured at the AREF pin with a high impedant voltmeter.

Note that VREF is a high impedant source, and only a capacitive load should be connected in a system.

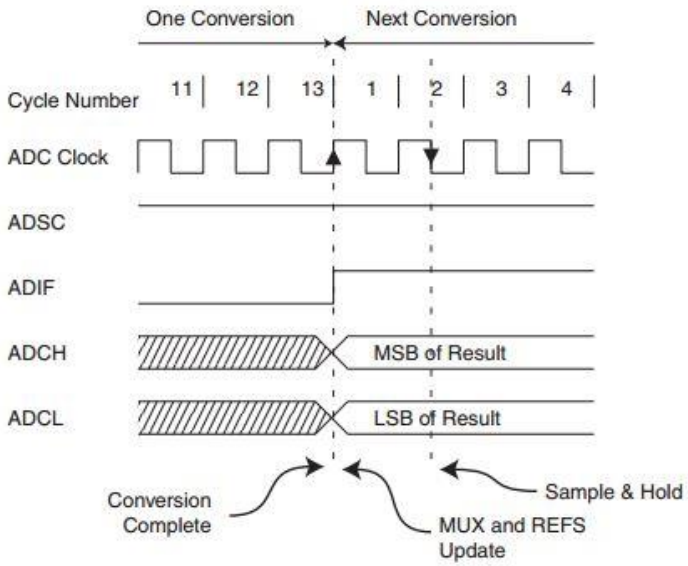
If the user has a fixed voltage source connected to the AREF pin, the user may not use the other reference voltage options in the application, as they will be shorted to the external voltage. If no external voltage is applied to the AREF pin, the user may switch between AVCC and 2.56V as reference selection. The first ADC conversion result after switching reference voltage source may be inaccurate, and the user is advised to discard this result.

Prescaling and Conversion Timing: By default, the successive approximation circuitry requires an input clock frequency between 50 KHz and 200 kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200 kHz to get a higher sample rate.

The ADC module contains a Prescalar, which generates an acceptable ADC clock frequency from any

CPU frequency above 100 kHz. The prescaling is set by the ADPS bits in ADCSRA.

The Prescalar starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The Prescalar keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.



In Free Running mode, a new conversion will be started immediately after the conversion completes, while ADSC remains high.ADC Timing Diagram, Free Running Conversion

ADC Conversion Result: After the conversion is complete (ADIF is high), the conversion result can be found in the ADC Result Registers (ADCL, ADCH).
For single ended conversion, the result is

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

Where VIN is the voltage on the selected input pin and V
REF the selected voltage reference. 0x000 represents ground, and 0x3FF represents the selected reference voltage minus one LSB.

Register Selection: ADC Multiplexer Selection Register – ADMUX

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7:6 – REFS1:0: Reference Selection Bits
These bits select the voltage reference for the ADC, as shown in Table 83. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete
(ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.
Bit 5 – ADLAR: ADC Left Adjust Result
The ADLAR bit affects the presentation of the ADC conversion result in the ADC Data Register.
Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the
ADLAR bit will affect the ADC Data Register immediately, regardless of any ongoing conversion.

Bits 4:0 – MUX4:0: Analog Channel and Gain Selection Bits

The value of these bits selects which combination of analog inputs are connected to the ADC. If these bits

are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

ADC Control and Status Register A – ADCSRA

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – ADEN: ADC Enable

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the

ADC off while a conversion is in progress, will terminate this conversion.

Bit 6 – ADSC: ADC Start Conversion

In Single Conversion mode, write this bit to one to start each conversion. In Free Running Mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC. ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

Bit 5 – ADATE: ADC Auto Trigger Enable

When this bit is written to one, Auto Triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in SFIOR.

Bit 4 – ADIF: ADC Interrupt Flag

This bit is set when an ADC conversion completes and the Data Registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are

set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a Read-Modify Write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

Bit 3 – ADIE: ADC Interrupt Enable

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

Bits 2:0 – ADPS2:0: ADC Prescalar Select Bits

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

The ADC Data Register – ADCL and ADCH

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

ADC Prescalar Selections Table

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers. When ADCL is read, the ADC Data Register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read
ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX, and the MUXn bits in ADMUX affect the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

CHAPTER-3

FLOW CHART

The Flowchart of the algorithm used in the project is shown below in which the processor "AVR" goes through the series of operations and after some time start iterating in a particular pattern till the conditions are true and then move to the next command. The functions to check surroundings of the ball, to move the ball and to check the accelerometer range are playing main role in the iteration field at each level.

Here the complete procedure, how the flow chart helped in writing the code is described below with description of each stage of flow chart…

3.1 Flow Chart 1.0: Flow chart 1.0 is designed to develop a suitable program to check the surroundings of the ball in the game. This surrounding function is implemented differently for all the three levels because of the difference in the patterns and happening of each level.

The main concept behind the every level's surrounding function is same and works according to the given flow chart. So, according to the concept whenever the surroundings function is called it will access the check bit function to check each pixel value but surrounding function is much more than checking a single pixel as the ball shape is in such a way that is covers set of three pixels in each direction. So the surroundings function check each pixel of this set separately using check bit function and produce a combined result for the set of three pixel present in a particular direction.

As function is returning here a single value so the information that a particular direction out of left, right, top and bottom is open to move or not for the ball is must be summed up in this single value, so for this purpose function will return a four digit number and each digit of this number will give the status of a particular direction and hence will cover all four direction with the code 1 and 0. Here if a particular bit is 1 implies the direction is open to move for the ball and if bit is 0 then it implies ball cannot make a move in that particular direction. The fourth digit from right representing the LEFT, third digit from the right representing the RIGHT, second digit representing the BOTTOM and the first digit representing the TOP direction of this four digit number.

Once this value is returned by the respective surrounding function the four digit number is encoded in the same pattern it was decoded in the main function. After encoding the processor pass the instruction of moving ball to the move ball function considering other factors at the same time.

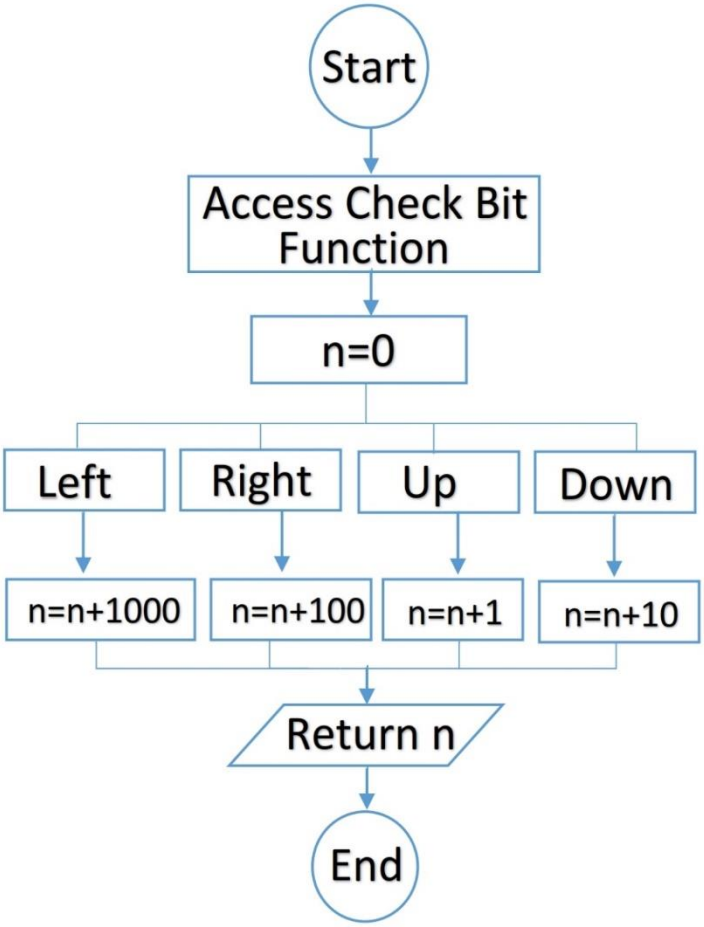


Fig12: Flow Chart 1.0

3.1 Flow Chart 2.0: Flow chart 2.0 is a complete prototype of the procedure, the program will work and this procedure is explained step by step by step below here…

According to the Algorithm the processor first choose a particular font (Aerial_Black_14) after starting for the printing purposes of Graphic LCD. A font .h file already has been attached which is created by the java program FontCreater 2.1.

A graphically designed welcome screen with a welcome note. It is designed to make the game box setup more interactive and attractive for the users.

Then the processor enters in a loop which iterates continuously till either END LEVEL condition or GAME OVER conditions are not true, this looping includes checking the surroundings of the ball by using a special algorithm shown by a Flowchart 1.0. Once the processor got the information about the surroundings either they are open to move or not then instantly it will move for checking the accelerometer range to get the desired moving direction and pass the parameters generated from this accelerometer range to the MOVE BALL function. The MOVE BALL function shifts the ball in particular direction by one pixel of Graphic LCD by encoding the parameters of the surrounding function and accelerometer range.

The iterating loop checks for the conditions that is for the LEVEL END and for COUNTER less than zero, if any of them are stands true then loop terminates and score or game over notifications is being displayed on the screen accordingly.

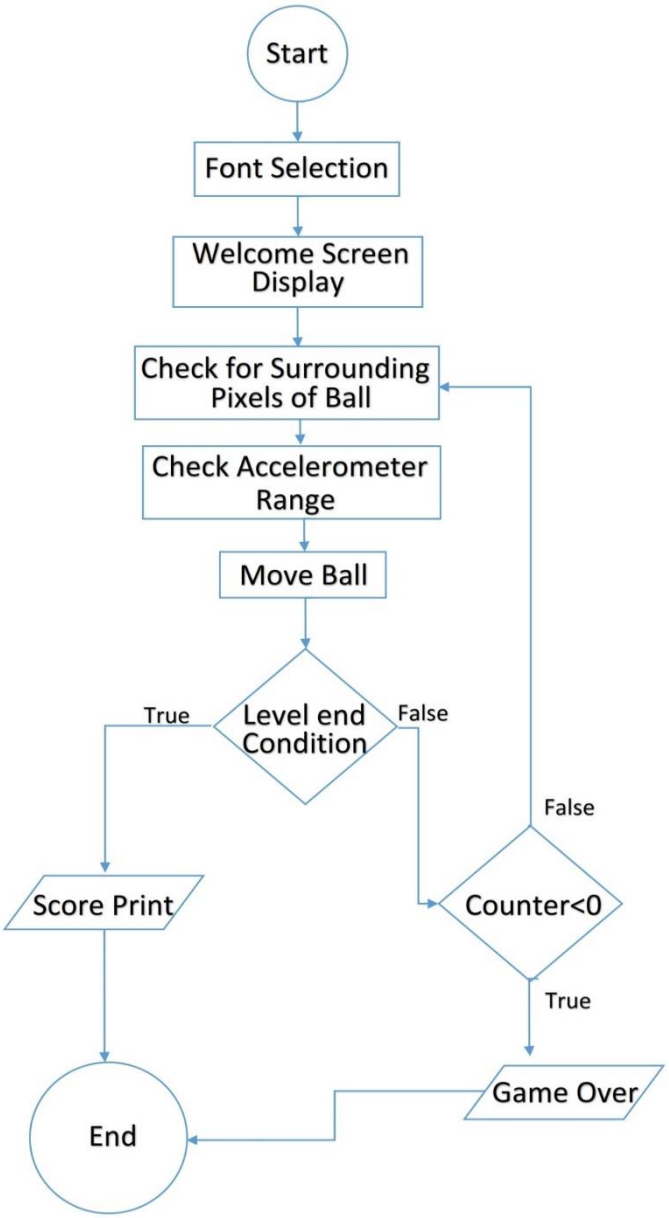


Fig13: Flow Chart 2.0

CHAPTER-4

GAMEPLAY AND FEATURES

4.1 INTRODUCTION:

The game comprises of 3 levels. These levels are designed in the increasing order of difficulty. As first level is the simplest one, second level is quite difficult having features like gate, bucket, moving bars and food, third level is somewhat different having four mazes. All of these levels are having time limits within which the player has to complete the level otherwise the game will get over. The Game Setup powers on with an interactive welcome screen and then level-1 of game starts, the brief description of each level is given below.

4.1.1 LEVEL 1: Level 1 is having a normal maze, without any dynamic hurdles and much complexity which is to be solved. The counter is being displayed in the right corner of the screen and level number in the left corner of the screen. The player has to move the ball from one corner of the maze to the other corner by solving it through the moments generated from tilting the box. This is to be done within the specified time limit. As soon as the counter reaches zero "GAME OVER" will be displayed on the screen and the score upto the previous level is being displayed. When the level is finishes the score adds up to the final score and the game moves to the next level, more difficult, more challenging with more fun.

4.1.2 LEVEL 2: Level 2 is basically consist of a maze with dynamic elements a bucket, a food key and a moving bar (hurdle). In this level the first step is to move the ball towards the bucket, as soon as it reaches inside the bucket, the bucket is closed from the top and start moving upward, as it reached to the top in the column it tilts itself and opened. Now in the second portion there is one food key and moving bar, this moving bar start moving as soon as the bucket opens. The ball has to grab the key to open the gate. But if the ball touches the moving bar it is reset to its original position in the bucket. As soon as the ball takes the food key the gate is opened and ball can pass through it, then the ball is moved to the bottom. This finishes the level 2, score adds up in the final score again and game moves to the next and final level of Game Box.

4.1.3 LEVEL 3: Level 3 is comprises of a special type of maze having 4 blocks interconnected through the bridges. On solving the first maze and grabbing the two food items, the bridge joining that block to the other block appears and the path is opened for the second block. Each maze is consist of 2 food items that are to be eaten by the ball in order to complete a particular block. As soon as the ball eats both food items the path for the second maze is generated and so on. The completion of all the four blocks takes the ball to the final end of the maze and hence finishes the level 3. In the end score is being displayed.

If the score is among the top three scores then it stored at its respective position and a menu appears on the screens with option of restart and display top three high scores.

CHAPTER-5

RESULTS (OUTPUTS)

3.1 CONNECTION OF THE PROJECT ON PROTEUS ISIS

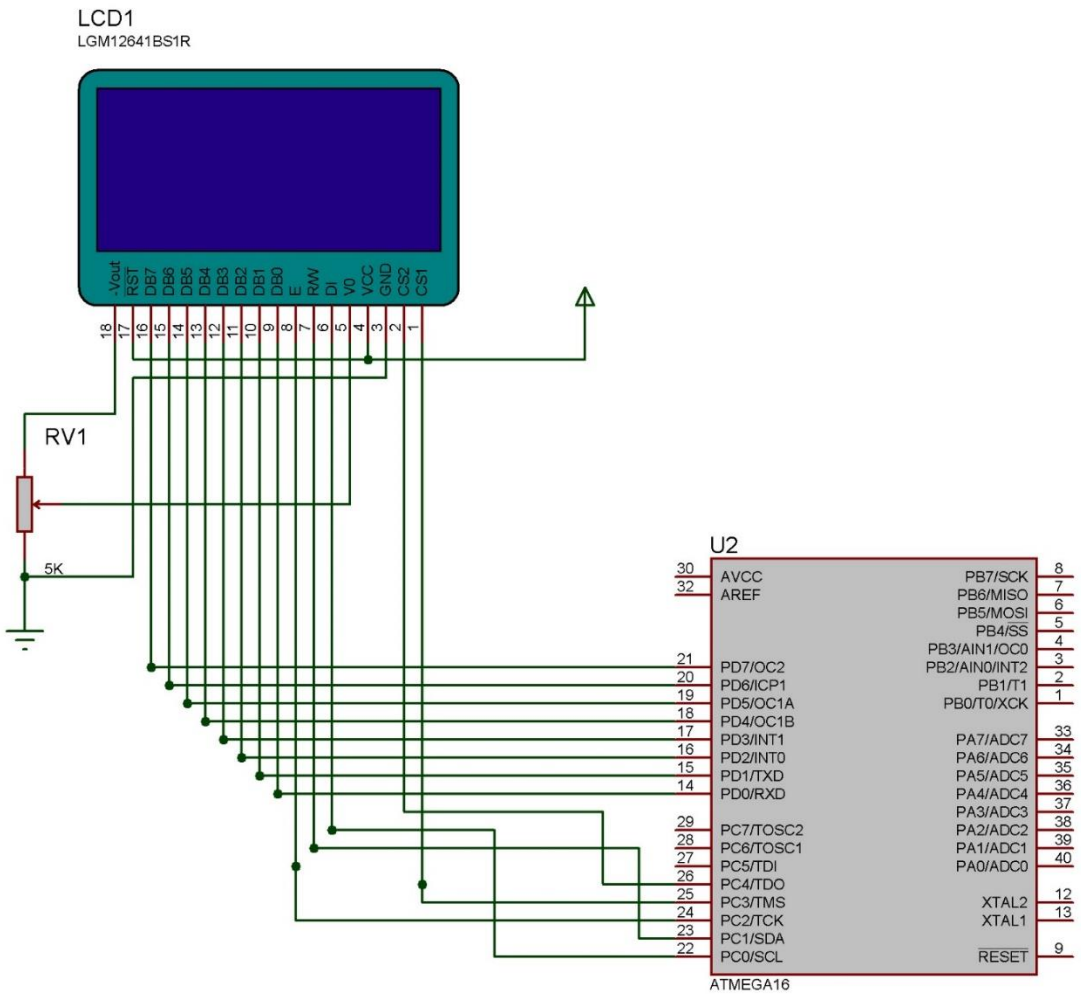


Fig14: pin connection of graphic LCD interfacing in microcontroller

3.2 COMPLETE HARDWARE SETUP OF GAME BOX

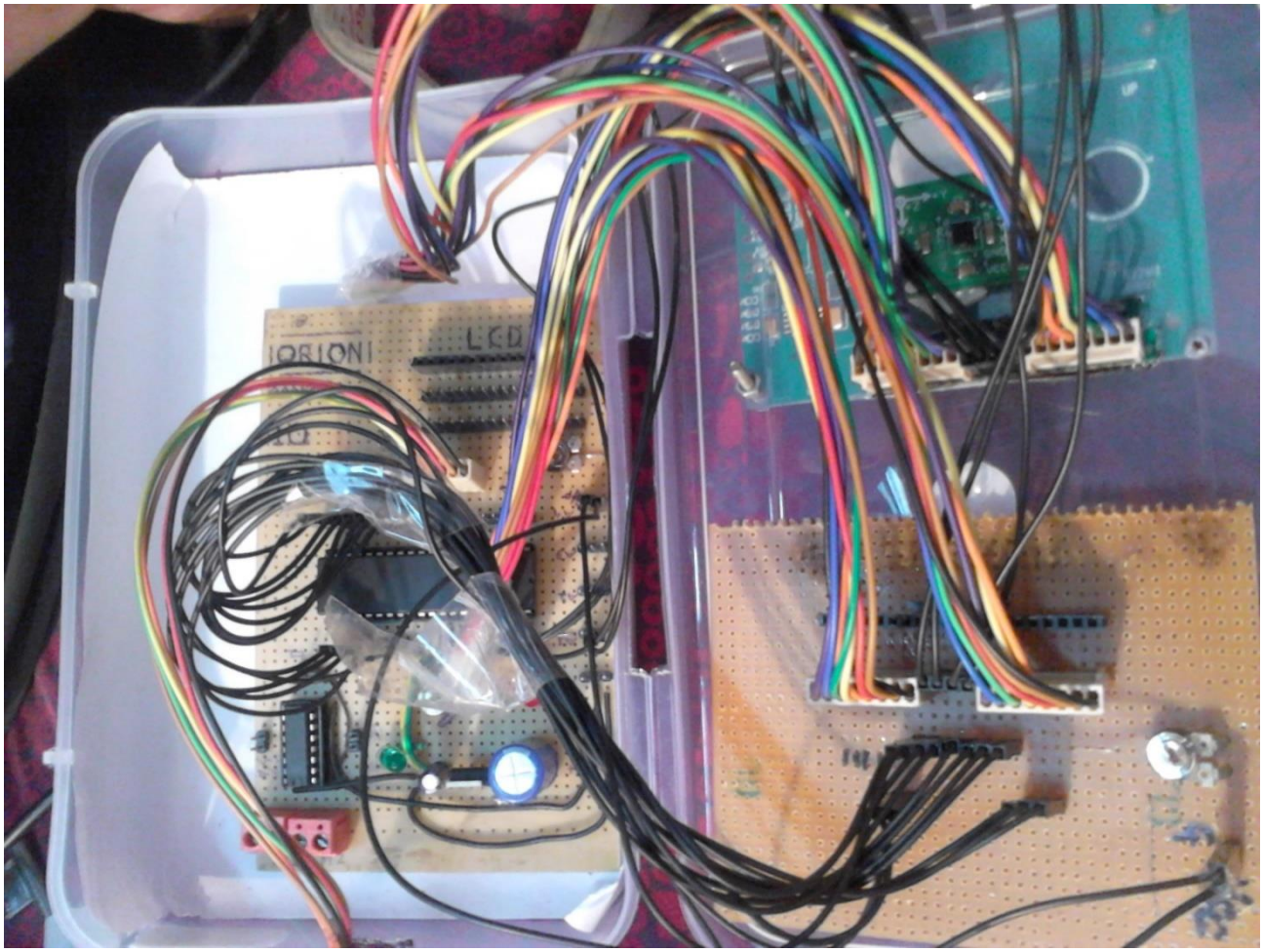


Fig 15: Complete Hardware setup Screenshot includes GLCD, Controller and Sensor setup.

3.3 GRAPHIC LCD OUTPUTS:

3.2.1 1st level output

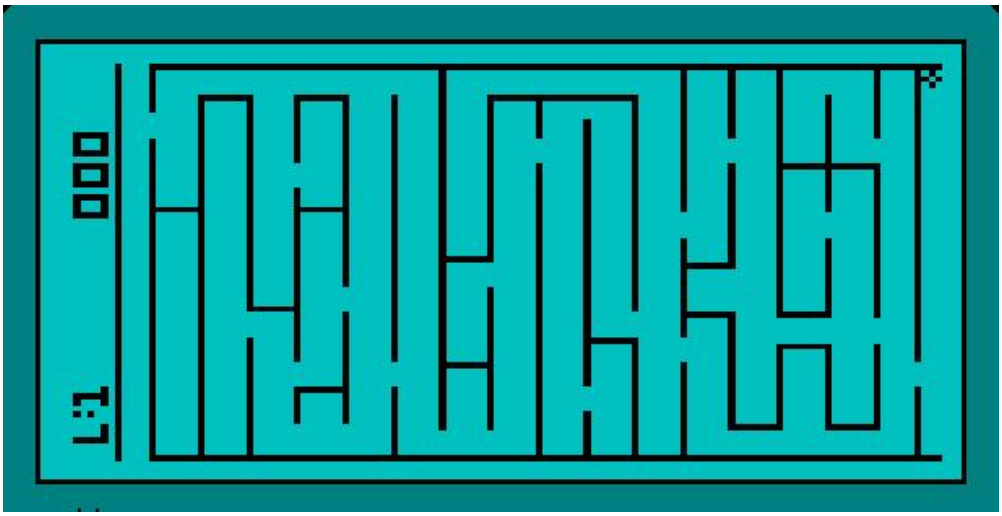


Fig 16: Output display of level 1

3.2.2 2ND level output

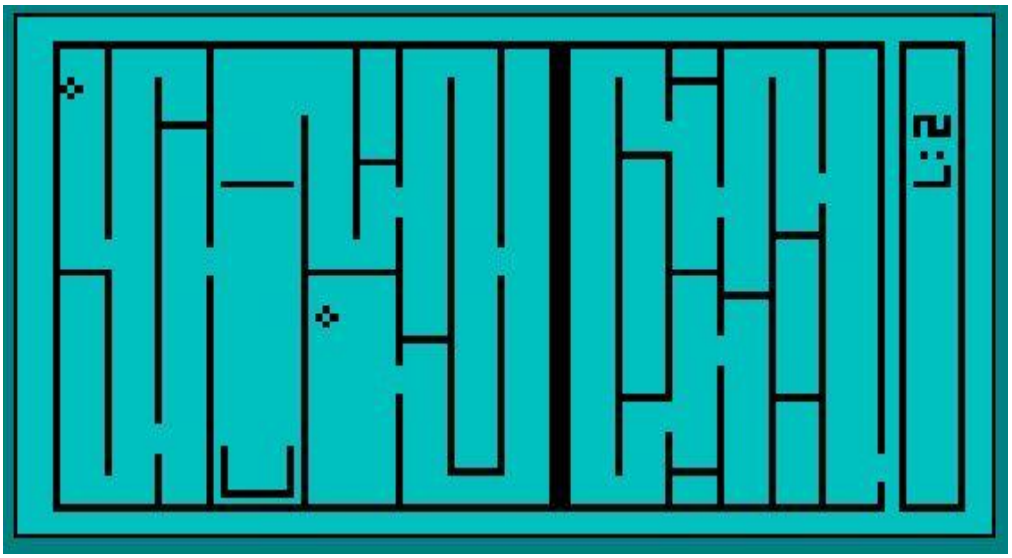


Fig 17: Output of level 2

3.2.3 3rd level output

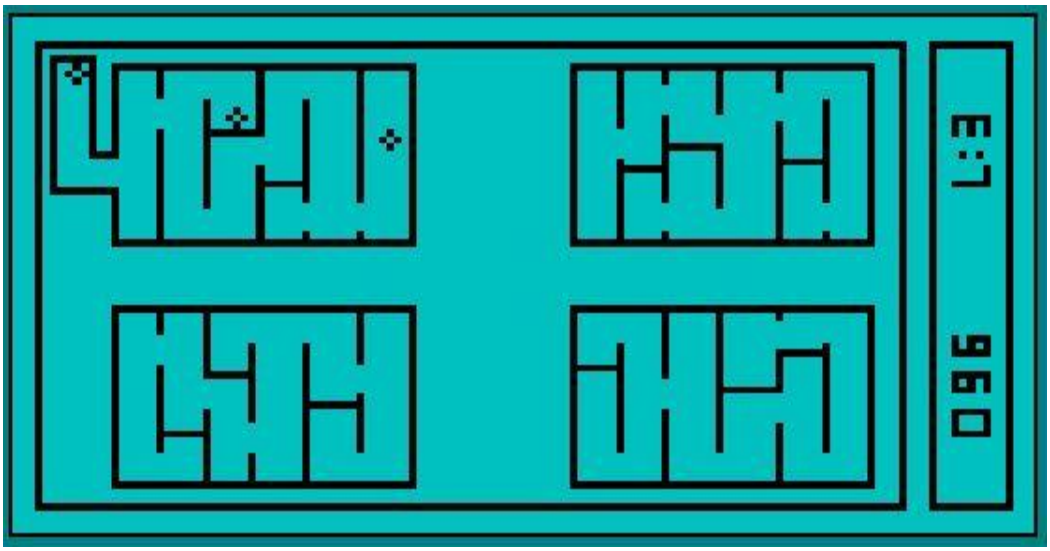


Fig 18: Output of level 3

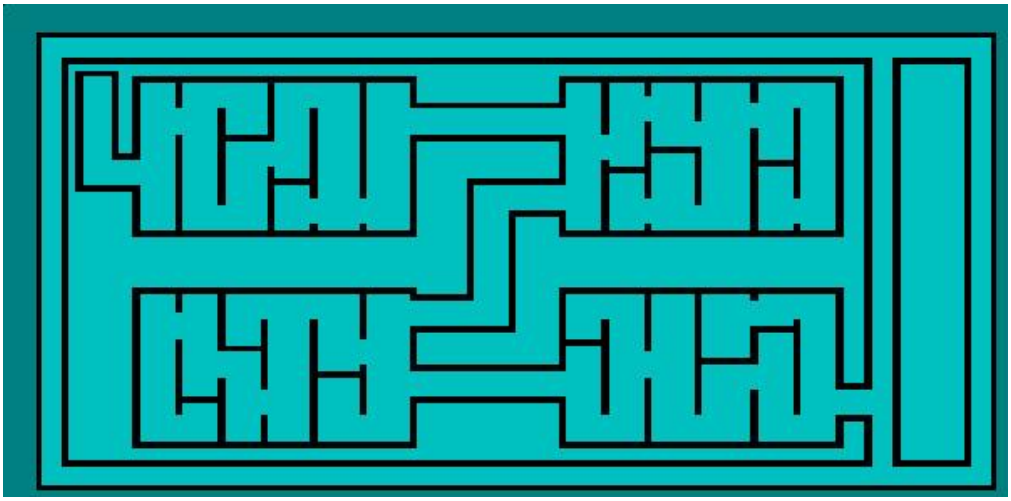


Fig 19: output of level 3 (when all tunnels are open)

3.2.4 Displaying High Score

Fig 20: Output display of Level Completed

3.2.5 Displaying Game Over

Fig 21: Displaying Gave Over

3.2.5 Displaying Menu

Fig 22: Displaying Menu

3.2.5 Displaying High Scores

Fig 23: Displaying High Score

Bibliography

[1] F. Maximilian Thiele, me@apetech.de
Library used for KS108B controller of Graphic LCD (128x64) to operates the LCD and perform a lots of function and Java Based Font Creator is completely written and shared by F. Maximilian Thiele.

[2] Avinash Gupta, eXtremeElectronics.co.in
Library of 16x2 LCD used for testing purposes and interfacing accelerometer is completely written and shared by Avinash Gupta, founder of eXtreme Electronics.

[3] AtMega 32 Datasheet, ATMEL Corporation
Datasheet of Atmega 32 is used mainly in literature study particularly Analog to Digital Converter and registers of Atmega 32, this datasheet is the copyright of ATMEL Corporation, provided as product manual.

[4] GLCD Datasheet, Electronic Theatre Controls Inc.
Datasheet of Graphic LCD of Product dose: LG128643-SMLYH6V is used for the Different function of LCD and PIN configuration during the project work.

[5] ADXL335 Datasheet
Accelerometer ADXL335 Datasheet is used to work with the accelerometer.

[6] AVR Microcontroller and Embedded Systems: Using Assembly and C
Text book on AVR microcontroller by Muhammad Ali Mazidi, Pearson Custom Electronic Media

[7] Google.com, Google Inc.
Google helped at each and every step of the project idea to project work and completion.

Code (Compact Disc)