كلية الهندسة
قسم هندسة الالكترونيات والاتصالات

Faculty of Engineering of Helwan University
Communication & Electronics Department

Hand talk (A Smart Hand Glove Interpreter)

Graduation project year 2020 -2021Communication and Electronics Department

Project Executive Team

Supervisor:

- **Dr. Eman Sedhom**

Executives:

- **Ahmed Gamal Shaker**
- **Asmaa Ahmed El-Sktey**
- **Asmaa Essam Said**
- **Abdullah Abd El-Kareem**
- **Mustafa Mohamed Fouad**

Table of Contents

Table of figure

ACKNOWLEDGMENTS:

This graduation project was completed at Helwan university_ faculty of engineering during the period of September 2020 - July 2021 basically describes our work and study in our graduation project.

First praise is to Allah, the Almighty, on whom ultimately, we depend for sustenance and guidance we are also deeply appreciate to our supervisor, Dr. Eman Sedhom for their invaluable trust kind cooperation sincere encouragement and invaluable guidance throughout the whole project making process. Most of all we are forever obliged to our families for their unconditional love, abundant support, and non-stop encouragement throughout our entire life.

Also, we would like to thank our professors in communication and electronics department, who made their best to teach us and guided us through the right path that led us to excellence. at the end we would like to express our profound gratitude to Allah who always we ask for his assistance and we owe him any success and progress we made in our life.

Abstract

In general, deaf people have difficulty in communicating with others who do not understand sign language. Even those who do speak aloud typically have a "deaf voice" of which they are self-conscious and that can make them reticent.

The Hand Talk glove is a normal, cloth driving glove fitted with flex sensors. Technology has always been of great help to the disabled and given them a helping hand to allow them to live a normal and healthy life like others. we have come up with a novel idea of a glove named Hand talk that will convert hand movements into text and allow the deaf to express themselves better.

The Hand talk glove needs to be worn on the hand by the deaf or mute person and depending on the variation of movement, the device will convert it intelligently into voice for the other person to comprehend it easily.

Communication between speakers and non-speakers of Sign Language can be problematic, inconvenient, and expensive. This project attempts to bridge the communication gap by designing a portable glove by translating the hand gestures into its corresponding meaning then output that meaning in an audio form and display it on a screen.

The project consists of three main parts, the first part is the glove itself which is designed and fabricated specifically for this purpose, the glove is equipped with flex sensors, contact sensors, and a Gyroscope to measure the rotation of the hand. The second part is containing the glove's AVR Atmega32 microcontroller which is the control panel that holds the controller and the main component at the back of the arm, finally the interface panel which holds the LCD screen and Arduino UNO microcontroller handle sound department through speaker. The controller collects readings from sensors and apply a programed algorithm to compare the collected data with the stored data if the motion is identified a signal is sent to the LCD and loudspeaker simultaneously to output the corresponding gesture sound.

Advantage: if the patient performs a gesture of fist (most often we people form the fist in case of fear, emergency), there would be a message displayed on the LCD: "CRITICAL SITUATION" or "EMERGENCY" along with a beep and that CRITICAL message would also be transmitted in the form of text or call to the Doctor's phone or the Relative's one.

Chapter 1: Introduction

1. Introduction

Communication is of paramount importance in today's data driven society. Shopping at a supermarket, asking for directions, event planning, or even national security depend on reliable communication. To the average person, everyday routines such as dentist appointments or having dinner with a friend would be difficult without the ability of speaking to convey ideas.

The sign language is one of the oldest means of communication and communication for the deaf and dumb, which appeared in Spain in the 17th century to deal with those without the ability to speak and hear. It is also one of the languages in which the language of the hands is used with specific signs, and the signs are according to each letter of the alphabet, through which sentences can be formed, and the sign language is not limited to the movement of hands, it includes facial expressions, lip movement and expressions with the movement of the body.

Sign languages are fully distinguished natural languages, structurally different from spoken languages, and there is also an international sign language, which Deaf people use in international meetings and informally when traveling and socializing. This is an uncomplicated form of sign language like natural sign language and has a limited dictionary. The Convention on the Rights of Persons with Disabilities recognizes and promotes the use of sign languages, as it makes clear that sign languages are equal in status regarding spoken languages and obliges states parties to facilitate the learning of sign language and enhance the language identity of the deaf community.

Many listeners believe that deaf people use a single sign all over the world, and that sign language is a unified language and are surprised when they know that each country has its own sign language that differs from the rest of the countries, not only this, but there may be more than one language in the same country. Each sign language differs from the other because it reflects the history, culture, customs of society and the environment in which the deaf live. Therefore, we find different signs - from one sign language to another - to global concepts such as: mother, father, day, night, tree, and so on. But who can use sign language in one country can communicate easily with sign language users of another country despite the two different languages, unlike Spoken language users who find it difficult to communicate if neither or both of they are fluent in both languages? As for the deaf, they are good at building bridges to bridge language gaps with the other and to communicate with him.

The firsthand Talk glove was designed by Ryan Patterson in the year 2001. Sign Language Translator consists of two separate components, a leather golf glove that has ten flexible sensors sewn into it which monitor the position of the fingers by measuring the electrical resistance created by the fingers as they bend. A small microcontroller on the back of the hand converts the change in the electrical current into digital signals and transmits them wireless to a computer The main disadvantage with this model was that a computer or a laptop was always required for its functioning which made it less portable. Sign language is the language used by deaf and mute people and it is a communication skill that uses gestures instead of sound to convey meaning simultaneously combining hand shapes, orientations and movement of the hands, arms or body and facial expressions to express fluidly a speaker's thoughts. Signs are used to communicate words and sentences to audience. A gesture in a sign language is a particular movement of the hands with a specific shape made of them. A sign language usually provides sign for whole words. It can also provide sign for letters to perform words that do not have corresponding sign in that sign language. In this project Flex Sensor Plays the major role, Flex sensors are sensors that change in resistance depending on the amount of bend on the sensor.

Figure 1: hand talk

This project uses PIC microcontroller to control all the processes and flex sensors along with accelerometer sensors will track the movement of fingers as well as entire palm. an LCD will be used to display the users' gesture and a speaker to translate the gesture into audio signal is planned, if possible, for execution. This project can be further developed to recognize complex like food, water, etc.

The importance of sign language is attributable to the deaf and dumb, as it works to communicate between the deaf and dumb and people and transmits mutual feelings between them; it also helps to works on the mental, oral and indicative development of people with special capabilities of the deaf and dumb, and it also helps to reduce It is the internal and psychological pressures that afflict those suffering from speechlessness and hearing; and it helps to get rid of fear, depression and frustration in the deaf and dumb, and it works to develop social, cognitive and cultural relations of individuals.

2. Problem statement

Globally 360 million people suffer from hearing loss that causes disability, and 34 million people are children, 7.5 million people suffer from hearing loss in Egypt. representing 10% of the deaf and dumb in the world, whose number is estimated at 72 million people worldwide, the latest statistics issued by the Egyptian Organization for Persons with Disabilities in 2000 indicate that their numbers in Egypt are 3 million people. more than 80% of whom live in developing countries It is estimated that by 2050 over 900 million people will experience hearing loss.

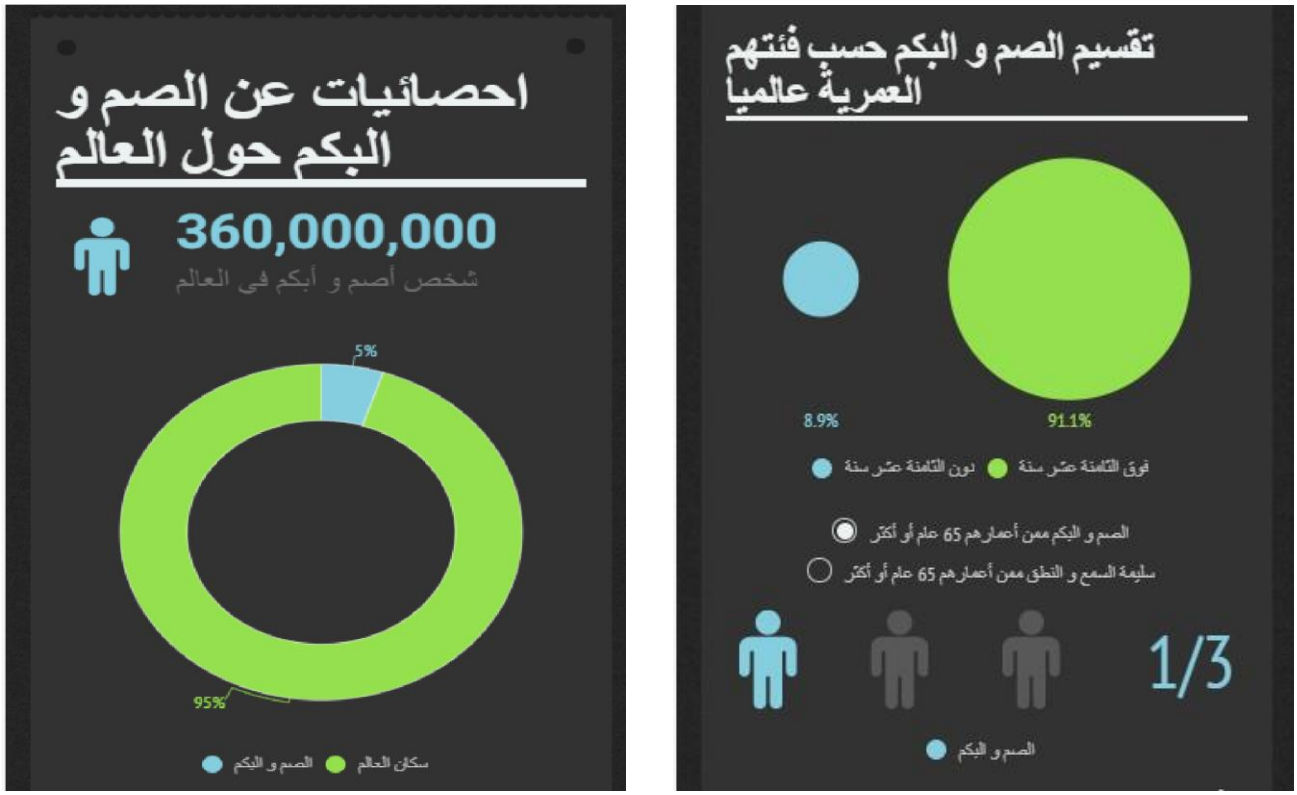


Figure 2 Statistics about deaf and dumb

Figure 3 Global production capacity to provide deaf hearing aids

Before talking about the situation of the deaf and dumb in Egypt, we note that sign language is one of the oldest means of communication and communication for the deaf and mute, and it appeared in Spain in the 17th century to deal with those who do not have the ability to speak and hear. It is also considered one of the languages in which the language of the hands is used with specific signs, and the signs are according to each letter of the alphabet, and through them it is possible to form sentences.

Sign language is the only communication tool used by deaf people to communicate to each other. However, normal people do not understand sign language, and this will create a large communication barrier between deaf people and normal people. In addition, the sign language is also not easy to learn due to its natural differences in sentence structure and grammar and there are a small number of people can learn or talk this language. Therefore, there is a need to develop a system which can help in translating the sign language into text and voice to ensure the effective communication can be easily take place in this community.

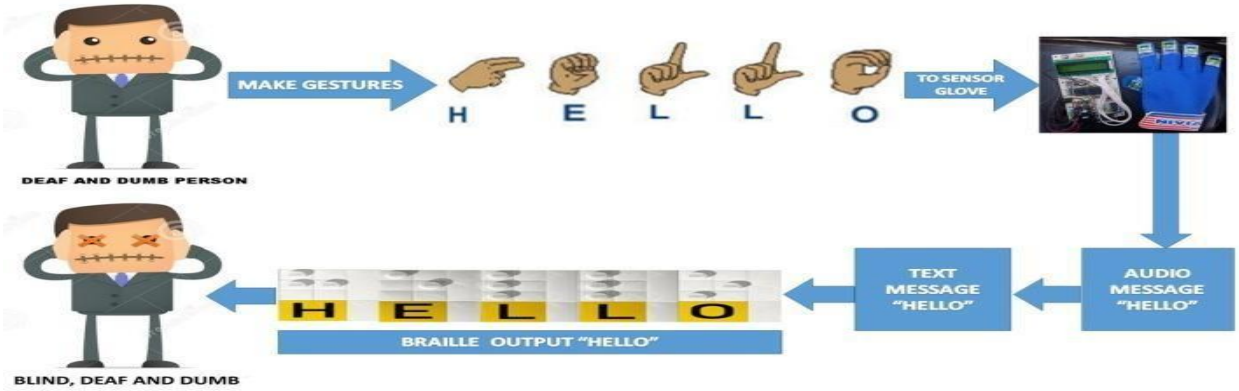


Figure 4 problem of dumb and deaf people

The Egyptian sign language

Figure 6 Some Arabic sign language

Figure 5 Shows the Arabic Alphabets in The Egyptian sign language.

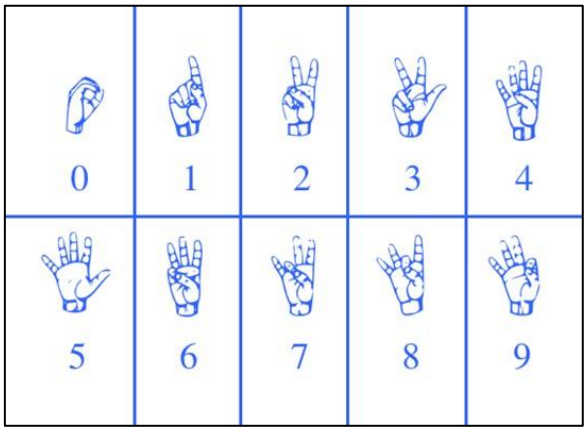


Figure 7 Numbers

3 Goals

3.1 Why hand talk gloves!

- ***The main objective of this project is to help dumb people by removing Communication barrier so that they are not restricted in a small social circle and are able to convey their feelings and emotions whenever they want.***
- ***Also, it would be helpful in educational and health issues related to dumb people.***
- ***Our project design aims to develop a user friendly, low powered glove to translate the LEM "Egyptian sign language" gesture into text and voice, The electronic glove will do that with combination of flex sensor, accelerometer, Gyroscope, and Arduino UNO microcontroller.***
- ***The glove system will interrupt the data from a variety of sensors to capture the four parameters of the Egyptian sign language. The four parameters that distinguish sign language gestures are hand shape, orientation of the palm, location of the hand and movement. Using a hand recognition algorithm, the gestures will be translated from signs language letters to words. Then it will be printed on an LCD and pronounced by a speaker.***
- ***With this glove, we are mainly aiming to help people with hearing and speech impairment. By taking into consideration the four parameters of the Egyptian sign language.***
- ***We design and build a glove to be worn on the right hand that uses a microcontroller to translate sign language into text and speech in real time. every person's hand has a unique size and shape, and we aim to create a device that could provide reliable translations regardless of those differences.***
- ***It requires fewer components, so its cost is low.***
- ***minimum size: Due to small size we can place its hardware on our hand easily.***
- ***Flexible to users.***
- ***It takes less power to operate system.***
- ***Easy to define gestures: we can add or define our own gestures.***
- ***To solve deaf-mute people problem by give a voice to the voiceless.***
- ***A new way for easy life, now you can talk and send messages to your friends without looking to your mobile screen.***

3.2Why Microcontroller ATMEGA 32 not Arduino!

- ***The high-performance, low-power Microchip 8-bit microcontroller combines 32 KB ISP flash memory with read-while-write capabilities.***
- ***1 KB EEPROM, 2 KB SRAM, 54/69 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for boundary-scan and on-chip debugging/programming,***
- ***three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a universal serial interface (USI) with start condition detector.***
- ***8-channel 10-bit A/D converter, programmable watchdog timer with internal oscillator, SPI.***
- ***Low Cost***

3.3 FUTURE WORK

It makes this easily portable and easier to use. Even though the less advanced flex sensors are used, still many sounds can be pre- recorded in the recorder and can be used through the programming. The flex sensors working in the two planes provide a lot of options for movement of the fingers and the thumb which is later transmitted into voice. and group of body sensors along with the pair of data gloves can be manufactured so that a deaf and dumb person can communicate to any normal person anywhere. Future work to do is:

- ***To give more advanced features Reduce the size of the project even more High-quality sensor can use the range can be increased***
- ***With the help of different gestures commands, different other commands can be added.***
- ***For more reliable and low complexity of the circuit microcontroller can be replaced by the Arduino or other Advanced Microcontrollers.***
- ***In this project many types of other applications can be added with using the different type of sensors like Heartbeat sensors for heartbeat monitoring and Temperature sensor for body temperature monitoring.***
- ***High quality sensor can use.***

Chapter 2: The Hardware

1. Structured of our system.

This diagram shown in figure 8 will be our structured of our project.

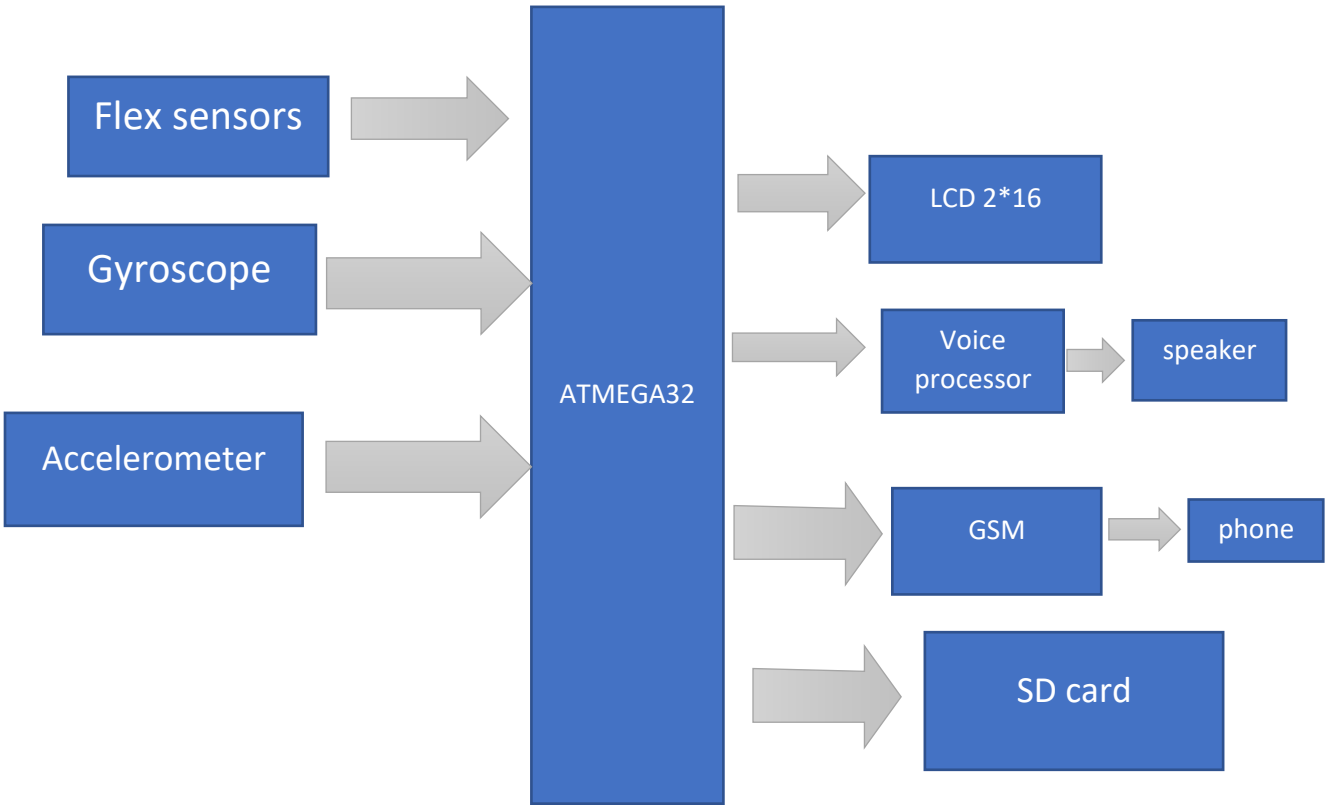


Figure 8 High-level Diagram of the design

The sensor system is made up of two main components: the flex sensor circuits, the MPU 6050 Accelerometer and Gyroscope Module.

2. Components in the project

1. ***AVR (ATMEGA32)***
2. ***Flex sensors***
3. ***MPU 6050 Accelerometer and Gyroscope Module.***
4. ***LCD 2*16***
5. ***GSM***
6. ***SD CARD Module***
7. ***Voice Processor (Arduino UNO)***
8. ***Speaker***
9. ***Gyroscope***
10. ***Accelerometer***

2.1 AVR ATMEGA32 Microcontroller

Figure 9 atmega32

Today's microcontrollers are much different from what it was in the initial stage, and the number of manufacturers is much more in count than it was a decade or two ago. At present some of the major manufacturers are Microchip (publication: PIC microcontrollers), Atmel (publication: AVR microcontrollers), Our interest is upon ATmega32. It belongs to Atmel's AVR series micro controller family.

PIN count: Atmega32 has got 40 pins. Two for Power (pin no.10: +5v, pin no. 11: ground), two for oscillator (pin 12, 13), one for reset (pin 9), three for providing necessary power and reference voltage to its internal ADC, and 32 (4×8) I/O pins.

About I/O pins: ATmega32 is capable of handling analogue inputs. Port A can be used as either DIGITAL I/O Lines or each individual pin can be used as a single input channel to the internal ADC of ATmega32, plus a pair of pins AREF, AVCC & GND (refer to ATmega32 datasheet) together can make an ADC channel.

. Timers: 3 Inbuilt timer/counters, two 8 bit (timer0, timer2) and one 16 bits (timer1).

. ADC: It has one successive approximation type ADC in which total 8 single channels are selectable. They can also be used as 7 (for TQFP packages) or 2 (for DIP packages) differential channels. Reference is selectable, either an external reference can be used or the internal 2.56V reference can be brought into action. There external reference can be connected to the AREF pin.

Communication Options

ATmega32 has three data transfer modules embedded in it. USART, Serial Peripheral Interface, I2C.

Atmega32 pin diagram

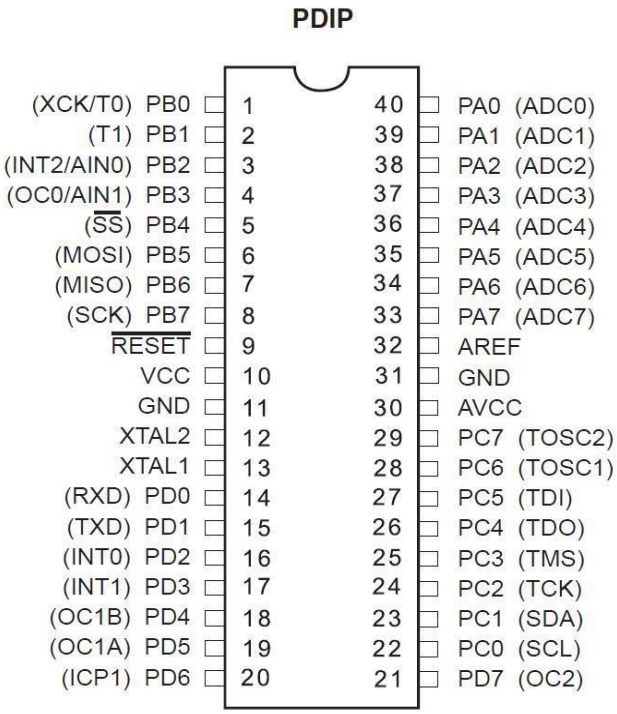


Figure 10 atmega32 pin

***Memory:** It has 32Kbytes of In-System Self-programmable Flash program memory, 1024 Bytes EEPROM, 2Kbytes Internal SRAM. Write/Erase Cycles: 10,000 Flash / 100,000 EEPROM.*

***Clock:** It can run at a frequency from 1 to 16 MHz Frequency can be obtained from external Quartz Crystal, Ceramic crystal, or an R-C network. Internal calibrated RC oscillator can also be used.*

***More Features:** Up to 16 MIPS throughput at 16MHz. Most of the instruction executes in a single cycle. Two cycle on-chip multiplication. 32 × 8 General Purpose Working Registers*

***Debug:** JTAG boundary scan facilitates on chip debug.*
***Programming:** Atmega32 can be programmed either by In-System Programming via Serial peripheral interface or by Parallel programming. Programming via JTAG interface is also possible. Programmer must ensure that SPI programming and JTAG are not disabled using fuse bits if the programming is supposed to be done using SPI or JTAG.*

2.2 Flex sensor

Flex sensors are usually available in two sizes. One is 2.2 inch, and another is 4.5 inches. Although the sizes are different the basic function remains the same. They are also divided based on resistance. There are LOW resistance, MEDIUM resistance, and HIGH resistance types. Choose the appropriate type depending on the requirement.

Figure 11 Flex sensor 2.2-inchFigure 10: flex sensor 4.5 inch

2.2.1 FLEX SENSOR function

FLEX SENSOR terminal resistance changes when it is bent.

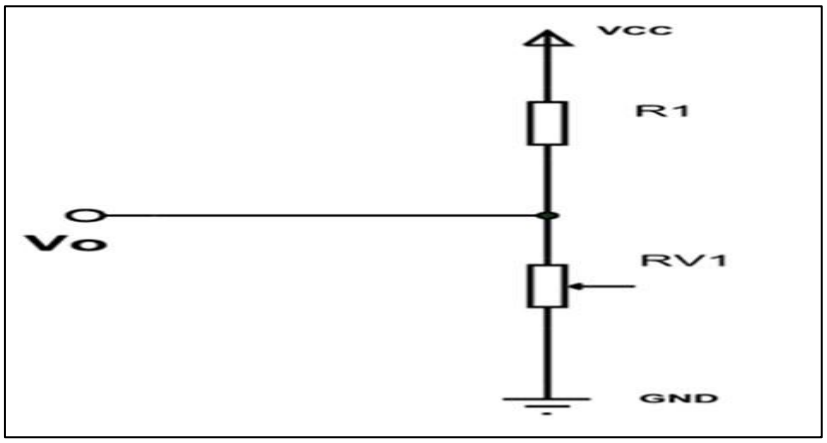


Figure 12 flex sensor function

Vo = VCC (Rx/ (R1+Rx)) #Rx: FLEX SENSOR resistance

2.2.2 FLEX SENSOR Pin Configuration

The flex sensor is a two-terminal device. The Flex sensor does not have polarized terminals like a diode. So, there is no positive and negative. There are two pins on it p1 and p2.

2.2.3 Features

I. ***Angle Displacement Measurement.***
II. ***Bends and Flexes physically with motion device.***

2.2.4 Possible Uses

I. ***Robotics - Gaming (Virtual Motion)***
II. ***Medical Devices***
III. ***Musical Instruments***
IV. ***Physical Therapy***
V. ***Simple Construction***

In this project we use the two 2.2 flex sensor.

1.Flex sensor 2.2 inch:
We use it for pinky thumb fingers.

Electrical specification:

I. ***Flat Resistance: 25K Ohms.***
II. ***Resistance Tolerance: ±30%.***
III. ***Bend Resistance Range: 45K to 125K Ohms (depending on bend radius).***
IV. ***Power Rating: 0.50 Watts continuous. 1 Watt Peak.***

Mechanical Specifications:

I. Life Cycle: >1 million.
II. ***Height: 0.43mm (0.017").***
III. ***Temperature Range: -35°C to +80°C.***

2.2.5 2D-Model

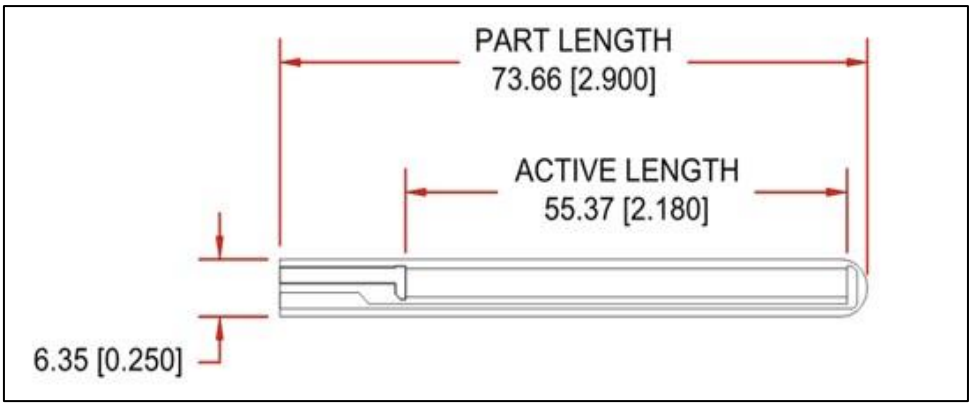


Figure 13 Measurements in millimeter (inches)

2.2.6 Characteristics of flex sensors

Figure 14 Bending Vs Resistance

Figure 15 Resistance Vs Voltage

2.2.7 how to use flex sensor

FLEX SENSOR is basically a VARIABLE RESISTOR whose terminal resistance increases when the sensor is bent. So, this sensor resistance increases depends on surface linearity. So, it is usually used to sense the changes in linearity.

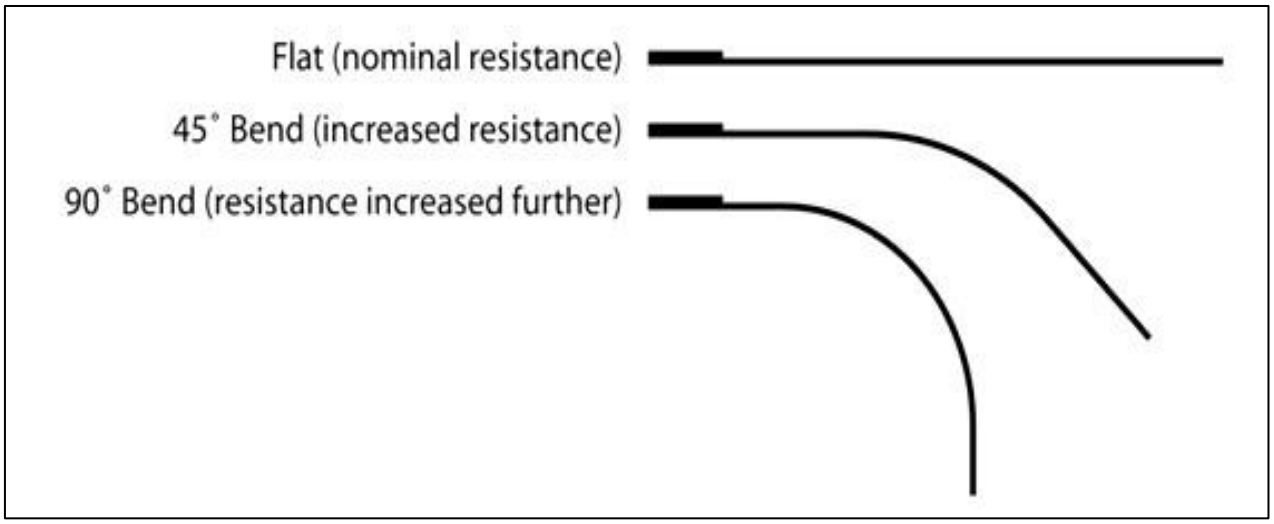


Figure 16 flex sensor 2.2 inch

2.3 MPU 6050

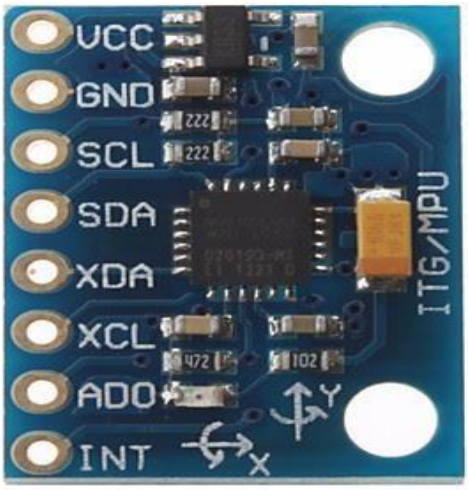


Figure 17 MPU 6050

MPU6050 sensor module is complete 6-axis Motion Tracking Device. It combines 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in small package. Also, it has additional feature of on-chip Temperature sensor. It has I2C bus interface to communicate with the microcontrollers.

It has Auxiliary I2C bus to communicate with other sensor devices like 3-axis Magnetometer, Pressure sensor etc.

If 3-axis Magnetometer is connected to auxiliary I2C bus, then MPU6050 can provide complete 9-axis Motion Fusion output.

2.3.1 MPU6050 inside sensors.

3-Axis Gyroscope

A gyroscope is a device for measuring or maintaining orientation, based on the principles of conservation of angular momentum. Device used to measure or maintain orientation.

- ***Works on the principals of angular momentum***
- ***Initial axis of rotation is conserved.***
- ***Consists of a spinning mass on an axel..***

Gyroscope

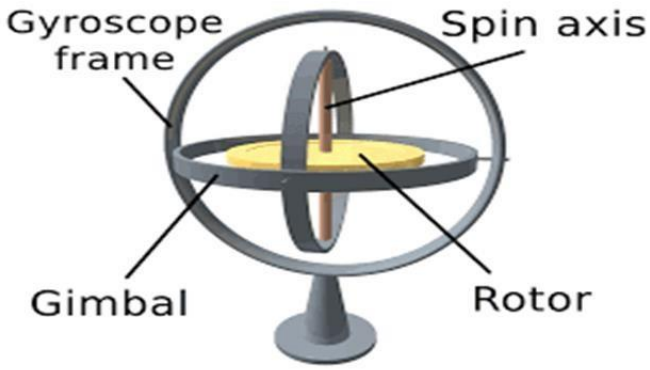


Figure 18 the rotor in gyroscope

The MPU6050 consist of 3-axis Gyroscope with Micro Electromechanical System (MEMS) technology. It is used to detect rotational velocity along the X, Y, Z axes

Gyroscope Working Theory:

- ***When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a MEM inside MPU6050.***
- ***The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate.***
- ***This voltage is digitized using 16-bit ADC to sample each axis.***
- ***The full-scale range of output are +/- 250, +/- 500, +/- 1000, +/- 2000.***
- ***It measures the angular velocity along each axis in degree per second unit.***

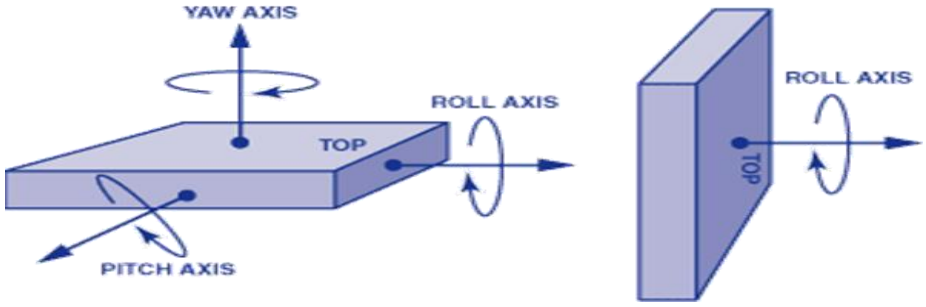


Figure 19 Gyroscope Working Theory

Accelerometer

An accelerometer is a device that measures direction and angle of rotation in three orthogonal directions: Roll, Pitch, and Yaw when hand fixed. An accelerometer is an electromechanical device used to measure acceleration forces.

Such forces may be static, like the continuous force of gravity or, as is the case with many Mobile devices, dynamic to sense movement or vibrations.

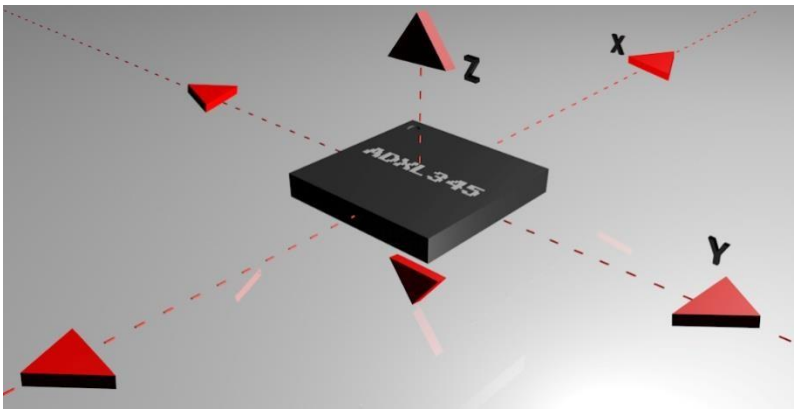


Figure 20 Accelerometer

The MPU6050 consist of 3-axis Accelerometer with Micro Electromechanical (MEMs) technology. It used to detect angle of tilt or inclination along the X, Y and Z axes

Modes of accelerometer:

- ***Acceleration along the axes deflects the movable mass.***
- ***This displacement of moving plate (mass) unbalances the differential capacitor which results in sensor output. Output amplitude is proportional to acceleration.***
- ***16-bit ADC is used to get digitized output.***
- ***The full-scale range of acceleration are +/- 2g, +/- 4g, +/- 8g, +/- 16g.***
- ***It measured in g (gravity force) unit.***
- ***When device placed on flat surface it will measure 0g on X and Y axis and +1g on Z axis.***

2.3.2 MPU-6050 Module

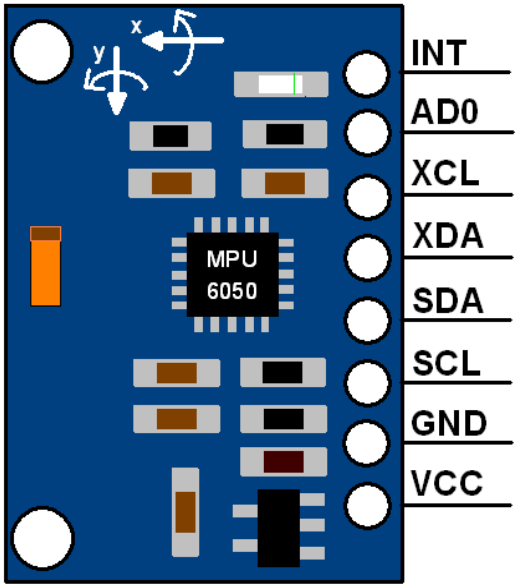


Figure 21 MPU-6050 Module

The MPU-6050 module has 8 pins,

INT: Interrupt digital output pin.

AD0: I2C Slave Address LSB pin. This is 0th bit in 7-bit slave address of device. If connected to VCC then it is read as logic one and slave address changes.

XCL: Auxiliary Serial Clock pin. This pin is used to connect other I2C interface enabled sensors SCL pin to MPU-6050.

XDA: Auxiliary Serial Data pin. This pin is used to connect other I2C interface enabled sensors SDA pin to MPU-6050.

SCL: Serial Clock pin. Connect this pin to microcontrollers SCL pin.

SDA: Serial Data pin. Connect this pin to microcontrollers SDA pin.

GND: Ground pin. Connect this pin to ground connection.

VCC: Power supply pin. Connect this pin to +5V DC supply.

MPU-6050 module has Slave address (When AD0 = 0, i.e., it is not connected to Vcc) as,

Slave Write address (SLA+W): 0xD0 & Slave Read address (SLA+R): 0xD1

MPU-6050 has various registers to control and configure its mode of operation.

2.4 GSM Module

Send Receive SMS & Call with SIM900 GSM Shield & Atmega32

A GSM modem or GSM module is a hardware device that uses GSM mobile telephone technology to provide a data link to a remote network. From the view of the mobile phone network, they are essentially identical to an ordinary mobile phone, including the need for a SIM to identify themselves to the network.

Before we get into what a GSM module is, let us get our fundamentals right and understand what GSM and GPRS are. A GSM module or a GPRS module is a chip or circuit that will be used to establish communication between a mobile device or a computing machine and a GSM or GPRS system

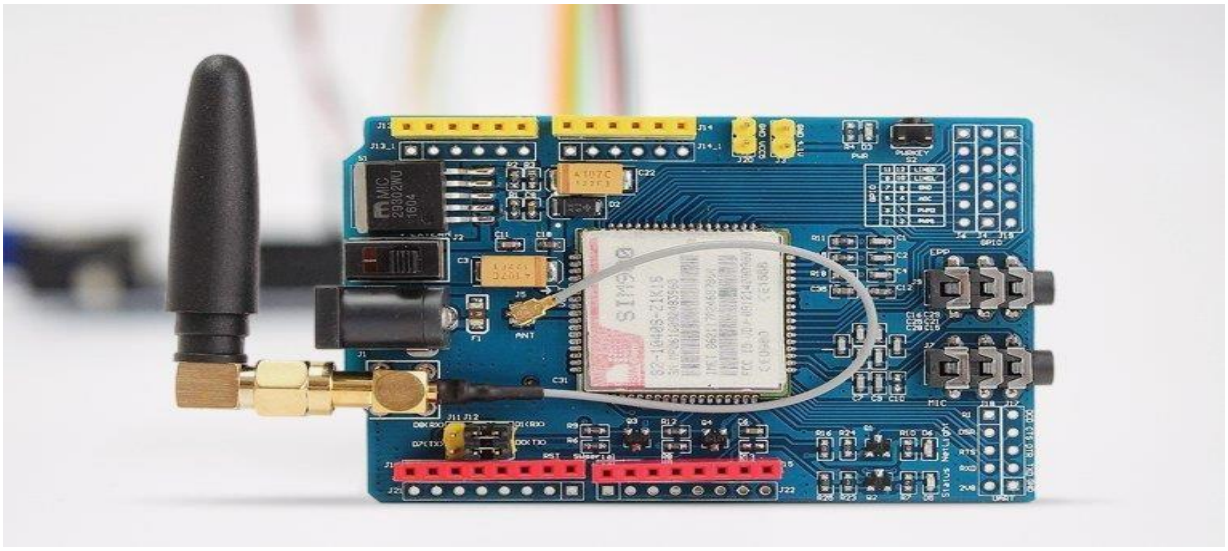


Figure 22 sim900 GSM shield

2.4.1 Hardware Overview of SIM900 GSM/GPRS Shield

The SIM900 GSM/GPRS shield is designed to surround the SIM900 chip with everything necessary to interface with microcontroller atmega32, plus a few extra goodies to take advantage of the chip's unique features.

Let us familiarize ourselves with these features and abilities of the shield.

Here is a quick overview:

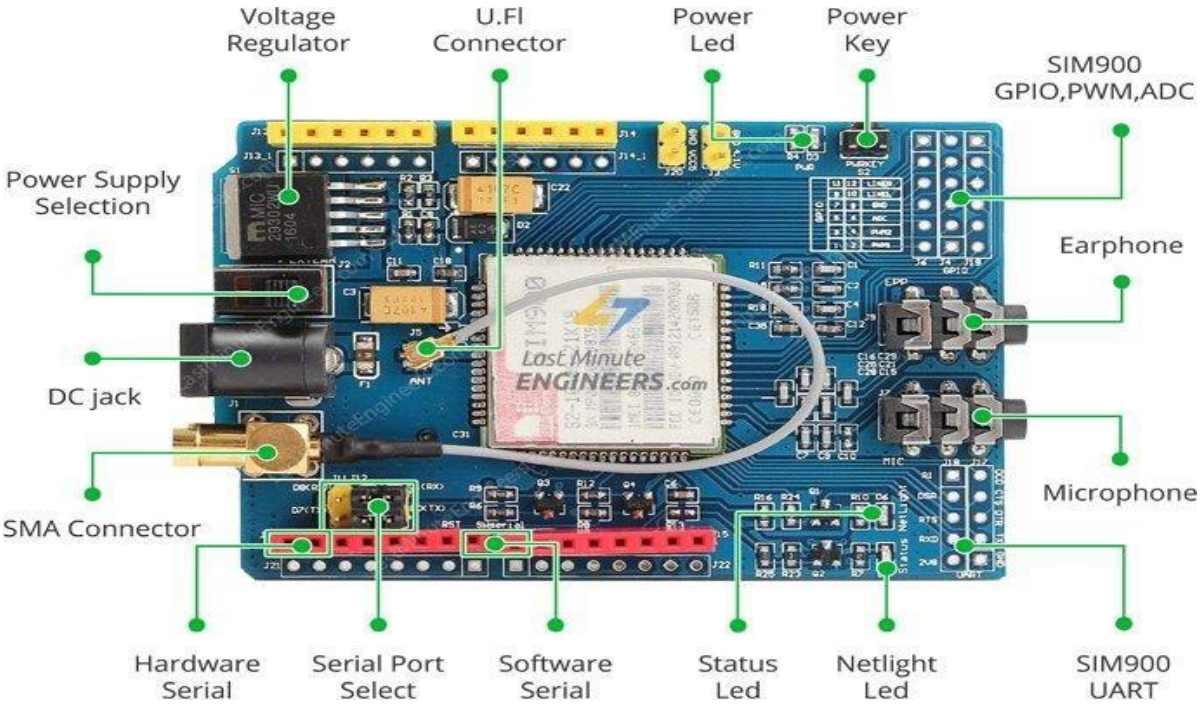


Figure 23 SIM900 GSM/GPRS Shield

2.4.2 LED Status Indicators

There are three LEDs on the SIM900 GSM/GPRS shield which indicates connectivity or power status. By observing these LEDs, you can get a visual feedback on what's going on with the shield.

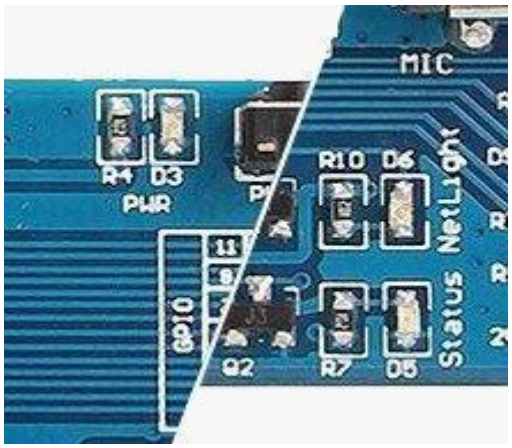


Figure 24 LED Status

Status LED: This LED indicates SIM900's working status. If this LED is on, the chip is in working mode.

Power LED: This LED is connected to the shield's power supply line. If this LED is on, the shield is receiving power.

Network light LED: This LED indicates the status of your cellular network. It will blink at various rates to show what state it is in. it should be blinking every 3 sec that means The SIM900 chip is registered to the cellular network & can send/receive voice and SMS.

Supplying Power for SIM900 Shield

One of the most important parts of getting the SIM900 shield working is supplying it with enough power. Depending on which state it is in, the SIM900 can be a relatively power-hungry device.

NOTE:
we can add an external power supply to the shield with the 5.5mm DC jack, to which we can connect any 5V-9V DC wall adapter we have, and, in our project, we used adapter with 9V DC. Next to the DC jack, is a Slide Switch to select the power source labeled EXTERN. To use external power source

Figure 25 DC jack of external power source

2.4.3 Antenna

An antenna is required to use the SIM900 for any kind of voice or data communications as well as some SIM commands. The shield usually comes with a 3dBi GSM antenna and allows us to put the shield inside a metal case (as long the antenna is outside).

Figure 26 antenna in GSM

2.4.4 Main Target of using GSM in the Project

we have to write the GSM code and attach the hand movement to it; As it is the main use of GSM in our project that when a deaf and dumb patient senses any disease, he must move his hands in a specific way, and then the GSM calls or sends an SMS to the doctor responsible for the patient's condition through or to any relative person to patient with the commands designated for GSM. By programming it with the controller in C language.

Figure 27 Feeling pain movement.

Figure 28 block diagram of GSM

SMS message: (Harry UP!! I Felt tired and need your help)

2.5 LCD

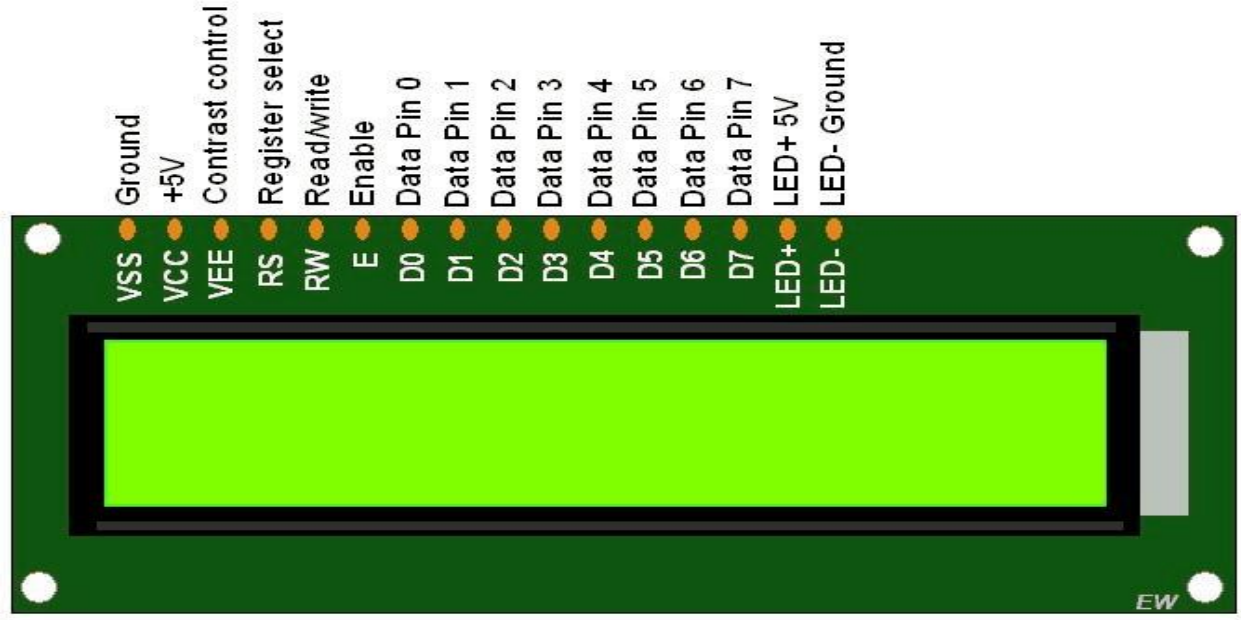


Figure 29 LCD

Interfacing LCD16x2 with AVR ATmega16/ATmega32 in 4-bit mode

LCDs (Liquid Crystal Displays) are used for displaying status or parameters in embedded systems.

LCD 16x2 is a 16-pin device which has 8 data pins (D0-D7) and 3 control pins (RS, RW, EN). The remaining 5 pins are for supply and backlight for the LCD.

The control pins help us configure the LCD in command mode or data mode. They also help configure read mode or write mode and when to read or write.

LCD 16x2 can be used in 4-bit mode or 8-bit mode depending on the requirement of the application. To use it, we need to send certain commands to the LCD in command mode and once the LCD is configured according to our need, we can send the required data in data mode.

2.5.1Features of 16×2 LCD module

- ***Operating Voltage is 4.7V to 5.3V***
- ***Current consumption is 1mA without backlight.***
- ***Alphanumeric LCD display module, meaning can display alphabets and numbers.***

- ***Consists of two rows and each row can print 16 characters.***
- ***Each character is built by a 5×8-pixel box.***
- ***Can work on both 8-bit and 4-bit mode.***
- ***It can also display any custom generated characters.***
- ***Available in Green and Blue Backlight***

2.5.2 Registers of LCD

A 16×2 LCD has two registers like data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is '0', then it is known as command register. Similarly, when the register set is '1', then it is known as data register.

Command Register

The main function of the command register is to store the instructions of command which are given to the display. So that predefined task can be performed such as clearing the display, initializing, set the cursor place, and display control. Here commands processing can occur within the register.

Data Register

The main function of the data register is to store the information which is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information which is to be exhibited on the screen of LCD. Whenever we send the information to LCD, it transmits to the data register, and then the process will be starting there. When register set =1, then the data register will be selected.

2.6 Micro SD Card Module

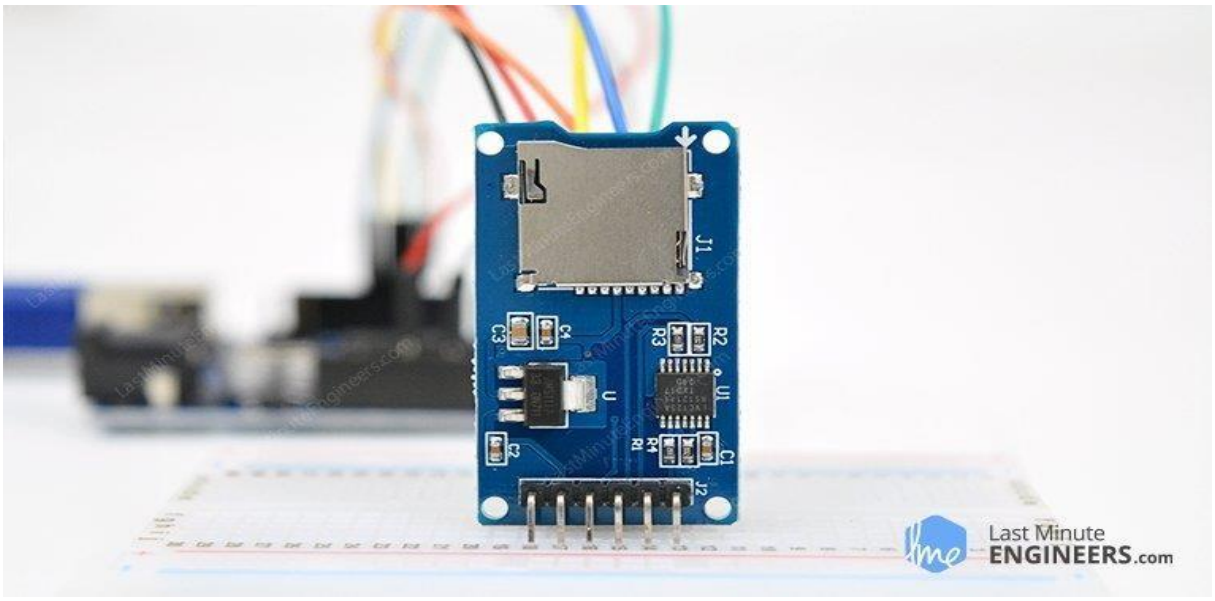


Figure 30 sd card module

The solution is to use what is found in every digital camera and mp3 player: Flash Cards! often called SD or micro-SD cards. Their ability to pack Gigabytes of data into a space smaller than a coin made them indispensable thing in our life.

2.6.1 Hardware Overview

The Micro-SD card module contains two main components that make it easy:

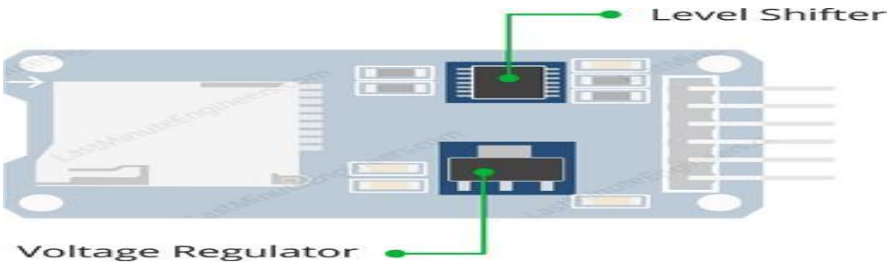


Figure 31 component of SD card

There are actually two ways to interface with micro-SD cards – SPI mode and SDIO mode. SDIO mode is way faster and is used in mobile phones, digital cameras etc. But it is more complex and requires signing non-disclosure documents. Instead, every SD card module is based on 'lower speed & less overhead' SPI mode that is easy for any microcontroller to use.

2.6.2 Micro SD Card Module Pinout

The micro-SD card module is fairly simple to connect. It has six pins:

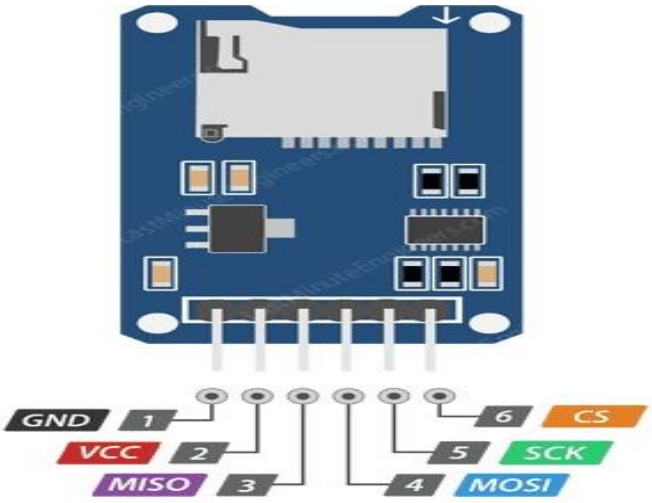


Figure 32 : micro sd card pinout

VCC pin supplies power for the module and should be connected to 5V pin on the Master.

GND should be connected to the ground of Master.

MISO (Master in Slave Out) is SPI output from the Micro SD Card Module.

MOSI (Master Out Slave In) is SPI input to the Micro SD Card Module.

SCK (Serial Clock) pin accepts clock pulses which synchronize data transmission generated by Master.

SS (Slave Select) pin is used by (Master) to enable and disable specific devices on SPI bus.

2.7 SD CARD

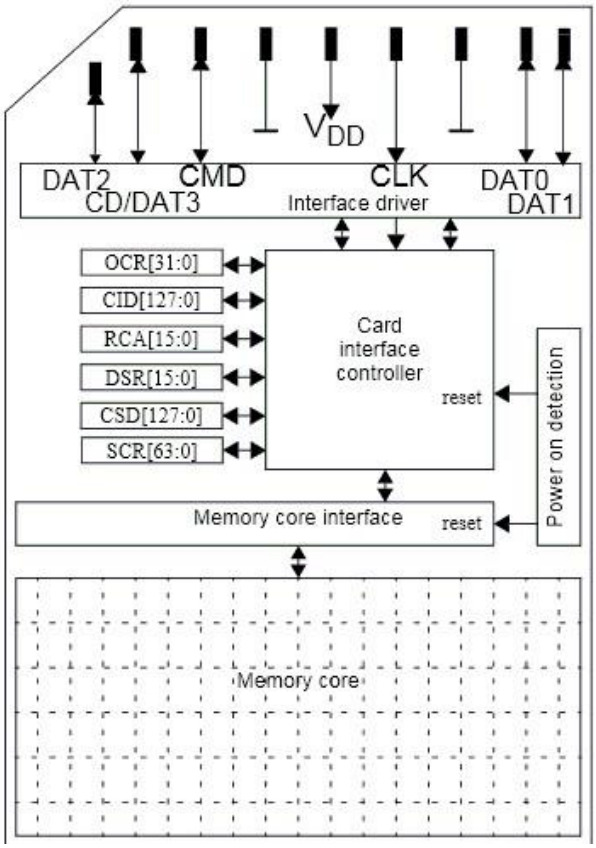


Figure 33 SD CARD

The SD card is consisting of two basic semiconductor sections, a 'memory core' and a 'SD card controller'.

***The 'memory** core' is the flash memory region where the actual data of the file is saved. When we format the SD card a file system will be written into this region*

***The capacity of the 'memory core'** is referred to as the size of the SD card.*

The 'SD card controller' helps to communicate the with external devices like microcontrollers. It can respond to certain set of standard SD commands and read or write data from the memory core in for the external device.

2.8 Speaker

Speakers are one of the most common output devices used with computer systems. Some speakers are designed to work specifically with computers, while others can be hooked up to any type of sound system. Regardless of their design, the purpose of speakers is to produce audio output that can be heard by the listener.

Speakers are transducers that convert electromagnetic waves into sound waves. The speakers receive audio input from a device such as a computer or an audio receiver. This input may be either in analog or digital form. Analog speakers simply amplify the analog electromagnetic waves into sound waves. Since sound waves are produced in analog form, digital speakers must first convert the digital input to an analog signal, then generate the sound waves.

Here are some of the important factors that must be considered in the design of an effective transducer this include efficiency, frequency response, linearity of the amplitude response, size, and cost:

- ***Would have an electroacoustic efficiency approaching 100%.***
- ***Would have an output response that is independent of frequency over the entire audible range.***
- ***Would introduce neither harmonic nor inter-modulation distortion into its output.***
- ***Would small size.***

Figure 34 speaker

Chapter 3: Software and Algorithm

1.driver of project

❖ ***ATmega32 is an 8-bit AVR microcontroller that takes input from flex sensors and then computes the output by using the algorithm which is stored inside the microcontroller.***

To program microcontroller we need to learn:

- ***Embedded C***
- ***Microcontroller interface***

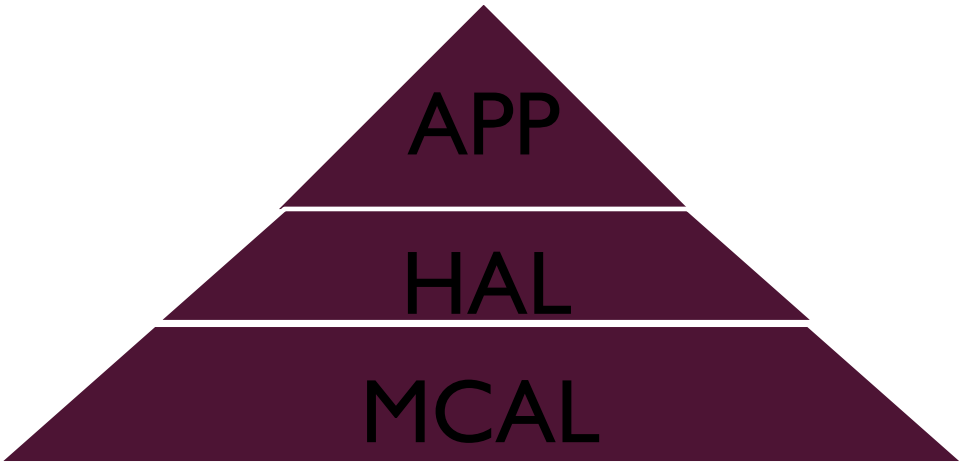


Figure 34 layer of embedded system

Every layer consists of number of drivers, each driver preforms specific operation.

1. *MCAL (Microcontroller Abstraction Layer):*
   - ***SPI driver used for SD Card Module***
   - ***GPIO driver used for LCD, Flex sensors and speaker.***
   - ***DAC driver used for speaker.***
   - ***USART driver used for GSM Module.***
   - ***Timer0***

2. ***HAL (Hardware Abstraction Layer):***

- ***drivers for LCD, GSM, SD Card***

Receiver Section:

This section is only for the display the message or the operation of different devices through the gesture. RF Receiver receives the command, and this command is catch by the microcontroller controls the actions of this sections.

LCD (16x2) is used for the displaying the messages or requirement of the patients at different places.

2. GPIO Driver

The General-Purpose Input/output Controller (GPIO) controls the I/O pins of the microcontroller. Each GPIO pin may be used as a general-purpose I/O or be assigned to a function of an embedded peripheral.

GPIO stands for General Purpose Input/Output. It is a standard interface used to connect microcontrollers to other electronic devices. For example, it can be used with sensors, diodes, displays, and System-on-Chip modules.

GPIO Register

All the GPIOs present in the microcontroller are grouped as Port X where X is A, B, C, D

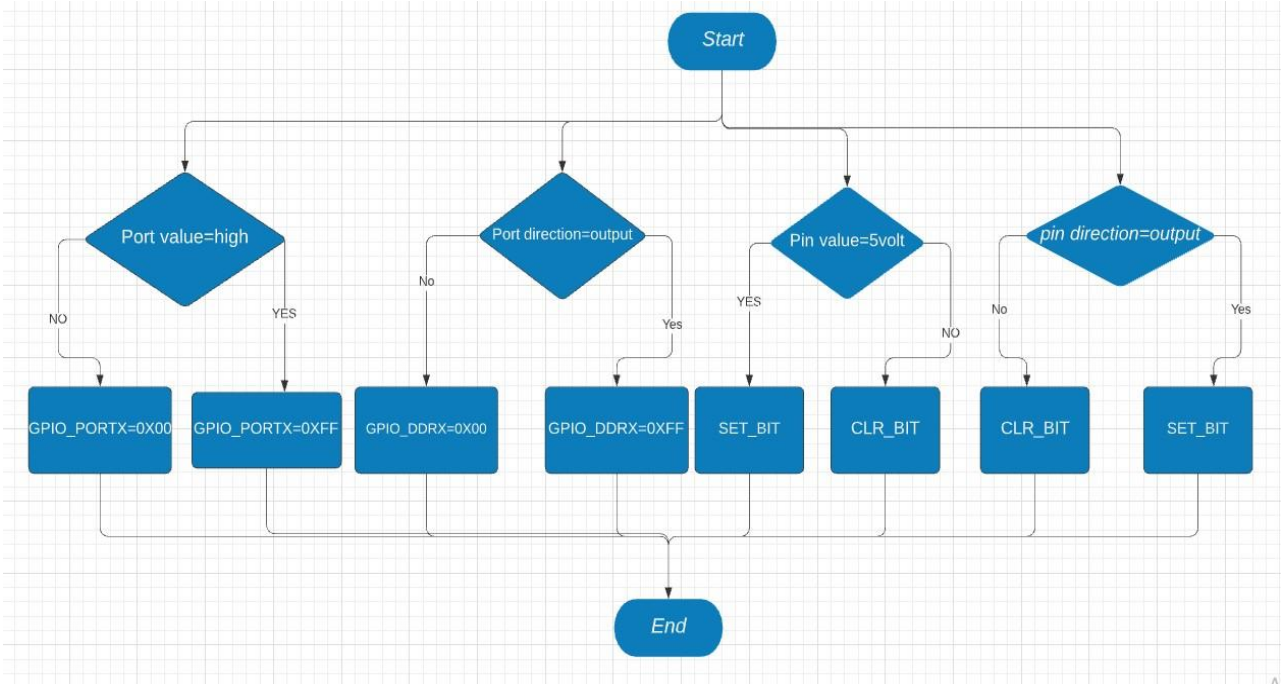


Figure 35 flow chart of GPIO

Function Definitions

- ***Interrupt Support***

The GPIO can be configured to generate an interrupt when it detects a change on a GPIO pin.

- ***Function GPIO set_pin_callback()***

Set call back for given GPIO pin.

- ***Function GPIO enable_pin_interrupt()***

Enable the interrupt of a pin.

- ***Function GPIO _disable_pin_interrupt()***

Disable the interrupt of a pin.

- ***Function GPIO _get_pin_interrupt_flag()***

Get the interrupt flag of a pin.

- ***Function GPIO _clear_pin_interrupt_flag()***

Clear the interrupt flag of a pin.

This driver has the following dependencies:

- ***Power Management:***

If the CPU enters a sleep mode that disables clocks used by the GPIO, the GPIO will stop functioning and resume operation after the system wakes up from sleep mode., the peripheral will be able to control the GPIO pin even if the GPIO clock is stopped.

Clocks: The GPIO is connected to a Peripheral Bus clock (CLK_GPIO). This clock is generated by the Power Manager. CLK_GPIO is enabled at reset and can be disabled by writing to the Power Manager.

- ***Interrupts***

The GPIO interrupt request lines are connected to the microcontroller.

- ***Debug Operation***

When an external debugger forces the CPU into debug mode, the GPIO continues normal operation. If the GPIO is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

3. Timer0 driver

In AVR ATmega32, Timer1 can be used as an input capture to detect and measure events happening outside the microcontroller.

Timers and counters are a very useful, and some may say required, timers indicate when an amount of time has elapsed. Timers can also count using the same internal registers. The difference between a timer and a counter is the source of the pulse that increments the counting register.

If the source is time-based, such as the oscillator clock that runs the microcontroller, then it is a timer.

Counting and timing become the basis for many other features can be completed with a Pulse Width Modulation (PWM)signal. That PWM signal is based on a timer. just need to know that "Operating frequency of the Microcontroller "and "Required delay" we will give the initial value to the timer register and start the timer. After starting the timer, we will continuously monitor the status of the timer register. If it overflows (reaches the maximum value) changes, that means the timer reaches its maximum value, therefore we will stop the timer.

$$Time=1/frequency$$

there are three timers:

- ***Timer0: 8-bit timer.***
- ***Timer1: 16-bit timer.***
- ***Timer2: 8-bit timer***

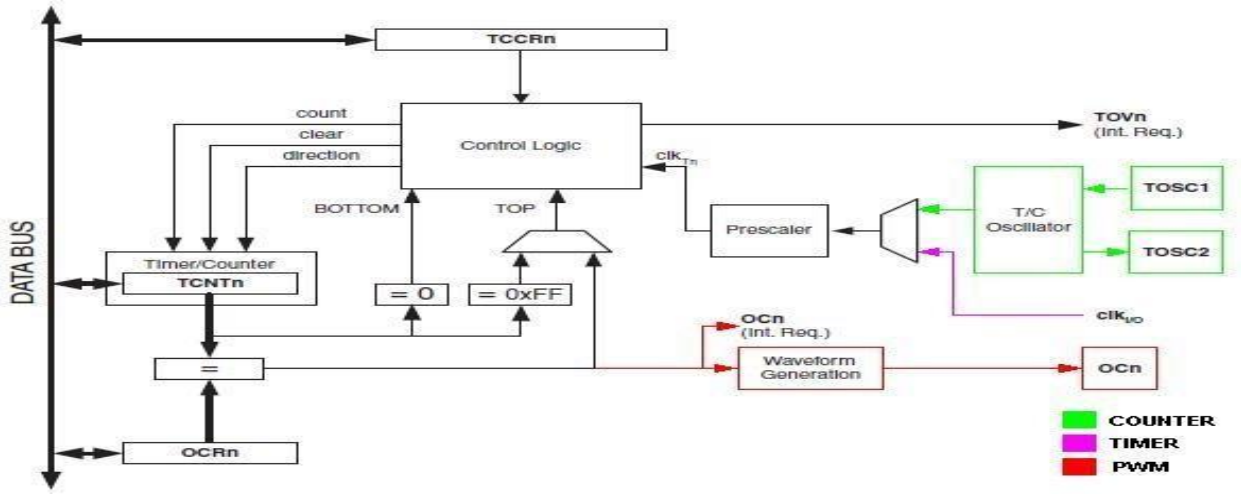


Figure 36 8-bit Timer/Counter Block Diagram

block diagram of the Timer0 module

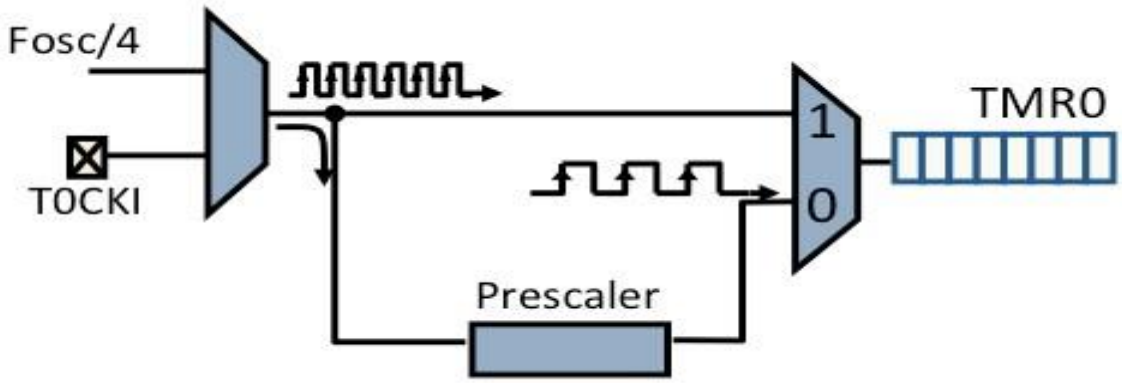


Figure 37 block diagram of the Timer0 module

It has all the following features of timer0:

- ***8-bit timer/counter register (TMR0)***
- ***8-bit Prescaler (independent of Watchdog Timer)***
- ***Programmable internal or external clock source***

- ***Programmable external clock edge selection***
- ***Interrupt on overflow***
- ***TMR0 can be used to gate Timer1.***

the basic registers of the Timer0

1. ***TCNT0: Timer / Counter Register***

It is an 8-bit register. It counts up with each pulse.

2. ***TCCR0: Timer / Counter Control register 0***

This is an 8-bit register used for the operation mode and the clock source selection.

7	6	5	4	3	2	1	0
FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

3.TIFR: Timer Counter Interrupt Flag register

7	6	5	4	3	2	1	0
OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0

4.TIMSK: Timer / Counter Interrupt Mask Register

7	6	5	4	3	2	1	0
OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0

We have to set the TOIE0 (Timer0 Overflow Interrupt Enable) bit in the TIMSK register to set the timer0 interrupt so that as soon as the Timer0 overflows, the controller jumps to the Timer0 interrupt routine.

3.1 Steps to Program Delay using Timer0.

1. ***Load the TCNT0 register with the initial value (let us take 0x25).***
2. ***For normal mode and the pre-scaler option of the clock, set the value in the TCCR0 register. As soon as the clock Prescaler value gets selected, the timer/counter starts to count, and each clock tick causes the value of the timer/counter to increment by 1.***

3.Timer keeps counting up, so keep monitoring for timer overflow i.e., TOV0 (Timer0 Overflow) flag to see if it is raised.

4. ***Stop the timer by putting 0 in the TCCR0 i.e., the clock source will get disconnected and the timer/counter will get stopped.***
5. ***Clear the TOV0 flag. Note that we must write 1 to the TOV0 bit to clear the flag.***
6. ***Return to the main function.***

4. TIMER 2 driver

TIMER2 is an 8-bit timer (like TIMER0), most of the registers are like that of TIMER0 registers. Apart from that, TIMER2 offers a special feature which other timers do not – Asynchronous Operation.

Methodology – Using Prescaler and Interrupt

we use a Prescaler of 256. For this, the overflow time is 4.096 Ms. Thus, the timer should overflow 12 times (MAX = 255) and count to 53 in the 13th iteration, and then reset the timer. The formula used is as follows:

$$Timer\ Count = \frac{Required\ Delay}{Clock\ Time\ Period} - 1$$

4.1 the TIMER2 registers.

TCCR2 Register

The Timer/Counter Control Register – TCCR2 is as follows:

Bit	7	6	5	4	3	2	1	0	
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	TCCR2
Read/Write	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	

TCCR2 Register

In the Timer/Counter Register – TCNT2, the value of the timer is stored. Since TIMER2 is an 8-bit timer, this register is 8 bits wide.

TCNT2 Register

In the Timer/Counter Register – TCNT2, the value of the timer is stored. Since TIMER2 is an 8-bit timer, this register is 8 bits wide.

Bit	7	6	5	4	3	2	1	0	
	TCNT2[7:0]								TCNT2
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	

TCNT2 Register

TIMSK Register

*The **Timer/Counter Interrupt Mask** – TIMSK Register is as follows. It is a register common to all the timers.*

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
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	

TIMSK Register

*Here we are concerned with the **6th bit – TOIE2 – Timer/Counter2 Overflow Interrupt Enable**. We set this to '1' in order to enable overflow interrupts.*

TIFR Register

*The **Timer/Counter Interrupt Flag Register** – TIFR is as follows. It is a register common to all the timers.*

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
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	

TIFR Register

*Here we are concerned with the **6th bit – TOV2 – Timer/Counter2 Overflow Flag**. This bit is set (one) whenever the timer overflows. It is cleared automatically whenever the corresponding Interrupt Service Routine (ISR) is executed. Alternatively, we can clear it by writing '1' to it.*

4.2 The timer in Fast PWM mode

In Fast PWM mode, you usually work with the pins associated with the timer. This is important for the

• Timer0: OC0A (=PD6, Arduino Pin 6) / OC0B (=PD5, Arduino Pin 5) • Timer2: OC2A (=PB3, Arduino Pin 11) / OC2B (=PD3, Arduino Pin 3) The PWM mode works in Mode 3 with a timer overflow after 255 (0xFF). In Mode 7, PWM mode works with a Compare Match. Top is the value stored in OCRxA. As an example, let's take Mode 7. The target is a square wave signal with a frequency of 1 kHz at OC2B. Per period, the signal should be HIGH 20% of the time and 80% LOW. Another expression for this is: the duty cycle is 20%.

4.3 Fast PWM on OC2B

We set the COM2B1 bit for this purpose. According to the table, this means: "clear OC2B at Compare Match, set OC2B at BOTTOM". The Compare Match refers to the value stored in OCR2B. Graphically, it looks like this:

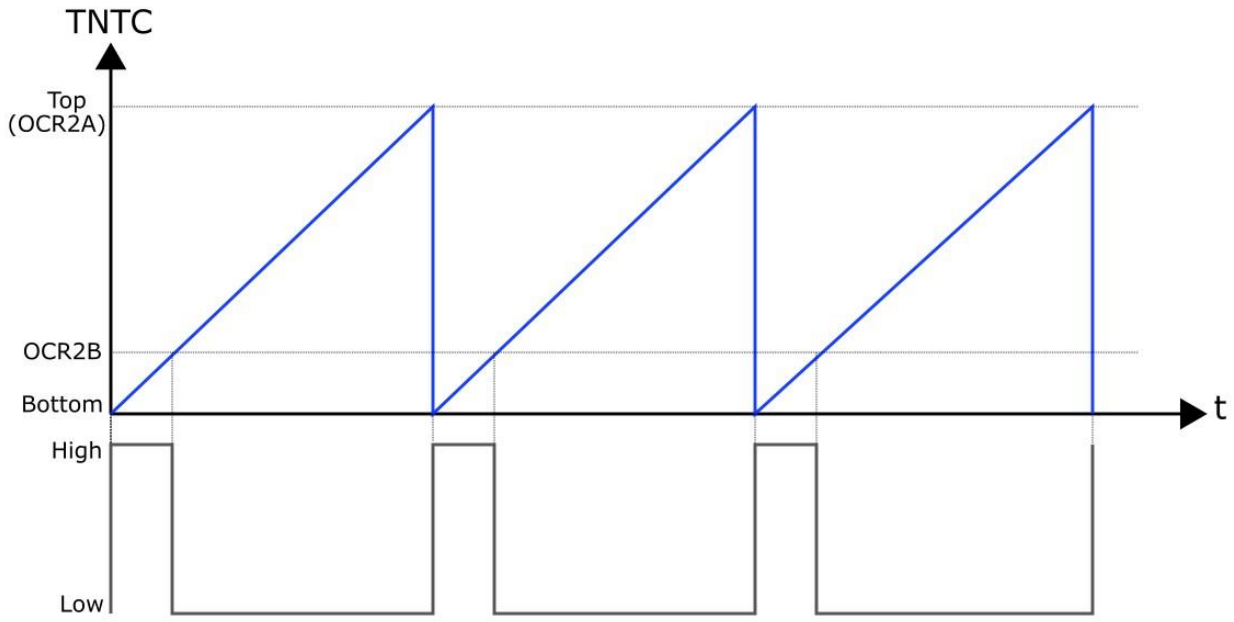


Figure 38 PWM signal on OC2B

First, we must do some calculations again. We need the top value for the frequency and the value for OCR2B for the duty cycle. Because of the high frequency, we don't need a scale factor. The following applies:

$$f_{desired} = \frac{system\_clock}{\text{prescaler} \cdot (1 + \text{Top})}$$

$$Top = \frac{16000}{prescaler} - 1 \text{ with clock} = 16000000 \text{ and } f = 1000$$

A presale of 64 results in 249 for top. That is 250 steps. One-fifth of it is 50. Since the counter starts at 0, OCR2B is 49, at least theoretically. In practice, you must try it out. I have hit the desired signal better with the pairing 249 / 50. Of course, it also depends on how exactly the microcontroller clocks.

5. LCD Code

Used timer 0 in the lcd code.

Programming LCD16x2 4-bit mode with AVR

ATmega16/Atmega32 Initialization

1. ***Wait for 15ms, Power-on initialization time for LCD16x2.***
2. ***Send 0x02 command which initializes LCD 16x2 in 4-bit mode.***
3. ***Send 0x28 command which configures LCD in 2-line, 4-bit mode, and 5x8 dots.***
4. ***Send any Display ON command (0x0E, 0x0C)***
5. ***Send 0x06 command (increment cursor)***

Command writes function.

1. ***First, send a higher nibble of command.***
2. ***Make RS pin low, RS=0 (command reg.)***
3. ***Make RW pin low, RW=0 (write operation) or connect it to ground.***
4. ***Give High to Low pulse at Enable (E).***
5. ***Send lower nibble of command.***
6. ***Give High to Low pulse at Enable (E).***

Data write function.

1. ***First, send a higher nibble of data.***
2. ***Make RS pin high, RS=1 (data reg.)***
3. ***Make RW pin low, RW=0 (write operation) or connect it to ground.***
4. ***Give High to Low pulse at Enable (E).***
5. ***Send lower nibble of data.***
6. ***Give High to Low pulse at Enable (E).***

5.1 Simulation

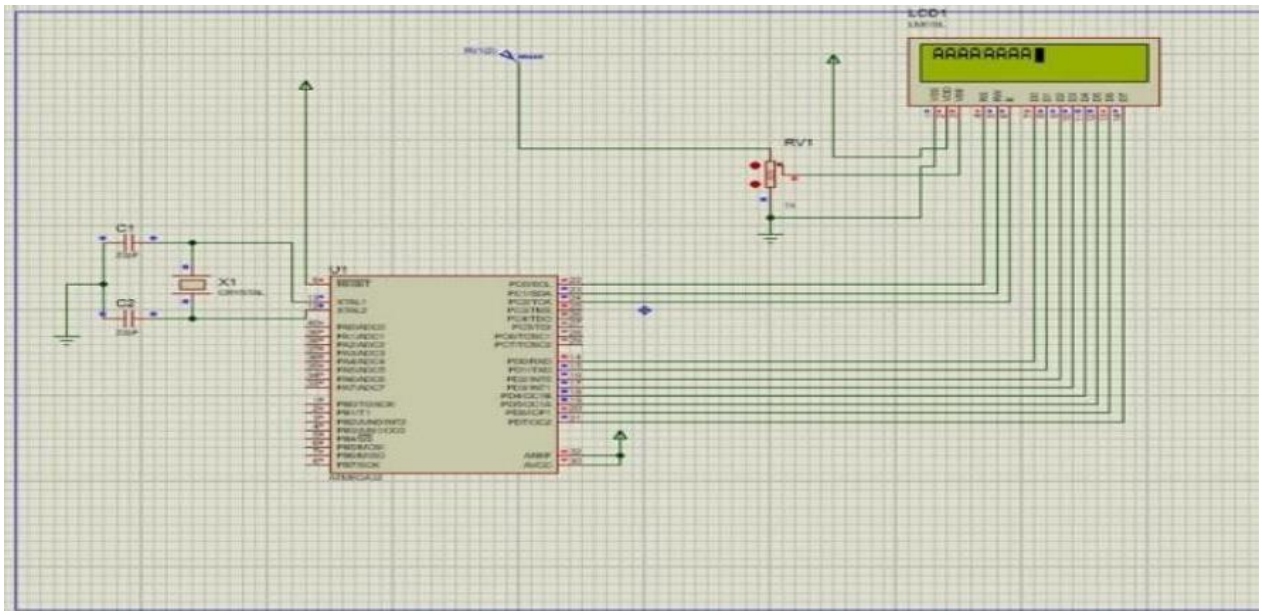


Figure 39 SIMULATION OF LCD

6. UART Communication driver

The SIM900 GSM/GPRS shield uses UART protocol to communicate with a microcontroller. The chip supports baud rate as default of 9600bps or 115200bps with Auto-Baud detection

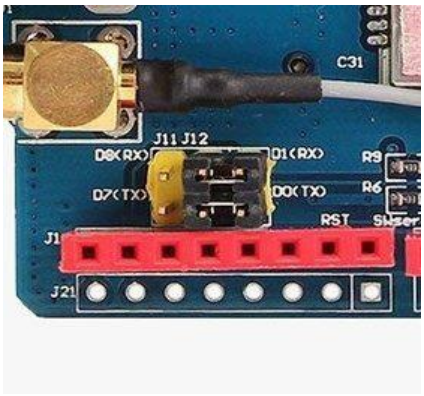


Figure 40 The SIM900 GSM/GPRS

With the help of jumpers we can connect (RX, TX) of the shield to either Software Serial (D8, D7) or Hardware Serial(D1,D0) of the controller.

RxD (Receiver) pin is used for serial communication.

TxD (Transmitter) pin is used for serial communication.

GND is the Ground Pin and needs to be connected to GND pin on the controller.

Working principles:

GSM is working with AT commands:

AT – It is the most basic AT command. It also initializes Auto-badger. If it works you should see the AT characters echo and then OK, telling you it is OK and it's understanding you correctly!

Commands to test the module:

AT+CSQ – Check the 'signal strength' – the first # is dB strength, it should be higher than around 5. Higher is better. Of course, it depends on your antenna and location!

AT+CREG? Check that you are registered on the network. The second # should be 1 or 5. 1 indicates you are registered to home network and 5 indicates roaming network. Other than these two numbers indicate you are not registered to any network.

The Target Command which will use in project to sending SMS is:

- ***AT+CMGS="+ZZxxxxxxxxxx" – Sends SMS to the phone number specified. The text message entered followed by a 'Ctrl+z' character is treated as SMS.***

- ***By replacing ZZ with country code and "xxxxxxxxxx" by the phone number***

7. GSM module driver

So that GSM can communicate with the Microcontroller Through communication protocol called (UART)

7.1 Working Principle

The transmit pin of first UART is connected to second device's receive pin and receive pin to that of transmit pin. So, data flows from transmit pin to the receive pin of the receiving UART.

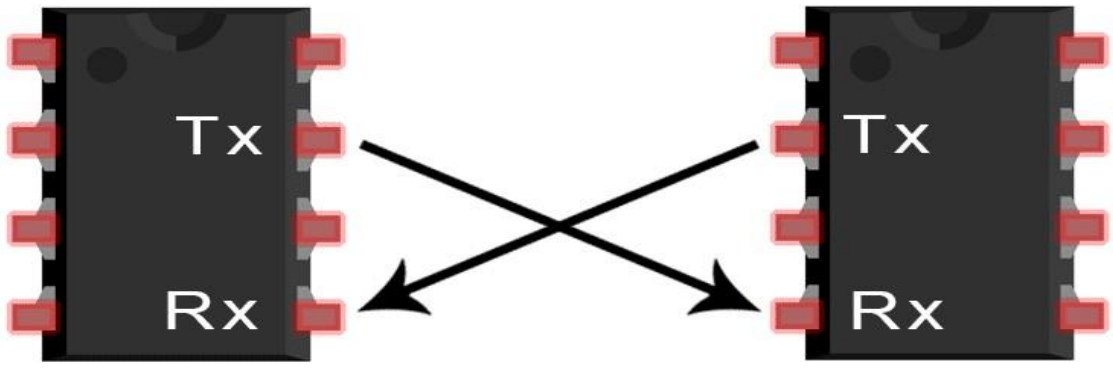
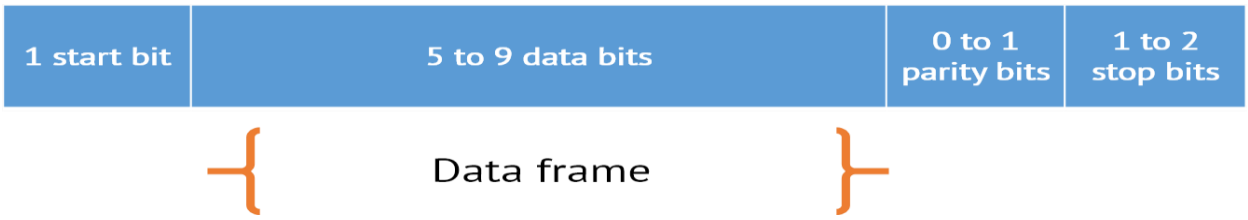


Figure 41 TX and RX

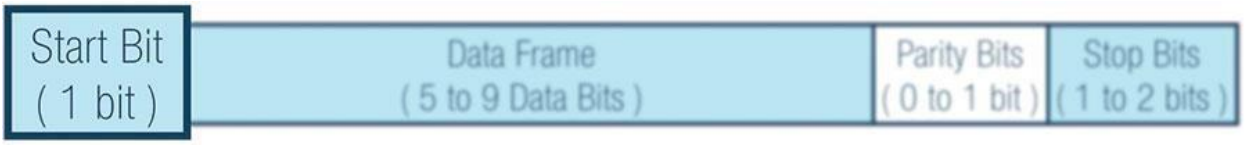
2-Baud rate = 9600bps.

7.2 Data Framing

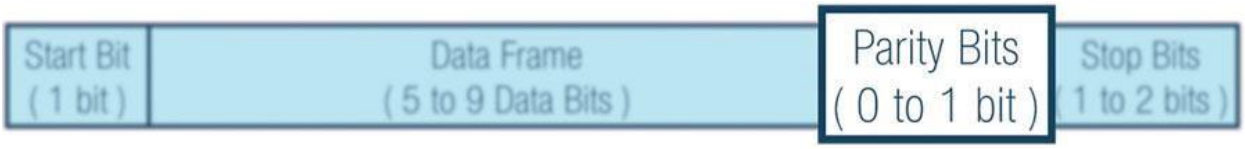
UART transmits data in packets. Each data packet may contain 1 start bit, 5 to 9 data bits, an optional parity bit and 1 or 2 stop bits.



Start Bit



Parity



Stop Bits

To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for one (1) to two (2) bit(s) duration.

7.3 Modes of UART

- ***Asynchronous normal mode (no clock line)***
- ***Asynchronous double mode (no clock line)***
- ***Synchronous mode (we need clock line)***

Advantages

- ***Error checking using parity bits.***
- ***Variable data packet.***
- ***Uses only two wires for data transmission and no clock***

7.4 Implementation of UART

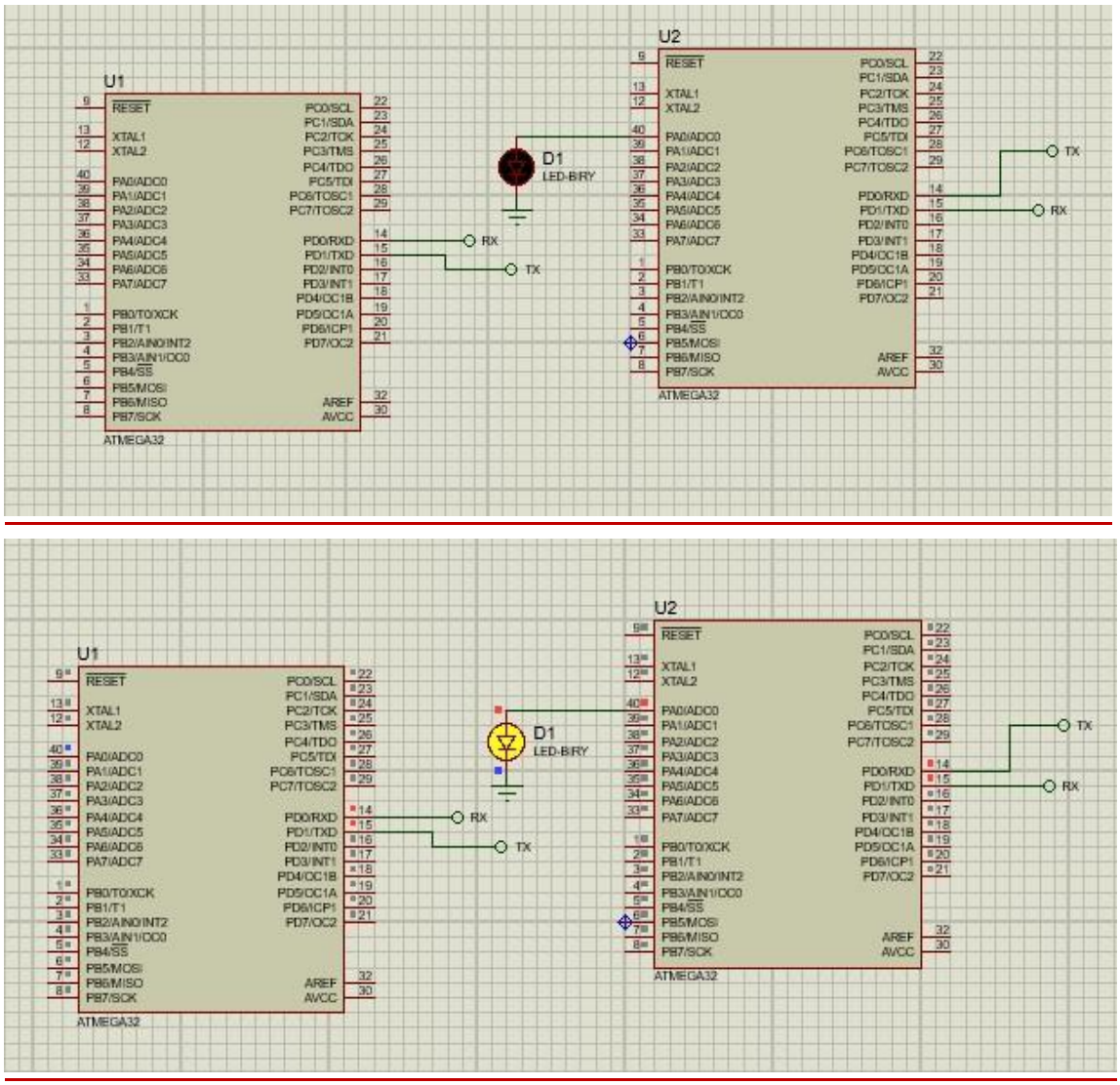


Figure 42 Implementation of UART

8. Flex sensor interfacing with AVR Microcontroller

ADC channels

The ADC in ATmega32 has 8 channels that means you can take samples from eight different terminal. From ADC0 to ADC7 every sensor connects to one channel.

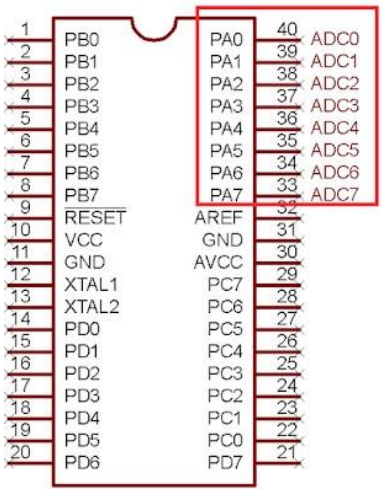


Figure 43 ADC channels

8.1 ADC Registers:

The ADC has only four registers.

1- ***ADC Multiplexer Selection Register – ADMUX.***

For selecting the reference voltage and the input channel.

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

2- ***ADC Control and Status Register A – ADCSRA.***

As the name says it has the status of ADC and is also use for controlling it.

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

3- ***The ADC Data Register – ADCL and ADCH.***

The result of conversion is here. ADLAR=0, the result is right Adjusted.

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	
R	R	R	R	R	R	R	R	
R	R	R	R	R	R	R	R	
0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	

8.2 ADC Prescaler:

The ADC needs a clock pulse to do its conversion. The ADC requires a frequency between 50KHz to 200KHz. At higher frequency, the conversion is fast while a lower frequency the conversion is more accurate. System clock can be divided by 2,4,16,32,64,128 by setting the Prescaler.

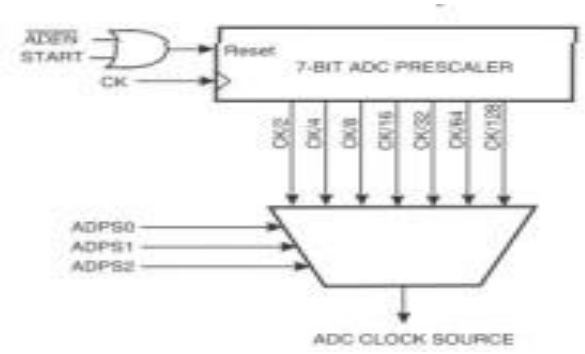


Figure 44 ADC Prescaler

A 10 Bit ADC has a range of 0-1023. (2^10=1024). The ADC also has a Reference voltage (ARef). When input voltage is GND (0V input) the output is 0 and when input voltage is equal to ARef the output is 1023.So the input range is from 0 up to ARef and digital output is from 0 up to 1023. Reference voltage Avref can be internal (2.5V) or external, from 0V up to 5V.

$$ADC = \frac{(Vin \times 1023)}{VRef}$$

8.3 Potential Divider Circuit

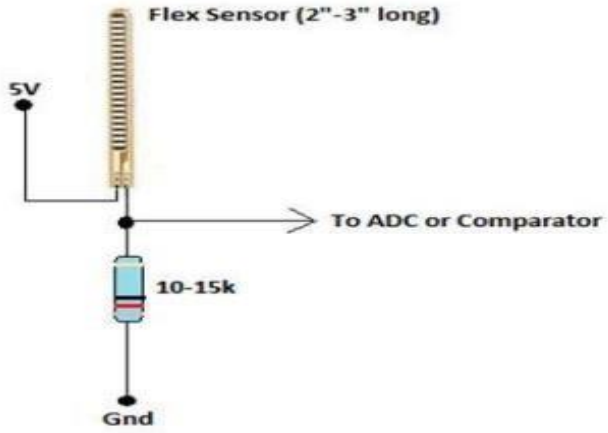


Figure 45 Potential Divider Circuit

8.4 Simulation

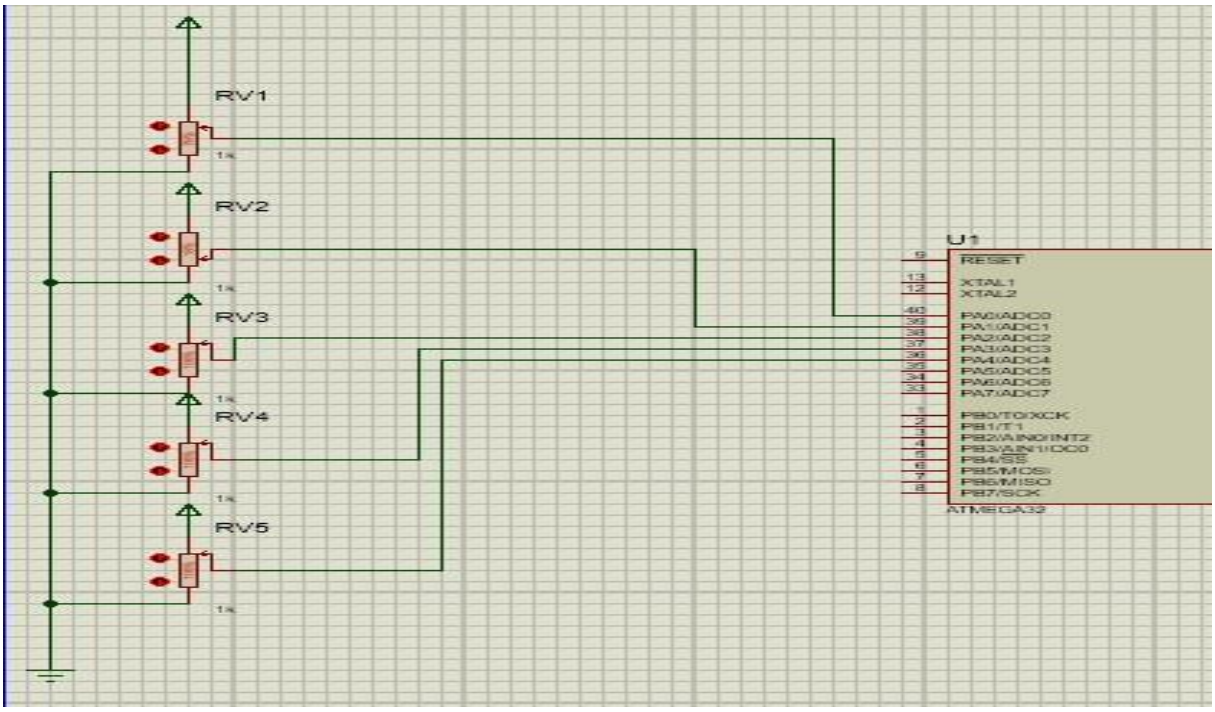


Figure 46 simulation of ADC

Note: Flex sensor as Variable resistor in the simulation.

9. SPI in AVR ATmega32

9.1 Introduction

The Serial Peripheral Interface (SPI) is a bus interface connection protocol originally started by Motorola Corp. It uses four pins for communication.

- ***SDI (Serial Data Input)***
- ***SDO (Serial Data Output)***
- ***SCLK (Serial Clock)***
- ***CS (Chip Select)***

It has two pins for data transfer called SDI (Serial Data Input) and SDO (Serial Data Output). SCLK (Serial Clock) pin is used to synchronize data transfer and Master provides this clock. CS (Chip Select) pin is used by the master to select the slave device.

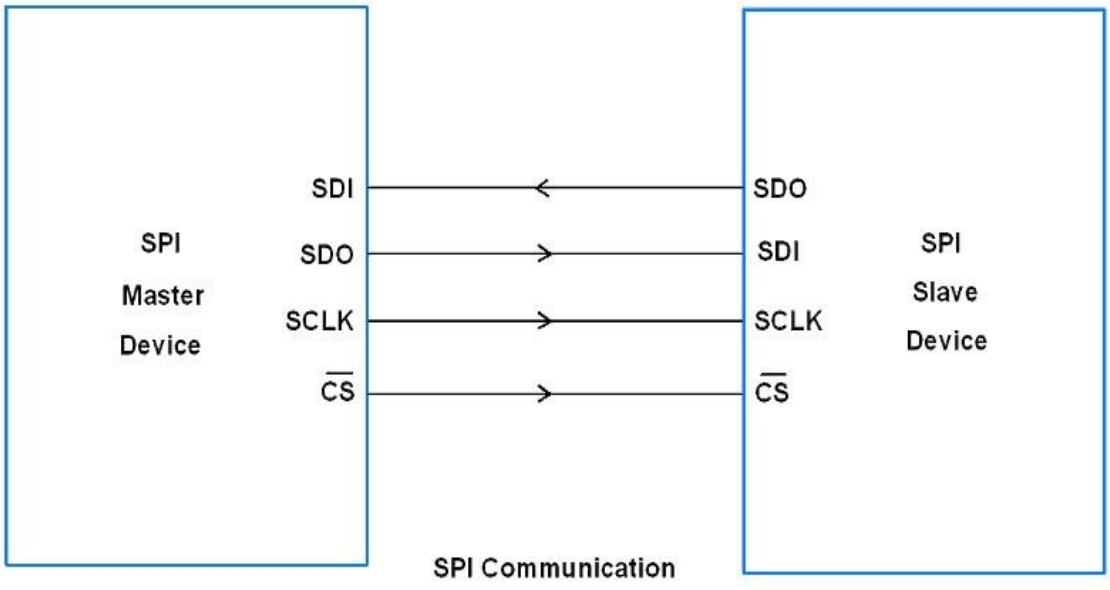


Figure 47 SPI communication

SPI devices have 8-bit shift registers to send and receive data. Whenever a master needs to send data, it places data on the shift register and generates a required clock. Whenever a master wants to read data, the slave places the data on the shift register and the master generates a required clock.

9.2 ATmega32 SPI Communication

ATmega16 has an inbuilt SPI module. It can act as a master and slave SPI device.

SPI communication pins in AVR ATmega32 are:

- ***MISO (Master in Slave Out)***

  The Master receives data, and the slave transmits data through this pin.
- ***MOSI (Master Out Slave In)***

  The Master transmits data, and the slave receives data through this pin.
- ***SCK (Shift Clock)***

  The only master can initiate a serial clock., which is used by the slave.
- ***SS (Slave Select)***

  Master can select slaves through this pin.

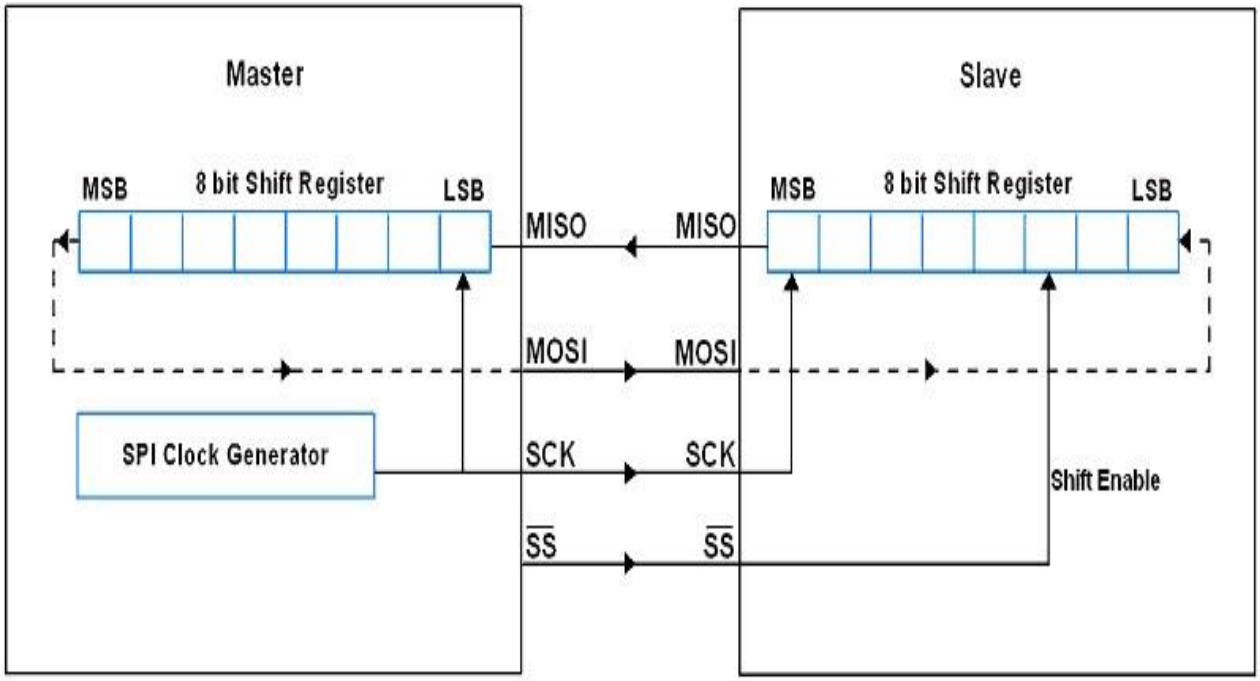


Figure 48 SPI master slave interconnection

Note: SPI is a full-duplex communication protocol. data on master and slave shift registers get interchanged at the same time.

9.3 ATmega32 SPI Control

AVR ATmega16 uses three registers to configure SPI communication that are SPI Control Register, SPI Status Register and SPI Data Register.

SPCR: SPI Control Register

Bit 7 – SPIE: SPI Interrupt Enable bit

Bit 6 – SPE: SPI Enable bit

Bit 5 – DORD: Data Order bit

Bit 4 – MSTR: Master/Slave Select bit

Bit 3 – CPOL: Clock Polarity Select bit

Bit 2 – CPHA: Clock Phase Select bit

Bit 1:0 – SPR1: SPR0 SPI Clock Rate Select bits

SPSR: SPI Status Register

Bit 7 – SPIF: SPI interrupt flag bit

This flag gets set when the serial transfer is complete.

Bit 6 – WCOL: Write Collision Flag bit

This bit gets set when SPI data register writes occur during previous data transfer.

Bit 0 – SPI2X: Double SPI Speed bit

When set, SPI speed (SCK Frequency) gets doubled.

SPDR: SPI Data Register

SPI Data register used to transfer data between the Register file and SPI Shift Register. Writing to the SPDR initiates data transmission.

9.4 ATmega32 SPI Point of views

When the device is in master mode

- ***Master writes data byte in SPDR. Writing to SPDR starts data transmission.***
- ***8-bit data starts shifting out towards slave and after the complete byte is shifted, SPI clock generator stops, and SPIF bit gets set.***

When the device is in slave mode

- ***THE Slave SPI interface remains in sleep if the SS pin is held high by the master.***
- ***It activates only when the SS pin is driven low. Data is shifted out with incoming SCK clock from master during a write operation.***
- ***SPIF is set after the complete shifting of a byte.***

SS pin functionality Master mode

- ***In master mode, the SS pin is used as a GPIO pin.***
- ***Make the SS pin direction as output to select a slave device using this pin.***
- ***Note that if the SS pin configured as input, then it must be set high for master operation.***
- ***If it is set as input in master mode and gets driven low by an external circuit, then the SPI system recognizes this as another master selecting SPI as a slave due to its active low behavior.***
- ***This will clear the MSTR bit in the SPCR register and SPI turns in slave mode.***

SS pin functionality Slave mode

- ***In slave mode, the SS pin is always configured as an input.***
- ***When it is low SPI activates.***
- ***And when it is driven high SPI logic gets reset and does not receive any incoming data.***

10. Interfacing SD Card with AVR

The aim of this layer is to read a file from the FAT32 file system of the SD card. The SD card interfaced with the microcontroller using serial 'SPI-buses'.

SD CARD

The SD card is consisting of two basic semiconductor sections, a 'memory core' and a 'SD card controller'.

The 'memory core'

is the flash memory region where the actual data of the file is saved. When we format the SD card a file system will be written into this region

The capacity of the 'memory core'

is referred to as the size of the SD card. The 'SD card controller' helps to communicate the with external devices like microcontrollers. It can respond to certain set of standard SD commands and read or write data from the memory core in for the external device.

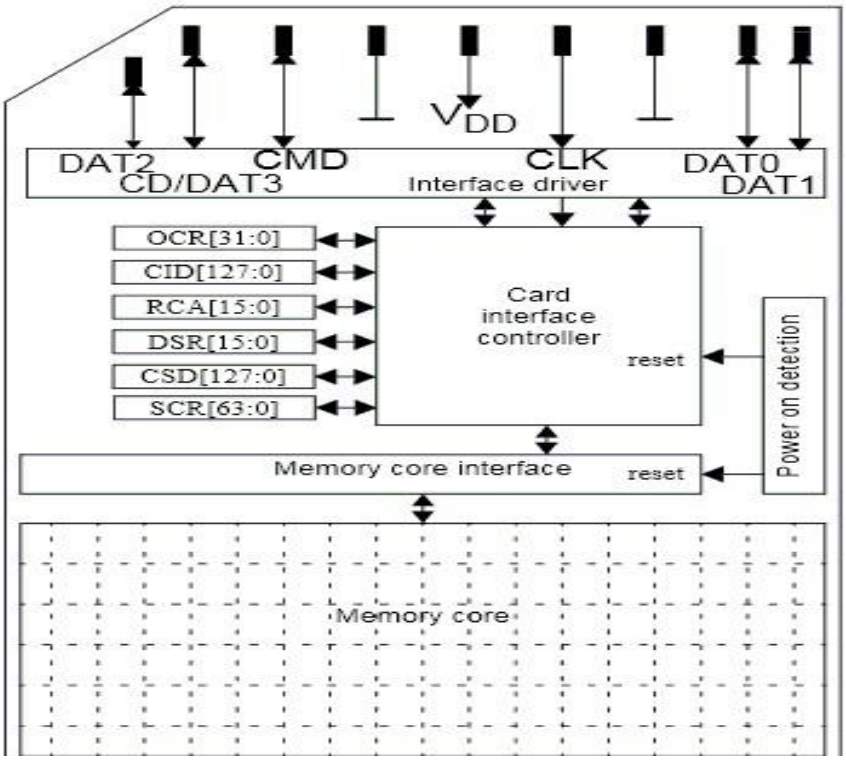


Figure 49 SD memory card architecture

10.1 SD CARD FUNCTIONAL LAYERS

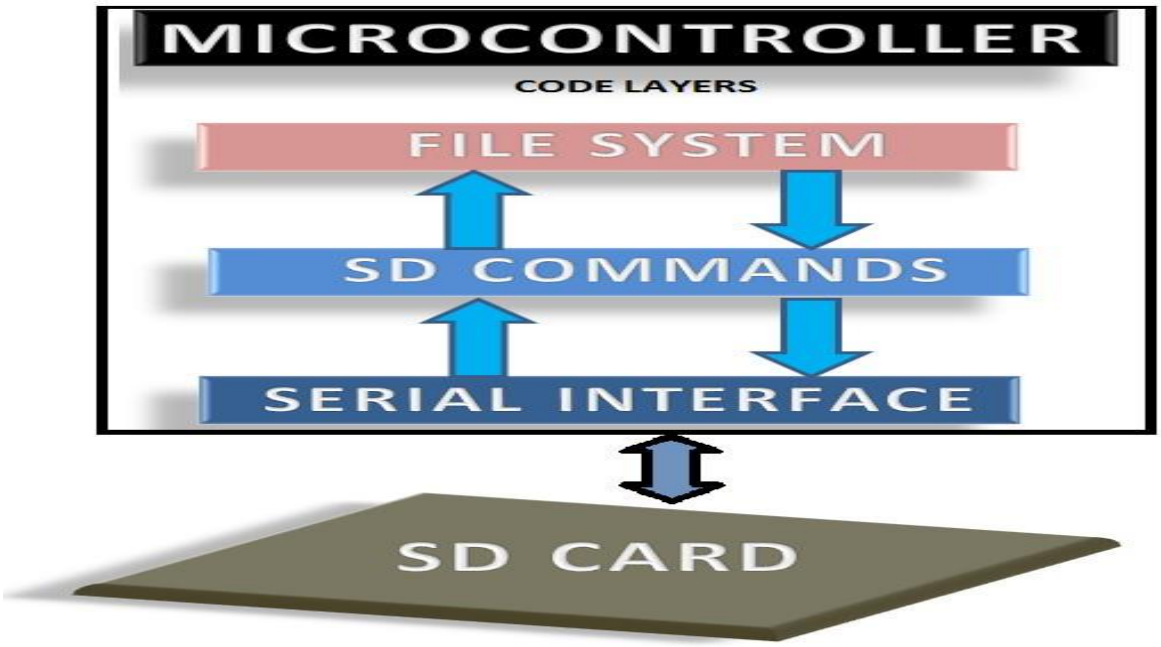


Figure 50 SD card functional layer

1) SERIAL INTERFACE LAYER

The microcontroller initiates all the data transfers. clock is also controlled. The microcontroller is free to choose between the SD cards by (Chip Select).

The data is transmitted from the microcontroller to the SD card using the MOSI (Master Output Slave Input) channel and the data is transferred by the SD card to the microcontroller using the MISO (Master Input Slave Output) channel.

The microcontroller used in this project runs on 5V power supply, but the SD card can take only up to 3.3V. This problem can be solved by using a bi-directional level converter introduced between the SPI pins of the microcontroller and the SD card.

2) SD COMMANDS LAYER

The SD card accepts only a set of standard SD commands. microcontroller can read the registers of the SD card, and read/write the 'Memory Core'.

COMMANDS FOR INITIALIZING THE MEMORY CARD

Before the memory card can respond to these commands, the memory card should be initialized in SPI mode. Certain commands initialize the SD card. SD interfacing mode on reset. Hence the first command send to the SD card should have the correct CRC byte included.

COMMANDS FOR READING DATA

The data can be read from the 'Memory Core' of the SD card using the commands given below.

READ_SINGLE_BLOCK – Read data from a single block.

READ_MULTIPLE_BLOCK – Read data from multiple blocks.

READ_SINGLE_BLOCK

3) SYSTEM FILE LAYER

The FAT32 file system is written into the 'Memory Core' when it was formatted. The FAT32 stands for File Allocation Table 32, means it has a file allocation table of length 32 bits.
The entire data of a file is scrambled across the Memory Core and the FAT (File Allocation Table) holds the location of next block corresponding to the location of the current block. The format of a FAT32 file system is as shown below:

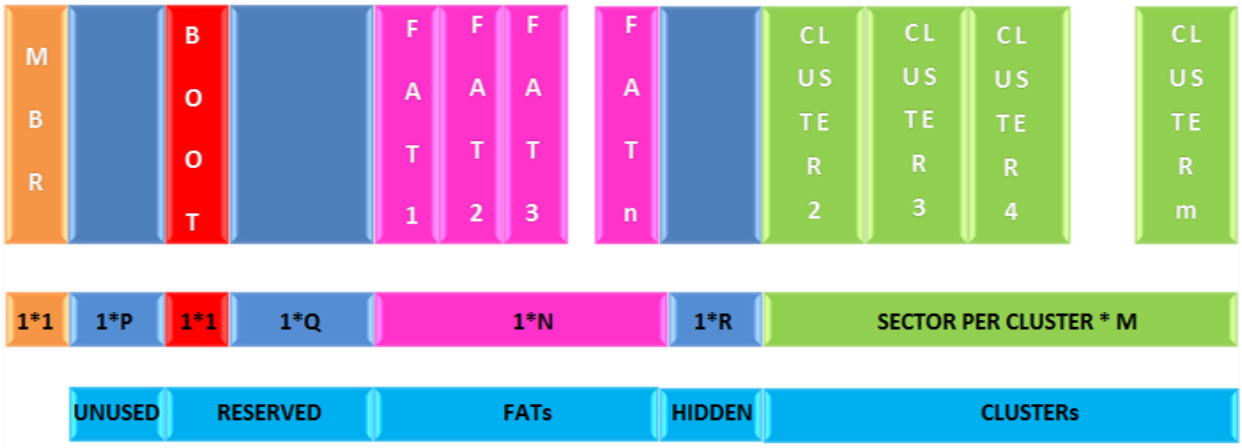


Figure 51 FAT32 file system

11. FAT32 Driver

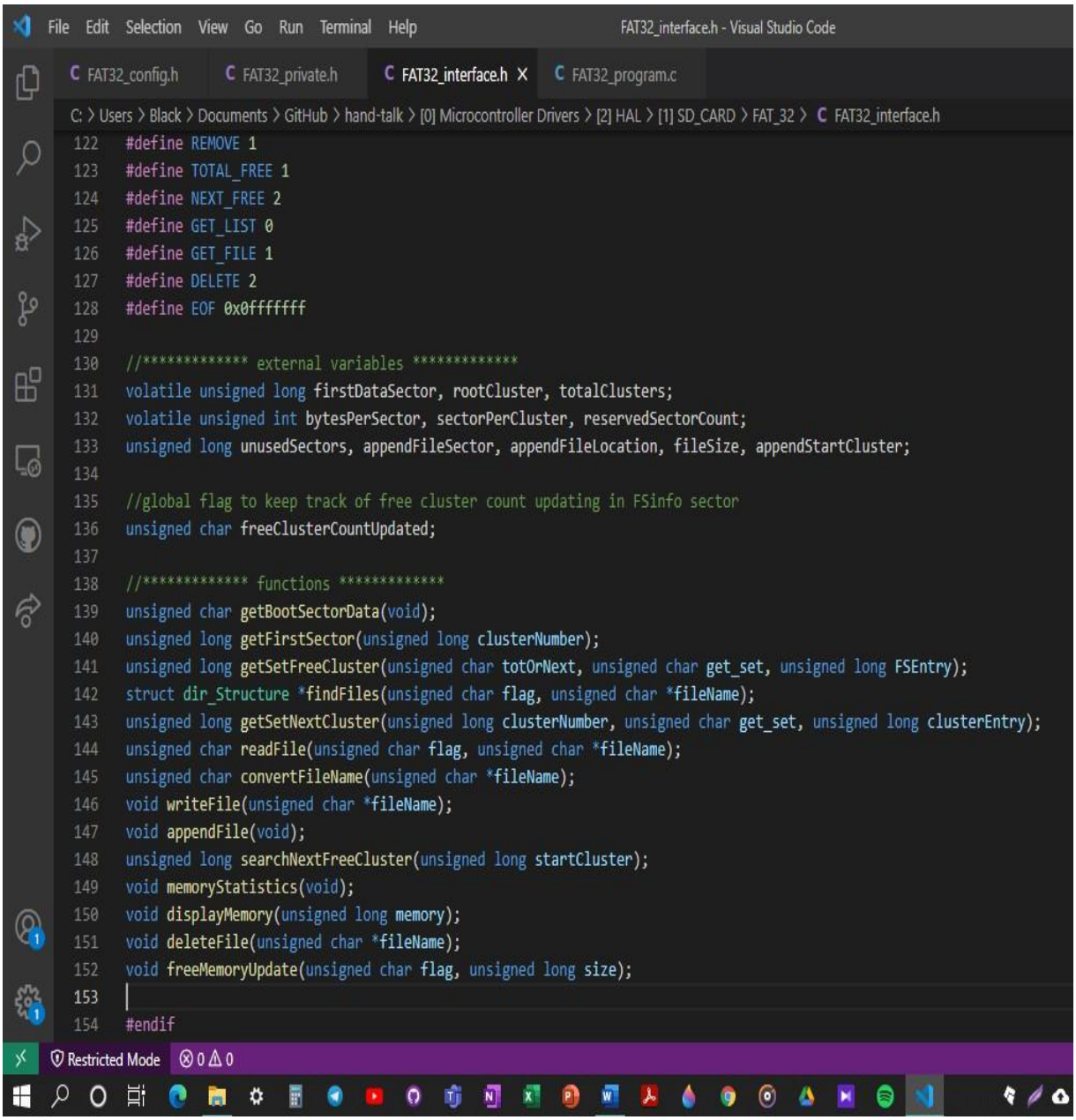


Figure 52 FAT32 Driver

12. I2C Communication

I2C communication is the short form for inter-integrated circuits. It is a communication protocol developed by Philips Semiconductors for the transfer of data between a central processor and multiple ICs on the same circuit board using just two common wires.

12.1 The physical I2C Bus

I2C Bus (Interface wires) consists of just two wires and are named as Serial Clock Line (SCL) and Serial Data Line (SDA). The data to be transferred is sent through the SDA wire and is synchronized with the clock signal from SCL. All the devices/ICs on the I2C network are connected to the same SCL and SDA lines as shown below:

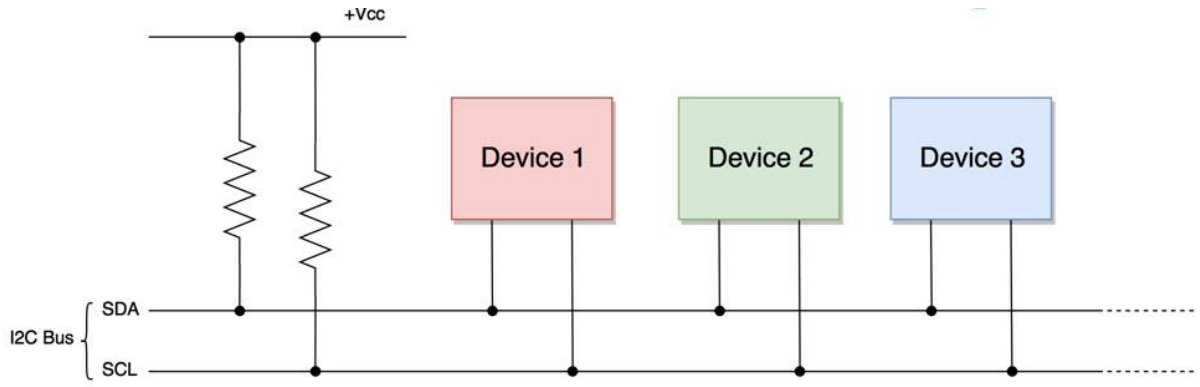


Figure 53 i2c Bus

The I2C bus lines (SDA, SCL) are operated as open drain drivers. It means that any device/IC on the I2C network can drive SDA and SCL low, but they cannot drive them high. So, a pull up resistor is used for each bus line, to keep them high (at positive voltage) by default.

The reason for using an open-drain system is that there will be no chances of shorting, which might happen when one device tries to pull the line high, and some other device tries to pull the line low.

12.2Data Transfer Protocol

The following protocol (set of rules) is followed by master device and slave devices for the transfer of data between them.

Data is transferred between the master device and slave devices through a single SDA data line, via patterned sequences of 0's and 1's (bits). Each sequence of 0's and 1's is termed as a transaction and the data in each transaction is structured as below:

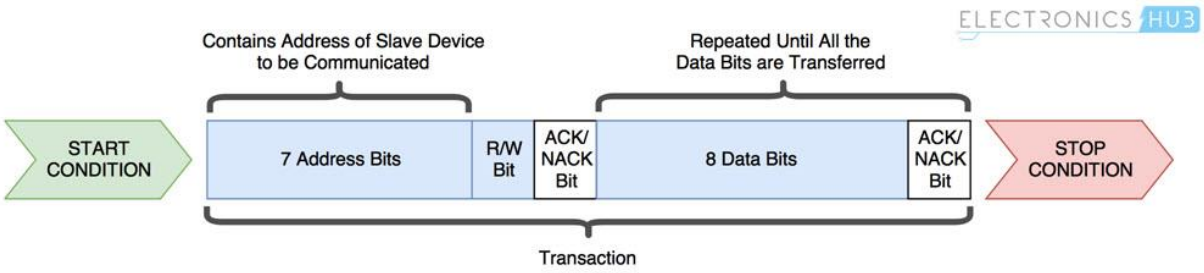


Figure 54 data transfer protocol

Start Condition

Whenever a master device/IC decides to start a transaction, it switches the SDA line from high voltage level to a low voltage level before the SCL line switches from high to low.Once a start condition is sent by the master device, all the slave devices get active even if they are in sleep mode, and wait for the address bits.

Address Block

It comprises of 7 bits and are filled with the address of slave device to/from which the master device needs send/receive data. All the slave devices on the I2C bus compare these address bits with their address.

Read/Write Bit

This bit specifies the direction of data transfer. If the master device/IC need to send data to a slave device, this bit is set to '0'. If the master IC needs to receive data from the slave device, it is set to '1'.

ACK/NACK Bit

It stands for Acknowledged/Not-Acknowledged bit. If the physical address of any slave device coincides with the address broadcasted by the master device, the value of this bit is set to '0' by the slave device. Otherwise, it remains at logic '1' (default).

Data Block

It comprises of 8 bits, and they are set by the sender, with the data bits it needs to transfer to the receiver. This block is followed by an ACK/NACK bit and is set to '0' by the receiver if it successfully receives data. Otherwise, it stays at logic '1'.

This combination of data block followed by ACK/NACK bit is repeated until the data is completely transferred.

Stop Condition

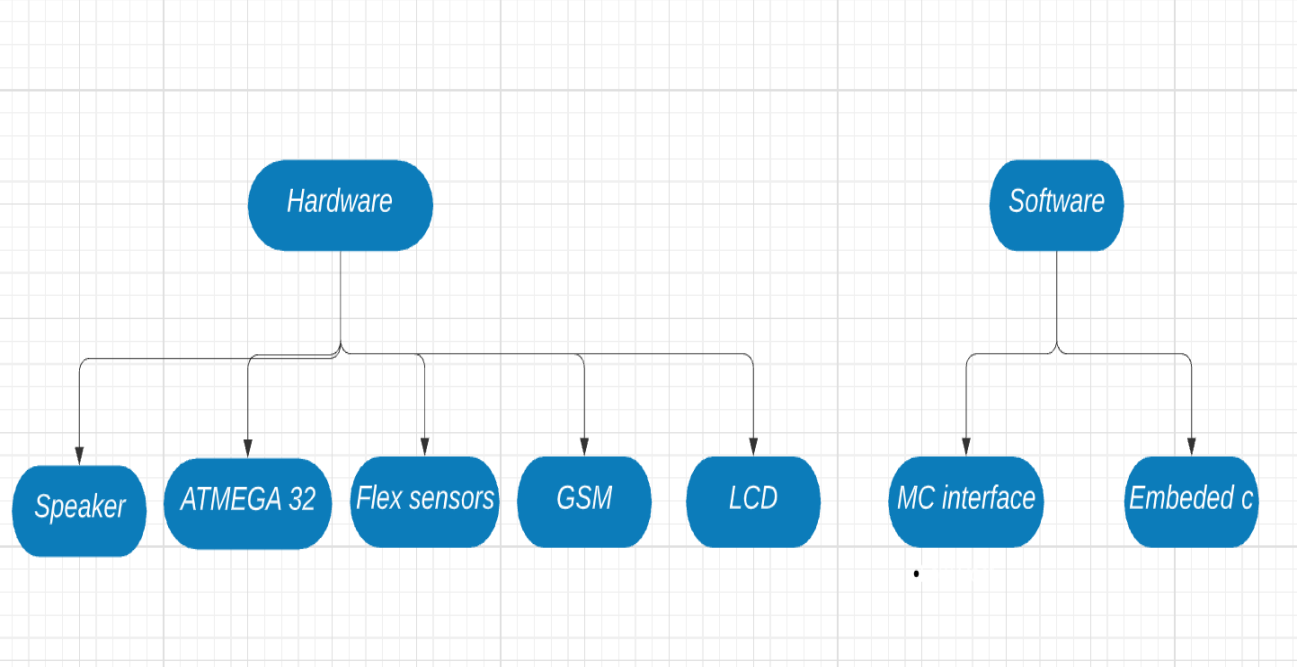
After required data blocks are transferred through the SDA line, the master device switches the SDA line from low voltage level to high voltage level before the SCL line switches from high to low.

Summary

Sign language is a useful tool to ease the communication between the deaf and the mute community and the normal people. Yet there is a communication barrier between these communities with normal people.

This project is useful for differently abled, speech-impaired, and paralyzed patients who cannot speak properly.

Smart gloves are designed to be a comfortable mean for better communication between humans, it consists of two main parts:



Hardware:

We can explain the hardware components according to:

Input elements:

- **Flex sensor → which is considered as an indicator of how much the finger in bended and it is measured according to the resistance value**
- **ATMEGA 32 → Main chip**

Output elements:

- **Speaker → it is for illustrating signals that deaf people will perform using our gloves into voice.**
- **LCD →to show the words that were translated from sign language by gloves.**

Software:

It is totally about creating a model that can understand the sign language and transform it into voice, it is done by studying mc interface and embedded c

This work is done to check the feasibility of recognizing sign language using flex sensor and accelerometer gloves and displaying the data, which proved to be an efficient system.

Reference

1. **8-bit Microcontroller with 32KBytes In-System Programmable Flash ATmega32 ATmega32L**
2. **Atmel Corporation,"8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash", Atmega2560 Datasheet, February 2014**
3. **Cull. (2011, November 15).Flexion [Online]. Available URL: http://www.sensorwiki.org/doku.php/sens ores/flexion**
4. **Sumathi M S, Ashwin B M, Balaji V, Sharath M Kumar, "Give Voice to the Voiceless Using Microcontroller and Digital Gloves", International Journal of Scientific & Engineering Research, Vol. 5, Issue 5, May-2014, ISSN 2229-5518.**
5. **M. Mohandas, M. Deric he, and J. Liu, "Image-Based and Sensor-Based Approaches to Arabic Sign Language recognition", IEEE Transactions on Human machine and computer systems, Vol. 44, No.4, August 2014.**
6. **Gunasekaran K, Manikandan R.," Sign Language to Speech Translation System Using PIC Microcontroller", International journal of engineering and technology, Vol.5, No. 2, Apr-May 2013.**
7. **Sakshi Goyal, Ishita Sharma, Shaun Sharma' "Sign language Recognition System for Deaf and Dumb People" International Journal of Engineering Research & Technology, vol. 2, issue 4, April 2013**