SMART BUS BOARDING SYSTEM FOR VISUALLY IMPAIRED PASSENGERS

Submitted in partial fulfilment of the requirements
of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND TELECOMMUNICATION

BY

SHAIKH SHAH HUSSAIN (14DET62)

SHAIKH NIZAMUDDIN (14DET60)

KHAN ARIF (14DET51)

SUPERVISOR

PROF. ZEESHAN ALI

UNIVERSITY OF MUMBAI

ANJUMAN I ISLAM KALSEKAR TECHNICAL CAMPUS

2016-2017

CERTIFICATE

This is to certify that the project entitled **"Real Time Smart Supervision System Using Internet Of Things"** is the bonafide work carried out by

SHAIKH SHAH HUSSAIN	14DET62
SHAIKH NIZAMUDDIN	14DET60
KHAN ARIF	14DET51

B.E EXTC students of Anjuman-I-Islam Kalsekar Technical Campus , Panvel ,during the year 2016-17, in partial fulfillment of the requirements for the Bachelor of engineering in Electronics and telecommunication engineering and is submitted to the Mumbai University. The project has been approved.

(H.O.D)	**(Examiner)**	**(Guide)**
Prof. Mujib Tamboli		**Prof. Zeeshan Ali**

APPROVAL SHEET

Project Report Approval for B. E.
This project report entitled (**Real Time *Smart supervision system using IoT***) by

SHAIKH SHAH HUSSAIN	14DET62
SHAIKH NIZAMUDDIN	14DET60
KHAN ARIF	14DET51

is approved for the degree of B.E EXTC.

Examiners

1.--

2.--

Supervisors

1.--

2.--

Chairman

Date:

Place: New Panvel

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(SHAIKH SHAH HUSSAIN 14DET62)

(SHAIKH NIZAMUDDIN 14DET60)

(KHAN ARIF 14DET51)

Date

ACKNOWLEDGEMENT

We appreciate the beauty of a rainbow, but never do we think that we need both the sun and the rain to make its colors appear. Similarly, this project work is the fruit of many such unseen hands. It's those small inputs from different people that have lent a helping to our project. It is a matter of great privilege for our team to present this project on "Smart Bus Boarding System for Visually Impaired Passengers".

I also take this opportunity to express a deep sense of gratitude to .Prof. **Mr. Mujib Tamboli,** HOD of EXTC department for his cordial support, valuable information and guidance, which helped us in completing this task through various stages.

I take this opportunity to express my profound gratitude and deep regards to our guide Prof**. Zeeshan Ali** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project work.

I am obliged to the staff member of AIKTC, for the valuable information provided by them in the respective fields. I am grateful for their cooperation during the period of my project work.

ABSTRACT

There are 161 million people visually impaired out of 6.7billion people in the world. Each visually impaired individual faces different challenges based on their specific level of vision. With the rise of various support-based organizations, more visually impaired people have been given the opportunity to education and many other means. But still the issues of navigation for the blind are very complex and troublesome especially when they walked down in street and also navigate to distant places by public transport system. In this project we propose a Passenger bus boarding system using wireless networks .

The visually impaired passengers are provided with a RFID tag which containes the information of the passenger and also its destination information. The Bus stops containes RFID reader and a RF module of 2.4 GHz frequency along with their master controller installed in the sits of the stop. The RFID reader will recognize the Visually impaired passenger by his RFID tag and forwards its destination information to the microcontroller, which will process the information and waits for the bus arriving signal from the RF module. When the bus, that also contains one RF module with its microcontroller comes into the range of bus stop, the RF module at the bus stop will recognize the bus no. and forwards it to the microcontroller. The microller will compare the data of the bus no and passenger’s destination and generate the alert signal accordingly to the voice module, by getting the voice alert the Passenger will know that their bus has arrived or not.

The driver at the bus will also get notification of the Visually impaired Passenger through LCD installed in the Bus unit. And in this way the visually impaired person will boared the bus easily

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The project aims in designing a system which is capable of alerting the Blind Person if the Bus has arrived, using RF networks. It takes the input destination of the Blind person through RFID and save it to the bus stop unit. When the bus arrives in the range of RF network, it detects the bus no.Then checks in the database and notify through voice alert and vibration. Blind can undoubtedly get the data about the transport to achieve destination, so travelling makes simple to him.

Two units namely the Bus stop unit and the bus unit consisting of different modules like the ISD1820 voice synthesizer, MFRC522 RFID reader, nRF24L01 RF module and a ATmega328 microcontroller. With the help of the RF module in both the units the System identifies the bus, Passenger has to board. The microcontroller, at the Bus stop unit controls the RF module and the RFID module that are used for tracking the Bus unit and knowing the desired location given by the passenger through RFID card respectively. If the location matches with the stored database,the microcontroller will wait for the signal from Rf module. Whwn the desired Bus arrive The microcontroller will give signal to voice synthesizer and the voice synthesizer will notify the passenger via speaker installed in the seats of the bus stop unit to get down at the desired location. The microcontroller will also send the signal (with the help of RF module) to the arrived bus unit in order to notify the bus driver about the Passenger's desire to board the bus. This is to ensure the safety and simplicity of the Passenger to board the bus without any difficulties. This project is also aimed at helping the elder people for independent navigation.

Chapter 2

BLOCK DIAGRAM

2.1 Block Diagram of Bus unit:

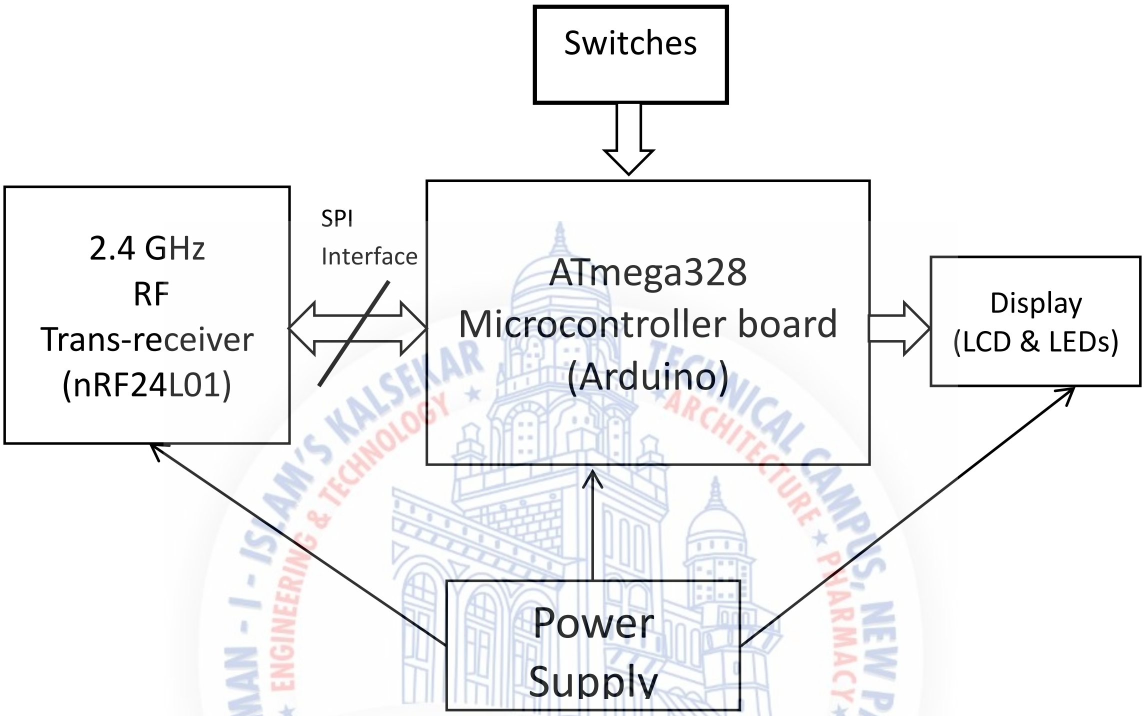


Fig. 1 Block Diagram of Bus Unit

2.2 Block Diagram of Bus Stop unit

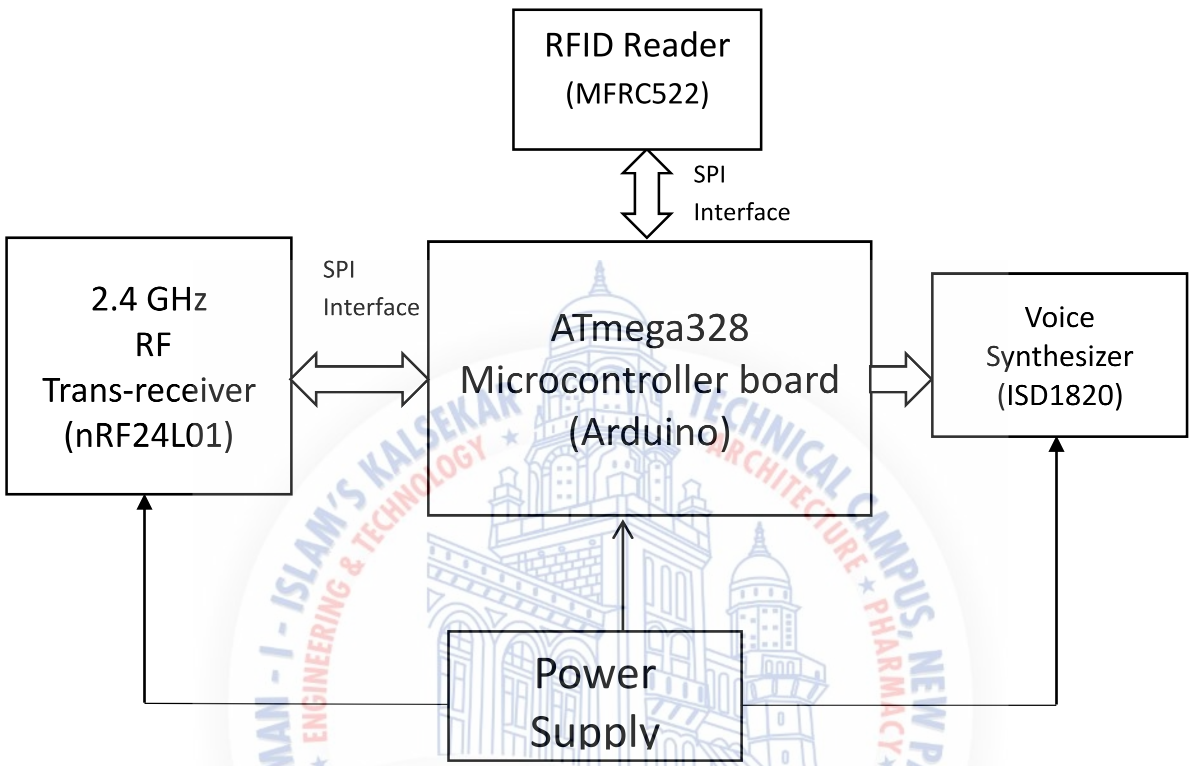


Fig. 2 Block Diagram of Bus stop Unit

Chapter 3

CIRCUIT DIAGRAM

3.1 Circuit Diagram of Bus unit

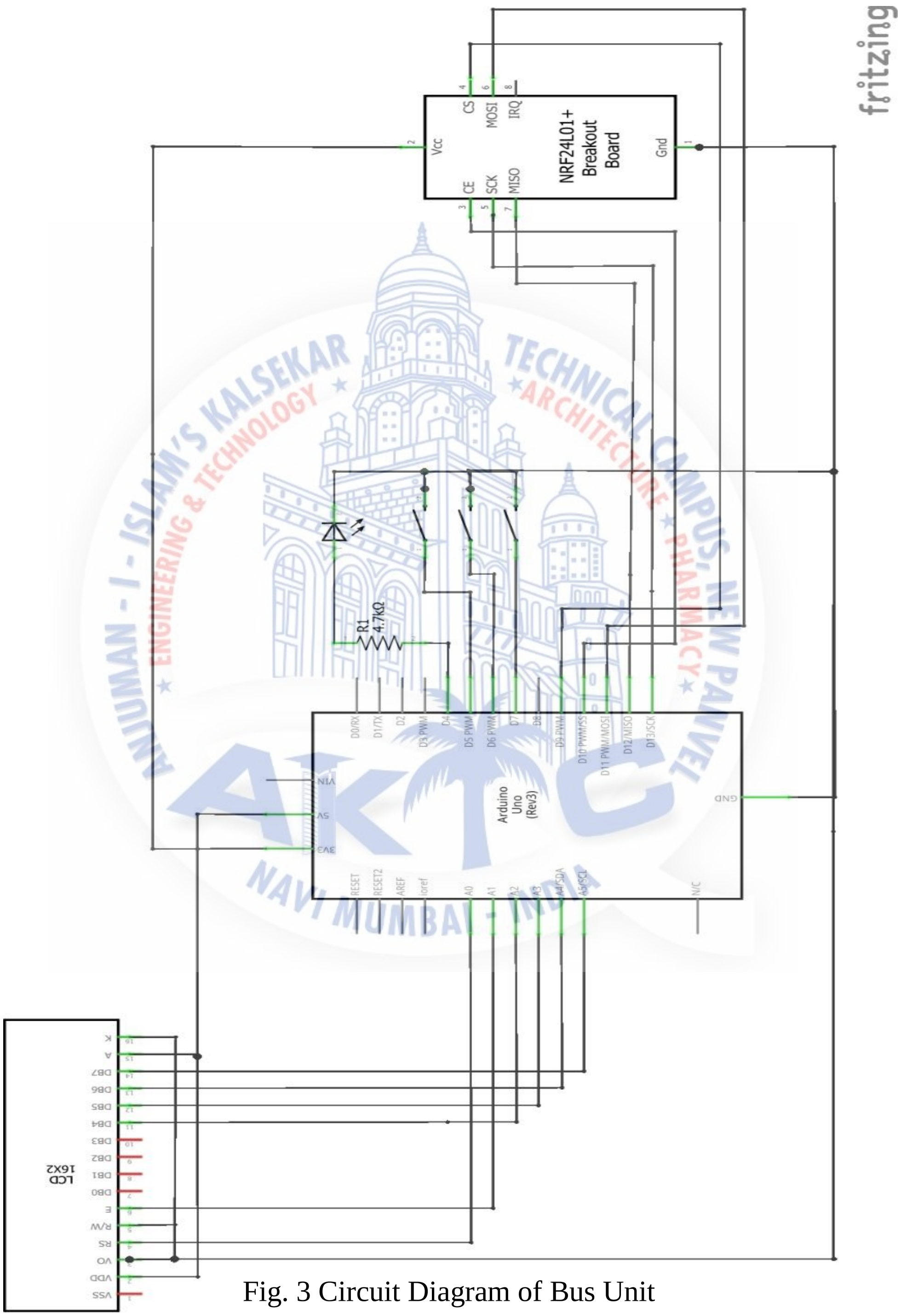


Fig. 3 Circuit Diagram of Bus Unit

3.2 Circuit Diagram of Bus Stop Unit

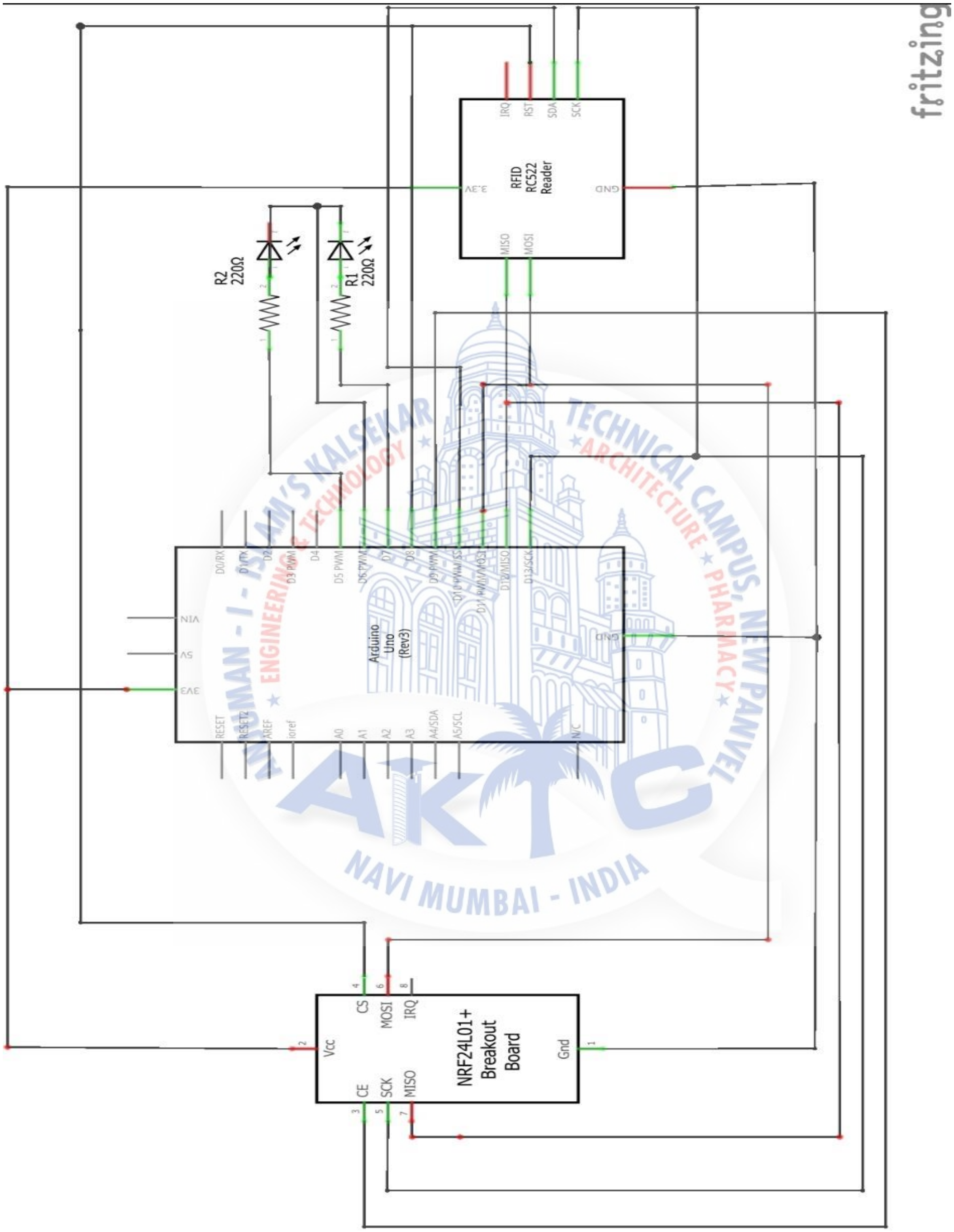


Fig. 4 Circuit Diagram of Bus Stop
Unit

Chapter 4

WORKING

4.1 SPI Interface

Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards. It uses separate clock and data lines, along with a select line to choose the device you wish to talk to. A common serial port, the kind with TX and RX lines, is called "asynchronous" (not synchronous) because there is no control over when data is sent or any guarantee that both sides are running at precisely the same rate.

SPI works in a slightly different manner. It's a "synchronous" data bus, which means that it uses separate lines for data and a "clock" that keeps both sides in perfect sync. The clock is an oscillating signal that tells the receiver exactly when to sample the bits on the data line. This could be the rising (low to high) or falling (high to low) edge of the clock signal; the datasheet will specify which one to use. When the receiver detects that edge, it will immediately look at the data line to read the next bit. Because the clock is sent along with the data, specifying the speed isn't important, although devices will have a top speed at which they can operate.

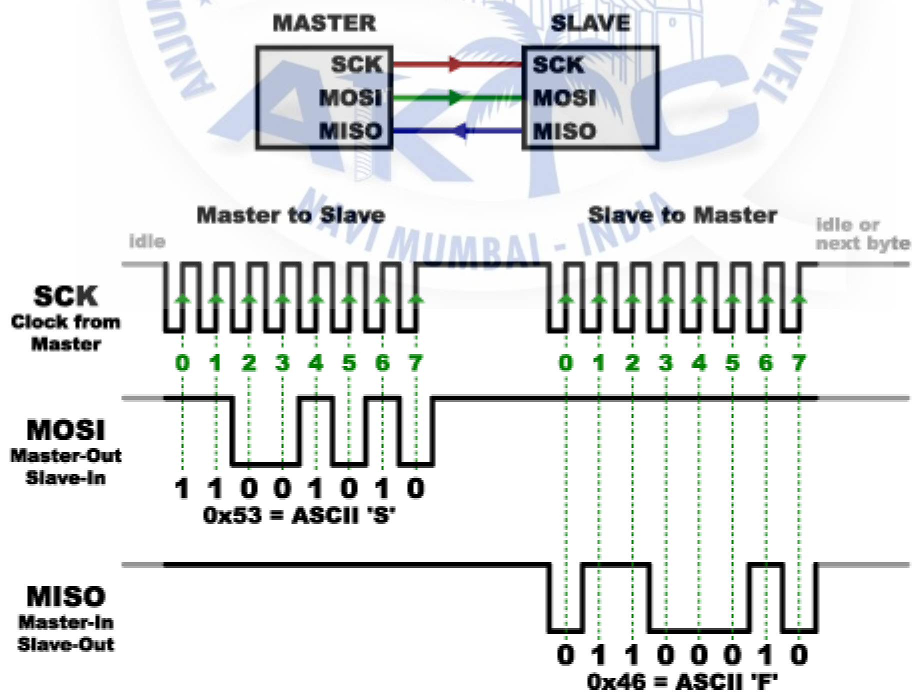


Fig. 5 SPI Interface

In SPI, only one side generates the clock signal (usually called CLK or SCK for Serial ClocK). The side that generates the clock is called the "master", and the other side is called the "slave". There is always only one master (which is almost always your microcontroller), but there can be multiple slaves (more on this in a bit).

When data is sent from the master to a slave, it's sent on a data line called MOSI, for "Master Out / Slave In". If the slave needs to send a response back to the master, the master will continue to generate a prearranged number of clock cycles, and the slave will put the data onto a third data line called MISO, for "Master In / Slave Out".

In our project there is SPI interface between Arduino and RF module in bus unit. Whereas in bus stop unit, Multiple SPI connections are implemented. Arduino acts as a master and RFID reader and RF module is its slave. The Slave Select (SS) pin is used.

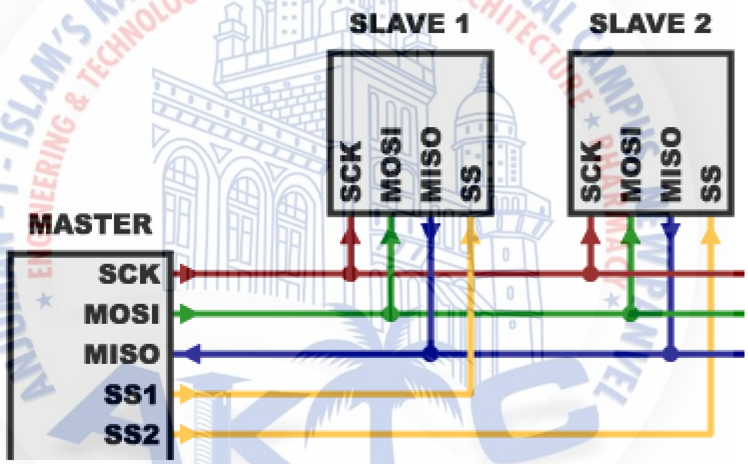


Fig. 6 Multiple SPI Interface

4.2 RFID Reading the Data from tag

In the bus stop unit RFID reader of 13.5MHz is installed. This Rfid Reader is connected and Controlled by its Master, which is none other than Microcontroller (Arduino). It uses SPI onterface to communicate with the Microcontroller, where Arduino is a Master and RFID Reader is a slave. Three wire is connected for communication namely, SCK (Serial clock), MOSI (Master Out Serial In), MISO (Master In Serial Out) and at last for multiple SPI connection SS (Slave Select) is Used first SS is Connected to RFID Reader and another Is connected to CS (Chip Select) of RF Transciever.

The Visually Impaired Pasenger carries the RFID card in which the Basic information of the Passenger and his Destination is stored. The RFID tag is a tag which contains a RF coil along with the Microchip. The Microchip of RFID tag is Integrated Circuit with a Memory of 1KB.

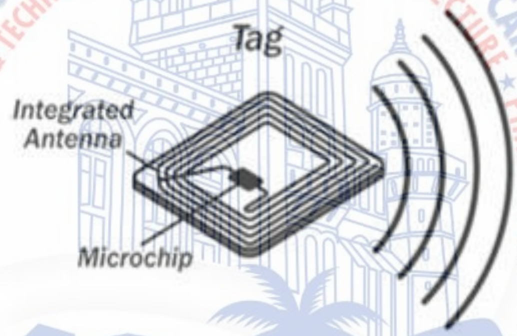


Fig. 7 RFID tag

The RFID tag used for the Project is MIFARE Classsic 1K. The RFID works on the Principle of Induction Process, where The rfid reader with the coil powers the rfid coil by induction of EM waves. First when the RFID reader gets the comand from interogator(usually Microcontroller), its initiates th command and send the signal through the coil. When the coil comes in the range of its wave the EM wave will be induced in the coil of the tag. Then the Microchip in the coil will be powered and sends the required data (such UID i.e. Unique ID) to the reader through the same induction process. The RFID reader will receive the UID of the tag and then it will verify for the validity of the tag and sends the data to interogator.

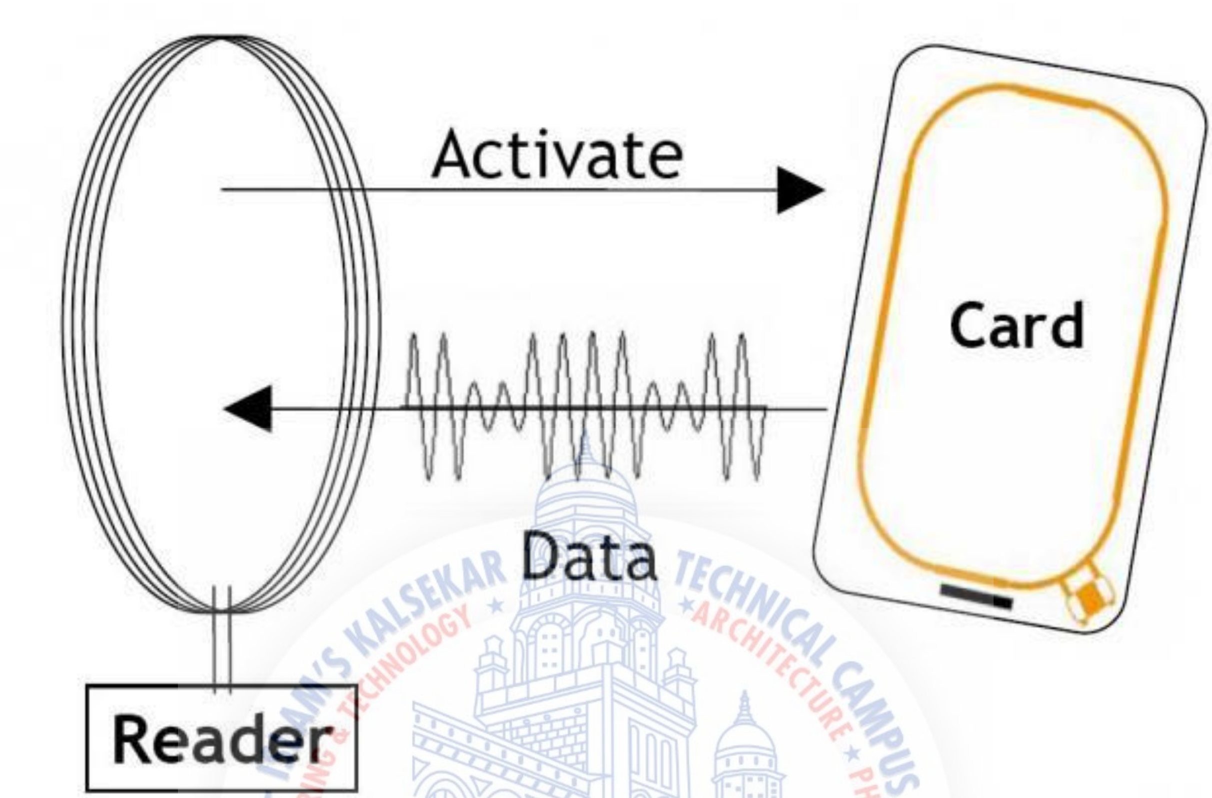


Fig. 8 Comunication between RFID reader and tag

At the Bus stop the Visually impaired passenger will carry the tag and will take a seat in which the RFID reader is installed. The tag will come in range of RFID and then the reading of data is done. The RFID reader will read the Specific block where the destination is stored. This data is given to the Arduino.

The Arduino containes the database of all the Destinations and its coresponding Bus no.

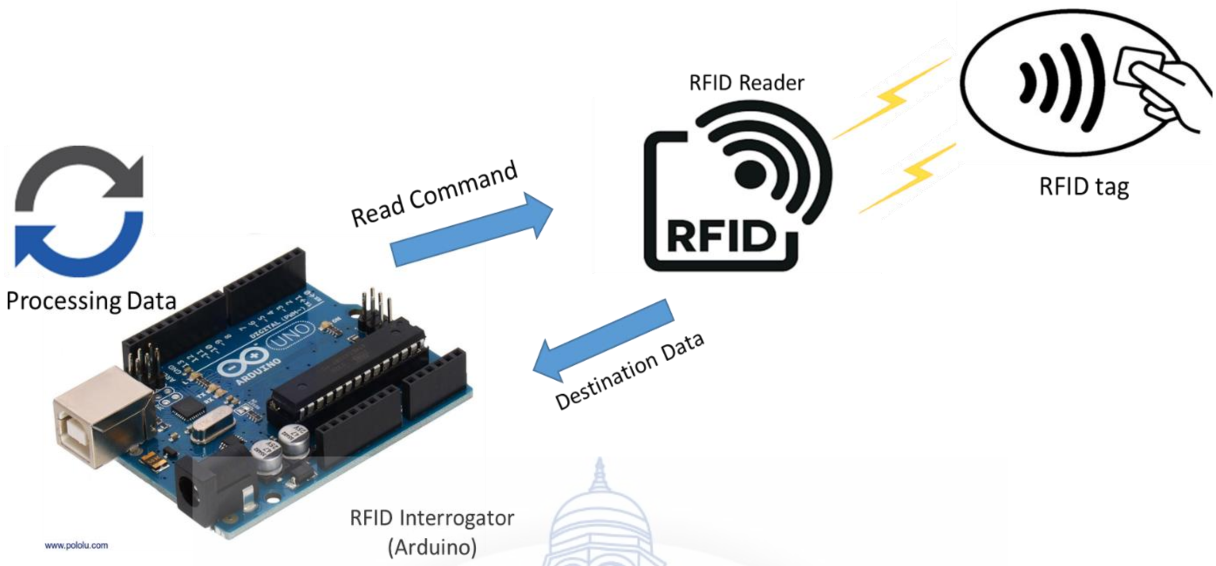


Fig. 9 Reading Data from tag

4.3 RF Communication between Bus and Bus Srop

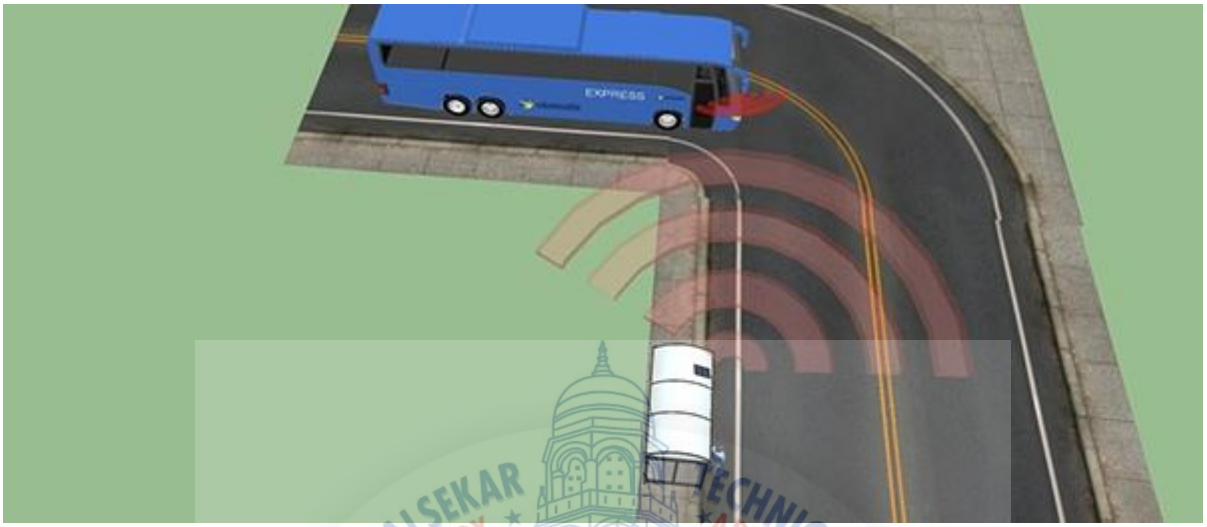


Fig. 10 RF Link between Bus and Bus Stop

The above figure shows the comunication between Bus stop and Bus. Both Bus Stop and Bus, are provided with the 2.4 GHz ranged RF Transciever. The RF Transciever at the Bus will continously send it's Bus no. On the other hand bus stop unit's RF transciever will be receiving the signal until the valid bus no, is bieng tracked.

After reading Data from the Tag and processing it, the Microcontroller(Arduino) will temporarily disables the RFID Communication and will enables the RF communication. After processing and comparing the data (from RFID tag) with the stored Database, the Arduino will conclude the required Bus no. the Passenger has to Board in order to reach his Destination. The Arduino will intructs the RF Transciever to continously Receive the Upcoming data from the upcoming Buses until the required bus has being arrived.

When the Bus(to be board) Arrived at the line of sight i.e. In the 30 meters range, the Microcontroller(Arduino) will instuct the RF transciever to establish the comunication with the Bus unit. The microcontroller at the same time will also give voice alert to the passenger waiting for the bus, this is achieved with the help of ISD1820 voice synthesizer and the mini speaker installed in the seat of the Passenger. After the RF communication gets Establishes between Bus and Bus stop, the full duplex mode is used, where the bus stop unit will send the signal that the

Visually Impaired Passenger wants to board the Bus to the Bus unit. The bus unit will receive this signal and give the notification to the driver in LCD display.

In this way 2 way comunication takes place to simplify the boarding of the bus by the visually impaired Passengers. It also ensures safety to the Passenger.

Fig. nRF24L01 Comunication

Chapter 5

HARDWARE REQUIREMENT

5.1 MFRC522 RFID Module

The MFRC522 is a highly integrated reader/writer IC for contactless communication at 13.56 MHz. The MFRC522 reader supports ISO/IEC 14443 A/MIFARE and NTAG.

The MFRC522's internal transmitter is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443 A/MIFARE cards and transponders without additional active circuitry. The receiver module provides a robust and efficient implementation for demodulating and decoding signals from ISO/IEC 14443 A/MIFARE compatible cards and transponders. The digital module manages the complete ISO/IEC 14443 A framing and error detection (parity and CRC) functionality.

The MFRC522 supports MF1xxS20, MF1xxS70 and MF1xxS50 products. The MFRC522 supports contactless communication and uses MIFARE higher transfer speeds up to 848 kBd in both directions

.

Fig. 11 MFRC522 RFID module

5.1.1 Features of MFRC522

- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers for connecting an antenna with the minimum number of external components
- Supports ISO/IEC 14443 A/MIFARE and NTAG ν
- Typical operating distance in Read/Write mode up to 50 mm depending on the antenna size and tuning
- Supports MF1xxS20, MF1xxS70 and MF1xxS50 encryption in Read/Write mode
- Supports ISO/IEC 14443 A higher transfer speed communication up to 848 kBd
- Supports MFIN/MFOUT
- Additional internal power supply to the smart card IC connected via MFIN/MFOUT
- Supported host interfaces
- SPI up to 10 Mbit/s
- I 2C-bus interface up to 400 kBd in Fast mode, up to 3400 kBd in High-speed mode
- RS232 Serial UART up to 1228.8 kBd, with voltage levels dependant on pin voltage supply
- FIFO buffer handles 64 byte send and receive
- Flexible interrupt modes
- Hard reset with low power function
- Power-down by software mode
- Programmable timer
- Internal oscillator for connection to 27.12 MHz quartz crystal
- 2.5 V to 3.3 V power supply
- CRC coprocessor
- Programmable I/O pins
- Internal self-test

5.1.2 Block Diagram of MFRC522

The analog interface handles the modulation and demodulation of the analog signals. The contactless UART manages the protocol requirements for the communication protocols in cooperation with the host. The FIFO buffer ensures fast and convenient data transfer to and from the host and the contactless UART and vice versa. Various host interfaces are implemented to meet different customer requirements.

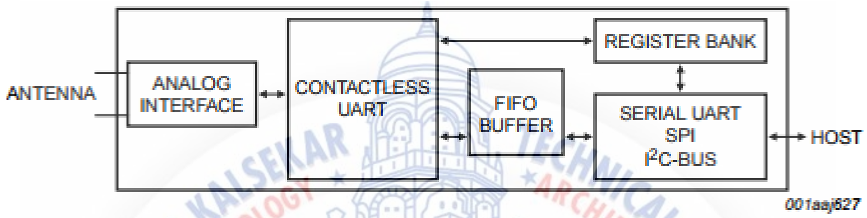


Fig. 12 Simplified Block Diagram of MFRC522

5.1.3 MFRC522 Pinout

RFID MODULE	ARDUINO
3.3v	Arduino Pin 3.3v
RST (Reset)	Arduino Pin 9
GND (Ground)	Arduino Pin GND
NC	No Connection
MISO	Arduino Pin 13
MOSI	Arduino Pin 11
SCK	Arduino Pin 13
SDA	Arduino Pin 10

Fig. 13 MFRC522 Arduino Pin Mapping

Types	Symbol	Description
MFRC522 MODULE PIN	3V3	POWER Supply 3V3
	5V	POWER Supply 5V
	GND	POWER Ground
	SCK	SPI Interface
	MISO	SPI Interface
	SI(MOSI)	SPI Interface
	SA(CS)	Serial Data Interface
	RS(NRSTPD)	Not Reset And Power-Down

Fig. 13 MFRC522 Pin description

5.1.4 Funtional Descriptions

The MFRC522 transmission module supports the Read/Write mode for ISO/IEC 14443 A/MIFARE using various transfer speeds and modulation protocols.

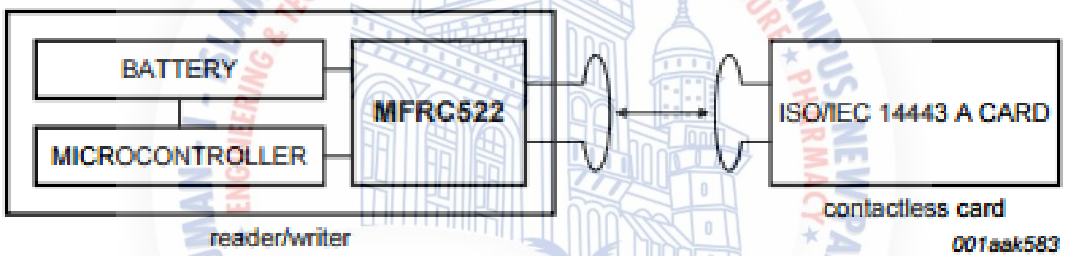
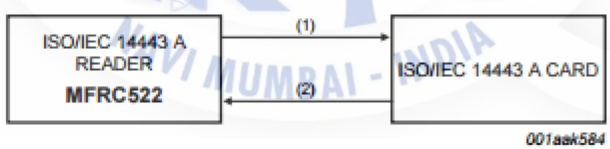


Fig. 14 Read/Write Operation



(1) Reader to card 100 % ASK, Miller encoded, transfer speed 106 kBd to 848 kBd.

(2) Card to reader subcarrier load modulation, Manchester encoded or BPSK, transfer speed 106 kBd to 848 kBd.

Fig. 15 ISO/IEC 14443 A/MIFARE Read/Write mode communication diagram

5.1.5 SPI Interface in MFRC522

A serial peripheral interface (SPI compatible) is supported to enable high-speed communication to the host. The interface can handle data speeds up to 10 Mbit/s. When communicating with a host, the MFRC522 acts as a slave, receiving data from the external host for register settings, sending and receiving data relevant for RF interface communication. An interface compatible with SPI enables high-speed serial communication between the MFRC522 and a microcontroller. The implemented interface is in accordance with the SPI standard.

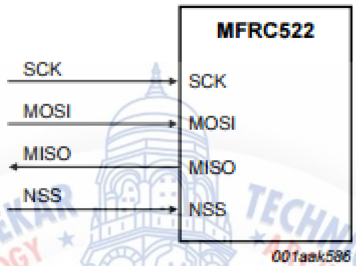


Fig. 16 SPI Interface in MFRC522

The MFRC522 acts as a slave during SPI communication. The SPI clock signal SCK must be generated by the master. Data communication from the master to the slave uses the MOSI line. The MISO line is used to send data from the MFRC522 to the master. Data bytes on both MOSI and MISO lines are sent with the MSB first. Data on both MOSI and MISO lines must be stable on the rising edge of the clock and can be changed on the falling edge. Data is provided by the MFRC522 on the falling clock edge and is stable during the rising clock edge.

5.1.6 SPI read

Reading data using SPI requires the byte order shown in Table 1 to be used. It is possible to read out up to n-data bytes. The first byte sent defines both the mode and the address.

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	address 1	address 2	...	address n	00
MISO	X[1]	data 0	data 1	...	data n – 1	data n

Table 1 MOSI and MISO byte order

5.1.7 SPI write

To write data to the MFRC522 using SPI requires the byte order shown in Table 7. It is possible to write up to n data bytes by only sending one address byte. The first send byte defines both the mode and the address byte

Line	Byte 0	Byte 1	Byte 2	To	Byte n	Byte n + 1
MOSI	address 0	data 0	data 1	...	data n – 1	data n
MISO	X[1]	X[1]	X[1]	...	X[1]	X[1]

Table 2 MOSI and MISO byte order

5.2 nRF24L01 RF Module

The nRF24L01 is a single chip 2.4GHz transceiver with an embedded baseband protocol engine (Enhanced ShockBurst™), designed for ultra low power wireless applications. The nRF24L01 is designed for operation in the world wide ISM frequency band at 2.400 - 2.4835GHz. An MCU (microcontroller) and very few external passive components are needed to design a radio system with the nRF24L01.

The nRF24L01 is configured and operated through a Serial Peripheral Interface (SPI.) Through this interface the register map is available. The register map contains all configuration registers in the nRF24L01 and is accessible in all operation modes of the chip.

Fig. 17 nRF24L01 module

The radio front end uses GFSK modulation. It has user configurable parameters like frequency channel, output power and air data rate.

The air data rate supported by the nRF24L01 is configurable to 2Mbps. The high air data rate combined with two power saving modes makes the nRF24L01 very suitable for ultra low power designs. Internal voltage regulators ensure a high Power Supply Rejection Ratio (PSRR) and a wide power supply range.

5.2.1 Features of nRF24L01

- Radio
- Worldwide 2.4GHz ISM band operation
- 126 RF channels
- Common RX and TX pins
- GFSK modulation
- 1 and 2Mbps air data rate
- 1MHz non-overlapping channel spacing at 1Mbps
- 2MHz non-overlapping channel spacing at 2Mbps
    - Transmitter
- Programmable output power: 0, -6, -12 or -18dBm
- 11.3mA at 0dBm output power
    - Receiver
- Integrated channel filters
- 12.3mA at 2Mbps
- -82dBm sensitivity at 2Mbps
- -85dBm sensitivity at 1Mbps
- Programmable LNA gain
    - RF Synthesizer
- Fully integrated synthesizer
- No external loop filer, VCO varactor diode or resonator
- Accepts low cost ±60ppm 16MHz crystal
    - Enhanced ShockBurst™
- 1 to 32 bytes dynamic payload length
- Automatic packet handling
- Auto packet transaction handling
- 6 data pipe MultiCeiver™ for 1:6 star networks
    - Power Management
- Integrated voltage regulator
- 1.9 to 3.6V supply range

- Idle modes with fast start-up times for advanced power management
- 22uA Standby-I mode, 900nA power down mode
- Max 1.5ms start-up from power down mode
- Max 130us start-up from standby-I mode
    - Host Interface
- 4-pin hardware SPI
- Max 8Mbps
- 3 separate 32 bytes TX and RX FIFOs
- 5V tolerant inputs
    - Compact 20-pin 4x4mm QFN package

5.2.3 Block Diagram of nRF24L01 chip

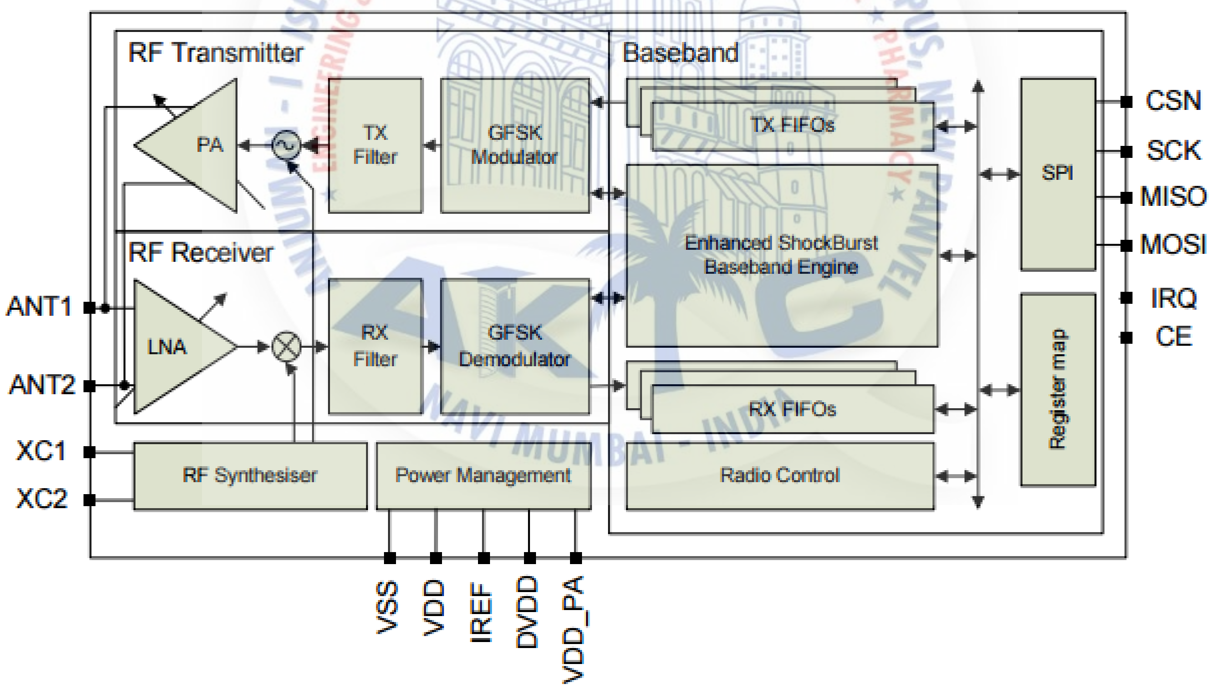


Fig. 18 Block Diagram of nRF24L01

5.2.4 Operational Modes

5.2.4.1 State diagram

The state diagram (Figure 19.) shows the modes the nRF24L01 can operate in and how they are accessed. The nRF24L01 is undefined until the VDD becomes 1.9V or higher. When this happens nRF24L01 enters the Power on reset state where it remains in reset until it enters the Power Down mode. Even when the nRF24L01 enters Power Down mode the MCU can control the chip through the SPI and the Chip Enable (CE) pin Three types of states are used in the state diagram. “Recommended operating mode” is a state that is used during normal operation. “Possible operating mode” is a state that is allowed to use, but it is not used during normal operation. “Transition state” is a time limited state used during start up of the oscillator and settling of the PLL.

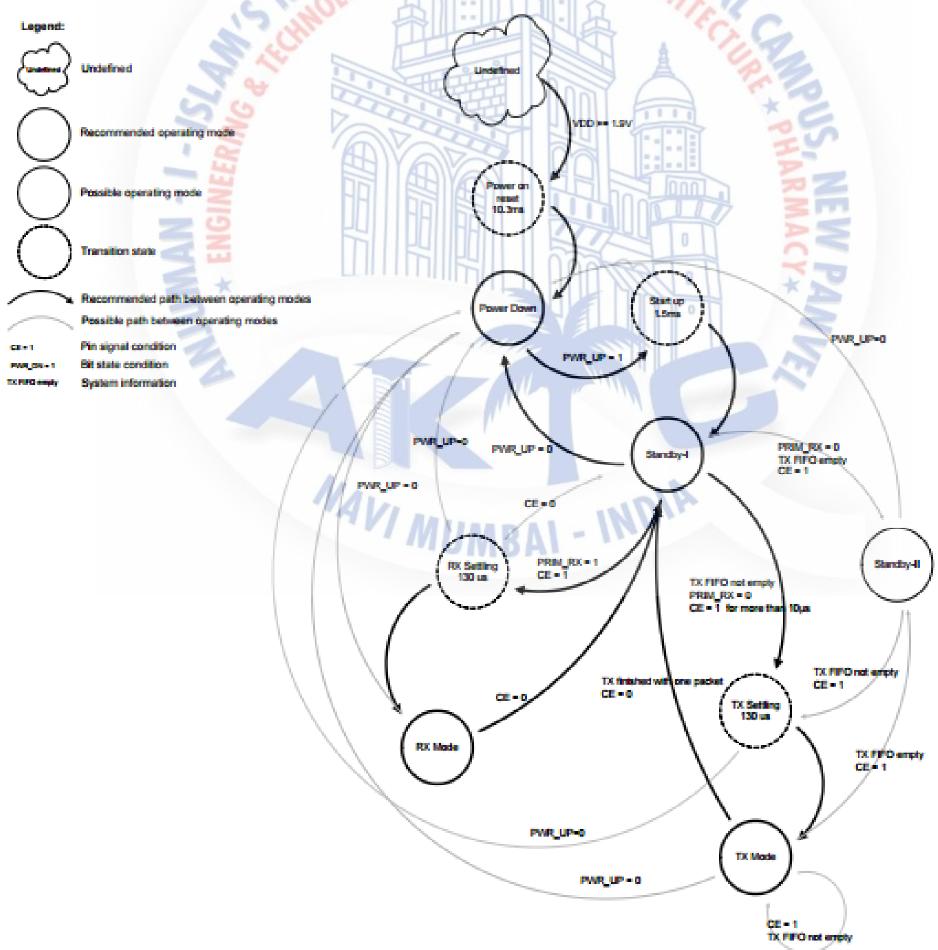


Fig. 19 State Diagram of nRF24L01

5.2.4.2 Power Down Mode

In power down mode nRF24L01 is disabled with minimal current consumption. In power down mode all the register values available from the SPI are maintained and the SPI can be activated. Power down mode is entered by setting the PWR_UP bit in the CONFIG register low.

5.2.4.3 Standby Modes

By settting the PWR_UP bit in the CONFIG register to 1, the device enters standby-I mode. Standby-I mode is used to minimize average current consumption while maintaining short start up times. In this mode part of the crystal oscillator is active. This is the mode the nRF24L01 returns to from TX or RX mode when CE is set low. In standby-II mode extra clock buffers are active compared to standby-I mode and much more current is used compared to standby-I mode. Standby-II occurs when CE is held high on a PTX device with empty TX FIFO. If a new packet is uploaded to the TX FIFO, the PLL starts and the packet is transmitted.

5.2.4.4 RX mode

The RX mode is an active mode where the nRF24L01 radio is a receiver. To enter this mode, the nRF24L01 must have the PWR_UP bit set high, PRIM_RX bit set high and the CE pin set high. In this mode the receiver demodulates the signals from the RF channel, constantly presenting the demodulated data to the baseband protocol engine. The baseband protocol engine constantly searches for a valid packet. If a valid packet is found (by a matching address and a valid CRC) the payload of the packet is presented in a vacant slot in the RX FIFO. If the RX FIFO is full, the received packet is discarded. The nRF24L01 remains in RX mode until the MCU configures it to standby-I mode or power down mode. If the automatic protocol features (Enhanced ShockBurst™) in the baseband protocol engine are enabled, the nRF24L01 can enter other modes in order to execute the protocol.

In RX mode a carrier detect signal is avaliable. The carrier detect is a signal that is set high when a RF signal is detected inside the receiving frequency channel. The signal must be FSK modulated for a secure detection. Other signals can also be detected. The Carrier Detect (CD) is set high when an RF signal is detected in RX mode, otherwise CD is low. The internal CD signal is filtered before presented to CD register. The RF signal must be present for at least 128µs before the CD is set high.

5.2.4.5 TX mode

The TX mode is an active mode where the nRF24L01 transmits a packet. To enter this mode, the nRF24L01 must have the PWR_UP bit set high, PRIM_RX bit set low, a payload in the TX FIFO and, a high pulse on the CE for more than 10µs. The nRF24L01 stays in TX mode until it finishes transmitting a current packet. If CE = 0 nRF24L01 returns to standby-I mode. If CE = 1, the next action is determined by the status of the TX FIFO. If the TX FIFO is not empty the nRF24L01 remains in TX mode, transmitting the next packet. If the TX FIFO is empty the nRF24L01 goes into standby-II mode.The nRF24L01 transmitter PLL operates in open loop when in TX mode. It is important to never keep the nRF24L01 in TX mode for more than 4ms at a time. If the auto retransmit is enabled, the nRF24L01 is never in TX mode long enough to disobey this rule.

Mode	PWR_UP register	PRIM_RX register	CE	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFO. Will empty all levels in TX FIFO[a].
TX mode	1	0	minimum 10µs high pulse	Data in TX FIFO.Will empty one level in TX FIFO[b].
Standby-II	1	0	1	TX FIFO empty
Standby-I	1	-	0	No ongoing packet transmission
Power Down	0	-	-	-

Table 3 Operational modes

5.3 Arduino

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

Fig. 20 Arduino Uno R3 Board

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board:
- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

5.4 ISD1820 voice synthesizer

This module is base on ISD1820, which a multiple-message record/playback device. It can offers true single-chip voice recording, no-volatile storage, and playback capability for 8 to 20 seconds. The sample is 3.2k and the total 20s for the Recorder.

This module use is very easy which you could direct control by push button on board or by Microcontroller such as Arduino, STM32, ChipKit etc. Frome these, you can easy control record , playback and repeat and so on.

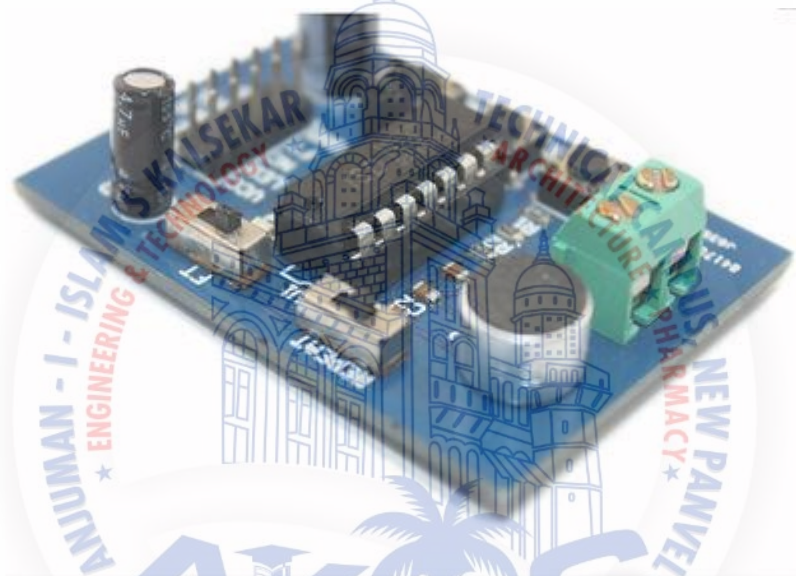


Fig. 21 ISD1820 voice synthesizer module

Feature

- Push-button interface, playback can be edge or level activated
- Automatic power-dwon mode
- On-chip 8Ω speaker driver
- Signal 3V Power Supply
- Can be controlled both manually or by MCU
- Sample rate and duration changable by replacing a single resistor
- Record up to 20 seconds of audio
- Dimensions: 37 x 54 mm

Pin definition and Rating

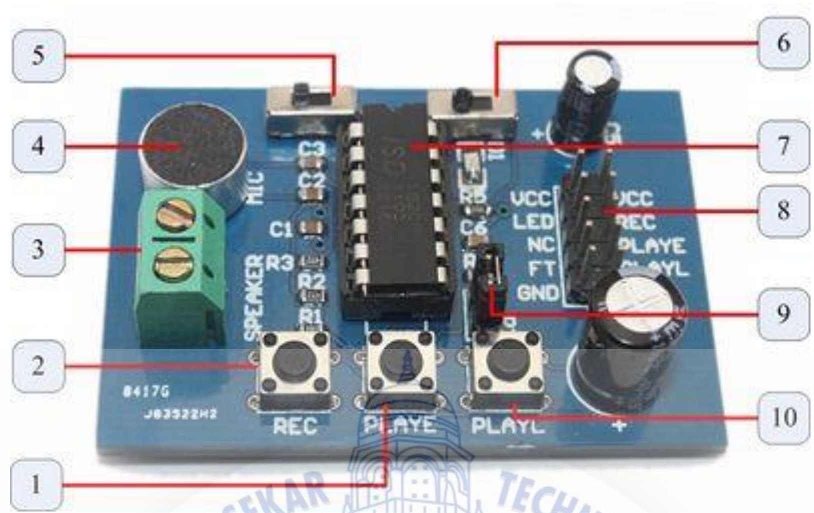


Fig. 22 ISD1820 pinout

Number	Descriptions
1	**PLAYE** - Playback, Edge-activated:When a HIGH-going transition is detected on continues until an End-of-Message (EOM) marker is encountered or the end of the memory space is reached. Upon completion of the playback cycle, the device automatically power down into standby mode Take PLAY LOW during a playback cycle will not terminate the current cycle. This pin has an internal pull-down device. Holding this pin HIGH will increase standby current consumption.
2	**REC** - The REC input is an active-HIGH record signal.The device records whenever REC is HIGH. This pin must remain HIGH for the duration of the recording. REC takes precedence over either playback(PLAYL or PLAYE) signal. If REC is pulled HIGH during a playback cycle, the playback immediately ceases and recording begins. A record cycles is completed when REC is pulled LOW. An End-of-Message(EOM) marker is internally recorded, enabling a subsequent playback cycle to terminate appropriately. The device automatically power down to standby mode when REC goes LOW. This pin has an internal pull-down device. Holding this pin HIGH will increase standby current consumption.
3	**Speaker Outputs** - The SP+ and SP- pins provide direct drive for loudspeakers with impedances as low as 8Ω. A single output may be used, but for direct-drive loud-

	speakers, the two opposite-polarity outputs provide an improvement in output power of up to four times over a single-ended connection will require an AC-coupling capacitor between the SP pin and the speaker. The SP+ pin and the SP- pin are internally connected through a 50KΩ resistance.When not in playback mode, they are floating.
4	**MIC** - Microphone Input, the microphone input transfers its signals to the on-chip preamplifier. An on-chip Automatic Gain Control (AGC) circuit controls the gain of the preamplifier. An external microphone should be AC coupled this pin via a series capacitor. The capacitor value, together with an internal 10KΩ resistance on this pin, determines the low-frequency cutoff for the 1800 passband.
5	**REPLAY** - loop play the record.
6	**FT** - Feed Through: This mode allows use of the speaker drivers for external signals. The signal between the MIC and MIC_REF pins will pass through the AGC, the filter and the speaker drivers to the speaker output SP+ and SP-. The input FT controls the feed through mode. TO operate this mode, the control pins REC, PLAYE and PLAYL are held LOW at Vss. The pin FT is held HIGH to Vcc. For normal operation of record, play and power down, the FT pin is held at Vss. The FT pin has a weak pull-down to Vss.
7	**ISD1820** - IC chip
8	Lead Out **IO** - VCC LED NC FT GND / VCC REC PLAYE PLAYL GND
9	**P2** - default short connection ROSC to 100kΩ resistance, that's means record duration is 10s
10	**PLAYL** - Playback, Level-activated, when this input pin level transits for LOW to HIGH, a playback cycle is initiated. Playback continues until PLAY is pulled LOW or an End-of-Message (EOM) marker is detected, or the end of the memory space is reached. The device automatically powers down to standby mode upon completion of the playback cycle. This pin has an internal pull-down device. Holding this pin HIGH will increase standby current consumption.

Table 4 Pin Description

Speaker

Fig. 23 8 Ohm 0.5 W Speaker

Features:

- Small Size
- Power rating: 0.5W
- Impedance: 8 ohm
- **Dimensions:** 50mm diameter, 16mm high, 28mm base diameter

5.5 LCD display (16*2)

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling displ+ay etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

Pin Diagram:

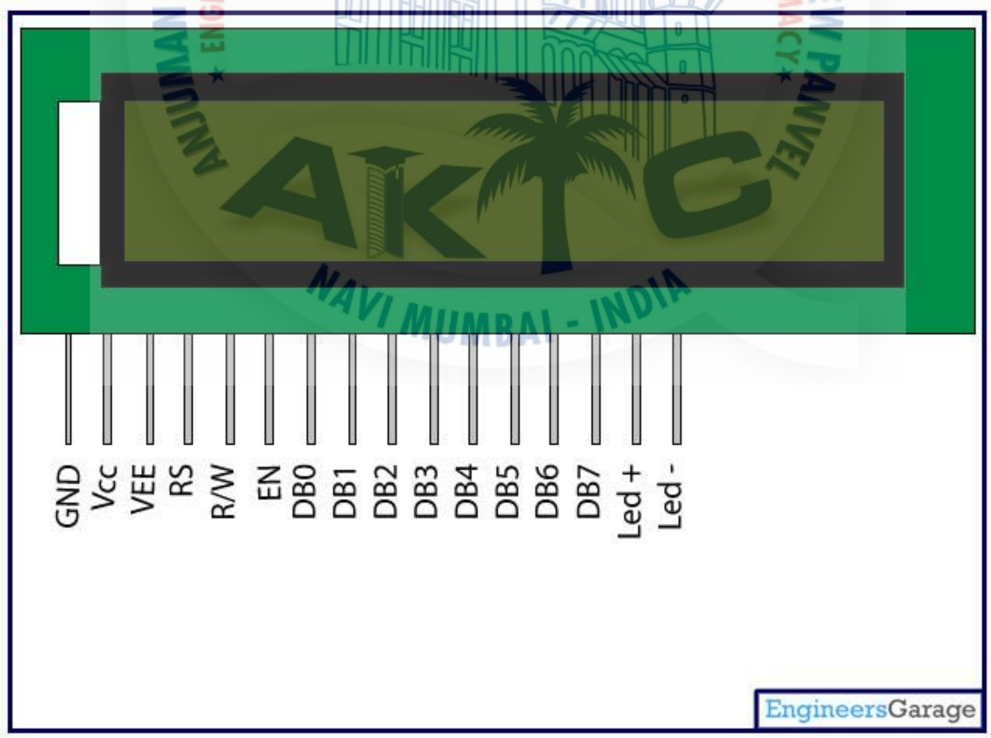


Fig. 23 LCD Pinout

Pin Description:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Table 5 Pin Description of LCD

Chapter 6

Software Requirements

6.1 Software Development in Arduino

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

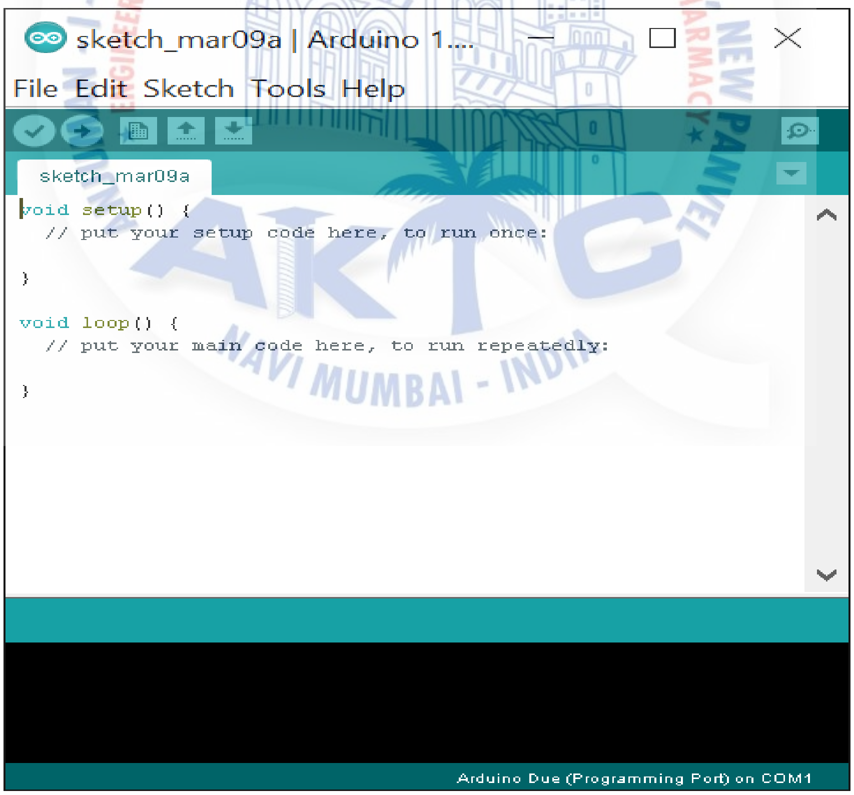


Fig. 24 Arduino Integrated Development Environment (IDE)

A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions:

- setup: This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.
- loop: After setup has been called, function loop is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Most Arduino boards contain a light-emitting diode (LED) and a load resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions A typical program for a beginning Arduino programmer blinks an LED repeatedly.

6.2 Bus Unit Program

```
 #include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
#include <LiquidCrystal.h>
 int led=7;
int s1=6;
int s2=5;
int s3=4;


RF24 radio(9,10);

const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };
unsigned long Command;


LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);

void setup()
{ pinMode(s1,INPUT);
  pinMode(s2,INPUT);
  pinMode(s3,INPUT);
  pinMode(led,OUTPUT);
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("   *Bus Unit*   ");
  lcd.setCursor(0,1);
  lcd.print("Initializing");
  radio.begin();
  delay(200);
  lcd.print(".");
  radio.openReadingPipe(1,pipes[1]);
  delay(100);
  lcd.print(".");
  radio.startListening();
  radio.printDetails();
  delay(100);
  lcd.print(".");
  radio.openWritingPipe(pipes[0]);
  radio.openReadingPipe(1,pipes[1]);
  delay(100);
  lcd.print(".");
  radio.stopListening();
  delay(500);
```

```
}

void loop(void)
{
 digitalWrite(s1,HIGH);
 digitalWrite(s2,HIGH);
 digitalWrite(s3,HIGH);
  up:
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Choose Bus no.  ");
  lcd.setCursor(0,1);
  lcd.print("(1)70 (2)7 (3)50");
  delay(2000);
 up2:

  if(digitalRead(s1)==LOW)
  {
    Command=1;
    delay(100);
  }
  else if(digitalRead(s2)==LOW)
  {
    Command=2;
    delay(100);
  }
  else if(digitalRead(s3)==LOW)
  {
    Command=3;
    delay(100);
  }
  else
  { goto up2;}
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Bus No.        ");
  lcd.setCursor(10,0);
  if(Command==1)
  {
    lcd.print("70");
    delay(200);
  }
  if(Command==2)
  {
    lcd.print("07");
    delay(200);
```

```
    }
    if(Command==3)
    {
      lcd.print("50");
      delay(200);
    }

again:  radio.stopListening();

   radio.write( &Command, sizeof(Command) );
   delay(200);
   radio.startListening();
    if(radio.available())
    {
     unsigned long data;
     radio.read( &data, sizeof(data) );
     if(data==4)
     { lcd.setCursor(0,1);
       lcd.print("VIP Available   ");

       for(int i=0;i<4;i++)
       {digitalWrite(led,HIGH);
        delay(500);}
    }
    }
    else if(digitalRead(s1)==LOW||digitalRead(s2)==LOW||digitalRead(s3)==LOW)
    {goto up;}
    else
    {goto again;}

   delay(500);

    }
```

6.3 Bus Stop unit Program

```
#include <SPI.h>

#include "nRF24L01.h"

#include "RF24.h"

#include <MFRC522.h>

#define SS1 10  //slave select pin

#define RST 8  //reset pin

#define SS2 9

MFRC522 mfrc522(SS1, RST);        // instatiate a MFRC522 reader object.

MFRC522::MIFARE_Key key;//create a MIFARE_Key struct named 'key', which will hold the
card information

RF24 radio(9,8);

int led1=5;

int led2=3;

int voice=6;

int gnd=4;

const uint64_t pipes[2] = { 0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL };

typedef enum { role_ping_out = 1, role_pong_back } role_e;

const char* role_friendly_name[] = { "invalid", "Ping out", "Pong back"};

role_e role = role_pong_back;


  void setup(void)

  {

  Serial.begin(9600);

  SPI.begin();

  pinMode(led1,OUTPUT);

  pinMode(led2,OUTPUT);
```

```
  pinMode(voice,OUTPUT);
  pinMode(gnd,OUTPUT);
  pinMode(SS1,OUTPUT);
  pinMode(SS2,OUTPUT);
  switched(1);
  radio.begin();
  radio.setRetries(15,15);
  radio.openReadingPipe(1,pipes[1]);
  radio.startListening();
  radio.printDetails();
  radio.openWritingPipe(pipes[1]);
  radio.openReadingPipe(1,pipes[0]);
  switched(0);
  mfrc522.PCD_Init();        // Init MFRC522 card (in case you wonder what PCD means:
proximity coupling device)
   for (byte i = 0; i < 6; i++)
   {
     key.keyByte[i] = 0xFF;//keyByte is defined in the "MIFARE_Key" 'struct' definition in the
.h file of the library
   }
  }

  int block=2;
  byte readbackblock[18];

  void loop()
  { digitalWrite(led1,LOW);
    digitalWrite(led2,LOW);
```

```
digitalWrite(voice,LOW);
digitalWrite(gnd,LOW);
switched(0);
if ( ! mfrc522.PICC_IsNewCardPresent())
{
  return;
}
if ( ! mfrc522.PICC_ReadCardSerial())
{
  return;
}
digitalWrite(led1,HIGH);
delay(500);
digitalWrite(led1,LOW);
Serial.println("card selected");
readBlock(block, readbackblock);
Serial.print("read block: ");
for (int j=0 ; j<16 ; j++)
{
  Serial.write (readbackblock[j]);
}
Serial.println("");

switched(1);
radio.startListening();

if ( radio.available() )
```

```
{
unsigned long data;
radio.read( &data, sizeof(unsigned long) );
Serial.println(data);
delay(20);
unsigned long sec=70;
if(data==sec)
{ digitalWrite(led2,HIGH);
  delay(5000);
}
}
}
```

Chapter 7

Advantages of the System

- Blind can undoubtedly get the data about the transport to achieve destination, so travelling makes simple to him.
- Can travel autonomous of any persons need.
- User – friendly interactions with the user.
- Easy to use.
- Audio and vibration alert can be Implemented.
- RFID based input for destination target, Hence no need to provide information additional
- This is not limited to just visually impaired individual it likewise helps senior individual.
- Communication is given between the visuallyimpaired and driver if there should be an occurrence of any crisis, hence safety for Passenger.
- RFID Module uses the encryption of data, hence security can be provided
- Can be used on the real time environment
- The system can be modified to give the automatic ticketing for the passengers, by using IOT.
- Future development of the system can be vast.
- The entire system is very cost effective.
- All the modules are very easy to handle.
- Convenient to the users.
- Low Power platform.

Chapter 8

Conclusion & Future Scope

8.1 Conclusion

In the previous system, the blind person used to carry a bus stop unit but according to the system proposed, the bus stop unit is made stationary at the bus stop. The combination of a voice synthesizer and the speaker system will help the blind at the bus stop, to find his/her bus that passes through a required/desired route. As soon as the bus approaches the bus stop, the LED in the bus starts blinking indicating the presence of blind person at the bus stop. And simultaneously the buzzer will turn on at the bus stop alerting the blind person about the approaching bus at the bus stop. These two tasks are accomplished with the aid of a RF unit which is in the bus (mobile) and at the bus stop (stationary).Finally a speaker system is used to convey the information about the bus like bus name, bus route and number. The aim of the proposed system is to provide a helping hand to the visually impaired for convenient navigation is fulfilled.

8.2 Future Scope

This system can further be improved by using GSM to provide communication between Visually Impaired Passenger and his/her relatives in case of any emergency about more realistic location of his arrival and destination.

Also by using IOT in the system, we can implement the automatic ticketing system not only for the Visually Impaired Passengers but for all the Passengers boarding the Bus in Advance. Here the RFID tag will include the Destination information similar to this System. The Bus stop unit will be connected to clouds through IOT device. Then the Digital ticket will be generated automatically in few seconds after some encryption algorithms. This digital ticket will be stored in the Passengers RFID tag. After barding the bus, this Digital ticket will be confirmed by another RFID reader at the Bus unit. Hence becomes easy to travel in the Bus.

This System will beIdentical to the one used in metro.

Chapter 9

REFERENCES

Papers:

1. Adarsh Holikatti, Dr. S. Mohan Kumar- Smart Bus Alert System for Easy Navigation of Blind: International Journal of Advanced Networking & Applications (IJANA)
2. Diravath Chander, M.Venkata Sireesha- PASSENGER BUS ALERT SYSTEM FOR EASY NAVIGATION OF BLIND: International Journal of Engineering & Science Research
3. Dhruvin Shah, Anvi Vora- Future Navigation System for Blind/Illiterate/Old People: International Journal of Computer Applications Volume 106 – No. 14, November 2014
4. G.Lavanya ME, Preethy. W, Shameem.A , Sushmitha.R- Passenger BUS Alert System for Easy Navigation of Blind: 2013 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2013]
5. Swapnil Gholap, Govind Ekshinge, Parag Naik, Prof.S.D.Chavan- Navigation of Blind People Using Passenger Bus Alert System: International Journal of Scientific and Research Publications, Volume 5, Issue 12, December 2015