ABSTRACT

The aim of this project is to ease the digital transactions by providing the flexibility of the use of individual's AADHAR and FINGER PRINTS for security and simplicity and it is the latest innovational project on the ease of doing "DIGITAL TRANSACTION" based on the 2- layer security which is "AADHAR" and "FINGER PRINT" of the user, where the user can add the fingerprints of his or her family members on his AADHAR. This scheme allows the other family members of the user to use the services of petrol pump while the payment is received from the account of registered AADHAR number of the family, when the payment is done a message of confirmation of transaction will go to the registered mobile number of the user indicating that the transaction is successfully done.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my guide, Dr PRASHANT MANI his valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this Project work.

SIDDHARTH KATIYAR

SHUBHAM VYAS

VIPIN KUMAR

CHAPTER 1

INTRODUCTION

One of the rarest and value added creation of nature is the petroleum products. It is most widely used commodity today, due to the increasing number of vehicles being used, as the mode of commutation, for our travel, on a regular basis. As India is now becoming digital, this proposed system will contribute to Digital India program .Current System at the petrol pump needs manual intervention for the fuel filling process i.e. the customer needs an attendant to fill the fuel in customer vehicle and pay the money for that transaction. These station attendants keep inhaling the petrol and diesel vapors along one of the rarest and value added creation of nature is the petroleum products. It is most widely used commodity today, due to the increasing number of vehicles being used, as the mode of commutation, for our travel, on a regular basis. As India is now becoming digital, this proposed system will contribute to Digital India program impact their lungs and hence their respiratory system. This is very dangerous, if the exposure continues for a prolonged period. In hilly and remote areas, the bunk attendants must face adverse weather conditions and safety issues like burglary during wee working hours. Due to direct cash transactions at the petrol pump, chances of robbery are high. Since the attendants operate the Handel of the nozzle of the fuel pumps in petrol stations, fraudulent transactions can occur, as exact quantity of petrol as per amount paid by customer may not done by the attendant .All the drawbacks of the current system are overcome by the automation process used in the proposed project. AADHAR cards given to the customers eliminate the need for station attendant, as the interaction is now between the customer and AADHAR card reader giving instructions to the customer, to proceed at each step, which is displayed on the LCD unit. These cards work on fingerprint technology. When manual intervention is eliminated, all issues faced by attendants is resolved. The automated process is programmed internally to calculate the exact amount of fuel to be filled, which will be equivalent to the amount paid. This helps in removing the possibility of fraudulent transactions. Many systems have been proposed to enhance this automation process of fuel filling.GSM number at the petrol pump. The customer sends the message in a format [ID user _id amount] like [ID110P250] which means the User_id of the customer is 110 and the fuel quantity demanded is equivalent to that of 250 Rupees. In response, customer receives a

password for the transaction from that GSM number at pump on his smart phone. Now when customer drives into the station , he will be prompted to enter the password in the keypad unit. After password authentication, the customer is prompted to press a specific button on keypad to initiate the fuel filling process. The entered amount of fuel is filled, by switching on pump attached to relay driver, which will pump the fuel, via fuel pipe into vehicle tank. This message when received at pump is sent to database in control cabin through the microcontroller. Thus, secure and unmanned fuel filling is attained using this system. The main components used in the project are ATMEGA 16 microcontroller, LCD display, keypad, GSM module, DC motor, Fingerprint scanner.

1.1 ATMEGA 16 MICROCONTROLLER.

The AVR ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. The throughput of AVR ATmega16 is about 1 MIPS per MHz using single clock per instruction allowing the system designed to optimize power consumption versus processing speed. The AVR has 32 general purpose working registers and rich instruction set. They are directly connected to the ALU, and allow two independent registers to be accessed in one single instruction executed in one clock cycle. We used Atmega16 because of the following features:

- It has low power CMOS 8-bit controller with AVR RISC
- Its throughput is up to 16MIPS per
- It has 32 General purpose registers directly connected to
- 16 Kbytes In-System programmable flash
- 512 bytes of EEPROM, 1k byte SRAM, JTAG
- Three timer/counter for comparison
- Internal and external interrupts
- Serial programmable USART+I2C protocol

1.1.1 ATmega16 Microcontroller Pin Descriptions

VCC Digital supply voltage

GND Ground.

Port A (PA7-PA0) It is used for analog inputs of A / D. But if A/D converter is not enabled it also serves as a self in 8-bit bi-directional port for input and output. Port pins can provide internal pull-up resistors (selected for each bit).

Port B (PB7-PB0) Port B is used as input/output 8-bit bi-directional port having internal pull- up resistors. The output buffers of Port B are of symmetrical drive characteristics having high sink and source capability. When acting as input pins, Port B pins; if pulled externally low; will source current if the pull-up resistors are activated.

Port C **(PC7-PC0)** Port C's special feature is JTAG interface. If the JTAG interface is enabled, the pull-up resistors on pins PC5 (TDI), PC3 (TMS) and PC2 (TCK) will be activated even if a reset occurs. Along with this Port C can also be used as input/output 8-bit bi-directional port having internal pull-up resistors.

Port D (PD7-PD0) Port D serves the functions of various special features of the ATmega16 like interrupts input, timer/counter output and UART. In addition to this Port D is used as input/output 8-bit bi-directional port having internal pull-up resistors.

1.1.2 ATmega16 Microcontroller CLOCK SOURCES

ATmega16 has the following clock source options, selectable by Flash Fuse bits as shown below. The clock from the selected source is input to the AVR clock generator, and routed to the appropriate modules. In device clocking option for External Crystal or Ceramic Resonator combination for bits is 1111 or 1010. For External Low-frequency Crystal it is 1001 and for External RC Oscillator 1000 or 0101. Calibrated Internal RC Oscillator 0100 or 0001 and for External Clock it is 0000.

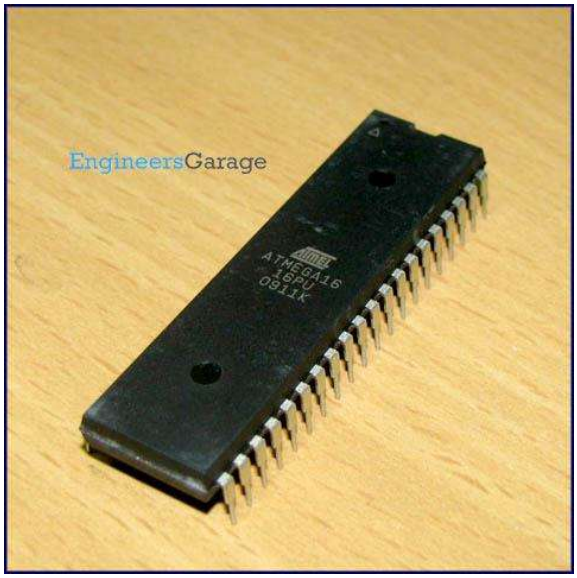


Figure 1.1

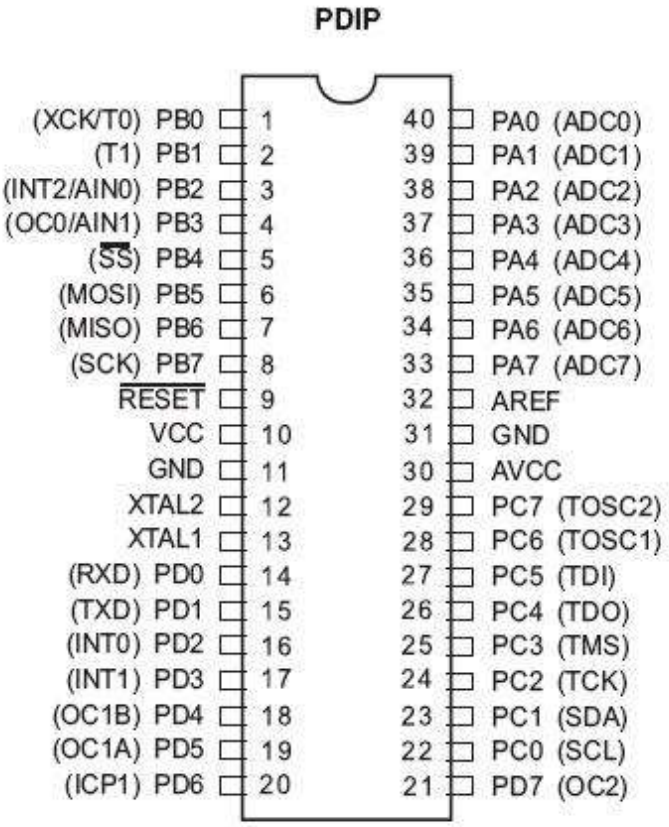


Figure 1.2

1.2 FINGERPRINT SCANNER

A fingerprint scanner is a type of technology that identifies and authenticates the fingerprints of an individual in order to grant or deny access to a computer system or a physical facility. It is a type of biometric security technology that utilizes the combination of hardware and software techniques to identify the fingerprint scans of an individual.

A fingerprint scanner typically works by first recording fingerprint scans of all authorized individuals for a particular system or facility. These scans are saved within a database. The user requiring access puts their finger on a hardware scanner, which scans and copies

the input from the individual and looks for any similarity within the already-stored scans. If there is a positive match, the individual is granted access.

Fingerprint scanners most commonly use an individual's thumbprint as identification.

Figure1.3

1.3 KEYPAD MATRIX

A keypad is a set of buttons or keys bearing digits, symbols and/or alphabetical letters placed in order on a pad, which can be used as an efficient input device. A keypad may be purely numeric, as that found on a calculator or a digital door lock, or alphanumeric as those used on cellular phone. Aside from the row of number keys found on the upper portion of a computer keyboard, a separate numerical pad is also located on the right side for efficient data entry.

Figure1.4

1.4 LCD DISPLAY

A liquid crystal display or LCD draws its definition from its name itself. It is combination of two states of matter, the solid and the liquid. LCD uses a liquid crystal to produce a visible image. Liquid crystal displays are super-thin technology display screen that are generally used in laptop computer screen, TVs, cell phones and portable video games. LCD's technologies allow displays to be much thinner when compared to cathode ray tube

(CRT) technology.

An LCD is either made up of an active matrix display grid or a passive display grid. Most of the Smartphone's with LCD display technology uses active matrix display, but some of the older displays still make use of the passive display grid designs. Most of the electronic devices mainly depend on liquid crystal display technology for their display. The liquid has a unique advantage of having low power consumption than the LED or cathode ray tube.

Liquid crystal display screen works on the principle of blocking light rather than emitting light. LCD's requires backlight as they do not emits light by them. We always use devices which are made up of LCD's displays which are replacing the use of cathode ray tube. Cathode ray tube draws more power compared to LCD's and are also heavier and bigger.

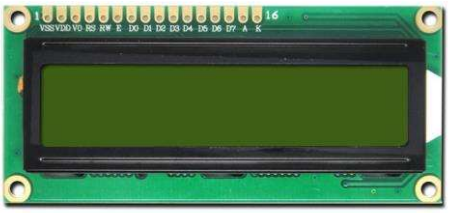


Figure 1.5

1.5 GSM MODULE

GSM/GPRS Modem-is built with Dual Band GSM/GPRS engine- SIM900A, works on frequencies 900/ 1800 MHz. The Modem is coming with interface, which allows you connect PC as well as microcontroller with Chip(MAX232). The baud rate is configurable from 9600-115200 through AT command. The GSM/ GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface. The

onboard Regulated Power supply allows you to connect wide range unregulated power supply . Using this modem you can make audio calls, SMS, Read SMS, attend the incoming calls and internet ect through simple AT commands.

Features:

- Dual-Band GSM/GPRS 900/ 1800 MHz
- RS232 interface for direct communication with computer or MCU kit
- Configurable baud rate.
- Power controlled using 29302WU IC.
- ESD Compliance.
- Enable with MIC and speaker socket.
- With slid in SIM card tray.
- With Stub antenna and SMA connector.
- Input Voltage: 12V DC.
- High quality PCB FR4 Grade with FPT Certified.

Applications:

- Industrial automation.
- GPRS based data logging.
- GPRS and GPS application.
- Home automation.
- Health monitoring.
- Agriculture automation.
- Vehicle tracking.
- Remote monitoring and controlling.
- GPRS based Weather report logging.
- GSM GPRS based Security alert.
- GPRS based remote terminal for file transfer.
- IVRS.
- Bulksms sending.

Figure 1.6

1.6 ARDUINO UNO R3.

It is the new Arduino Uno R3. In addition to all the features of the previous board, the Uno now uses an ATmega16U2 instead of the 8U2 found on the Uno (or the FTDI found on previous generations). This allows for faster transfer rates and more memory. No drivers needed for Linux or Mac (inf file for Windows is needed and included in the Arduino IDE), and the ability to have the Uno show up as a keyboard, mouse, joystick, etc.

The Uno R3 also adds SDA and SCL pins next to the AREF. In addition, there are two new pins placed near the RESET pin. One is the IOREF that allow the shields to adapt to the voltage provided from the board. The other is a not connected and is reserved for future purposes. The Uno R3 works with all existing shields but can adapt to new shields which use these additional pins.

Arduino is an open-source physical computing platform based on a simple i/o board and a development environment that implements the Processing/Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer (e.g. Flash, Processing, MaxMSP). The open-source IDE can be downloaded for free (currently for Mac OS X, Windows, and Linux).

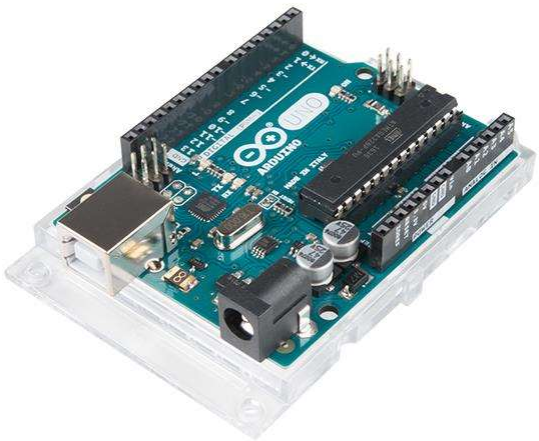


Figure 1.7

1.7 DC MOTOR

Electrical motors are everywhere around us. Almost all the electro-mechanical movements we see around us are caused either by a AC or a DC motor. Here we will be exploring DC motors. This is a device that converts DC electrical energy to a mechanical energy.
This DC or direct current motor works on the principal, when a current carrying conductor is placed in a magnetic field, it experiences a torque and has a tendency to move. This is known as motoring action. If the direction of current in the wire is reversed, the direction of rotation also reverses. When magnetic field and electric field interact they produce a mechanical force, and based on that the working principle of DC motor is established.

Figure 1.8

CHAPTER 2

LITERATURE SURVEY

In the year 1883,sylvanius F. Bowser invented an idea drawn water from a well by using a wooden plunger.Around 1885, he used this idea in case of petrol pump and became founder of S.F.Bowser Pump Company. At start the unit was very small consisting of storage barrel, the plunger,a hand lever and an upright faucet lever. This station got huge success and very soon it became a “filling station”[4]. In the year 1890, he started to pump gasoline along with the kerosene and in this way the first gas pump was invented. The company S.F.Bowser continues the refining, improvement.This “filling station” was sold to general stores in the year 1893.Before 1893, the motorist used to filled the vehicle tanks by using“drum and measure” method. The gasoline was stored in big steel drum and kept at height above the ground. The cans would be feed by using gravity and they these canspour directly into vehicle tank by using funnels. The funnels were design properly so that it should not damage the engine or vehicle[5].This process was quit lengthy, improper and dangerous.After inventing “filling station pump” this process become updated and people started to used“filling station” to dispense the kerosene,petrol. When some other oil company issued mapped Gulf was the only one company that issued the map. After this various innovations made and the modern petrol pump are developed which are treading now a day.A part from this we have added some new features into the existing system such as smoke detection, automatic payment using Smart card, indication of fuel level etc. Hence this system is more efficient than the existing once.

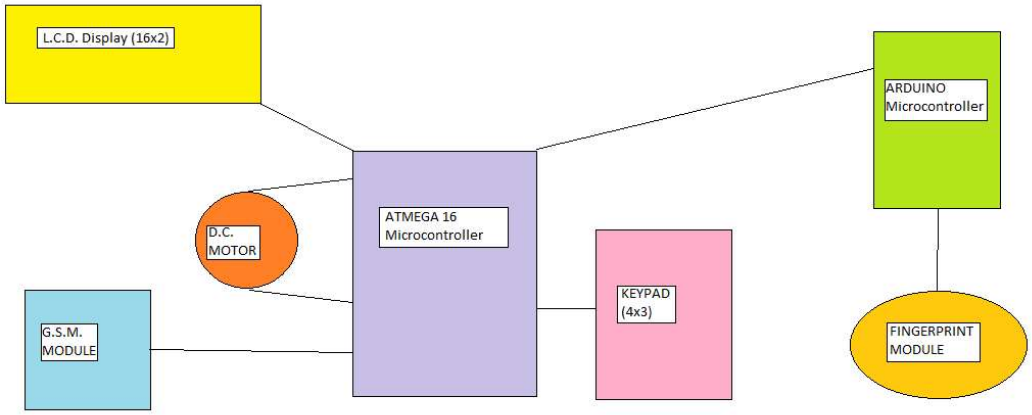
User can recharge these card at recharge points. When customer scratches the card through RFID reader, he will get his amount details on LCD screen then he has to enter the amount of petrol which has to fill up in rupees [6]. Hence the amount will be deducted automatically from the user card and the amount Details of customer will be displayed on the LCD screen again. The proposed system consists of four samples of

RFID card. Out of which 3 cards are registered and rest are unregistered. RFID card reader will accept the card if it is authorized i.e. registered. Then it will ask for amount for the

petrol to be dispensed. In this way the system will work. When customer will swipe the unauthorized card i.e.unregistered card then reader will display the error message as the card is unauthorized. In these ways we have secured the system.Much more development is being taking place in the process of making the petrol pump automated.Till now linking of AADHAR card to the petrol pump is not being implemented .But in the near future some of the Indian psu such as IOCL , BPCL with the help of Honeywell automation is looking to implement this particular system in the near future.

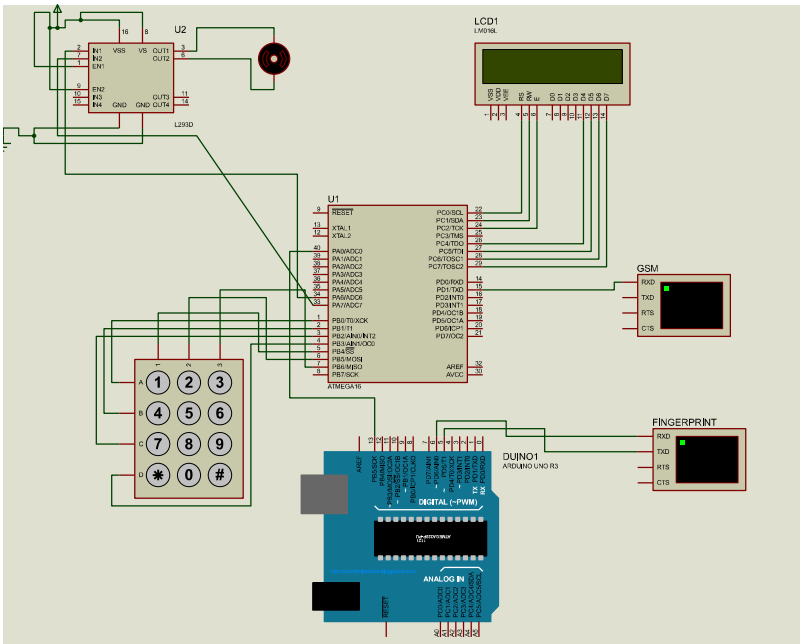
CHAPTER 3

SYSTEM DESIGN



Block diagram

Figure 3.1



Circuit diagram

Figure 3.2

- **PORT A:** Pins(PA0-PA6) are interfaced with the 4x3 keypad
- **PORT B:** Pins(PB4-PB7) are used as the data pins and are connected with the (D4-D7) of the LCD.Pins(PB0-PB2) of the controller are connected with the (D0-D2) of the LCD.
- **PORT C:** Pins(PC0 and PC1) are connected with the SDA and SCL pins of the G.S.M. Module (I2C) respectively, and Pins(PC2 and PC3) are connected to the terminals of the D.C. motor.
- **PORT D:** Pins(PD0 and PD1) of ATMEGA 16 are connected with the RxD and TxD pins of the Finger Print Module.

3.1 GSM MODULE.

GSM/GPRS Modem-RS232 is built with Dual Band GSM/GPRS engine- SIM900A, works on frequencies 900/ 1800 MHz. The Modem is coming with RS232 interface, which allows you connect PC as well as microcontroller with RS232 Chip(MAX232). The baud rate is configurable from 9600-115200 through AT command. The GSM/GPRS Modem is having internal TCP/IP stack to enable you to connect with internet via GPRS. It is suitable for SMS, Voice as well as DATA transfer application in M2M interface. The onboard Regulated Power supply allows you to connect wide range unregulated power supply . Using this modem, you can make audio calls, SMS, Read SMS, attend the incoming calls and internet ect through simple AT commands.

Features

 Dual band GSM/GPRS 900/1800MHz.

 Configurable baud rate.

 SIM card holder.
 Built in network status LED.
 Inbuilt powerful TCP/IP protocol stack for internet data transfer over GPRS.

Applications

 Access control devices.
 Supply chain management.

Specifications:

Parameter	**Value**
Operating voltage	+12 V DC
Weight	<140 gm

Pin Specification:

Pin	**Name**	**Details**
1	GROUND	Power supply details
2	TX	Transmitter
3	RX	Reciever
4	Line_r& Line_1	Line input
5	Spk_p & Spk _n	Speaker (+) and (-)
6	Mic _p& mic_n	Mic positive & negative
7	DTR	Data terminal ready
8	CTS	Clear to send
9	RTS	Request to send

Working

Unlike mobile phones, a GSM modem doesn't have a keypad and display to interact with. It just accepts certain commands through a serial interface and acknowledges for those. These commands are called as AT commands. There are a list of AT commands to instruct the modem to perform its functions. Every command starts with "AT". That's why they are called as AT commands. AT stands for attention. In our simple project, the program waits for the mobile number to be entered through the keyboard. When a ten digit mobile number is provided, the program instructs the modem to send the text message using a sequence of AT commands.

3.2 LCD DISPLAY (16X2)

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in colour or monochrome. LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and 7-segment displays, as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

LCDs are used in a wide range of applications including LCD televisions, computer monitors, instrument panels, aircraft cockpit displays, and indoor and outdoor signage. Small LCD screens are common in portable consumer devices such as digital cameras, watches, calculators, and mobile telephones, including smartphones. LCD screens are also used on consumer electronics products such as DVD players, video game devices and clocks. LCD screens have replaced heavy, bulky cathode ray tube (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and plasma displays, with LCD screens available in sizes ranging from tiny digital watches to huge, big-screen television sets.

Since LCD screens do not use phosphors, they do not suffer image burn-in when a static image is displayed on a screen for a long time (e.g., the table frame for an aircraft

schedule on an indoor sign). LCDs are, however, susceptible to image persistence. The LCD screen is more energy-efficient and can be disposed of more safely than a CRT can. Its low electrical power consumption enables it to be used in battery-powered electronic equipment more efficiently than CRTs can be. By 2008, annual sales of televisions with LCD screens exceeded sales of CRT units worldwide, and the CRT became obsolete for most purposes.

Each pixel of an LCD typically consists of a layer of molecules aligned between two transparent electrodes, and two polarizing filters (parallel and perpendicular), the axes of transmission of which are (in most of the cases) perpendicular to each other. Without the liquid crystal between the polarizing filters, light passing through the first filter would be blocked by the second (crossed) polarizer. Before an electric field is applied, the orientation of the liquid-crystal molecules is determined by the alignment at the surfaces of electrodes. In a twisted nematic (TN) device, the surface alignment directions at the two electrodes are perpendicular to each other, and so the molecules arrange themselves in a helical structure, or twist. This induces the rotation of the polarization of the incident light, and the device appears gray. If the applied voltage is large enough, the liquid crystal molecules in the center of the layer are almost completely untwisted and the polarization of the incident light is not rotated as it passes through the liquid crystal layer. This light will then be mainly polarized perpendicular to the second filter, and thus be blocked and the pixel will appear black.

Item	Symbol	Standard			Unit
Power voltage	V_{DD}-V_{SS}	0	-	7.0	V
Input voltage	V_{IN}	VSS	-	VDD	
Operating temperature range	V_{OP}	0	-	+50	℃
Storage temperature range	V_{ST}	-10	-	+60	

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply voltage for LCD	V_{DD}-V_0	Ta =25℃	-	3.0	-	V
Input voltage	V_{DD}		3.1	3.3	3.5	
Supply current	I_{DD}	Ta=25℃, V_{DD}=3.3V	-	1.5	2.5	mA
Input leakage current	I_{LKG}		-	-	1.0	uA
"H" level input voltage	V_{IH}		2.2	-	V_{DD}	V
"L" level input voltage	V_{IL}	Twice initial value or less	0	-	0.6	
"H" level output voltage	V_{OH}	LOH=-0.25mA	2.4	-	-	
"L" level output voltage	V_{OL}	LOH=1.6mA	-	-	0.4	
Backlight supply voltage	V_F		-	3.0		
Backlight supply current	I_{LED}	V_{LED}=3.3 V R=25 Ω			16	mA

Parameter	Symbol	Test pin	Min.	Typ.	Max.	Unit
Enable cycle time	t_c	E	500	-	-	ns
Enable pulse width	t_w		300	-	-	
Enable rise/fall time	t_r, t_f		-	-	25	
RS; R/W setup time	t_{su1}	RS; R/W	100	-	-	
RS; R/W address hold time	t_{h1}	RS; R/W	10	-	-	
Read data output delay	t_{su2}	DB0~DB7	60	-	-	
Read data hold time	t_{h2}		10	-	-	

3.3 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 crystal oscillator, 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 Atmega8U2 programmed as a USB-to-serial converter.

3.3.1 Specifications

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB

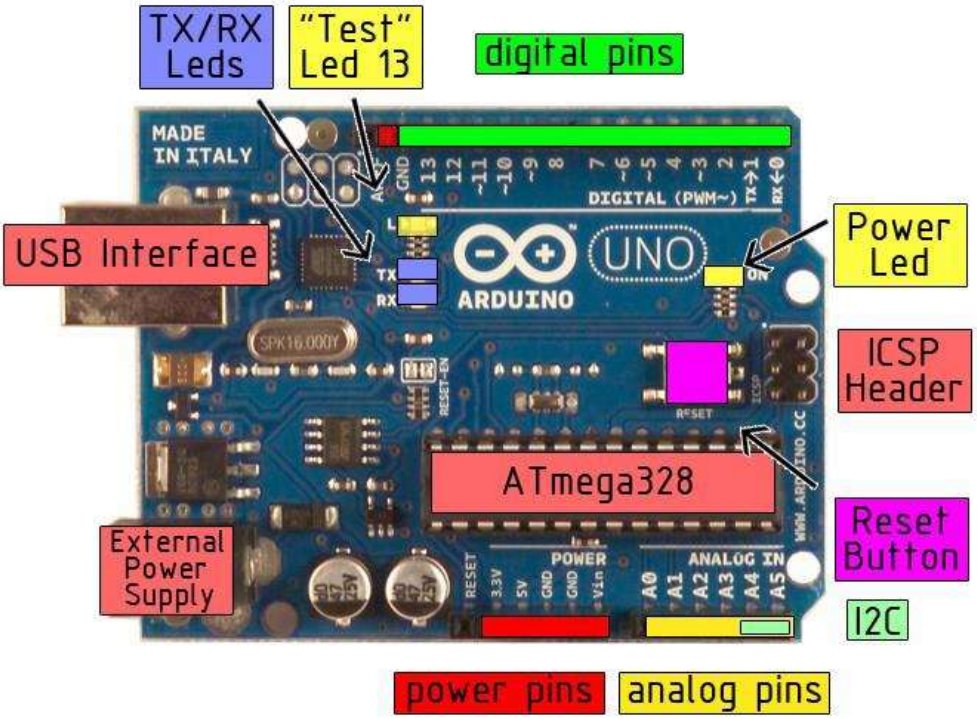


Figure 3.3

3.3.2 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.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- **GND.** Ground pins.

3.3.3 Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library.

3.3.4 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. TThese 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, which, although provided by the underlying hardware, is not currently included in the Arduino language.

· **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.

3.4 ATMEGA 16

Features

• High-performance, Low-power AVR® 8-bit Microcontroller
• Advanced RISC Architecture
– 131 Powerful Instructions – Most Single-clock Cycle Execution
– 32 x 8 General Purpose Working Registers
– Fully Static Operation
– Up to 16 MIPS Throughput at 16 MHz
– On-chip 2-cycle Multiplier
• High Endurance Non-volatile Memory segments
– 16K Bytes of In-System Self-programmable Flash program memory
– 512 Bytes EEPROM
– 1K Byte Internal SRAM
– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
– Data retention: 20 years at 85°C/100 years at 25°C(1)
– Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
– Programming Lock for Software Security
• JTAG (IEEE std. 1149.1 Compliant) Interface
– Boundary-scan Capabilities According to the JTAG Standard
– Extensive On-chip Debug Support
– Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
• Peripheral Features

– Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes

– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

– Real Time Counter with Separate Oscillator

– Four PWM Channels

– 8-channel, 10-bit ADC

8 Single-ended Channels

7 Differential Channels in TQFP Package Only

2 Differential Channels with Programmable Gain at 1x, 10x, or 200x

– Byte-oriented Two-wire Serial Interface

– Programmable Serial USART

– Master/Slave SPI Serial Interface

– Programmable Watchdog Timer with Separate On-chip Oscillator

– On-chip Analog Comparator

• Special Microcontroller Features

– Power-on Reset and Programmable Brown-out Detection

– Internal Calibrated RC Oscillator

– External and Internal Interrupt Sources

– Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

• I/O and Packages

– 32 Programmable I/O Lines

– 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF

• Operating Voltages

– 2.7 - 5.5V for ATmega16L

– 4.5 - 5.5V for ATmega16

• Speed Grades

– 0 - 8 MHz for ATmega16L

– 0 - 16 MHz for ATmega16

• Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L

– Active: 1.1 mA

– Idle Mode: 0.35 mA

– Power-down Mode: < 1 μs.

3.5 4*3 Keypad

3.5.1General Specification

- Contact rating:20mA,24VDC
- Contact resistance:200 ohm max
- Life:1,000,000 cycles per key
- Operating Temperature: -20 to +60
- Storage Temperature: -40 to +65

3.5.2 Standard Matrix Circuit Diagram

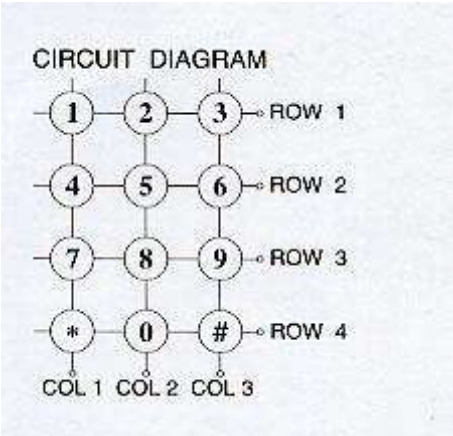


Figure 3.4

OUTPUT ARRANGEMENT

OUTPUT PIN NO.	SYMBOL
1	COL 2
2	ROW 1
3	COL 1
4	ROW 4
5	COL 3
6	ROW 3
7	ROW 2

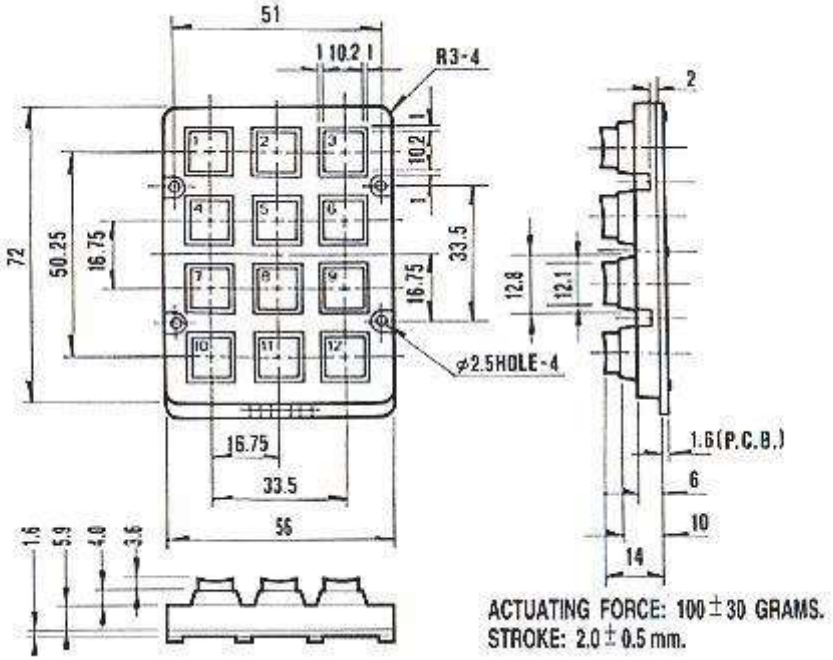


Figure 3.5

CHAPTER 4

SYSTEM ANALYSIS

Working (Steps)

1.Provide the power 5 volts D.C. to the AVR development board.

2.Provide the required 12 volts/1 ampere power to the G.S.M. Module by the specified 12 volts/1 ampere adopter.

3. Provide the required 5 volts power supply to Arduino UNO board.

4.Now a message will be displayed on the screen of LCD, "Welcome to the self service petrol pump".

5.In the next step the LCD will display "Enter your fingerprint" indicating the user to authenticate/press his/her fingerprint on the fingerprint module.

6.If the user is not an authorized user i.e his/her fingerprint doesn't matches with the stored fingerprints then the system doesn't provide access to that user for further steps and the LCD will go in its previous steps and show"welcome to the self service petrol pump".

7.If the user is authenticated user ,then the system will provide further access to the user and the LCD will show "Enter your aadhar password" for further confirmation.

8.Now the user can enter his/her unique aadhar password with the help of a 4x3keypad.

9.If the password is not correct then the LCD will show "access denied" and the user is blocked from further steps.

10. If the user's entered password is correct then the system will proceed to the next step and the LCD will going to display "Enter the amount" the amount of petrol in rupees to be filled.

11.Now the user enters his/her desired amount in rupees of petrol to be filled with the help of keypad.

12.As soon as the user had entered the amount ,the D.C.motor starts rotating for a particular time interval ,indicating that the petrol is being filling to the vehicle.

13.As the Petrol is being filled up a message is sent to the authenticated user's mobile number by the GSM module ,Indictating that the given amount is being deducted from the user's account.

14.D.C. Motor stops rotating.

15.The process is now completed successfully and the LCD will show "welcome to the self service petrol pump" welcoming the next user in queue.

CHAPTER 5

CODING, TESTING

The coding of this project is done in Embedded on PROTEUS ISIS and the code is executed on Atmel studio 6.1

5.1 AVR CODE:

```
#include<avr/io.h>
#include<util/delay.h>
voidlcdcmd(chara)
{
        PORTC=0xF0&a;
        PORTC&=~(1<<0);
        PORTC&=~(1<<1);
        PORTC|=(1<<2);
        _delay_ms(100);
        PORTC&=~(1<<2);
        _delay_ms(100);
        a=a<<4;

        PORTC=0xF0&a;
        PORTC&=~(1<<0);
        PORTC&=~(1<<1);
        PORTC|=(1<<2);
        _delay_ms(100);
        PORTC&=~(1<<2);
        _delay_ms(100);
```

```
}

voidlcddata(chara)
{
        PORTC=0xF0&a;
        PORTC|=(1<<0);
        PORTC&=~(1<<1);
        PORTC|=(1<<2);
        _delay_ms(100);
        PORTC&=~(1<<2);
        _delay_ms(100);
        a=a<<4;

        PORTC=0xF0&a;
        PORTC|=(1<<0);
        PORTC&=~(1<<1);
        PORTC|=(1<<2);
        _delay_ms(100);
        PORTC&=~(1<<2);
        _delay_ms(100);

}

voiddisplay(chary[])
{
        inti;
        for(i=0;y[i]!='\0';i++)
        {
                lcddata(y[i]);
        }
}

voidtx(charx)
```

```
{
        UDR=x;
        while((UCSRA&0b01000000)==0b00000000);
        UCSRA=0x00;
        _delay_ms(10);
}

voidstrtx(chary[])
{
        inti;
        for(i=0;y[i]!='\0';i++)
        {
                tx(y[i]);
        }
}
intflag=1;


inta[10],i,j=0;;
main()
{
        DDRA=0b11100000;
        DDRB=0B00001111;
        DDRC=0xFF;
        lcdcmd(0x02);

        lcdcmd(0x28);
        lcdcmd(0x0E);
        PORTB=0xFF;
        UBRRL=51;
        UCSRA=0b00000000;
        UCSRB=0b00011000;
        UCSRC=0b10000110;
        chart[10];
```

```
a:
i=0,j=0;
flag=1;
lcdcmd(0x80);
display("WELCOME TO SELF ");
lcdcmd(0xC0);
display("SERVICE STATION ");
_delay_ms(2000);
lcdcmd(0x01);
display("PLACE YOUR FINGER");
lcdcmd(0xC0);
display("PRINT ");
_delay_ms(5000);
lcdcmd(0x01);
if(PINA==0b00000001)
{
        display("AUTHENTICATED");
        _delay_ms(1000);
        lcdcmd(0x01);
        lcdcmd(0x80);
        display("ENTER YOUR ADHAR ");
        lcdcmd(0xC0);
        display("NUMBER..... ");
        _delay_ms(2000);
        lcdcmd(0x01);
        while(1)
        {
                PORTB=0b11111110;
                if(PINB==0b11101110)
{
        lcddata('1');
        a[i]=1;
        i++;
        _delay_ms(2000);
```

```
}
if(PINB==0b11011110)
{
        lcddata('2');
        a[i]=2;
        i++;
        _delay_ms(2000);
}

if(PINB==0b10111110)
{
        lcddata('3');
        a[i]=3;
        i++;
        _delay_ms(2000);
}


PORTB=0b11111101;
if(PINB==0b11101101)
{
        lcddata('4');
        a[i]=4;
        i++;
        _delay_ms(2000);
}
if(PINB==0b11011101)
{
        lcddata('5');
        a[i]=5;
        i++;
        _delay_ms(2000);
}
```

```
if(PINB==0b10111101)
{
        lcddata('6');
        a[i]=6;
        i++;
        _delay_ms(2000);
}

PORTB=0b11111011;
if(PINB==0b11101011)
{
        lcddata('7');
        a[i]=7;
        i++;
        _delay_ms(2000);
}
if(PINB==0b11011011)
{
        lcddata('8');
        a[i]=8;
        i++;
        _delay_ms(2000);
}

if(PINB==0b10111011)
{
        lcddata('9');
        a[i]=9;
        i++;
        _delay_ms(2000);
}
if(i==3)
{
```

```
lcdcmd(0x01);
if((a[0]==2)&&(a[1]==3)&&(a[2]==5)&&(flag==1))
{
        flag=0;
        lcdcmd(0x01);
        display("ENTER THE AMOUNT...");
        _delay_ms(500);

        ///6lcdcmd(0x01);
        lcdcmd(0xc0);
}

if(flag==0)
{
        PORTB=0b11111110;
        if(PINB==0b11101110)
        {

                lcddata('1');
                t[j]='1';
                j++;
                _delay_ms(2000);
        }
        if(PINB==0b11011110)
        {
                lcddata('2');
                t[j]='2';
                j++;
                _delay_ms(2000);
        }

        if(PINB==0b10111110)
        {
                lcddata('3');
```

```
                t[j]='3';
                j++;
                _delay_ms(2000);
            }


            PORTB=0b11111101;
            if(PINB==0b11101101)
            {
        lcddata('4');
        t[j]='4';
        j++;
        _delay_ms(2000);
}
if(PINB==0b11011101)
{
        lcddata('5');
        t[j]='5';
        j++;
        _delay_ms(2000);
}

if(PINB==0b10111101)
{
        lcddata('6');
        t[j]='6';
        j++;
        _delay_ms(2000);
}

PORTB=0b11111011;
if(PINB==0b11101011)
{
        lcddata('7');
```

```
        t[j]='7';
        j++;
        _delay_ms(2000);
}
if(PINB==0b11011011)
{
        lcddata('8');
        t[j]='8';
        j++;
        _delay_ms(2000);
}

if(PINB==0b10111011)
{
        lcddata('9');
        t[j]='9';
        j++;
        _delay_ms(2000);
}

if((j==3))
{
        i=0;
        j=0;
        PORTA=0b01000000;
        _delay_ms(2000);
        PORTA=0b10000000;
        _delay_ms(2000);
        PORTA=0b00000000;
        _delay_ms(200);
        strtx("AT\n\r");
        _delay_ms(100);
        strtx("AT+CMGF=1\n\r");
        _delay_ms(2000);
```

```
        strtx("AT+CMGS=\"+917017242854\"\n\r");
        _delay_ms(2000);
        strtx("AMOUNT DETECTED \n\r");
        strtx(t);

        strtx(" /Rs");
        _delay_ms(2000);
        tx(0x1A);
        lcdcmd(0x01);
        break;
}
}
if((flag==2)&&(a[0]!=1)&&(a[1]!=3)&&(a[2]!=5))
{
        lcdcmd(0x01);
        display("incorrect");
        strtx("AT\n\r");
        _delay_ms(100);
        strtx("AT+CMGF=1\n\r");
        _delay_ms(2000);
        strtx("AT+CMGS=\"+917017242854\"\n\r");
        _delay_ms(2000);
        strtx("WRONG PASSWORD\n\r");
        _delay_ms(2000);
        tx(0x1A);


        _delay_ms(2000);
        lcdcmd(0x01);
        break;
}


}
```

```
}
}
else
{
        display("ACCESS  DENIED");
        _delay_ms(3000);
        lcdcmd(0x01);

}


gotoa;
}
```

5.2 CODE FOR STORING FINGERPRINT ARDUINO

```
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"
//FPS connected to pin 4 and 5 - see previous schemas
FPS_GT511C3 fps(4, 5);
void setup()
{
Serial.begin(9600);
//display messages on the classical serial teminal - DEBUG
fps.UseSerialDebug = true;
fps.Open();
//call Enroll to add fingerprint
enroll();
}
void enroll()
{
// get the first available id for new finger print
```

```
intenrollid = 0;
boolusedid = true;
while (usedid == true)
{
usedid = fps.CheckEnrolled(enrollid);
if (usedid==true) enrollid++;
}
//enrollment start here with the first free id
fps.EnrollStart(enrollid);
Serial.print("Press finger to Enroll #");
Serial.println(enrollid);
// ***** FIRST MEASURE *****
// wait for finger press
while(fps.IsPressFinger() == false) delay(100);
boolbret = fps.CaptureFinger(true);
intiret = 0;
if (bret != false)
{
//has a finger print captured
Serial.println("Remove finger");
// Enroll step 1
fps.Enroll1();
//wait for finger removed
while(fps.IsPressFinger() == true) delay(100);
// ***** SECOND MEASURE *****
// Now we need to check the finger print
// wait for finger press
Serial.println("Press same finger again");
while(fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
Serial.println("Remove finger");
//enroll step 2
```

```
fps.Enroll2();
//wait for finger removed
while(fps.IsPressFinger() == true) delay(100);
// ***** THIRD MEASURE *****
//Check Again the finger print
Serial.println("Press same finger yet again");
while(fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
Serial.println("Remove finger");
iret = fps.Enroll3();
if (iret == 0)
{
//*** SUCCESS third measure are the same -> NOW finger print is stored
Serial.println("Enrolling Successfull");
}
else
{
//*** FAIL For some reason -> NOTHING STORED
Serial.print("Enrolling Failed with error code:");
Serial.println(iret);
}
}
elseSerial.println("Failed to capture third finger");
}
elseSerial.println("Failed to capture second finger");
}
elseSerial.println("Failed to capture first finger");
}
//loop is useless here
void loop()
{
delay(100000);
```

```
}
```

5.3 CODE FOR COMPARE FINGERPRINT (ARDUINO)

```
#include "FPS_GT511C3.h"
#include "SoftwareSerial.h"
//FPS connected to pin 4 and 5 - see previous schemas
FPS_GT511C3 fps(4, 5);
void setup()

Serial.begin(9600);
//display messages on the classical serial teminal - DEBUG
fps.UseSerialDebug = true;
fps.Open();
//call Enroll to add fingerprint
enroll();
}
void enroll()
{
// get the first available id for new finger print
intenrollid = 0;
boolusedid = true;
while (usedid == true)
{
usedid = fps.CheckEnrolled(enrollid);
if (usedid==true) enrollid++;
}
//enrollment start here with the first free id
fps.EnrollStart(enrollid);
Serial.print("Press finger to Enroll #");
Serial.println(enrollid);
// ***** FIRST MEASURE *****
// wait for finger press
while(fps.IsPressFinger() == false) delay(100);
boolbret = fps.CaptureFinger(true);
```

```
intiret = 0;
if (bret != false)
{
//has a finger print captured
Serial.println("Remove finger");
// Enroll step 1
fps.Enroll1();
//wait for finger removed
while(fps.IsPressFinger() == true) delay(100);
// ***** SECOND MEASURE *****
// Now we need to check the finger print
// wait for finger press
Serial.println("Press same finger again");
while(fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
Serial.println("Remove finger");
//enroll step 2
fps.Enroll2();
//wait for finger removed
while(fps.IsPressFinger() == true) delay(100);
// ***** THIRD MEASURE *****
//Check Again the finger print
Serial.println("Press same finger yet again");
while(fps.IsPressFinger() == false) delay(100);
bret = fps.CaptureFinger(true);
if (bret != false)
{
Serial.println("Remove finger");
iret = fps.Enroll3();
if (iret == 0)
{
//*** SUCCESS third measure are the same -> NOW finger print is stored
```

```
Serial.println("Enrolling Successfull");
}
else
{
//*** FAIL For some reason -> NOTHING STORED
Serial.print("Enrolling Failed with error code:");
Serial.println(iret);
}
}
elseSerial.println("Failed to capture third finger");
}
elseSerial.println("Failed to capture second finger");
}
elseSerial.println("Failed to capture first finger");
}
//loop is useless here
void loop()
{
delay(100000);
}
```

CHAPTER 6

CONCLUSION

Implementing the project is very much customer friendly, easy operation at finger tips using AADHAR CARD which is a must . Use of AADHAR card technology makes the system secure as it gives unique identification to the customer. Once after installing, maintenance is very easy. Long queues, time and efforts is saved. The main aim of the project is to give the business, substantial savings in labor, time and staff training costs. The software coding in Embedded C code gives the required flexibility to design the transactions as per customer's requirement and convenience. Due to automation, fraudulent transactions are eliminated.

Following are the advantages of the AADHAR based self service petrol pump

- Improves the use of GSM module in digital transactions.
- Increases the efficiency of the services of petrol pump (especially in late night hours).
- Enhances the best utilization of TIME.
- ~~People below18 years of age will not able to~~ access the petrol pump service
- Increases the overall profit of the petrol pump

CHAPTER 7

FUTURE ENHANCEMENT

- The project can be linked with the real- time database of the car user with the help of IOT (Internet of things).
- An option of “NEW REGISTERATION” can also be included in the project asking by the user to add his details like AADHAR, fingerprint , car number etc.
- By linking the project with database available on the national website and AADHAR the user can directly pay the money from his bank account which is linked to it’s AADHAR number.
- An additional ADC based “NO SMOKE SYSTEM” can also be added to this project.

REFERENCES

1. Mayur Gawade, Sandesh Gawde, Sonal Kanade, “A review paper on Automated Fuel Pump Security System”,International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 3, Issue: 11, Page No.: 6156 – 6158, November 2015

2. Peter Adole, Joseph M. Mom, Gabriel A. Igwue,” RFID Based Security Access Control System with GSM Technology”, American Journal of Engineering Research, Vol.5, Issue-7, pp-236-242

3. Sanal Malhotra, “Banking locker System with Odor identification and security question using RFID and GSM technology”, International Journel of Advances in Electronics Engineering – IJAEE Volume 4: Issue 3 [ISSN 2278-215X

4 . Fawzi M. Al-Naima, Mohammad M. Hasan, “Design and Implementation RFID-based Fuel Dispensing System”, International Journal of Computing and Network Technology,ISSN 2210-1519, 12 June 20