**INTRODUCTION**

For The innovation of technologies advances at an exponential rate in this period of fast change. Many businesses are making investments in innovations that guarantee consumer happiness at all levels. Technology in customer service, which offers new and reliable systems for customers, is one of the most common developments. Shopping carts, usually referred to as shopping trolleys, are a way to temporarily transfer your purchases before checking out. Since their creation, shopping carts have seen relatively few alterations. Most of the expansions have been made to change its weight and capacity. But as technology has advanced, some of the company's research has led to the creation of a user-friendly shopping system. The shopping cart, for instance, has a touch panel and a RFID Reader attached. The customer can use the touch panel to learn about the details, promotions and the location of products.

In addition, when a consumer places an item into the trolley, the RFID reader scans it, and the details of the product is presented on the touch panel display. Without waiting in a long line for the payment process, the consumer will only need to make a payment depending on the amount displayed to the cashier. The described upgrade to the shopping cart can save client shopping time, money and provide a better shopping experience. Due to primarily fast expanding application to track products through the food supply chain, RFID and associated technologies have seen an explosion in attention over the past two years. Due to the relatively high cost of RFID deployment and the extremely low profit margin of supermarket goods, item-level tagging was not yet practicable, therefore these applications instead monitor Store-Keeping Units (SKU) rather than specific product items. One can easily imagine a scenario in which every item in a supermarket is marked with an RFID label, shopping carts have RFID readers and perhaps even on-board computers that can identify items placed in the cart and display information and promotions that have been wirelessly or wired retrieved from the system's back-end. The introduction of RFID technology at the item level would also enable fast checkout lanes that scan all merchandise at once, eliminating lines consistently cited as one of the worst elements of grocery shopping

**OBJECTIVES & SCOPE OF STUDY**

* To develop the abilities such as working in groups, sharing responsibilities, initiative, perseverance
* To make a device which can be easily handled by all

**Scope / Objectives of the project:**

1. A Smart Cart System refers to a technological advanced and automated system use to enhance the human as well as industrial experience.
2. It typically involves the integration of various technologies such as RFID technology, sensors and user-friendly interfaces into carts. For example, in shopping mall the system aims to streamline the shopping process by providing features like assistance, security, store management, etc. The system also aims to reduce human efforts.
3. With advancements in technology, the scope of the Smart Cart system can continue to expand, offering even more innovative features and benefits.

* Few more areas were Smart Carts can be used,
* Hospitality Industry
* Manufacturing and Logistics
* Airports
* Libraries
* Retail inventory Management

1. **Market survey**

|  |  |  |  |
| --- | --- | --- | --- |
| **S NO** | **Question** | **Yes** | **No** |
| **1** | **Whether this project is cost effective** | **9** | **1** |
| **2** | **Whether this project will serve the purpose** | **8** | **2** |
| **3** | **Whether this project will reduce the human efforts** | **7** | **3** |
| **4** | **Whether its eco friendly** | **9** | **1** |
| **5** | **Whether skill operator required** | **0** | **10** |
| **6** | **Whether all the parts are available** | **10** | **0** |
| **7** | **Whether its safety** | **10** | **0** |
| **8** | **Whether its multi functioning** | **9** | **1** |
| **9** | **Whether it will serve the requirements** | **10** | **0** |
| **10** | **Whether mechanism is simple** | **10** | **0** |

**SCHEDULE OF THE PROJECT**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **JAN** | | **FEB** | | **MAR** | |
| **Sl No** |  |  |  |  |  |  |  |
| 1 | Data collection |  |  |  |  |  |  |
| 2 | Literature survey |  |  |  |  |  |  |
| 3 | planning |  |  |  |  |  |  |
| 4 | designing |  |  |  |  |  |  |
| 5 | drafting |  |  |  |  |  |  |
| 6 | purchase |  |  |  |  |  |  |
| 7 | manufacturing |  |  |  |  |  |  |
| 8 | testing |  |  |  |  |  |  |
| 9 | report |  |  |  |  |  |  |

**Expected Outcome of the project:**

1. The expected outcome of Smart Cart System using ultrasonic sensor and RFID technology is to create a cart that can autonomously follow a human user while avoiding obstacles and to generate bill automatically.
2. This system is expected to provide a convenient and personalized experience by eliminating the need for the user to push or navigate the cart manually.
3. By utilizing ultrasonic sensor, the system is expected to detect the presence of a human in front of it and calculate the distance to maintain a safe following distance.
4. By RFID technology, the system is expected to identify the item present in the cart and generate the total bill.
5. The Smart Cart System is expected to track the user’s movements and adjust its speed and direction according.

Overall the expected outcome of the Smart Cart System is to create auser-friendly, efficient and personalized experience that simplifies the task.

**PROJECT DELIVERABLES:**

1**. Fully Functional**:

- High-speed in accurate

- Modular design allowing for easy upgrades and maintenance.

2. **User-Friendly Interface:**

- Intuitive software interface for seamless user interaction.

- the application is easy and can be operated by anyone with minimum knowledge

3. **Material Compatibility:**

- The material used is standard material.

- The material is easy available in market.

4. **Safety Features:**

- Implementation of safety mechanisms to ensure secure operation.

- User manual outlining safety protocols and guidelines.

5. **Energy Optimization:**

- Efficient energy consumption with a focus on sustainability.

- Energy consumption report and recommendations for improvement.

**KEY MILESTONES:**

1. **Project Initiation:**

- Define project scope, objectives, and team roles.

- Conduct initial literature review on existing agro technologies.

2. **Design Phase:**

- Develop detailed specifications for the machine.

- Select and source materials for fabrication

- Proper design parameters are used to select the material

3. **Prototyping:**

- Test basic functionality and identify potential design improvements.

- To look for field stability and environment condition

4**. User Interface Development:**

- Implement a user-friendly application for operation.

- Conduct usability testing and gather feedback for improvements.

5. **Material Integration:**

- Explore and integrate advanced materials suitable for various applications.

- the material are painted in order to protect from corrosion

6. **Safety Features and Testing**

- Implement safety mechanisms in both hardware and software.

- Conduct thorough safety testing and ensure compliance with standards.

**CONSTRAINTS:**

1. Limited Materials. ...
2. Restricted Build Size. ...
3. Post Processing. ...
4. Large Volumes. ...
5. Part Structure. ...
6. Reduction in Manufacturing Jobs. ...
7. Design Inaccuracies. ...
8. Copyright Issues.

**WORK BREAKDOWN STRUCTURE:**

The work of the project is divided into:

* Literature review:

The brief literature study is carried out in this phase. This involves studying the various litearature papers and then arriving at the problem definition.

* Material selection:

After the designing is complete, the brief market survey is done to study different materials available in the market. The most optimum materials are chosen for the project.

* fabrication:

The system which forms the components to house all the other components is fabricated in this phase.with soldering and fitting

* Testing

Finally everything is tested for performance.

**RISK ANALYSIS:**

Risk analysis is a crucial aspect of project management to identify potential challenges that may arise during the development of the project. Here is a comprehensive risk analysis for the project:

1. Technical Risks:

as the technology used to manufacture the product is a risk because the machine may not be available to machine like laser machine and all so the technology plays the important role

2. Environmental Risks:

As the working environment many also effect the manufacturing process because of the environment condition if its cold climate or rainy the corrosion take place and in the summer due to excess heat the oil viscosity may also vary due to which lead of leak of lubricant oil

3. Resource Risks:

* Material Availability: Delays in the availability of specific materials may impede the fabrication process and extend project timelines.
* Budget Overruns: due to the amount factors the amount may not be available to purchase all the material at once

4. Operational Risks:

* Maintenance Challenges: Ensuring the long-term functionality of the machine vehicle its components may be challenging, leading to increased maintenance requirements.

5. Regulatory and Compliance Risks:

* Environmental Regulations
* Safety Standards: Ensuring the machine complies with safety standards, especially if deployed in public spaces, could be a regulatory challenge.

6. Project Management Risks:

* Timeline Delays: Unexpected setbacks in any phase of the project could lead to delays in the overall timeline.
* Communication Issues: Ineffective communication among team members may result in misunderstandings and hinder the project's progress.

7. Testing and Validation Risks:

* Performance Variability: Variability in factory conditions may impact the performance of the system, requiring extensive testing across diverse scenarios.

**COMPONENTS**

1. Ardiuno uno
2. LCD
3. I2C module
4. Ultrasonic sensor
5. L298 Driver
6. RFID ID CARD
7. Battery
8. motors
9. wires
10. form sheet
11. tray

**ARDIUNO CONTROLLER**



The Uno is a microcontroller board based on the [ATmega328P.](http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf) It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, 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.. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

**Technical specs**

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](http://www.atmel.com/Images/doc8161.pdf) |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 Ma |
| Flash Memory | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |

**Programming:**

The Uno can be programmed with the [Arduino Software](https://www.arduino.cc/en/Main/Software) (IDE). Select "Arduino/Genuino Uno" from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](https://www.arduino.cc/en/Reference/HomePage) and [tutorials](https://www.arduino.cc/en/Tutorial/HomePage).

The ATmega328 on the Uno comes preprogrammed with a [bootloader](https://www.arduino.cc/en/Hacking/Bootloader?from=Tutorial.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](http://www.atmel.com/Images/doc2525.pdf), [C header files](http://www.atmel.com/dyn/resources/prod_documents/avr061.zip)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using [Arduino ISP](https://www.arduino.cc/en/Main/ArduinoISP) or similar; see [these instructions](https://www.arduino.cc/en/Hacking/Programmer) for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in the Arduino repository. 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 rese ing 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](http://www.atmel.com/products/microcontrollers/default.aspx) (Windows) or the [DFU programmer](http://dfu-programmer.github.io/) (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](http://forum.arduino.cc/index.php/topic,111.0.html) for more information.

**Warnings**

The Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

**Differences with other boards**

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.

**Power**

The Uno board 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 from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become 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 Uno 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.

IOREF. This pin on the Uno board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

**Memory**

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](https://www.arduino.cc/en/Reference/EEPROM)).

Input and Output

See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical.

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode()](https://www.arduino.cc/en/Reference/PinMode), [digitalWrite()](https://www.arduino.cc/en/Reference/DigitalWrite), and [digitalRead()](https://www.arduino.cc/en/Reference/DigitalRead) functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

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 driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

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.  
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 Uno has a number of facilities for communicating with a computer, another Uno board, 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](https://www.arduino.cc/en/Guide/Windows#toc4). The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the 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](https://www.arduino.cc/en/Reference/SoftwareSerial) allows serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the [documentation](https://www.arduino.cc/en/Reference/Wire) for details. For SPI communication, use the [SPI library](https://www.arduino.cc/en/Reference/SPI).

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](http://forum.arduino.cc/index.php/topic,22974.0.html) for details.

Revisions

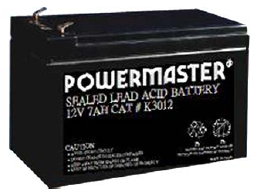
Revision 3 of the board has the following new features:

1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

Stronger RESET circuit.

Atmega 16U2 replace the 8U2.

**Battery**

**Features**

• Multi-cell design for economy of installation and maintenance

• Individual valve for each cell

• High quality ABS case and cover

• Absorbent Glass Mat (AGM) technology for efficient gas recombination of

up to 99% and freedom from electrolyte maintenance

• Not restricted for air transport

• Not restricted for surface transport

• Long life

• Float/cycle use

• Low self-discharge rate

• Use in any position

Keep batteries at a comfortable temperature - between about 50 and 80 degrees Fahrenheit, and ideally around 75 degrees. Don't let batteries freeze, and keep them off cold concrete floors. If the batteries are kept in the home, they should be in a separate, sealed and well-ventilated space out of reach of children.  Promptly recharge partially drained batteries. Some watt-hour meters show the percentage of charge left in the battery (much like most laptop computers). A battery that is only drained to about 90 percent of its capacity will last more than 10 times longer than a battery that is regularly completely drained

**16X2 LCD interface (JHD162A – HD44780 compatible Display Controller):-**

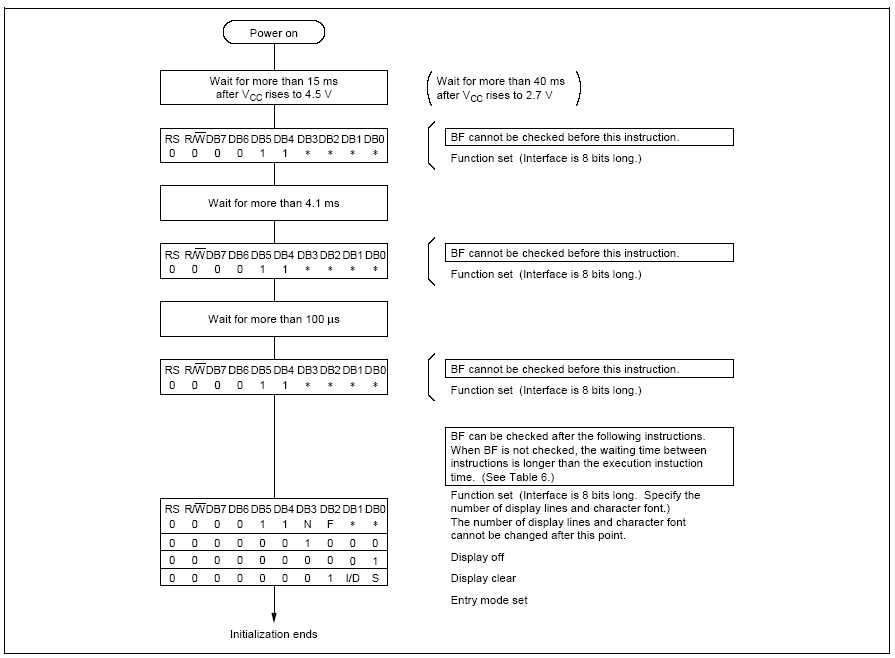
The LCD interfaced to the system was 16 characters and 2 lines type JHD162A from Jin Hang Displays Co. China. The LCD module consists of a Hitachi HD44780 type LCD controller with a Character generator ROM and Display Data Ram. The module consists of 16 interface pins and signals as listed below.

|  |  |  |
| --- | --- | --- |
| **Pin No.** | **Signal** | **Description** |
| 1 | GND | Power Ground |
| 2 | VCC | +5VDC Power Input |
| 3 | VEE | Contrast Voltage |
| 4 | RS | Register Select  (0=Command, 1=Data) |
| 5 | R/W’ | Read/Write Select  (0=Write, 1=Read) |
| 6 | EN | Strobe Input  (A high to low transition latches data to input registers) |
| 7-14 | D0-D7 | Data Input / Output Lines |
| 15 | A | Backlight +5V Power |
| 16 | K | Backlight Power Ground |

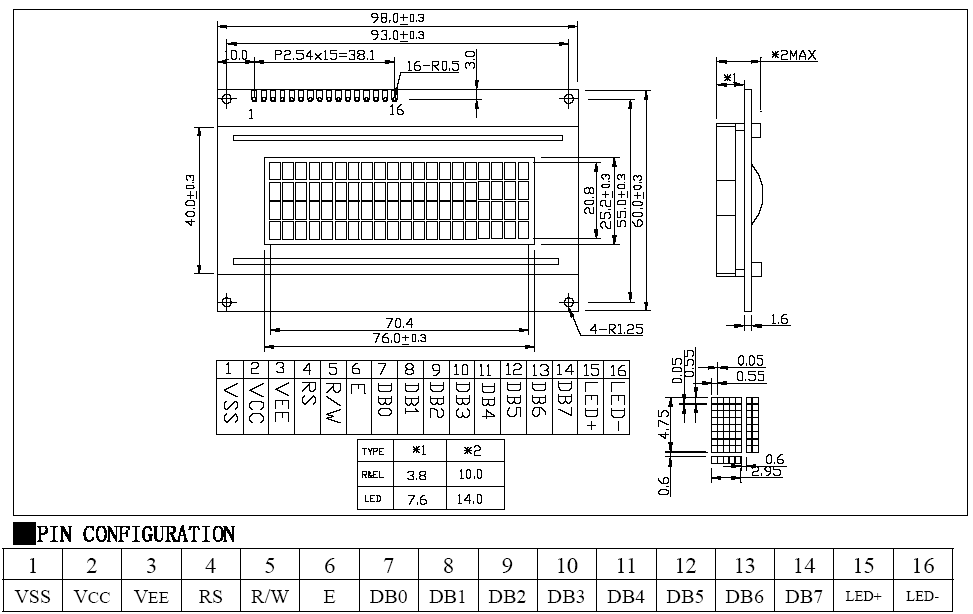


**Photograph showing the 16X2 LCD type JHD162A**

The LCD display can be interfaced to the system in either 4 bit or 8 bit data interface length mode but we have selected an 8 bit interface mode for faster display data update. The previous mode could be used in applications where the number of I/O pins availability is limited i.e. A smaller MCU with less I/O Pins. The LCD is interfaced to the MCU with data lines connected to port 1 of the 8051 and control lines RS, EN connected to port 3.2, 3.3 respectively. The R/W control line is permanently tied to ground for write only operations to the LCD display. The program is coded BASCOM8051 basic assembler for 8051 MCU’s. The assembled program in the form of an Intel HEX file is generated by the assembler which is further uploaded to the 8051 target board using a In circuit programming software – Flash Magic developed by ESA, for in circuit programming of NXP micro controllers.



***LCD Display Initialization Procedure***

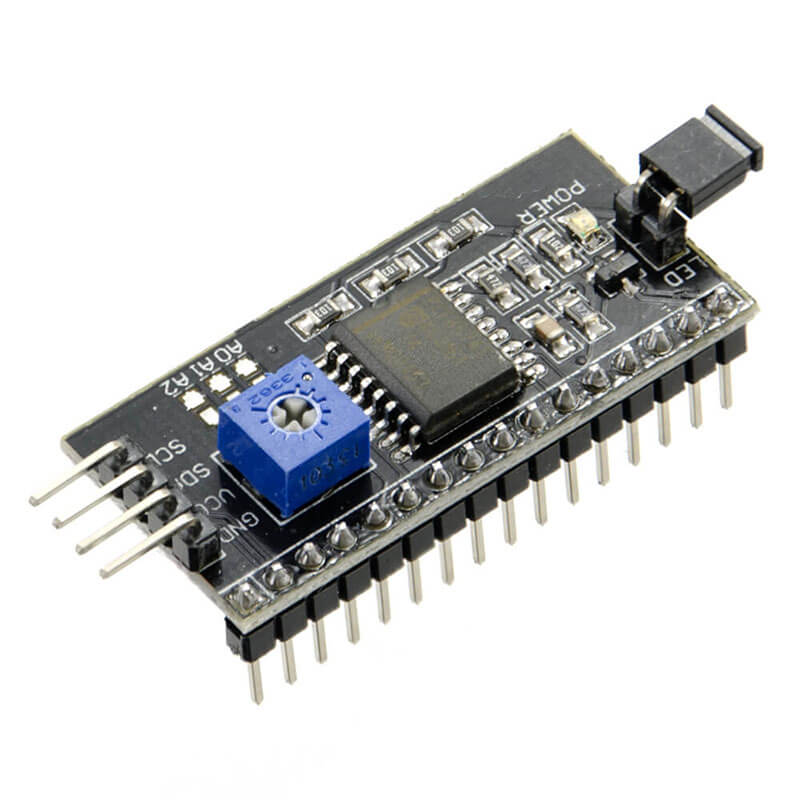
****

***LCD Display Module Mechanical Dimensions & Pin Configurations***

**I2C MODULE**

Very useful module to interface serial connection to parallel data, specially used for LCD displays etc. I2C Module has inbuilt PCF8574 I2C chip that converts I2C serial data to parallel data for the LCD display. I2C modules are currently supplied with a default I2C address of either 0x27 or 0x3F, you can check which version by verifying underside of the module. If there a 3 sets of pads labelled A0, A1, & A2 then the default address will be 0x3F. If there are no pads the default address will be 0x27.

The module has a contrast adjustment pot on the underside of the display. This may require adjusting for the screen to display text correctly.



**Specifications:**

* Operating Voltage: 5V
* Backlight and Contrast is adjusted by potentiometer
* Serial I2C control of LCD display using PCF8574
* Come with 2 IIC interface, which can be connected by Dupont Line or IIC dedicated cable
* Compatible for 16x2 LCD
* This is another great IIC/I2C/TWI/SPI Serial Interface
* With this I2C interface module, you will be able to realize data display via only 2 wires.

**Applications**

* Used for various display control units/projects.

|  |  |
| --- | --- |
|  |  |
|  |  |

**HC-SR04 ULTRASONIC (US) SENSOR**

As shown above the **HC-SR04 Ultrasonic sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

**Distance = Speed × Time**

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.



**HC-SR04 Sensor Features**

* Operating voltage: +5V
* Theoretical  Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

### **Ultrasonic Sensor Pin Configuration**

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Vcc | The Vcc pin powers the sensor, typically with +5V |
| 2 | Trigger | Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave. |
| 3 | Echo | Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor. |
| 4 | Ground | This pin is connected to the Ground of the system |

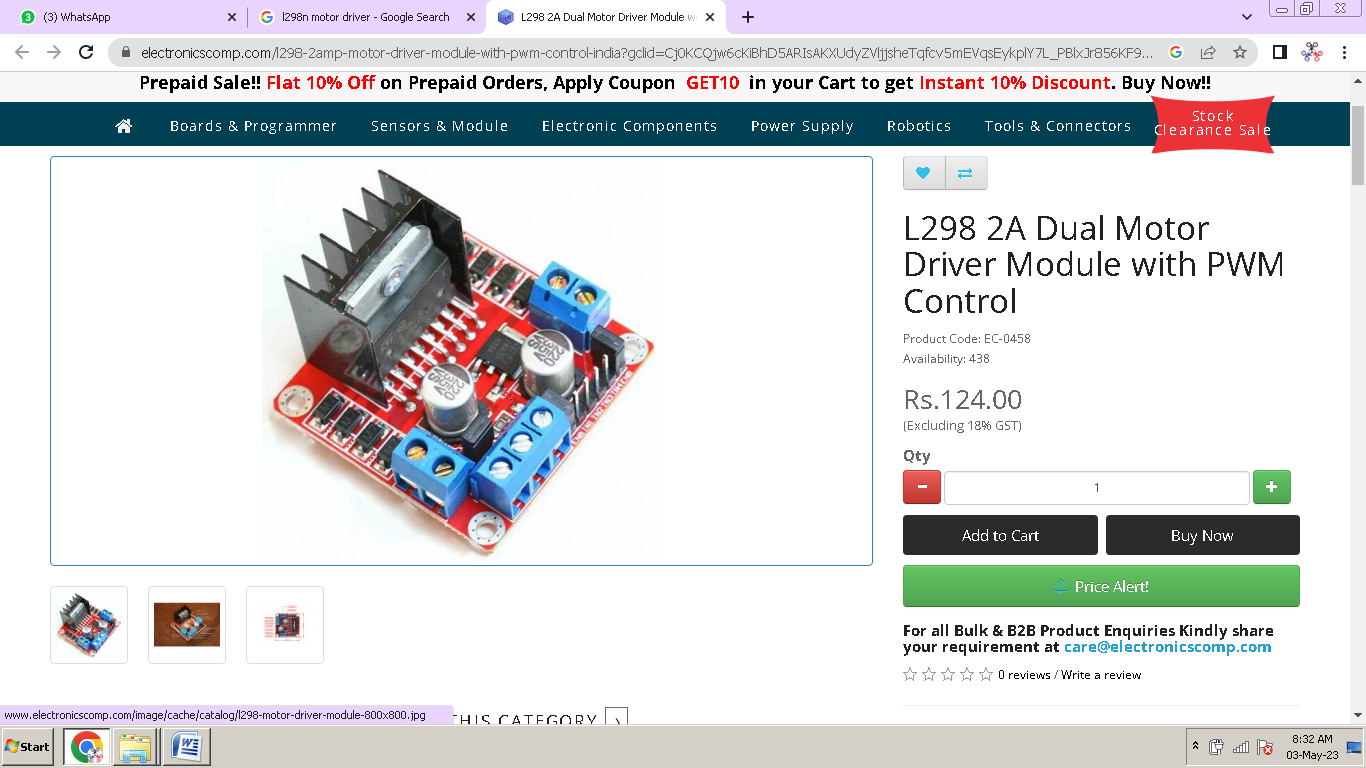
**RFID CARDS**

RC522 - RFID Reader / Writer 13.56MHz with Cards Kit includes a 13.56MHz RF reader cum writer module that uses an RC522 IC and two S50 RFID cards. The MF RC522 is a highly integrated transmission module for contact-less communication at 13.56 MHz. RC522 supports ISO 14443A/MIFARE mode. RC522 - RFID Reader features an outstanding modulation and demodulation algorithm to serve effortless RF communication at 13.56 MHz. The S50 RFID Cards will ease up the process helping you to learn and add the 13.56 MHz RF transition to your project. The module uses SPI to communicate with microcontrollers. The open-hardware community already has a lot of projects exploiting the RC522 – RFID Communication, using Arduino.



**Features:-**

* Integrated MF RC522
* 13.56MHz contactless communication card chip.
* Low-voltage, low-cost, small size of the non-contact card chip to read and write.
* Suitable for Smart meters and portable handheld devices.
* Advanced modulation and demodulation concept completely integrated in all types of 13.56MHz passive contactless communication methods and protocols.
* 14443A compatible transponder signals.
* ISO14443A frames and error detection.
* Supports rapid CRYPTO1 encryption algorithm, terminology validation MIFARE products.
* MFRC522 support MIFARE series of high-speed non-contact communication, two-way data transmission rate up to 424kbit/s.
* Low cost, and ideal for user equipment development.
* The reader and RF card terminal design meets advanced applications development and production needs.
* Can be directly loaded into the various reader molds, very convenient.  
  **Specifications:-**
* Operating Current :13-26mA / DC 3.3V
* Idle Current :10-13mA / DC 3.3V
* Sleep Current: < 80uA
* Peak Current: < 30mA
* Operating Frequency: 13.56MHz
* Supported card types: mifare1 S50, mifare1 S70 MIFARE Ultralight, mifare Pro, MIFARE DESFire
* Environmental Operating Temperature: -20 - 80 degrees Celsius
* Environmental Storage Temperature: -40 - 85 degrees Celsius
* Relative humidity: relative humidity 5% - 95%
* Reader Distance: ≥ 50mm / 1.95" (mifare 1)
* Module Size: 40mm × 60mm
* Module interface: SPI
* Data transfer rate: Maximum 10Mbit/s

**L298  MOTOR DRIVER**

This L298 Based Motor Driver Module is a high power motor driver perfect for driving DC Motors and Stepper Motors. It uses the popular L298 motor driver IC and has the onboard 5V regulator which it can supply to an external circuit. It can control up to 4 DC motors, or 2 DC motors with directional and speed control

This motor driver is perfect for robotics and mechatronics projects and perfect for controlling motors from microcontrollers, switches, relays, etc. Perfect for driving DC and Stepper motors for micro mouse, line following robots, robot arms, etc.

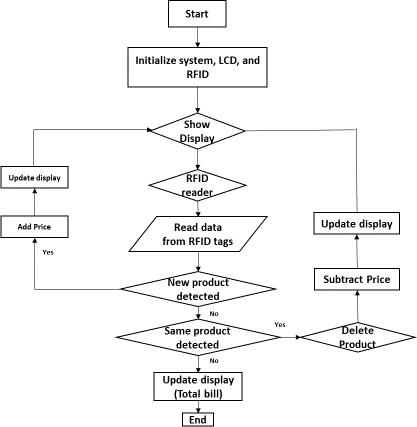
An H-Bridge is a circuit that can drive a current in either polarity and be controlled by Pulse Width Modulation (PWM).

Pulse Width Modulation is a means of controlling the duration of an electronic pulse. In motors try to imagine the brush as a water wheel and electrons as the flowing droplets of water. The voltage would be the water flowing over the wheel at a constant rate, the more water flowing the higher the voltage. Motors are rated at certain voltages and can be damaged if the voltage is applied to heavily or if it is dropped quickly to slow the motor down. Thus PWM. Take the water wheel analogy and think of the water hitting it in pulses but at a constant flow. The longer the pulses the faster the wheel will turn, the shorter the pulses, the slower the water wheel will turn. Motors will last much longer and be more reliable if controlled through PWM.

**WORKING**

.

1. First switch on button to start the all component like Arduino, LCD, ultrasonic sensor, RFID reader etc.
2. Every Product has different and RFID tag which contains unique id.
3. When shopper drops any product in the cart, first he to scan the product and RFID reader read the tag.
4. The information of the product is extracted and the same time billing information is updated.
5. Rate of product displays on LCD and at last sum is also displayed.
6. Product that are not needed are taken out but we have to re-scan the product in front of RFID reader to subtract the price.
7. Updated price is displayed on LCD.
8. Final bill is ready and we can do payment on counter

****

**BLOCK DIAGRAM**

**Ardiuno**

**LCD**

**RFID**

**Battery**

**Ultrasonic sensor 1**

**Ardiuno**

**UNO**

**Ultrasonic sensor 2**

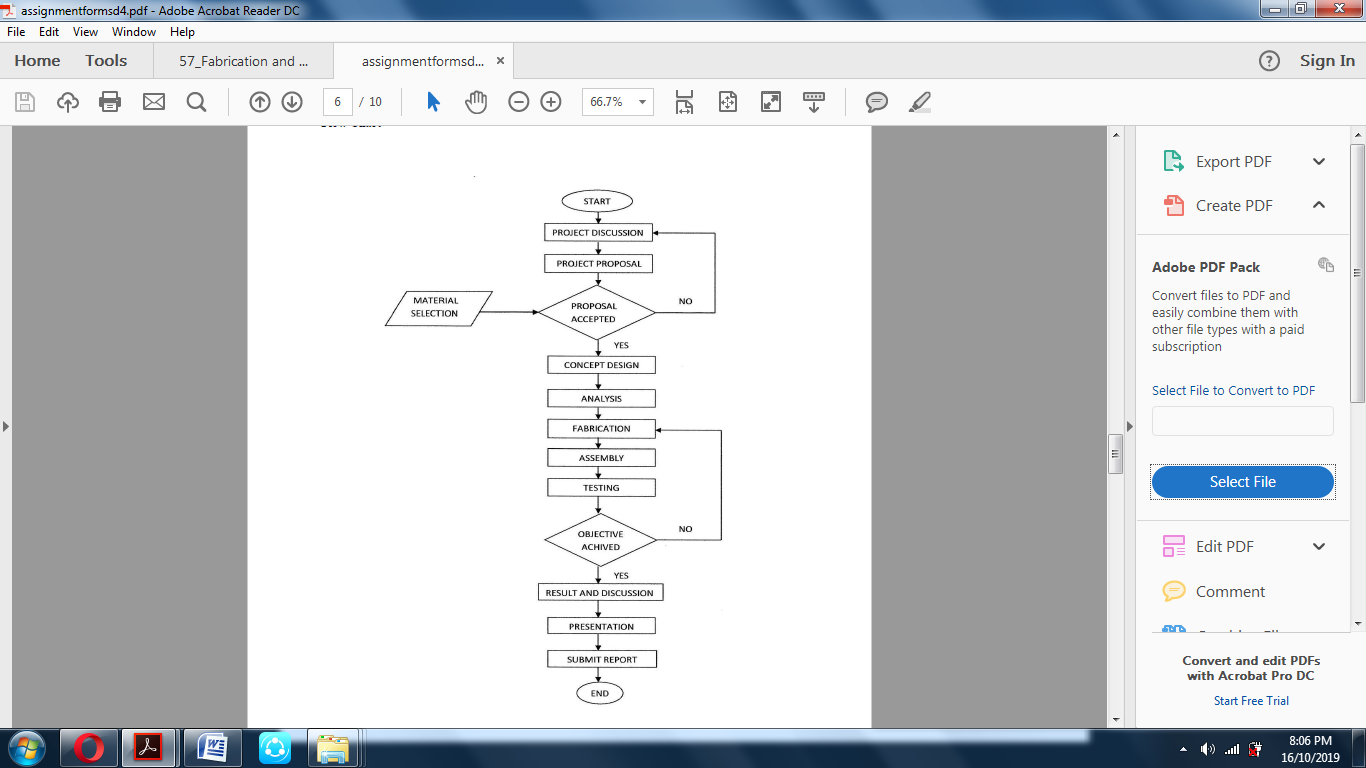
**Ultrasonic sensor 3**

**L298 driver**

**Motor 1**

**Motor 2**

**METHODOLOGY**

****

**CODING**

* 1. ROBOT

#include <SoftwareSerial.h>

#include "NewPing.h"

int LmotorA=8;

int LmotorB=9;

int RmotorA=10;

int RmotorB=11;

#define TRIGGER\_PINL 2

#define ECHO\_PINL 3

NewPing sonarL(TRIGGER\_PINL, ECHO\_PINL);

#define TRIGGER\_PINC 4

#define ECHO\_PINC 5

NewPing sonarC(TRIGGER\_PINC, ECHO\_PINC);

#define TRIGGER\_PINR 6

#define ECHO\_PINR 7

NewPing sonarR(TRIGGER\_PINR, ECHO\_PINR);

void setup()

{

pinMode(LmotorA,OUTPUT);

pinMode(LmotorB,OUTPUT);

pinMode(RmotorA,OUTPUT);

pinMode(RmotorB,OUTPUT);

Serial.begin(9600);

}

void loop()

{

if(sonarC.ping\_cm()<50)

{

Serial.println("C: ");

Serial.println(sonarC.ping\_cm());

// delay(200);

Forward();

}

else if(sonarR.ping\_cm()<20)

{

Serial.println("R: ");

Serial.println(sonarR.ping\_cm());

//delay(200);

Right();

}

else if(sonarL.ping\_cm()<20)

{

Serial.println("L: ");

Serial.println(sonarL.ping\_cm());

// delay(500);

Left();

}

else

{

Stop();

}

}

void Forward()

{

digitalWrite(LmotorA,HIGH);

digitalWrite(LmotorB,LOW);

digitalWrite(RmotorA,HIGH);

digitalWrite(RmotorB,LOW);

Serial.println("F");

delay(50);

}

void Right()

{

digitalWrite(LmotorA,HIGH);

digitalWrite(LmotorB,LOW);

digitalWrite(RmotorA,LOW);

digitalWrite(RmotorB,HIGH);

Serial.println("R");

delay(500);

}

void Left()

{

digitalWrite(LmotorA,LOW);

digitalWrite(LmotorB,HIGH);

digitalWrite(RmotorA,HIGH);

digitalWrite(RmotorB,LOW);

Serial.println("L");

delay(500);

}

void Stop()

{

digitalWrite(LmotorA,LOW);

digitalWrite(LmotorB,LOW);

digitalWrite(RmotorA,LOW);

digitalWrite(RmotorB,LOW);

Serial.println("S");

}

* 1. CART

#include <SPI.h>

#include <MFRC522.h>

#define SS\_PIN 10

#define RST\_PIN 9

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

#include <Wire.h>

int itemcount1;

int itemqty1;

int itemcount2;

int itemqty2;

int itemcount3;

int itemqty3;

void setup()

{

Serial.begin(9600);

SPI.begin();

mfrc522.PCD\_Init();

Serial.println("Approximate your card to the reader...");

Serial.println();

lcd.backlight();

lcd.init();

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("Angadi");

lcd.setCursor(0, 1);

lcd.print("College");

delay(3000);

lcd.clear();

lcd.setCursor(1, 0);

lcd.print(" E/C");

lcd.setCursor(0, 1);

lcd.print(" DEPARTMENT");

delay(3000);

lcd.clear();

}

void loop()

{

a();

product();

// totalcost();

}

void a()

{

lcd.setCursor(1, 0);

lcd.print("ADD THE PRODUCT");

lcd.setCursor(0, 1);

lcd.print(" IN CART");

}

void product()

{

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

return;

}

if ( ! mfrc522.PICC\_ReadCardSerial())

{

return;

}

Serial.print("UID tag :");

String content= "";

byte letter;

for (byte i = 0; i < mfrc522.uid.size; i++)

{

Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");

Serial.print(mfrc522.uid.uidByte[i], HEX);

content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));

content.concat(String(mfrc522.uid.uidByte[i], HEX));

}

Serial.println();

Serial.print("Message : ");

content.toUpperCase();

if (content.substring(1) == "F3 D7 62 AC")

{

itemcount1++;

itemqty1++;

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("COLGATE");

lcd.setCursor(0, 1);

lcd.print(" 50Rs/pc");

delay(2000);

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("QUANTITY: ");

lcd.print(itemqty1);

lcd.setCursor(0, 1);

lcd.print("COLGATE COST:");

lcd.print(itemcount1\*50);

delay(2000);

lcd.clear();

}

else if (content.substring(1) == "43 A0 32 0E")

{

itemcount2++;

itemqty2++;

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("SHAMPO");

lcd.setCursor(0, 1);

lcd.print(" 20Rs/pc");

delay(2000);

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("QUANTITY: ");

lcd.print(itemqty2);

lcd.setCursor(0, 1);

lcd.print("SHAMPO COST: ");

lcd.print(itemcount2\*20);

delay(2000);

lcd.clear();

}

else if (content.substring(1) == "C3 92 31 0E")

{

itemcount3++;

itemqty3++;

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("SOAP");

lcd.setCursor(0, 1);

lcd.print(" 15Rs/pc");

delay(2000);

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("QUANTITY: ");

lcd.print(itemqty3);

lcd.setCursor(0, 1);

lcd.print("SOAP COST: ");

lcd.print(itemcount3\*15);

delay(2000);

lcd.clear();

}

else if (content.substring(1) == "63 72 EF A6")

{

lcd.clear();

lcd.setCursor(1, 0);

lcd.print("TOTAL BILLING");

delay(3000);

lcd.clear();

totalcost();

}

else

{

}

}

void totalcost()

{

int sum=itemcount1\*50+itemcount2\*20+itemcount3\*15;

int sum1=itemqty1+itemqty2+itemqty3;

lcd.clear();

lcd.print("TOTAL QUANTITY");

lcd.setCursor(0,1);

lcd.print(sum1);

delay(2000);

lcd.clear();

lcd.setCursor(1,0);

lcd.print("Cart Total:");

lcd.setCursor(0,1);

lcd.print(sum);

lcd.print(" Rs");

delay(4000);

lcd.clear();

lcd.setCursor(1,0);

lcd.print("THANK YOU FOR");

lcd.setCursor(0,1);

lcd.print("SHOPPING WITH US");

delay(4000);

lcd.clear();

}

**ADVANTAGES**

1. Multi operation
2. Easy in operation.
3. Low cost
4. Simple construction.
5. Adaptable.
6. High capacity.
7. Performance.
8. Automatic operated.
9. Environmental friendly.
10. Easy to setup
11. Light weight.
12. Easy maintenance.

**COST EXPENDITURE**

|  |  |  |
| --- | --- | --- |
| **Materials Cost** | | |
| **SL No** | **Particulars** | **Cost in Rs** |
| 1 | Ardiuno | 600 |
| 2 | motor | 400 |
| 3 | L298 driver | 1200 |
| 4 | LCD | 200 |
| 5 | I2C Module | 250 |
| 6 | RFID | 500 |
| 7 | Ultrasonic sensor | 200 |
| 8 | Battery | 600 |
| 9 | Wire | 200 |
| 10 | Other | 2000 |
| **Process Cost** | | |
| 1 | Coding | 200 |
| 2 | Fabrication | 200 |
|  | | |
| 1 | Project Report | 1000 |
| 2 | Miscellaneous | 2000 |
| **Total** | | **Rs /-** |

**FABRICATION**

**Soldering Guide:-**

**How to Solder:-**

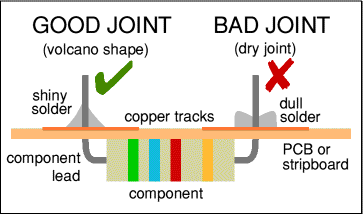
**First a few safety precautions:**

* **Never touch the element or tip of the soldering iron.**   
  They are very hot (about 400°C) and will give you a nasty burn.
* **Take great care to avoid touching the mains flex with the tip of the iron.**   
  The iron should have a heatproof flex for extra protection. An ordinary plastic flex will melt immediately if touched by a hot iron and there is a serious risk of burns and electric shock.
* **Always return the soldering iron to its stand when not in use.**   
  Never put it down on your workbench, even for a moment!
* **Work in a well-ventilated area.**   
  The smoke formed as you melt solder is mostly from the flux and quite irritating. Avoid breathing it by keeping you head to the side of, not above, your work.
* **Wash your hands after using solder.**   
  Solder contains lead which is a poisonous metal.

**Preparing the soldering iron:**

* **Place the soldering iron in its stand and plug in.**   
  The iron will take a few minutes to reach its operating temperature of about 400°C.
* **Dampen the sponge in the stand.**   
  The best way to do this is to lift it out the stand and hold it under a cold tap for a moment, then squeeze to remove excess water. It should be damp, not dripping wet.
* **Wait a few minutes for the soldering iron to warm up.**   
  You can check if it is ready by trying to melt a little solder on the tip.
* **Wipe the tip of the iron on the damp sponge.**   
  This will clean the tip.
* **Melt a little solder on the tip of the iron.**   
  This is called 'tinning' and it will help the heat to flow from the iron's tip to the joint. It only needs to be done when you plug in the iron, and occasionally while soldering if you need to wipe the tip clean on the sponge.

**You are now ready to start soldering:-**



* + **Hold the soldering iron like a pen, near the base of the handle.**   
    Imagine you are going to write your name! Remember to never touch the hot element or tip.
* **Touch the soldering iron onto the joint to be made.**   
  Make sure it touches both the component lead and the track. Hold the tip there for a few seconds and...
* **Feed a little solder onto the joint.**   
  It should flow smoothly onto the lead and track to form a volcano shape as shown in the diagram. Apply the solder to the joint, not the iron.
* **Remove the solder, then the iron, while keeping the joint still.**   
  Allow the joint a few seconds to cool before you move the circuit board.
* **Crocodile clip, photograph © Rapid ElectronicsInspect the joint closely.**   
  It should look shiny and have a 'volcano' shape. If not, you will need to reheat it and feed in a little more solder. This time ensure that **both** the lead and track are heated fully before applying solder.

|  |
| --- |
| Crocodile clip |

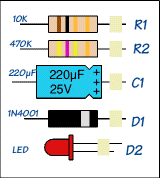
**Using a heat sink:-**

Some components, such as transistors, can be damaged by heat when soldering so if you are not an expert it is wise to use a heat sink clipped to the lead between the joint and the component body. You can buy a special tool, but a standard crocodile clip works just as well and is cheaper.   
**Further information:-**

For a much more detailed guide to soldering, including troubleshooting, please see the [Basic Soldering Guide](http://www.epemag.wimborne.co.uk/solderfaq.htm) on the Everyday Practical Electronics Magazine website.

**Soldering Advice for Components:-**

It is very tempting to start soldering components onto the circuit board straight away, but please take time to identify all the parts first. You are much less likely to make a mistake if you do this!

1. **Stick all the components onto a sheet of paper using sticky tape.**
2. **Identify each component** and write its name or value beside it.
3. **Add the code (R1, R2, C1 etc.) if necessary.**   
   Many projects from books and magazines label the components with codes (R1, R2, C1, D1 etc.) and you should use the project's parts list to find these codes if they are given.
4. **Resistor values** can be found using the resistor colour code which is explained on our [Resistors](http://www.kpsec.freeuk.com/components/resist.htm) page. You can print out and make your own [Resistor Colour Code Calculator](http://www.kpsec.freeuk.com/components/rescal.htm) to help you.
5. **Capacitor values** can be difficult to find because there are many types with different labelling systems! The various systems are explained on our [Capacitors](http://www.kpsec.freeuk.com/components/capac.htm) page.

Some components require special care when soldering. Many must be placed the correct way round and a few are easily damaged by the heat from soldering. Appropriate warnings are given in the table below, together with other advice which may be useful when soldering.

For more detail on specific components please see the [Components](http://www.kpsec.freeuk.com/compon.htm) page or click on the component name in the table.

**For most projects it is best to put the components onto the board in the order given below:-**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Components** | **Pictures** | **Reminders and Warnings** |
| 1 | [IC Holders](http://www.kpsec.freeuk.com/components/ic.htm) (DIL sockets) | IC holder | Connect the correct way round by making sure the notch is at the correct end.  Do NOT put the ICs (chips) in yet. |
| 2 | [Resistors](http://www.kpsec.freeuk.com/components/resist.htm) | resistor | No special precautions are needed with resistors. |
| 3 | [Small value capacitors](http://www.kpsec.freeuk.com/components/capac.htm) (usually less than 1µF) | small value capacitors | These may be connected either way round.  Take care with polystyrene capacitors because they are easily damaged by heat. |
| 4 | [Electrolytic capacitors](http://www.kpsec.freeuk.com/components/capac.htm) (1µF and greater) | electrolytic capacitor | Connect the correct way round. They will be marked with a + or - near one lead. |
| 5 | [Diodes](http://www.kpsec.freeuk.com/components/diode.htm) | diodes | Connect the correct way round.  Take care with germanium diodes (e.g. OA91) because they are easily damaged by heat. |
| 6 | [LEDs](http://www.kpsec.freeuk.com/components/led.htm) | LED | Connect the correct way round.  The diagram may be labeled a or + for anode and k or - for cathode; yes, it really is k, not c, for cathode! The cathode is the short lead and there may be a slight flat on the body of round LEDs. |
| 7 | [Transistors](http://www.kpsec.freeuk.com/components/tran.htm) | transistors | Connect the correct way round.  Transistors have 3 'legs' (leads) so extra care is needed to ensure the connections are correct.  Easily damaged by heat. |
| 8 | [Wire Links](http://www.kpsec.freeuk.com/components/connect.htm) between points on the circuit board. | single core wire  single core wire | Use single core wire; this is one solid wire which is plastic-coated.  If there is no danger of touching other parts you can use tinned copper wire, this has no plastic coating and looks just like solder but it is stiffer. |
| 9 | [Battery clips](http://www.kpsec.freeuk.com/components/connect.htm), **buzzers** and other parts with their own wires |  | Connect the correct way round. |
| 10 | [Wires](http://www.kpsec.freeuk.com/components/connect.htm) to parts off the circuit board, including [switches](http://www.kpsec.freeuk.com/components/switch.htm), [relays](http://www.kpsec.freeuk.com/components/relay.htm), [variable resistors](http://www.kpsec.freeuk.com/components/vres.htm) and **loudspeakers**. | stranded wire  stranded wire | You should use stranded wire which is flexible and plastic-coated.  Do not use single core wire because this will break when it is repeatedly flexed. |
| 11 | [ICs (chips)](http://www.kpsec.freeuk.com/components/ic.htm) | 555 timer IC | Connect the correct way round.  Many ICs are static sensitive.  Leave ICs in their antistatic packaging until you need them, then earth your hands by touching a metal water pipe or window frame before touching the ICs.  Carefully insert ICs in their holders: make sure all the pins are lined up with the socket then push down firmly. |

**What is solder:-**

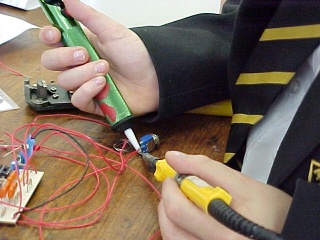
****

|  |
| --- |
|  |
| **Reels of solder** |

Solder is an alloy (mixture) of tin and lead, typically 60% tin and 40% lead. It melts at a temperature of about 200°C. Coating a surface with solder is called 'tinning' because of the tin content of solder. Lead is poisonous and you should always wash your hands after using solder.

Solder for electronics use contains tiny cores of flux, like the wires inside a mains flex. The flux is corrosive, like an acid, and it cleans the metal surfaces as the solder melts. This is why you must melt the solder actually on the joint, not on the iron tip. Without flux most joints would fail because metals quickly oxidize and the solder itself will not flow properly onto a dirty, oxidized, metal surface.

**Disordering:-**

At some stage you will probably need to desolder a joint to remove or re-position a wire or component.

**There are two ways to remove the solder:**

|  |
| --- |
|  |
| **Using a desoldering pump (soldersucker)** |

**With a disordering pump (solder sucker)**

* Set the pump by pushing the spring-loaded plunger down until it locks.
* Apply both the pump nozzle and the tip of your soldering iron to the joint.
* Wait a second or two for the solder to melt.
* Then press the button on the pump to release the plunger and suck the molten solder into the tool.
* Repeat if necessary to remove as much solder as possible.
* The pump will need emptying occasionally by unscrewing the nozzle

**With solder remover wick (copper braid):-**

* Apply both the end of the wick and the tip of your soldering iron to the joint.
* As the solder melts most of it will flow onto the wick, away from the joint.
* Remove the wick first, then the soldering iron.
* Cut off and discard the end of the wick coated with solder.

|  |
| --- |
|  |
|  |

After removing most of the solder from the joint(s) you may be able to remove the wire or component lead straight away (allow a few seconds for it to cool). If the joint will not come apart easily apply your soldering iron to melt the remaining traces of solder at the same time as pulling the joint apart, taking care to avoid burning yourself.

**First Aid for Burns:-**

Most burns from soldering are likely to be minor and treatment is simple:

* Immediately cool the affected area under gently running cold water.   
  Keep the burn in the cold water for at least 5 minutes (15 minutes is recommended). If ice is readily available this can be helpful too, but do not delay the initial cooling with cold water.
* Do not apply any creams or ointments. The burn will heal better without them. A dry dressing, such as a clean handkerchief, may be applied if you wish to protect the area from dirt.
* Seek medical attention if the burn covers an area bigger than your hand.

**To reduce the risk of burns:-**

* Always return your soldering iron to its stand immediately after use.
* Allow joints and components a minute or so to cool down before you touch them.
* Never touch the element or tip of a soldering iron unless you are certain it is cold.

# FUTURE SCOPE OF THE PROJECT

We feel the project that we have done has a good future scope in any the main constraint of this device is the high initial cost but has low operating costs.

Savings resulting from the use of this device will make it pay for itself with in short period of time & it can be a great companion in any field

The device affords plenty of scope for modifications, further improvements & operational efficiency, which should make it commercially available & attractive. If taken up for commercial production and marketed properly, we are sure it will be accepted in the industry.

**CONCLUSION**

We have taken up this project as real challenge, as we were not experience in the field. We started our work on this project facing new hurdles initially.

The maneuverability of the device is quite good and the handling is quite simple. For commercial purpose one can improve the efficiency of the device effectively by increasing the size of the device.

**REFERANCE**

[1] Bachelor’s thesis Business Administration, Business Academy 20 Smart Shopping Cart System. Author: -r: Jussie Phakinin

[2] IJDER | VOLUME 5 | ISSN:2321-9939 (2016) Smart Shopping Cart For Automatic Billing in Supermarket

[3] 7th International Conference on Communication , Computing and Virtualization (2017) Smart Cart wth Automatic Billing product Information , Product Recommendation using RFID & Zigbee with Anti-Theft.

[4] ISSN : 2456-3307 (www.ijsrcsit.com)(2018) Review on Smart Shopping Cart

[5] Development of an Intelligent Smart Shopping Cart System(2019)

[6] Smart Cart with Automatic Billing product Information ,Product Recommendation using RFID & Zigbee with AntiTheft(2019)

[7] Automatic voice-activated adjustment of shopping cart(2019) by- Ashish Duggal

[8] Journal of Automation and Automobile Engineering (e-ISSN:2582-3159) {2019} Smart Shopping Trolley

[9] RFID Cloud Smart Cart System.(apj@ieee.org) (2020) Implementation of RFID Cloud Smart cart System

[10] Design and Construction of a Smart shopping Trolley 2020 Capstone project BSC Electrical & Electronical Engineering

[11] Smart Trolley using Smart Phone and Arduino January 2017 (Journal of Electrical & Electronic Systems)

[12] Modelling of Future Automatic Trolley SystemGRD Journals | Global Research and Development Journal for Engineering National Conference on Emerging Research Trend in Electrical and Electronics Engineering (ERTE’19) | May 2019

[13] RFID Based Advanced Shopping trolley for Super Market Journal of Chemical and Pharmaceutical Sciences (June 2017)

[14] RFID Based Advanced Shopping trolley for Super Market Journal of Chemical and Pharmaceutical Sciences (June 2017)

[15] RFID-Cloud smart cart system [Publisher: IEEE (2016)]