CAFE MANAGEMENT

Aim:

To design an automation system that will allow a digital menu to be displayed on an LCD monitor. The customer's order will be received directly at the kitchen.

Team members:

Lokhandwala Burhanuddin Haiderbhai
Mohammed Umar
Motwani Aniket
Prashant Paliwal
Preeti Jannu
Singh Saurav
Zeal Shah

Mentors:

Chandu G
Devanshee Tanti
Prabhakar Jaiswal
Pranav Premlani

COMPONENTS USED IN THIS PROJECT

- ATmega32 microcontroller (2 nos)
- Hc-05 Bluetooth (2 nos)
- Keypad
- LCD (2 nos)
- Pulldown resistors

WORKING :-

- Firstly as our customer will give us orders, we got our data which we had to transmit to the kitchen. Then we send our data to the microcontroller ATMEGA32 via Keypad. As soon as the microcontroller get's the data as input from the keypad it is ready to transmit the data to the kitchen. Then the microcontroller transmits the data to another microcontroller which is in the kitchen with the help of Bluetooth module. And here we are using UART protocol for transmitting the data to the kitchen. As soon as the microcontroller is ready, it accepts the data from the first microcontroller. After receiving the data, this microcontroller is interfaced with LCD and hence this microcontroller sends the data to LCD & hence LCD shows the order given by the customer.

ATmega32 :-

- It is a microcontroller having 4 ports each having 8 pins and 8 special pins summing up as 40 pins.
- It has 32kb of programmable flash memory.
- ATmega 32 is a 8 bit high performance microcontroller.
- It can work in a wide range of voltage input, i.e, 0V to 5V.

HC-05 BLUETOOTH :-

- HC-05 Bluetooth Module is an easy to use Bluetooth module, designed for transparent wireless serial connection setup.
- It can be used both as slave and master.

KEYPAD(3x4 mobile keypad):-
Input device used to take the order from the customers.

LCD(16x2) :-
Used to display the menu to the customers.

LCD(20x4) :-
Used to display the order given by the customers to the chefs.

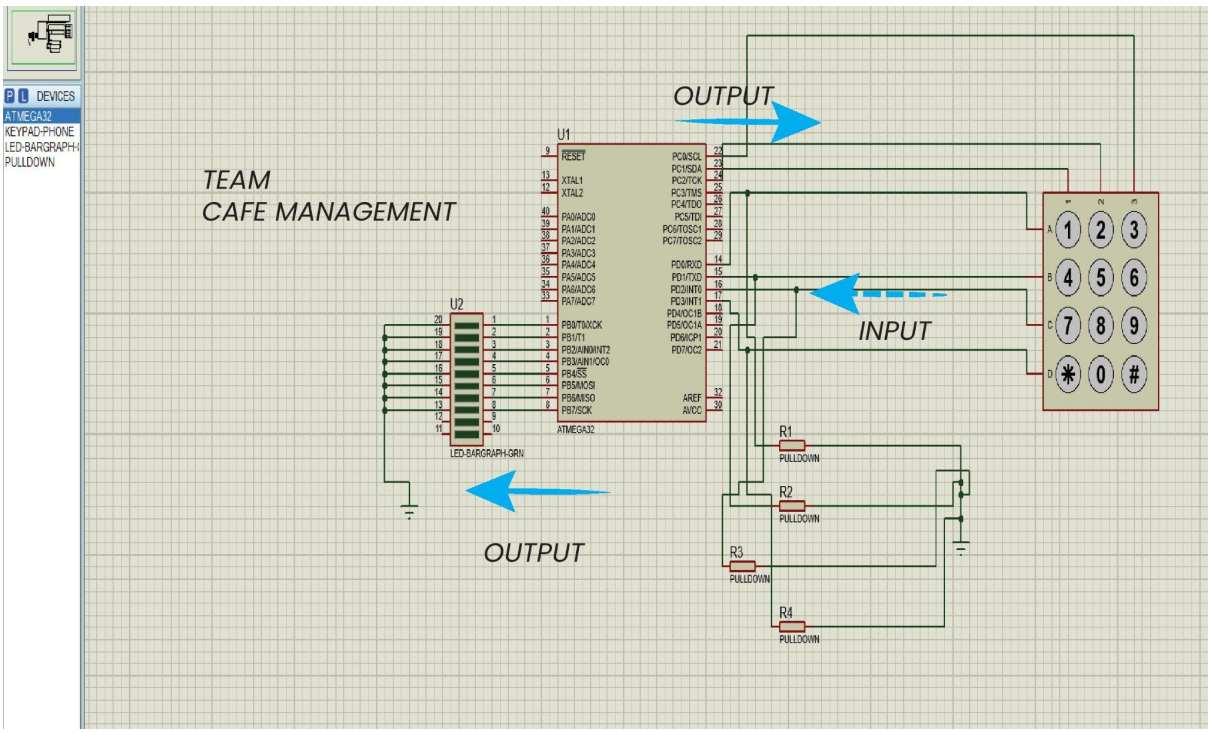
PULLDOWN RESISTORS:-
Used in Keypad interfacing to make flow excess current to the ground voltage.

KEYPAD INTERFACING :-

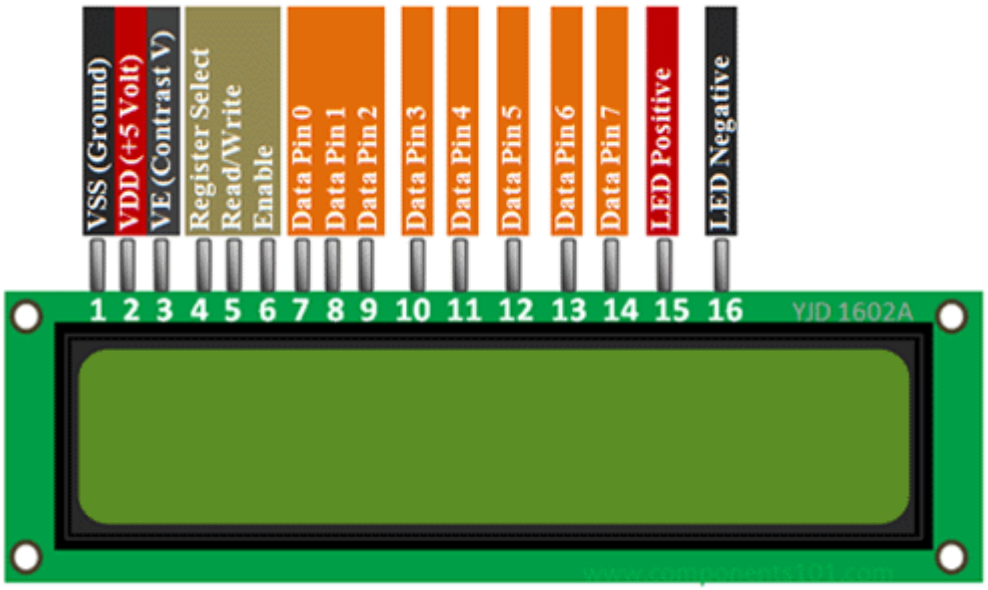
The keypad can be simply broken down as a matrix of 12 elements having a specific row number and a column number.

We used this logic in order to get info regarding which key is pressed.

For this what we did was provide power or made each column high one by one and took the input from the row which was pressed at the time of a specific column being high..



LCD INTERFACING :-



- The term LCD stands for liquid crystal display.
- We used a 16×2 LCD module .

It is named 16×2 LCD because it has 16 Columns and 2 Rows. So, it can display (16×2=32) 32 characters in total.

Pinouts :-

- Pin1 (Ground/Source Pin): Connected to ground.
- Pin2 (VCC/Source Pin): Connected to power supply.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): Used to select the command or data register.For command mode(RS=0) and for data mode(RS=1)
- Pin5 (Read/Write/Control Pin): Toggles between read and write mode.
- Pin 6 (Enable/Control Pin): This pin should be held high to execute the Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): Used for sending and receiving data and commands from the microcontroller.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.

Working :-

1. Busy flag: First check if the LCD is busy or not by using a busy flag.
    a. When the pin D7=1,the LCD is busy taking care of internal operations and will not accept any new information.
    b. When the pin D7=0, the LCD is ready to receive new information.

2.Read/Write mode: Then selecting the R/W pin allows the user to write information to the LCD or read information from it.

a.R/W =1 when reading

b.R/W =0 when writing

3.RS, register select: Next,select the RS pin.A 16×2 LCD has two registers like data register and command register.

a. If RS=0,the instruction command register is selected ,allowing the user to send a command such as clear display,cursor at home,etc.

b. If RS=1,the data register is selected,allowing the user to send data to be displayed on the LCD.

4.E,enable :When data is supplied to data pins, a high-to-low pulse must be applied to this pin in order for the LCD to latch in the data present at the data pins.This pulse must be a minimum of 450 ns wide.
5. Send information: After selecting the modes and registers, now to send actual data or command use the data pins(D0-D7).
6.Delay: LCD is very busy performing a lot of tasks at the same time.So should give it some time delay for processing data and implementing it.

16×2 LCD Commands :-

The commands of LCD 16X2 include the following.

- For 0x01, the LCD command will be the clear LCD screen
- For 0x02, the LCD command will be returning home
- For 0x04, the LCD command will be decrement cursor
- For 0x06, the LCD command will be Increment cursor
- For 0x05, the LCD command will be Shift display right
- For 0x07, the LCD command will be Shift display left
- For 0x08, the LCD command will be Display off, cursor off
- For 0x0A, the LCD command will be cursor on and display off
- For 0x0C, the LCD command will be cursor off, display on
- For 0x0E, the LCD command will be cursor blinking, Display on
- For 0x0F, the LCD command will be cursor blinking, Display on
- For 0x10, the LCD command will be Shift cursor position to left
- For 0x14, the LCD command will be Shift cursor position to the right
- For 0x18, the LCD command will be Shift the entire display to the left
- For 0x1C, the LCD command will be Shift the entire display to the right
- For 0x80, the LCD command will be Force cursor to the beginning (1st line)
- For 0xC0, the LCD command will be Force cursor to the beginning (2nd line)
- For 0x38, the LCD command will be 2 lines and 5×7 matrix

Referred Website:https://www.electronicwings.com/avr-atmega/interfacing-lcd-16x2-in-4-bit-mode-with-atmega-16-32-

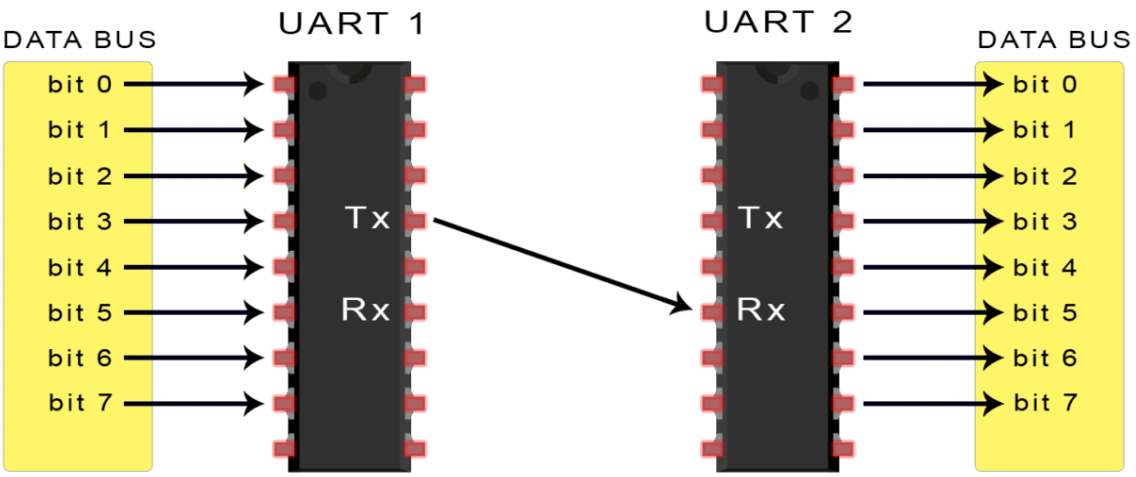
UART :-

- Uart stands for Universal Asynchronous receiver transmitter. It's not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or a stand-alone IC. A UART's main purpose is to transmit and receive serial data.
- Asynchronous Communication: Transfer of a byte data in the framed structure at a time.

❖ Frame structure in Asynchronous communication:

- **START bit:** It is a bit with which serial communication starts and it is always low.
- **Data bits packet**: Data bits can be 5 to 9 bits packet. Normally we use 8 data bit packets, which is always sent after the START bit.
- **STOP bit**: This is one or two bits. It is sent after the data bits packet to indicate the end of the frame. The stop bit is always logic high.

  In an asynchronous serial communication frame, the first START bit followed by the data byte and at last STOP bit forms a 10-bit frame. Sometimes the last bit is also used as a parity bit.

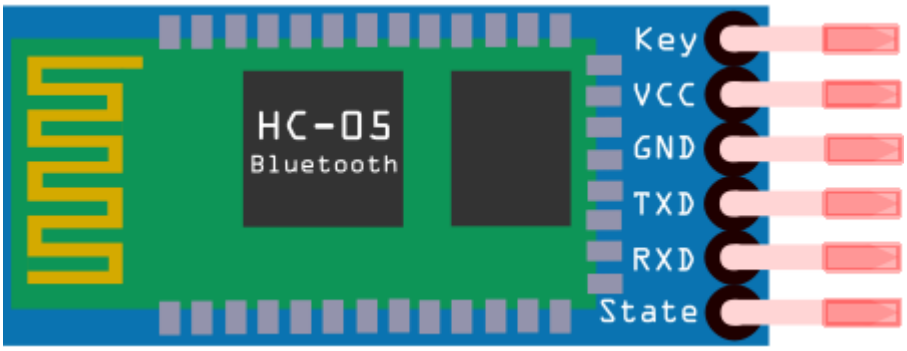


From Sender controlling device we get a set of parallel data through data bus and then that parallel data is converted to serial data with help of shift registers then, transmission is done of that serial data from Tx terminal of UART1 and received at Rx terminal of UART2 their another shift register is used to convert serial data to parallel data and the data is passed to the receiving controlling device via data bus.

Referred Website:

[Basics of UART Communication (circuitbasics.com)]

BLUETOOTH



There are two bluetooth modules HC05 and HC06 modules.

- The HC-05 can be a master or slave. This means the HC-05 can initiate a connection to another device.

- The HC-06 is a slave only.The HC-06 can only accept a connection from another device.
- We used HC05 module at both transmission end and receiving end.

Pinouts

1. **KEY/En:**This pin is used to bring the Bluetooth module in AT commands mode. By default, this pin operates in data mode. The Key/EN pin should be high to operate Bluetooth in command mode. In HC-05, the default baud speed in command mode is 38400bps and 9600 in data mode.
2. **VCC:**Used to power the Bluetooth module. Give 5V / 3.3 V to this pin
3. **GND:**The ground pin of the module
4. **TXD:**Connect this pin with the RXD pin of the Microcontroller. This pin transmits Serial data (wireless signals received by the Bluetooth module are converted by module and transmitted out serially on this pin)
5. **RXD:**Connect this pin to the TXD pin of the Microcontroller. The HC-05 Bluetooth module receives the data from this pin and then transmits it wirelessly.
6. **STATE:**It is used to check if the module is connected or not. It acts as a status indicator.

Working:

1. Connect pins of bluetooth module with microcontroller:

   RXD-TXD

   TXD-RXD

2. To pair the two HC05 bluetooth module:

   Go to the virtual serial port driver and create a new pair between COM1 and COM2.By default this will make one module as master and other as slave.

3. Set the baud rate:

   Open properties of the bluetooth module and set the baud rate in both.Also set the physical port as COM1 for one and COM2 for the other.

4. And our bluetooth module is ready to transfer information.

ERRORS WE FACED

- Never use assignment operator "==" in the arguments of conditional statements i.e. if-else statements. As the datatype on the LHS and RHS of assignment operators must be the same… Rather we can use
  1. Bit_is_set operator
  2. Logical and (&) or (|) as well as Bitwise Shift << >> operators.

  Like .. : PIND&(1<<2)
- X-Y Mirror the LED Bargraph as it is set as default having anode on the RHS but in the schematic we required it on the LHS side so we mirrored it.
- Use Pull-Down Registers : They allow the excess current to flow through the ground thus our bargraph works seamlessly.
- While giving the data from keypad we declared it as char data type but we were giving it the value of a number(int data type).
- While setting the baud rate in the bluetooth module, set it as same as it is in the microcontroller.And also check the frequency that has been set in the microcontroller.It should be in accordance with the baud rate set.
- We got stuck when we used the two bluetooth modules with the same name, so the name must be different for both the bluetooth modules.
- Before importing our hex file into microcontroller, we need to make sure that our atmel project folder consists of only one .c code file, because we saved multiple code files under the same project and were getting an error in our proteus simulation.