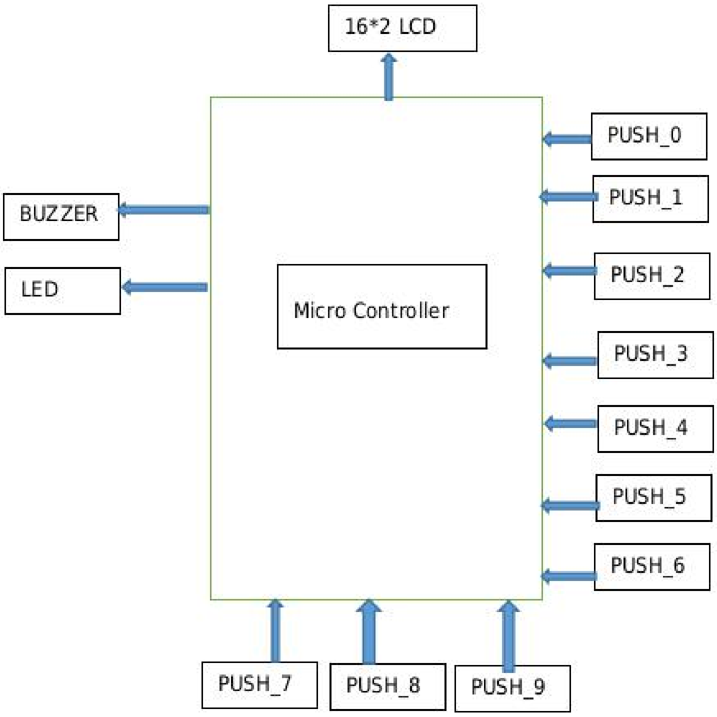
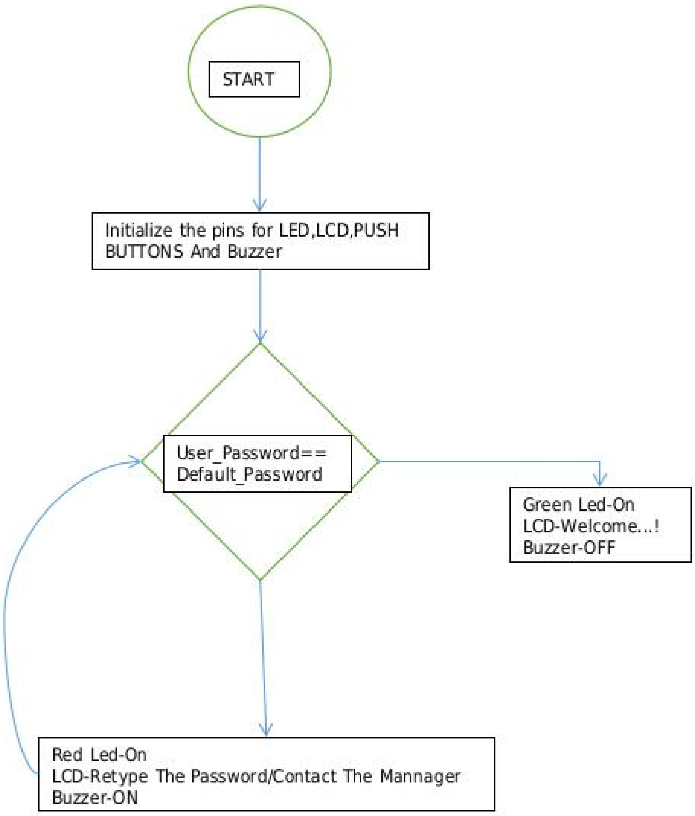
1.Password Protected Lock System

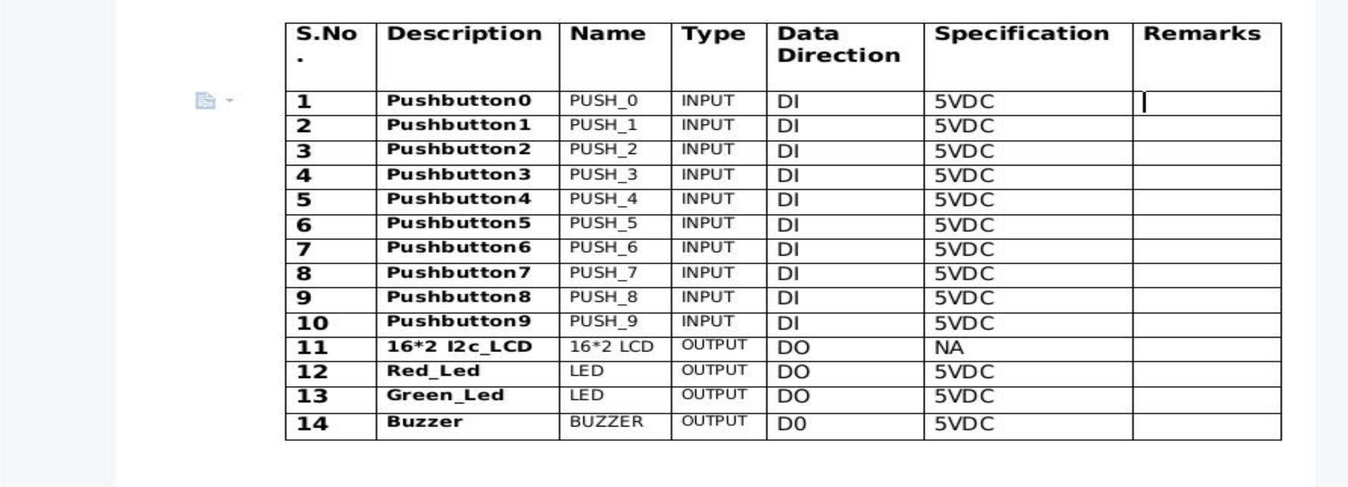
1. Block Diagram



2. FlowChart


START
Initialize the pins for LED,LCD,PUSH BUTTONS And Buzzer
User_Password==
Default_Password
Green Led-On
LCD-Welcome...!
Buzzer-OFF
Red Led-On
LCD-Retype The Password/Contact The Mannager
Buzzer-ON

3. Table



S.No.	Description	Name	Type	Data Direction	Specification	Remarks
1	**Pushbutton0**	PUSH_0	INPUT	DI	5VDC	
2	**Pushbutton1**	PUSH_1	INPUT	DI	5VDC	
3	**Pushbutton2**	PUSH_2	INPUT	DI	5VDC	
4	**Pushbutton3**	PUSH_3	INPUT	DI	5VDC	
5	**Pushbutton4**	PUSH_4	INPUT	DI	5VDC	
6	**Pushbutton5**	PUSH_5	INPUT	DI	5VDC	
7	**Pushbutton6**	PUSH_6	INPUT	DI	5VDC	
8	**Pushbutton7**	PUSH_7	INPUT	DI	5VDC	
9	**Pushbutton8**	PUSH_8	INPUT	DI	5VDC	
10	**Pushbutton9**	PUSH_9	INPUT	DI	5VDC	
11	**16*2 I2c_LCD**	16*2 LCD	OUTPUT	DO	NA	
12	**Red_Led**	LED	OUTPUT	DO	5VDC	
13	**Green_Led**	LED	OUTPUT	DO	5VDC	
14	**Buzzer**	BUZZER	OUTPUT	D0	5VDC	

4. C Code

```
#include <LiquidCrystal.h>
#include<string.h>
const int rs = A5, en = A4, d4 = A3, d5 = A2, d6 = A1, d7 = A0;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
unsigned int arduino_button_pins[]={0,1,2,3,4,5,6,7,8,9};
unsigned int button_present_values[]={0,0,0,0,0,0,0,0,0,0};
unsigned int button_past_values[]={0,0,0,0,0,0,0,0,0,0};
const int Green_LED=10;
const int Red_LED=11;
const int Buzzer=12;

int key_pressed()
{
  for(uint8_t button=0;button<10;button++)
    {
      int present_state = digitalRead(arduino_button_pins[button]);
      int previous_state= button_past_values[button];

      if(present_state)
      {
        if(present_state != previous_state)
        {
          button_past_values[button] = present_state;
          char str[10];
          sprintf(str,"KEY:%d",button);//Mearge the Strings
          lcd.setCursor(0,1);
          lcd.write(str);
          return button;
        }
      }
      else
      {
        button_past_values[button] = 0;
      }
      delay(50);
    }

}

int press_button()
{

if(digitalRead(0)||digitalRead(1)||digitalRead(2)||digitalRead(3)||digitalRead(
4)||digitalRead(5)
  ||digitalRead(6)||digitalRead(7)||digitalRead(8)||digitalRead(9))
  { return 1; }
  else
  { return 0; }
}

void setup()
{
  for(int i=0; i<10;i++)
  {
    pinMode(i,INPUT);
  }
  pinMode(Green_LED,OUTPUT);
  pinMode(Red_LED,OUTPUT);
  pinMode(Buzzer,OUTPUT);
  lcd.begin(16, 2);
  lcd.write("ENTER PIN");
}
  const int Preset_Pin=2332; //Password prefixed saved
  int pinByUser[] ={0,0,0,0} ;
  int keySequence = 0;
  int Final_Pin = 0 ;
void loop()
{
   while(press_button())
    {
      if(keySequence<4){
         pinByUser[keySequence]=key_pressed();
         lcd.setCursor(6,1);
         char pin[4];
         sprintf(pin,"DIGIT%d-%d",keySequence+1,pinByUser[keySequence]);
         lcd.write(pin);
      }
      else if(keySequence==4){
          lcd.setCursor(6,1);
          for(int a=0; a<4; a++)
          {
            Final_Pin = (Final_Pin * 10) + pinByUser[a];
          }
          char pin[4];
         sprintf(pin,"PIN:%d",Final_Pin);
         lcd.write(pin);
        delay(50);
        if(Final_Pin == Preset_Pin){
          lcd.setCursor(6,1);
        lcd.write("Welcome");
        lcd.setCursor(0,0);
        lcd.write("ACCESS AUTHORIZED !!");
       digitalWrite(Green_LED,HIGH);
        }else{
           lcd.setCursor(6,1);
        lcd.write("-!FAILED!-");
        digitalWrite(Red_LED,HIGH);
        }
      }
      else{
        lcd.setCursor(0,0);
        lcd.write("MAX LIMIT REACHED");
      }
      delay(1200);
      keySequence++;
    }
}
```

5. Simulation Circuit