CircuitDigest
(/)

Home (/) / Call and Message using Arduino and GSM Module
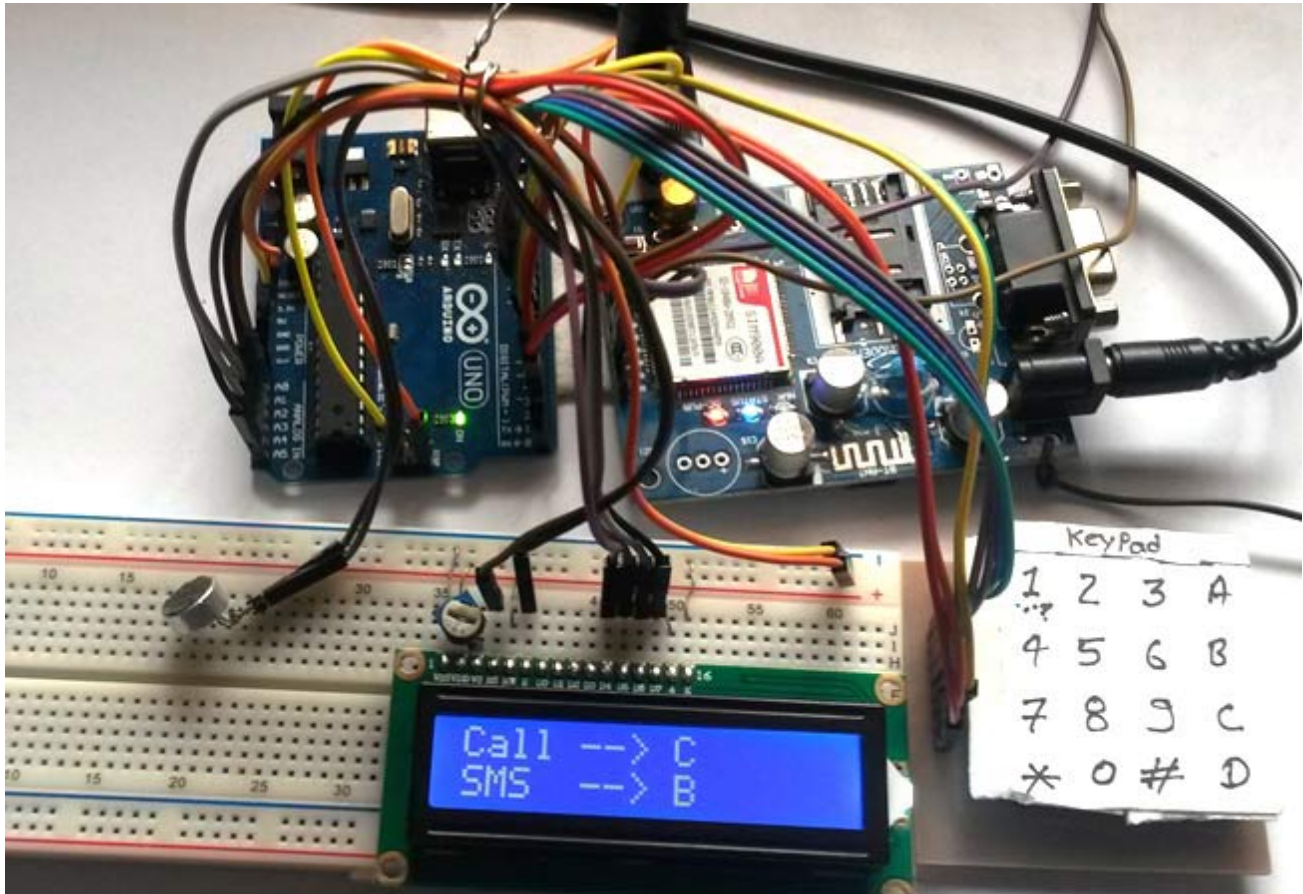
Call and Message using Arduino and GSM Module (/microcontroller-projects/call-and-message-using-arduino-sim900-gsm)

By Saddam (/users/saddam)   7 Comments (/microcontroller-projects/call-and-message-using-arduino-sim900-gsm/#comments)

Make/answer call and Read/send SMS using Arduino

Sometimes people find it difficult to use the **GSM Module** for its basic functions like calling, texting etc., specifically with the Microcontrollers. So here we are going to build a **Simple Mobile Phone using Arduino**, in which GSM Module is used to **Make the Call, answer the Call, send SMS, and read SMS,** and also this Arduino phone has Mic and Speaker to **talk over this Phone.** This project will also serve as a proper **interfacing of GSM Module with Arduino**, with all the Code needed to operate any Phone's basic functions.

Arduino Code

Arduino GSM Module

Components Required:

- Arduino Uno
- GSM Module SIM900
- 16x2 LCD
- 4x4 Keypad
- Breadboard or PCB
- Connecting jumper wire
- Power supply
- Speaker
- MIC
- SIM Card

Working Explanation:

In this **Arduino Mobile Phone Project**, we have used **Arduino Uno** to control whole system's features and interfacing all the components in this system. A **4x4 Alphanumeric Keypad** is used for taking all kind of inputs like: Enter mobile number, type messages, make a call, receive a call, send SMS, read SMS etc. **GSM Module** is used to communicate with the network for calling and messaging purpose. We have also interfaced a **MIC and a Speaker for Voice Call** and Ring sound and a **16x2 LCD** is used for showing messages, instructions and alerts.

Alphanumeric is a method to enter numbers and alphabets both by using same keypad. In this method, we have interfaced 4x4 keypad with Arduino (http://circuitdigest.com/microcontroller-projects/keypad-interfacing-with-arduino-uno) and written Code for accepting alphabets too, check the Code in **Code section** below.
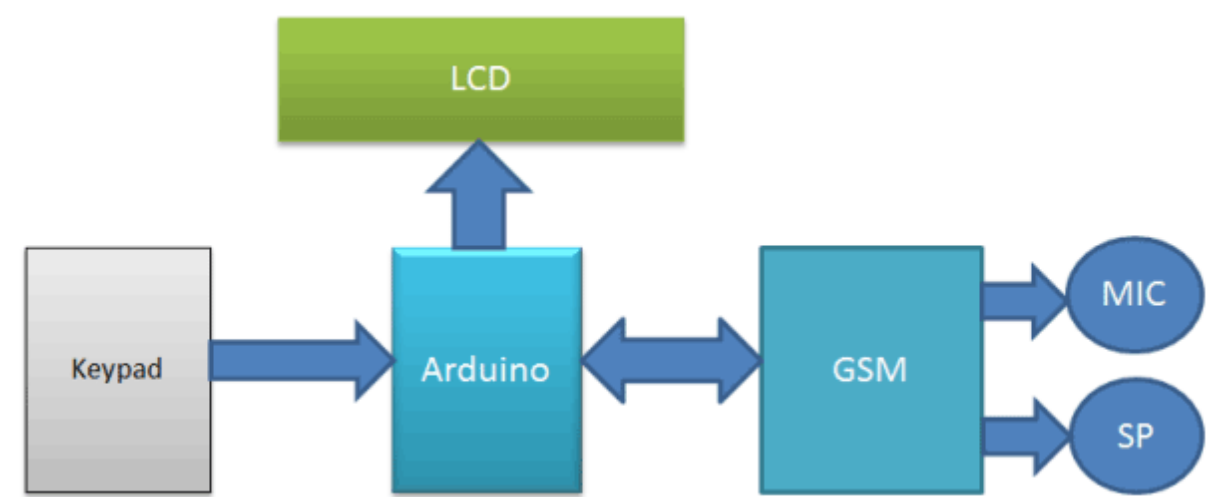

Working of this project is easy. All the features will be performed by Using Alphanumeric Keypad. Check the **Full code and a Demo Video** below to properly understand the process. Here we are going to explain all the four features of the projects below.

Explaining Four Features of Arduino Mobile Phone:

1. Make a Call:

To make a call by using our Arduino based Phone, we have to press 'C' and then need to enter the Mobile Number on which we want to make a call. Number will be entered by using alphanumeric keypad. After entering the number we again need to press 'C'. Now Arduino will process for connecting the call to the entered number by using AT command:
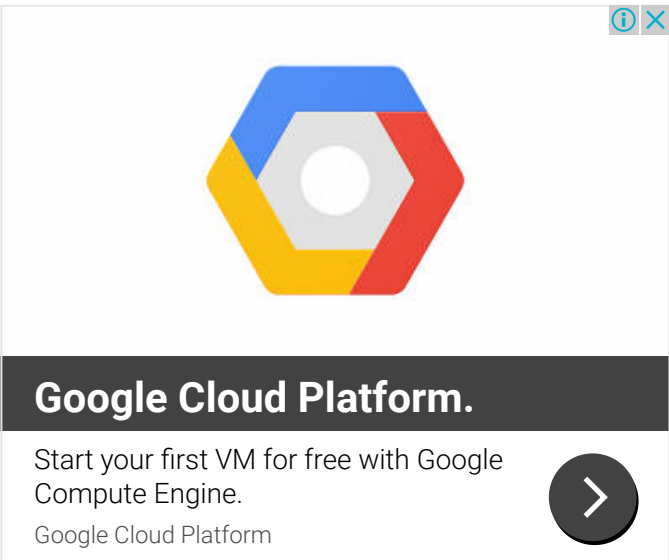
```
ATDxxxxxxxxxx; <Enter>      where xxxxxxxxx is entered Mobile Number.
```

2. Receive a Call:

Receiving a call is very easy. When someone is calling to your system SIM number, which is there in GSM Module, then your system will show 'Incoming...' message over the LCD with incoming number of caller. Now we just need to Press 'A' to attend this call. When we press 'A', Arduino will send given command to GSM Module:

```
ATA <enter>
```

3. Send SMS:

When we want to send a SMS using our Arduino based Mobile Phone, then we need to Press 'B'. Now System will ask for Recipient Number, means 'to whom' we want to send SMS. After entering the number we need to press 'D' and now LCD asks for message. Now we need to type the message, like we enter in normal mobile, by using keypad and then after entering the message we need to press 'D' to send SMS. To Send SMS Arduino sends given command:

```
AT+CMGF=1 <enter>
AT+CMGS="xxxxxxxxxx" <enter>      where: xxxxxxxxxx is entered mobile number
```

And send 26 to GSM to send SMS.

4. Receive and Read SMS:

This feature is also simple. In this, GSM will receive SMS and stores it in SIM card. And Arduino continuously monitors the received SMS indication over UART. We just need to Press 'D', to read the SMS, when we see the New Message symbol (like a envelope: See the **Video** at the end) on the LCD. Below is the **SMS Received indication displayed on the Serial port** is:

```
+CMTI: "SM" <SMS stored location>
+CMTI: "SM",6  Where 6 is message location where it stored in SIM card.
```

When Arduino gets this 'SMS received' indication then it extracts SMS storing location and sends **command to GSM to read the received SMS**. And show a 'New Message Symbol' over the LCD.

```
AT+CMGR=<SMS stored location><enter>
AT+CMGR=6
```

Now GSM sends stored message to Arduino and then Arduino extract main SMS and display it over the LCD and then after reading this SMS Arduino Clear the 'New SMS symbol' from the LCD.
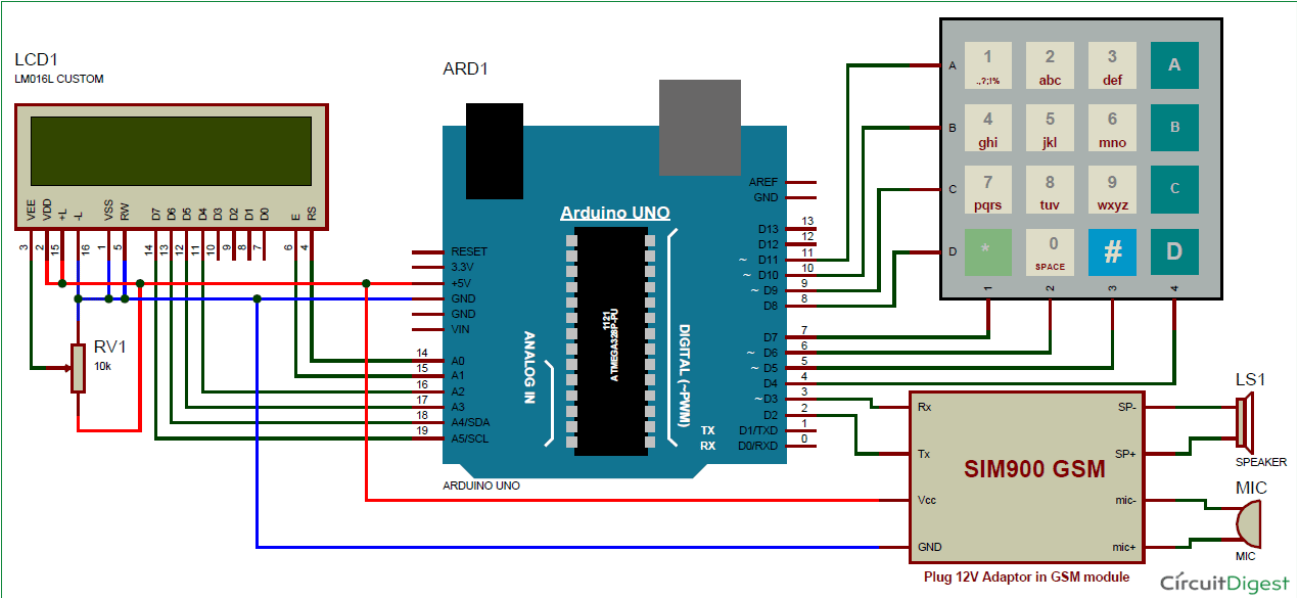
Note: There is no coding for MIC and Speaker.

Check the **Full code and a Demo Video** below to properly understand the process.

Circuit Diagram and Explanation:

[(/fullimage?i=circuitdiagram_mic/call-and-sms-using-arduino-and-gsm-circuit-diagram.png)](/fullimage?i=circuitdiagram_mic/call-and-sms-using-arduino-and-gsm-circuit-diagram.png)

Circuit Diagram of this for interfacing GSM SIM900 and Arduino is given above. 16x2 LCD pins RS, EN, D4, D5, D6 and D7 are connected with pin number 14, 15, 16, 17, 18 and 19 of Arduino respectively. GSM Module's Rx and Tx pins are directly connected with Arduino's pin D3 and D2 respectively (Ground of Arduino and GSM must be connected with each other). 4x4 keypad Row pins R1, R2, R3, R4 are directly linked to pin number 11,10, 9, 8 of Arduino and Colum pins of keypad C1, C2, C3 are linked with pin number 7, 6, 5, 4 of Arduino. MIC is directly connected at mic+ and mic- of GSM Module and Speaker is directly connected at SP+ and SP- pins for GSM Module.

Programming Explanation:

Programming part of this project is little complex for beginners. In this code we have used keypad library *#include <Keypad.h>* for interfacing simple keypad for entering numbers. And for entering alphabets with the same keypad, we have created function *void alfakey()*. Means we have made every key multi functioning and we can enter any character or integer by using only 10 keys.

Like if we press key 2 (abc2), it will show 'a' and if we presses it again then it will replace 'a' to 'b' and if again we press three times then it will show 'c' at same place in LCD. If we wait for some time after pressing key, cursor will automatic move to next position in LCD. Now we can enter next char or number. The same procedure is applied for other keys.

```
#include <Keypad.h>
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char hexaKeys[ROWS][COLS] =
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {11, 10, 9, 8}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {7, 6, 5, 4}; //connect to the column pinouts of the keypad

//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
```

```
void alfakey()
{
 int x=0,y=0;
 int num=0;
  while(1)
  {
    lcd.cursor();
    char key=customKeypad.getKey();
    if(key)
    {
        if(key=='1')
        {
          num=0;
          lcd.setCursor(x,y);
          .... .....
          ........ ....
```

Apart from operating keypad, we have created many other functions like *void call()* for calling feature of Phone, *void sms()* for messaging feature, void *lcd_status()* for display LCD status *void gsm_init()* for initializing the GSM Module etc. Check below all other function related to **make & receive Call and send & read SMS using GSM Module and Arduino**. All the functions are self-explanatory and understandable.

Arduino Uno Shield

What Is an Arduino Shield

Code:

```
#include <SoftwareSerial.h>
SoftwareSerial Serial1(2, 3); // RX, TX

#include<LiquidCrystal.h>
LiquidCrystal lcd(14,15,16,17,18,19);

byte back[8] =
{
  0b00000,
  0b00000,
  0b11111,
  0b10101,
  0b11011,
  0b11111,
  0b00000,
  0b00000
};

String number="";
String msg="";
String instr="";
String str_sms="";
String str1="";
int ring=0;
int i=0,temp=0;
int sms_flag=0;
char sms_num[3];
int rec_read=0;
int temp1=0;

#include <Keypad.h>
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char hexaKeys[ROWS][COLS] =
{
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {11, 10, 9, 8}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {7, 6, 5, 4}; //connect to the column pinouts of the keypad

//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

String ch="1,.?!@abc2def3ghi4jkl5mno6pqrs7tuv8wxyz90 ";

void setup()
{
  Serial1.begin(9600);
  lcd.begin(16,2);
  lcd.createChar(1, back);
  lcd.print("Simple Mobile ");
  lcd.setCursor(0,1);
  lcd.print("System Ready..");
  delay(1000);
  gsm_init();
  lcd.clear();
  lcd.print("System Ready");
  delay(2000);
}
void loop()
{
  serialEvent();
```

```
if(sms_flag==1)
{
  lcd.clear();
  lcd.print("New Message");
  int ind=instr.indexOf("+CMTI: \"SM\",");
  ind+=12;
  int k=0;
  lcd.setCursor(0,1);
  lcd.print(ind);
  while(1)
  {
    while(instr[ind]!= 0x0D)
    {
      sms_num[k++]=instr[ind++];
    }
    break;
  }
  ind=0;
  sms_flag=0;
  lcd.setCursor(0,1);
  lcd.print("Read SMS --> D");
  delay(4000);
  instr="";
  rec_read=1;
  temp1=1;
  i=0;
}

if(ring == 1)
{
  number="";
  int loc=instr.indexOf("+CLIP: \"");
  if(loc > 0)
  {
    number+=instr.substring(loc+8,loc+13+8);
  }
  lcd.setCursor(0,0);
  lcd.print("Incomming...    ");
  lcd.setCursor(0,1);
  lcd.print(number);
  instr="";
  i=0;
}
else
{
serialEvent();
lcd.setCursor(0,0);
lcd.print("Call --> C      ");
lcd.setCursor(0,1);
lcd.print("SMS  --> B    ");
if(rec_read==1)
{
  lcd.write(1);
  lcd.print("   ");
}
else
lcd.print("     ");
}

 char key=customKeypad.getKey();
if(key)
{
  if(key== 'A')
  {
    if(ring==1)
    {
    Serial1.println("ATA");
    delay(5000);
    }
  }
  else if(key=='C')
  {
    call();
  }
```

```
      else if(key=='B')
     {
       sms();
     }
      else if(key == 'D'  && temp1==1)
     {
       rec_read=0;
       lcd.clear();
       lcd.print("Please wait...");
       Serial1.print("AT+CMGR=");
       Serial1.println(sms_num);
       int sms_read_flag=1;
       str_sms="";
       while(sms_read_flag)
       {
         while(Serial1.available()>0)
         {
           char ch=Serial1.read();
           str_sms+=ch;
           if(str_sms.indexOf("OK")>0)
           {
             sms_read_flag=0;
             //break;
           }
         }
       }
       int l1=str_sms.indexOf("\"\r\n");
       int l2=str_sms.indexOf("OK");
       String sms=str_sms.substring(l1+3,l2-4);
       lcd.clear();
       lcd.print(sms);
       delay(5000);
       }
       delay(1000);
     }
}


void call()
{
  number="";
  lcd.clear();
  lcd.print("After Enter No.");
  lcd.setCursor(0,1);
  lcd.print("Press C to Call");
  delay(2000);
  lcd.clear();
  lcd.print("Enter Number:");
  lcd.setCursor(0,1);
  while(1)
  {
    serialEvent();
    char key=customKeypad.getKey();
    if(key)
    {
      if(key=='C')
      {
        lcd.clear();
        lcd.print("Calling........");
        lcd.setCursor(0,1);
        lcd.print(number);
        Serial1.print("ATD");
        Serial1.print(number);
        Serial1.println(";");
        long stime=millis()+5000;
        int ans=1;
        while(ans==1)
        {
          while(Serial1.available()>0)
          {
            if(Serial1.find("OK"))
            {
              lcd.clear();
              lcd.print("Ringing....");
              int l=0;
```

```
        str1="";
        while(ans==1)
        {
          while(Serial1.available()>0)
          {
            char ch=Serial1.read();
            str1+=ch;
            if(str1.indexOf("NO CARRIER")>0)
            {
              lcd.clear();
              lcd.print("Call End");
              delay(2000);
              ans=0;
              return;
            }
          }
          char key=customKeypad.getKey();
          if(key == 'D')
          {
            lcd.clear();
            lcd.print("Call End");
            delay(2000);
            ans=0;
            return;
          }
          if(ans==0)
          break;
        }
      }
    }
  }
  else
  {
    number+=key;
    lcd.print(key);
  }
}
}
}
void sms()
{
 lcd.clear();
 lcd.print("Initilising SMS");
 Serial1.println("AT+CMGF=1");
 delay(400);
 lcd.clear();
 lcd.print("After Enter No.");
 lcd.setCursor(0,1);
 lcd.print("Press D        ");
 delay(2000);
 lcd.clear();
 lcd.print("Enter Rcpt No.:");
 lcd.setCursor(0,1);
 Serial1.print("AT+CMGS=\"");
 while(1)
 {
  serialEvent();
  char key=customKeypad.getKey();
  if(key)
  {
   if(key=='D')
   {
    //number+="";
    Serial1.println("\"");
    break;
   }
   else
   {
    //number+=key;
    Serial1.print(key);
    lcd.print(key);
   }
  }
 }
}
```

```
  lcd.clear();
  lcd.print("After Enter MSG ");
  lcd.setCursor(0,1);
  lcd.print("Press D to Send ");
  delay(2000);
  lcd.clear();
  lcd.print("Enter Your Msg");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0,0);
  alfakey();
}
void alfakey()
{
 int x=0,y=0;
 int num=0;
  while(1)
  {
   lcd.cursor();
   char key=customKeypad.getKey();
   if(key)
   {
     if(key=='1')
     {
      num=0;
      lcd.setCursor(x,y);
      lcd.print(ch[num]);
      for(int i=0;i<3000;i++)
      {
       lcd.noCursor();
       char key=customKeypad.getKey();
       if(key=='1')
       {
        num++;
        if(num>5)
        num=0;
        lcd.setCursor(x,y);
        lcd.print(ch[num]);
        i=0;
        delay(200);
       }
      }
      x++;
      if(x>15)
      {
       x=0;
       y++;
       y%=2;
      }
      msg+=ch[num];
     }
     else if(key=='2')
     {
      num=6;
      lcd.setCursor(x,y);
      lcd.print(ch[num]);
      for(int i=0;i<3000;i++)
      {
       lcd.noCursor();
       char key=customKeypad.getKey();
       if(key=='2')
       {
        num++;
        if(num>9)
        num=6;
        lcd.setCursor(x,y);
        lcd.print(ch[num]);
        i=0;
        delay(200);
       }
      }
      x++;
      if(x>15)
      {
       x=0;
```

```
     y++;
     y%=2;
    }
    msg+=ch[num];
   }
  else if(key=='3')
  {
    num=10;
    lcd.setCursor(x,y);
    lcd.print(ch[num]);
    for(int i=0;i<3000;i++)
    {
    lcd.noCursor();
    char key=customKeypad.getKey();
    if(key=='3')
    {
    num++;
    if(num>13)
    num=10;
    lcd.setCursor(x,y);
    lcd.print(ch[num]);
    i=0;
    delay(200);
    }
    }
    x++;
    if(x>15)
    {
    x=0;
    y++;
    y%=2;
    }
    msg+=ch[num];
   }
  else if(key=='4')
  {
    num=14;
    lcd.setCursor(x,y);
    lcd.print(ch[num]);
    for(int i=0;i<3000;i++)
    {
    lcd.noCursor();
    char key=customKeypad.getKey();
    if(key=='4')
    {
    num++;
    if(num>17)
    num=14;
    lcd.setCursor(x,y);
    lcd.print(ch[num]);
    i=0;
    delay(200);
    }
    }
    x++;
    if(x>15)
    {
    x=0;
    y++;
    y%=2;
    }
    msg+=ch[num];
   }
       else if(key=='5')
  {
    num=18;
    lcd.setCursor(x,y);
    lcd.print(ch[num]);
    for(int i=0;i<3000;i++)
    {
    lcd.noCursor();
    char key=customKeypad.getKey();
    if(key=='5')
    {
```

```
num++;
if(num>21)
num=18;
lcd.setCursor(x,y);
lcd.print(ch[num]);
i=0;
delay(200);
}
}
x++;
if(x>15)
{
x=0;
y++;
y%=2;
}
msg+=ch[num];
}
else if(key=='6')
{
num=22;
lcd.setCursor(x,y);
lcd.print(ch[num]);
for(int i=0;i<3000;i++)
{
lcd.noCursor();
char key=customKeypad.getKey();
if(key=='6')
{
num++;
if(num>25)
num=22;
lcd.setCursor(x,y);
lcd.print(ch[num]);
i=0;
delay(200);
}
}
x++;
if(x>15)
{
x=0;
y++;
y%=2;
}
msg+=ch[num];
}
else if(key=='7')
{
num=26;
lcd.setCursor(x,y);
lcd.print(ch[num]);
for(int i=0;i<3000;i++)
{
lcd.noCursor();
char key=customKeypad.getKey();
if(key=='7')
{
num++;
if(num>30)
num=26;
lcd.setCursor(x,y);
lcd.print(ch[num]);
i=0;
delay(200);
}
}
x++;
if(x>15)
{
x=0;
y++;
y%=2;
```

```
 }
 msg+=ch[num];
 }
else if(key=='8')
{
 num=31;
 lcd.setCursor(x,y);
 lcd.print(ch[num]);
 for(int i=0;i<3000;i++)
 {
 lcd.noCursor();
 char key=customKeypad.getKey();
 if(key=='8')
 {
 num++;
 if(num>34)
 num=31;
 lcd.setCursor(x,y);
 lcd.print(ch[num]);
 i=0;
 delay(200);
 }
 }
 x++;
 if(x>15)
 {
 x=0;
 y++;
 y%=2;
 }
 msg+=ch[num];
 }
else if(key=='9')
{
 num=35;
 lcd.setCursor(x,y);
 lcd.print(ch[num]);
 for(int i=0;i<3000;i++)
 {
 lcd.noCursor();
 char key=customKeypad.getKey();
 if(key=='9')
 {
 num++;
 if(num>39)
 num=35;
 lcd.setCursor(x,y);
 lcd.print(ch[num]);
 i=0;
 delay(200);
 }
 }
 x++;
 if(x>15)
 {
 x=0;
 y++;
 y%=2;
 }
 msg+=ch[num];
 }
 else if(key=='0')
{
 num=40;
 lcd.setCursor(x,y);
 lcd.print(ch[num]);
 for(int i=0;i<3000;i++)
 {
 lcd.noCursor();
 char key=customKeypad.getKey();
 if(key=='0')
 {
 num++;
 if(num>41)
```

```
          num=40;
          lcd.setCursor(x,y);
          lcd.print(ch[num]);
          i=0;
          delay(200);
         }
        }
       x++;
       if(x>15)
       {
       x=0;
       y++;
       y%=2;
       }
       msg+=ch[num];
      }
      else if(key=='D')
      {
       lcd.clear();
       lcd.print("Sending SMS....");
      // Serial1.print("AT+CMGS=");
      // Serial1.print(number);
      // delay(2000);
       Serial1.print(msg);
       Serial1.write(26);
       delay(5000);
       lcd.clear();
       lcd.print("SMS Sent to");
       lcd.setCursor(0,1);
       lcd.print(number);
       delay(2000);
       number="";
       break;
      }
     }
    }
   }
}
void send_data(String message)
{
  Serial1.println(message);
  delay(200);
}
void send_sms()
{
  Serial1.write(26);
}
void lcd_status()
{
  lcd.setCursor(2,1);
  lcd.print("Message Sent");
  delay(2000);
 //lcd.setCursor()
 //lcd.print("")
 //return;
}
void back_button()
{
 //lcd.setCursor(0,15);
}
void ok_button()
{
  lcd.setCursor(0,4);
  lcd.print("OK");
}
void call_button()
{
  lcd.setCursor(0,4);
  lcd.print("CALL");
}
void sms_button()
{
  lcd.setCursor(0,13);
```

```
   lcd.print("SMS");
}
void gsm_init()
{
  lcd.clear();
  lcd.print("Finding Module..");
  boolean at_flag=1;
  while(at_flag)
  {
   Serial1.println("AT");
   while(Serial1.available()>0)
   {
    if(Serial1.find("OK"))
    at_flag=0;
   }

   delay(1000);
  }
  lcd.clear();
  lcd.print("Module Connected..");
  delay(1000);
  lcd.clear();
  lcd.print("Disabling ECHO");
  boolean echo_flag=1;
  while(echo_flag)
  {
   Serial1.println("ATE1");
   while(Serial1.available()>0)
   {
    if(Serial1.find("OK"))
    echo_flag=0;
   }
   delay(1000);
  }
  lcd.clear();
  lcd.print("Echo OFF");
  delay(1000);
  lcd.clear();
  lcd.print("Finding Network..");
  boolean net_flag=1;
  while(net_flag)
  {
   Serial1.println("AT+CPIN?");
   while(Serial1.available()>0)
   {
    if(Serial1.find("+CPIN: READY"))
    net_flag=0;
   }
   delay(1000);
  }
  lcd.clear();
  lcd.print("Network Found..");
  delay(1000);
  lcd.clear();
}
void serialEvent()
{
  while(Serial1.available())
  {
   char ch=Serial1.read();
   instr+=ch;
   i++;
   if(instr[i-4] == 'R' && instr[i-3] == 'I' && instr[i-2] == 'N' && instr[i-1] == 'G' )
   {
     ring=1;
   }
   if(instr.indexOf("NO CARRIER")>=0)
   {
     ring=0;
     i=0;
   }
   if(instr.indexOf("+CMTI: \"SM\"")>=0)
   {
```

```
        sms_flag=1;
      }
    }
}
```

Video:
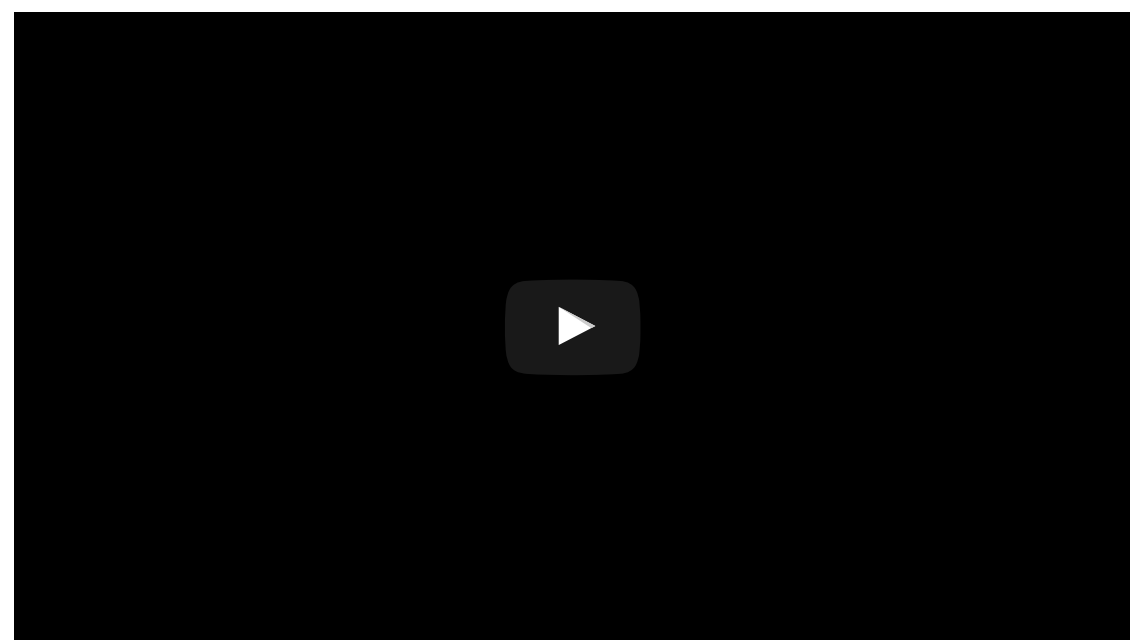
CircuitDigest highly recommends **EasyEDA** for Circuit Design and PCB Prototype (https://easyeda.com/order)
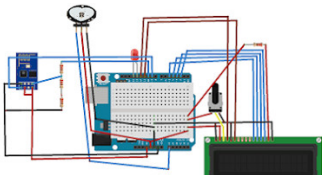
From Schematic Drawing to PCB Production, Just Need One Tool

(https://easyeda.com/)

Low Cost Prototype PCB: 10 pcs 2 layers only $10,quick delivery,100% E-test

(https://easyeda.com/order)
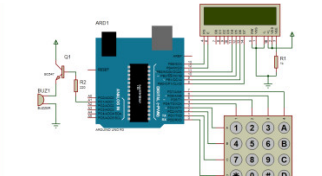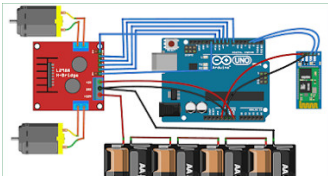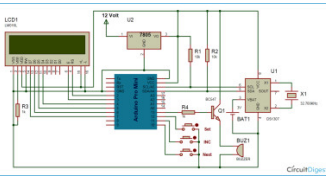
Add new comment (/microcontroller-projects/call-and-message-using-arduino-sim900-gsm#comment-form)

Comments (7)

teja

reply (/comment/reply/517/5507)

Arduino: 1.6.4 (Windows 8.1), Board: "Arduino Duemilanove or Diecimila, ATmega328"

mobilefull2.ino: In function 'void setup()':

mobilefull2:51: error: 'gsm_init' was not declared in this scope

mobilefull2.ino: In function 'void loop()':

mobilefull2:58: error: 'serialEvent' was not declared in this scope

mobilefull2:135: error: 'sms' was not declared in this scope

mobilefull2.ino: In function 'void call()':

mobilefull2:183: error: 'serialEvent' was not declared in this scope

'gsm_init' was not declared in this scope

mobilefull2.ino: In function 'void setup()':

mobilefull2:51: error: 'gsm_init' was not declared in this scope

mobilefull2.ino: In function 'void loop()':

mobilefull2:58: error: 'serialEvent' was not declared in this scope

mobilefull2:135: error: 'sms' was not declared in this scope

mobilefull2.ino: In function 'void call()':

mobilefull2:183: error: 'serialEvent' was not declared in this scope

'gsm_init' was not declared in this scope

This report would have more information with

"Show verbose output during compilation"

enabled in File > Preferences.

Sep 05, 2016

gunjan vyas   reply (/comment/reply/517/6645)

after initializing the display shows only one message "finding module" and there is no further process being done by the circuit. please give proper solution.

Oct 06, 2016

harish kumar   reply (/comment/reply/517/8109)

when i compiling the code it show,

error compiling for arduino uno board

what i do sir

Nov 10, 2016

harish kumar   reply (/comment/reply/517/8208)

what type of speaker u used sir ?

Nov 12, 2016

Mark   reply (/comment/reply/517/10715)

Looks good.

Feb 07, 2017

Karlos   reply (/comment/reply/517/11063)

after initializing the display shows only one message "finding module.." and there is no further process being done by the circuit. please give proper solution.

what's next? please

Feb 18, 2017

sachit kumaar   reply (/comment/reply/517/11375)

Tell me the command code for displaying messages through LCD via gsm900A...pls rply sir as soon as possible
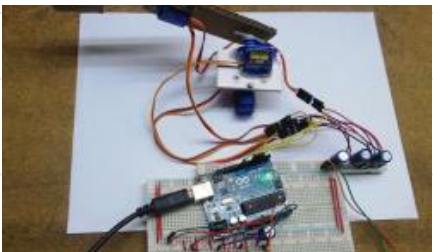
Feb 27, 2017

Leave a comment

Your name *

E-mail *

FEATURED PROJECTS AND CIRCUITS

The content of this field is kept private and will not be shown publicly.

Subj

Com

(/microcontroller-projects/arduino-smoke-detector-on-pcb-using-mq2-gas-sensor)

(/microcontroller-projects/diy-arduino-robotic-arm-tutorial)

(/microcontroller-projects/raspberry-pi-iot-intru alert-system)

No HTML tags allowed.

Web page addresses and e-mail addresses turn into links automatically.

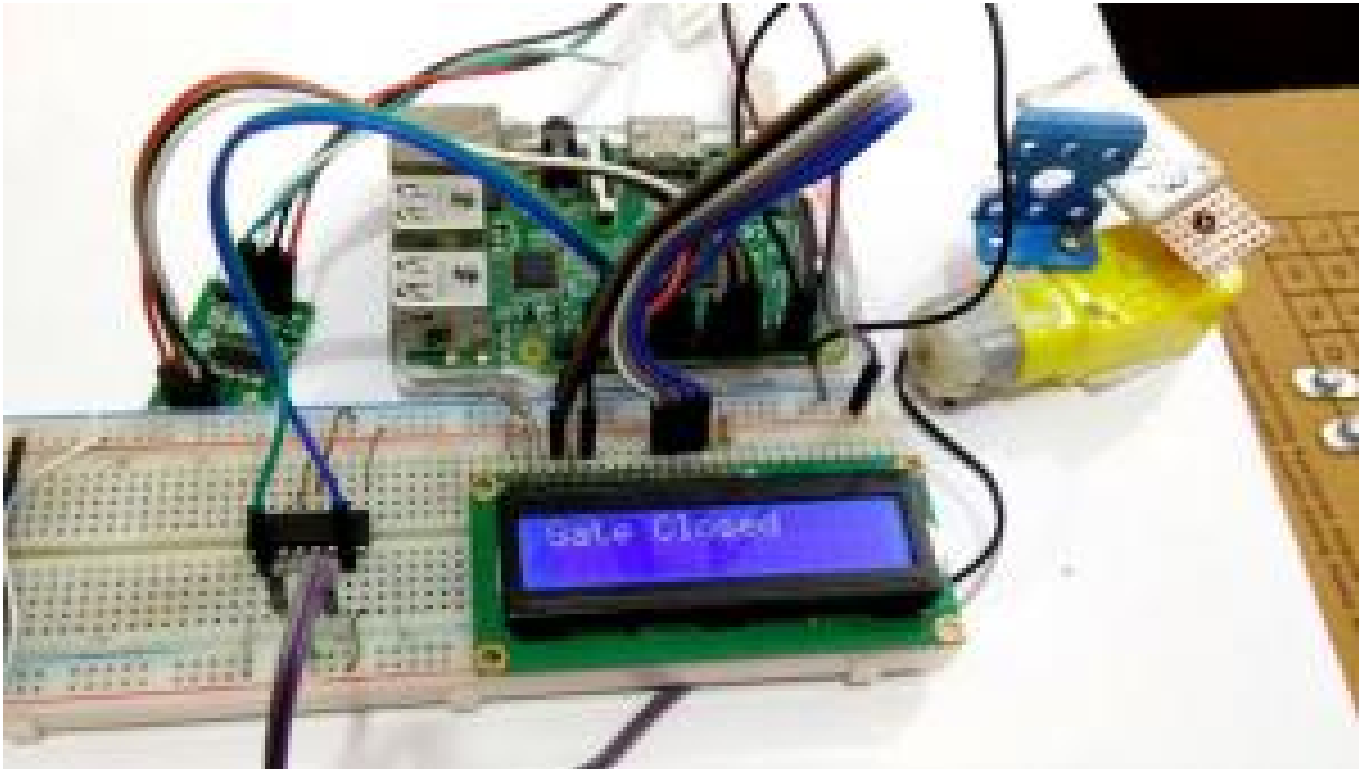Lines and paragraphs break automatically.
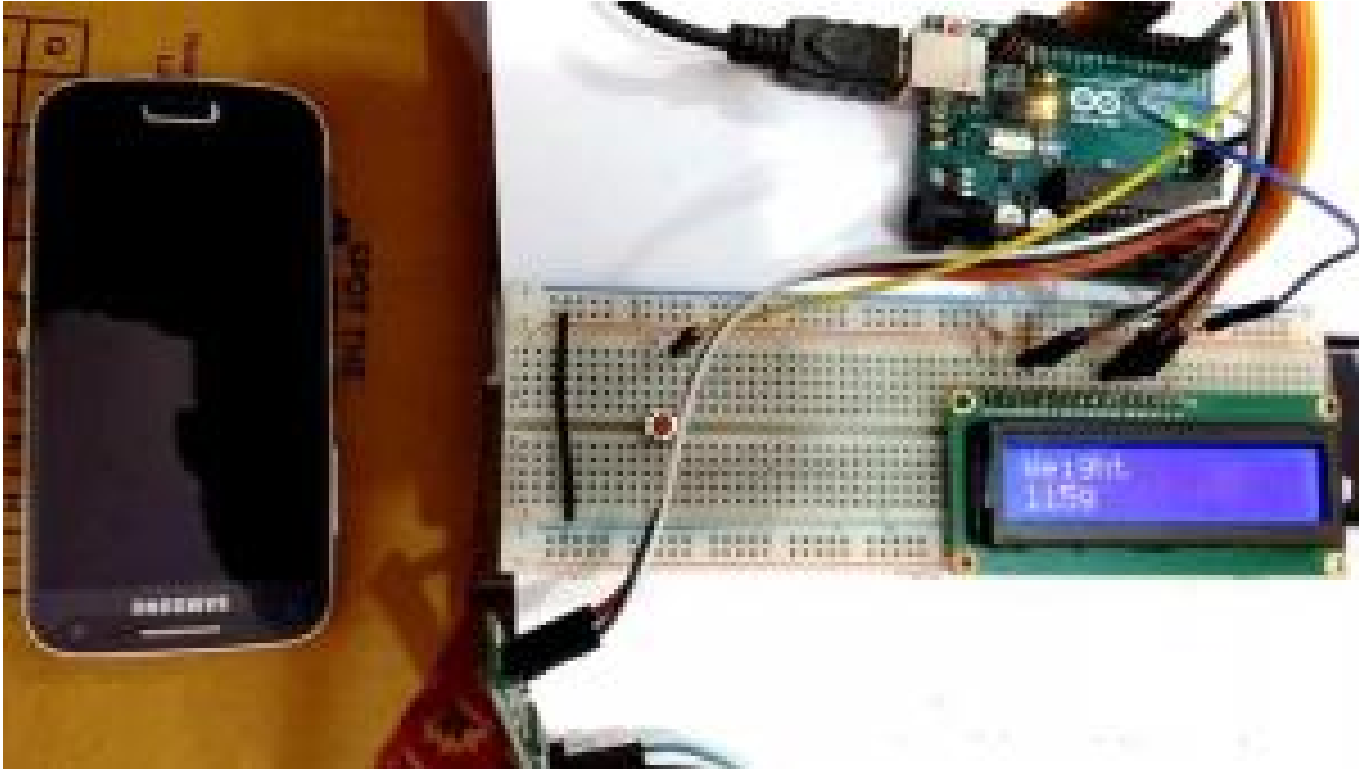
1
Shares

| Save | Preview |

RELATED CONTENT

Arduino Weight Measurement using Load Cell and HX711 Module (/microcontroller-projects/arduino-weight-measurement-using-load-cell)

Arduino Solar Tracker using LDR and Servo Motor (/microcontroller-projects/arduino-solar-panel-tracker)

Smoke Detector using MQ2 Gas Sensor and Arduino (/microcontroller-projects/arduino-smoke-detector-on-pcb-using-mq2-gas-sensor)

Arduino Robotic Arm (/microcontroller-projects/diy-arduino-robotic-arm-tutorial)

Mobile Phone Controlled Robot Car using G-Sensor and Arduino (/microcontroller-projects/g-sensor-controlled-robot-car-arduino)

LATEST POSTS

(/microcontroller-projects/raspberry-pi-weight-sensing-automatic-gate)

Raspberry Pi Based Weight Sensing Automatic Gate (/microcontroller-projects/raspberry-pi-weight-sensing-automatic-gate)

(/microcontroller-projects/arduino-weight-measurement-using-load-cell)

Arduino Weight Measurement using Load Cell and HX711 Module (/microcontroller-projects/arduino-weight-measurement-using-load-cell)