LAMPIRAN

CODING YANG DI GUNAKAN

```
#include <SoftwareSerial.h>
#include <GSM_Shield.h>
#include <SPI.h>       //SPI.h must be included as DMD is written by SPI (the IDE complains otherwise)
#include <DMDMega.h>       //
#include <TimerOne.h>   //
#include "SystemFont5x7.h"
#include "Arial_black_16.h"
#include <LiquidCrystal.h>
/*
 The circuit:
 * LCD RS pin to digital pin 12 => 22
 * LCD Enable pin to digital pin 11 =23
 * LCD D4 pin to digital pin 5 => 24
 * LCD D5 pin to digital pin 4 => 25
 * LCD D6 pin to digital pin 3  => 26
 * LCD D7 pin to digital pin 2 => 27
 * LCD R/W pin to ground
 * LCD VSS pin to ground
 * LCD VCC pin to 5V
 * 10K resistor:
 * ends to +5V and ground
 * wiper to LCD VO pin (pin 3)
*/

//Fire up the DMD library as dmd
#define DISPLAYS_ACROSS 1
#define DISPLAYS_DOWN 1
DMD dmd(DISPLAYS_ACROSS, DISPLAYS_DOWN);

boolean stringComplete = false;  // whether the string is complete
boolean sms_valid = false;
boolean man_panel = false;  // whether the string is complete
char newstr[256];
char newstr_lcd[256];

int  sLength;

//**************************************************************************
char number_incoming[20];
char number[]="+6285720061205";  //+6285720061205 Destination number
char text[]="[Auto Reply] SMS Diterima ";  //SMS to send
byte type_sms=SMS_ALL;      //Type of SMS
byte del_sms=1;             //0: No deleting sms - 1: Deleting SMS
//**************************************************************************

GSM gsm;
char sms_rx[122]; //Received text SMS
String teks_panel="";
```

```
String lcdpesan="";
//int inByte=0;    //Number of byte received on serial port

//LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
LiquidCrystal lcd(22, 23, 24, 25, 26, 27);

int call;
int error;

void ScanDMD()
{
 dmd.scanDisplayBySPI();
}


void setup()
{
 Serial.begin(9600);
 Serial.println("system startup");

 lcd.begin(16, 2);
 // Print a message to the LCD.
 lcd.setCursor(0,0);
 lcd.print(" System Startup");

 Timer1.initialize( 5000 );          //period in microseconds to call ScanDMD. Anything longer
than 5000 (5ms) and you can see flicker.
 Timer1.attachInterrupt( ScanDMD );

 lcd.setCursor(0,1);
 lcd.print("               ");
 lcd.print(" GSM Turn On.... ");

 gsm.TurnOn(9600);          //module power on
 gsm.InitParam(PARAM_SET_1);//configure the module
 gsm.Echo(0);               //enable AT echo

 dmd.clearScreen( true );

 lcd.clear();
}


void loop()
{
 char inSerial[5];
 int i=0;
 lcd.setCursor(0,0);
 lcd.print("==System Ready==n ");
 //
```

```
Check_Call(); //Check if there is an incoming call
Check_SMS();  //Check if there is SMS

//Check data serial com
if (Serial.available() > 0)
{
  while (Serial.available() > 0) {
    inSerial[i]=(Serial.read()); //read data
    i++;
  }
  inSerial[i]='\0';
 Check_Protocol(inSerial);
}
if (!man_panel) {
teks_panel=sms_rx;
}
if (sms_valid)
{
if (teks_panel=="")
lcdpesan="Pesan : -";
else
lcdpesan="Pesan : "+teks_panel;

LCD_status(lcdpesan,true);
}

if (stringComplete)
{ if (!man_panel)
  {
    teks_panel=sms_rx;
    Check_Protocol("s"); //auto reply
  }
 sLength=teks_panel.length();
 teks_panel.toCharArray(newstr,sLength+1);
 Serial.print("Send To Display Panel \"");
 Serial.print(teks_panel);
 Serial.println("\"");
 LCD_status("Send To Display",false);
 stringComplete = false;
 man_panel = false;
}

//dmd.selectFont(SystemFont5x7);
dmd.selectFont(Arial_Black_16);
dmd.clearScreen( true );
dmd.drawMarquee(newstr,sLength,(32*DISPLAYS_ACROSS)-1,0);
long start=millis();
long timer=start;
```

```
    boolean ret=false;
    while(!ret){
     if ((timer+30) < millis()) {
       ret=dmd.stepMarquee(-1,0);
       timer=millis();
     }
    }
    delay(1000);
}

void LCD_status(String inStr,boolean teksjalan)

{ if (teksjalan) {
  inStr=inStr+" ";
  sLength=inStr.length();
  inStr.toCharArray(newstr_lcd,sLength+1);
  runningText(16,newstr_lcd,1,200);
}
else
{
  lcd.setCursor(0,1);
  lcd.print(inStr);
}
return;

}

void runningText(int sizeColLcd, char message[], int row, int timerefresh){
  int textSize=strlen(message);
  char tempLcd[sizeColLcd];
  int tempPos =0;
  int iter;

  iter=textSize+sizeColLcd;

  for(int i=0; i<=iter; i++){
    if(i<sizeColLcd){
      memcpy(tempLcd, &message[0], i+1);
      lcd.setCursor(sizeColLcd-(i+1),row);
      lcd.print(tempLcd);
    }
    else {
      memcpy(tempLcd, &message[tempPos], sizeColLcd);
      lcd.setCursor(0,row);
      lcd.print(tempLcd);
      tempPos +=1;
    }

    delay(timerefresh);
```

```
  }
}

void Check_Protocol(String inStr)
{
     Serial.print("Command: ");
     Serial.println(inStr);

 Serial.println("Check_Protocol");

   switch (inStr[0])
    {
     case 't':
           stringComplete = true;
           Serial.println("Tes Manual Panel");
           teks_panel="Test Display OK";
           man_panel = true;
         break;

    case 'a' :  //Answer
       if (gsm.CallStatus()==CALL_INCOM_VOICE){
        gsm.PickUp();
        Serial.println("Answer");
       }
       else
       {
        Serial.println("No incoming call");
       }
      break;

    case 'c': // C  //Call
      Serial.println(word(inStr.length()));
      if (inStr.length()==2)  //To call variable 'number'   comand   c
      {
       Serial.print("Calling ");
       Serial.println(number);
       gsm.Call(number);
      }
      if (inStr.length()>2)  //To call number in phone book position   comand   cx where x is
the SIM position
      {
        error=gsm.GetPhoneNumber(inStr[1],number);
        if (error!=0)
        {
         Serial.print("Calling ");
         Serial.println(number);
         gsm.Call(number);
        }
        else
        {
```

```
      Serial.print("No number in pos ");
      Serial.println(inStr[1]);
     }
   }
   break;

 case 'h': //H //HangUp if there is an incoming call
   if (gsm.CallStatus()!=CALL_NONE)
   {
    Serial.println("Hang");
    gsm.HangUp();
   }
   else
   {
    Serial.println("No incoming call");
   }
   break;


 case 's': //S //Send SMS
   Serial.print("Send SMS to ");
   Serial.println(number);
   error=gsm.SendSMS(number,number_incoming);
   if (error==0)  //Check status
   {
     Serial.println("SMS ERROR \n");
   }
   else
   {
     Serial.println("SMS OK \n");
   }
   break;

 case 'p':  //Read-Write Phone Book
   if (inStr.length()==3)
   {

     switch (inStr[1])
     {
      case 'd':  //Delete number in specified position  pd2
        error=gsm.DelPhoneNumber(inStr[2]);
        if (error!=0)
        {
         Serial.print("Phone number position ");
         Serial.print(inStr[2]);
         Serial.println(" deleted");
        }
        break;

      case 'g':  //Read from Phone Book position      pg2
```

```
          error=gsm.GetPhoneNumber(inStr[2],number);
          if (error!=0)  //Find number in specified position
          {
           Serial.print("Phone Book position ");
           Serial.print(inStr[2]);
           Serial.print(": ");
           Serial.println(number);
          }
          else  //Not find number in specified position
          {
           Serial.print("No Phone number in position ");
           Serial.println(inStr[2]);
          }
          break;
         case 'w':  //Write from Phone Book Position    pw2
          error=gsm.WritePhoneNumber(inStr[2],number);
          if (error!=0)
          {
           Serial.print("Number ");
           Serial.print(number);
           Serial.print(" writed in Phone Book position ");
           Serial.println(inStr[2]);
          }
          else Serial.println("Writing error");
          break;
       }

      }
      break;

    }

  delay(1500);

  return;
}


void Check_Call()  //Check status call if this is available
{
   call=gsm.CallStatus();
   switch (call)
   {
    case CALL_NONE:
      Serial.println("Standby/no call");
      break;
    case CALL_INCOM_VOICE:
      gsm.CallStatusWithAuth(number_incoming,0,0);
      Serial.print("incoming voice call from ");
      Serial.println(number_incoming);
```

```
      break;
    case CALL_ACTIVE_VOICE:
      Serial.println("active voice call");
      break;
    case CALL_NO_RESPONSE:
      Serial.println("GSM no response");

      break;
   }
  return;
}


void Check_SMS()  //Check if there is an sms 'type_sms'
{
  char pos_sms_rx;  //Received SMS position
  String num_masuk="";
  String num_master="";
  pos_sms_rx=gsm.IsSMSPresent(type_sms);
  if (pos_sms_rx!=0)
  {
   //Read text/number/position of sms
   gsm.GetSMS(pos_sms_rx,number_incoming,sms_rx,120);
   Serial.print("Received SMS from ");
   Serial.print(number_incoming);
   Serial.print("(sim position: ");
   Serial.print(word(pos_sms_rx));
   Serial.println(")");
   Serial.println(sms_rx);
   num_masuk=number_incoming;
   num_master=number;
   LCD_status("SMS Masuk dari "+num_masuk,true);
   if (num_masuk!=num_master)
    { sms_valid=false;
     if (del_sms==1)
     { error=gsm.DeleteSMS(pos_sms_rx);
      if (error==0)
       Serial.println("SMS deleted numb asing");
      else
       Serial.println("SMS not deleted numb asing");
      }
     return;
    }

   sms_valid=true;
   stringComplete = true;

   if (del_sms==1)
   { error=gsm.DeleteSMS(pos_sms_rx);
    if (error==0)
```

```
        Serial.println("SMS deleted");
        else
        Serial.println("SMS not deleted");
      }
    }
   return;
}
```