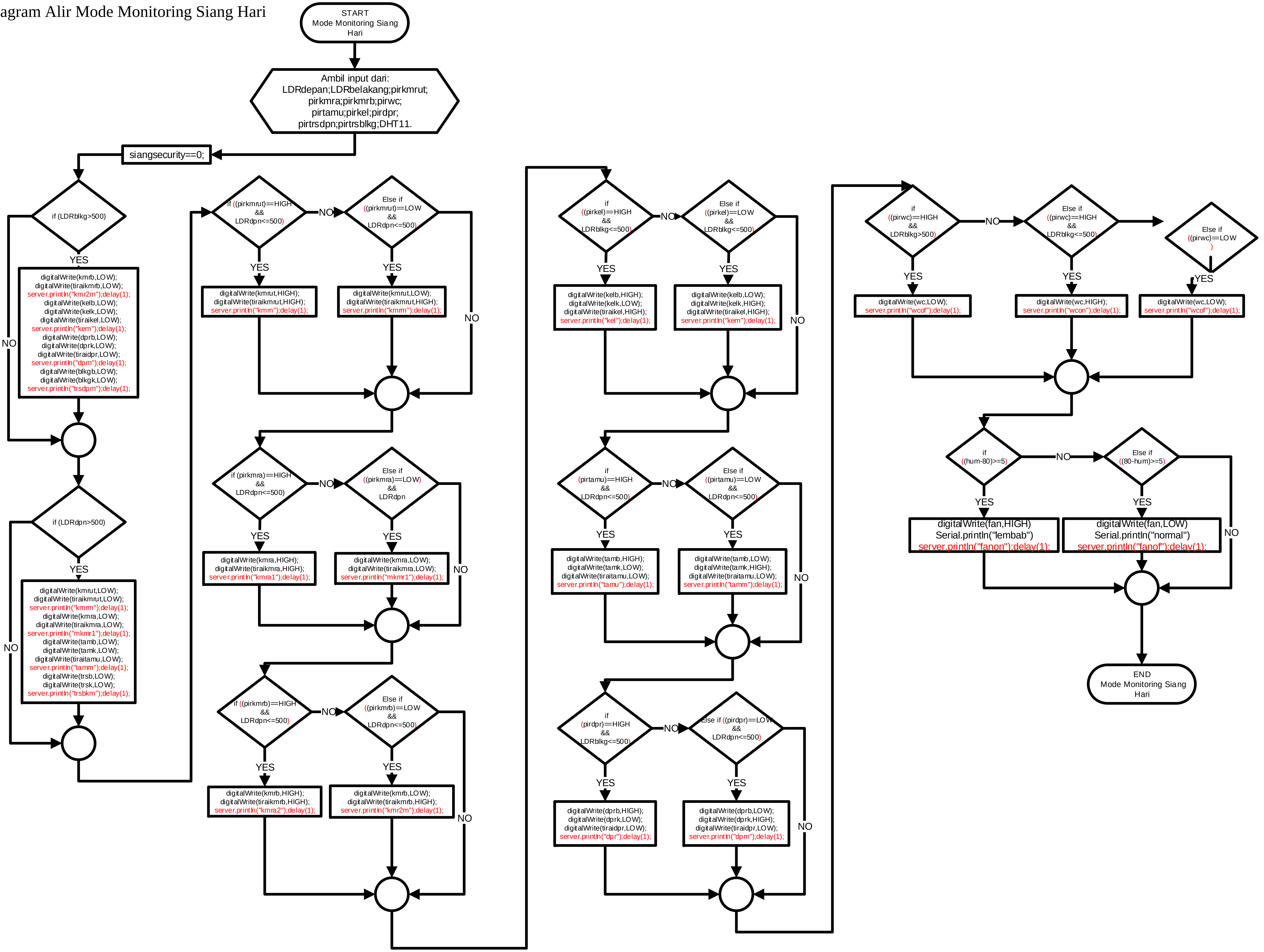
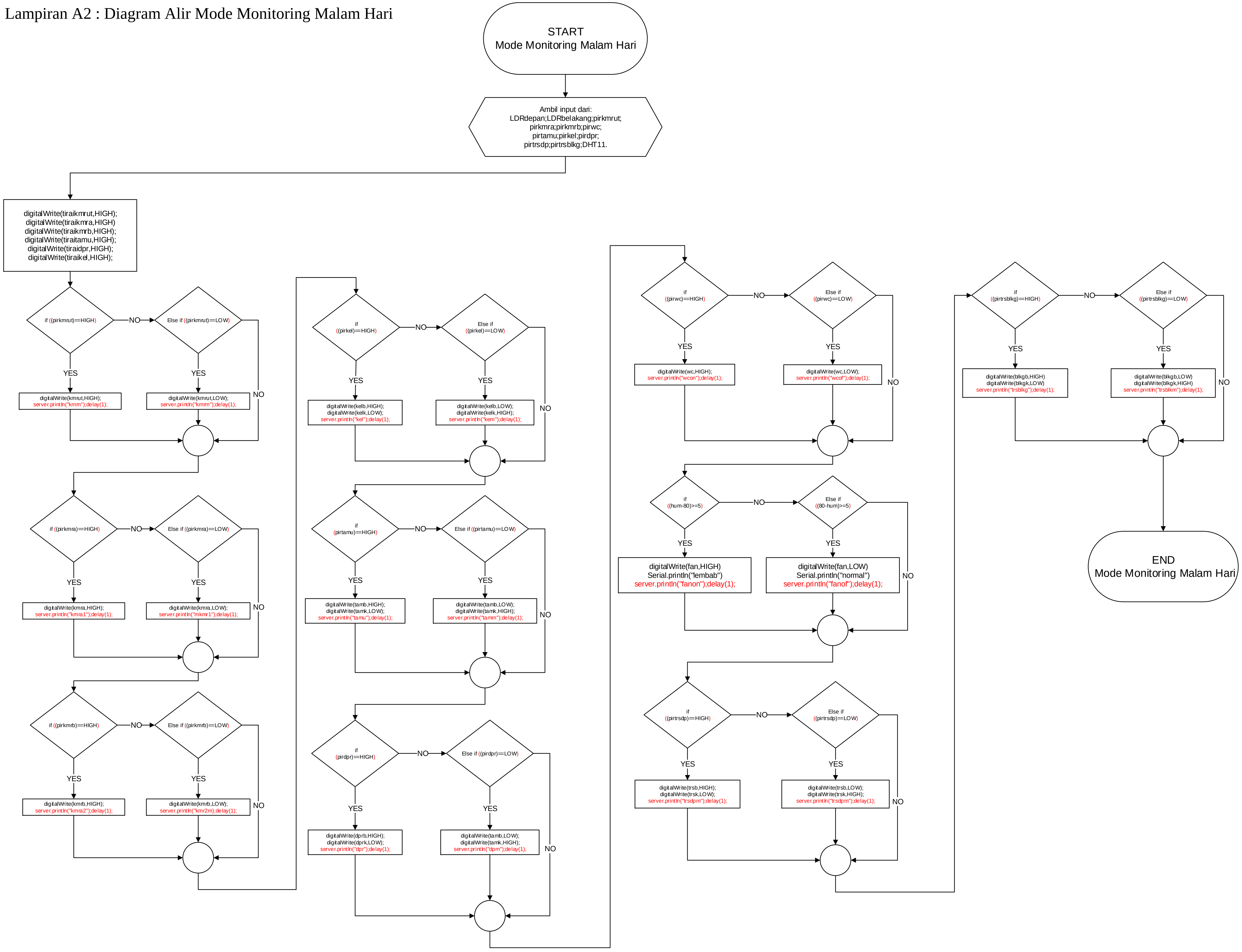
LAMPIRAN A
DIAGRAM ALIR

Lampiran A

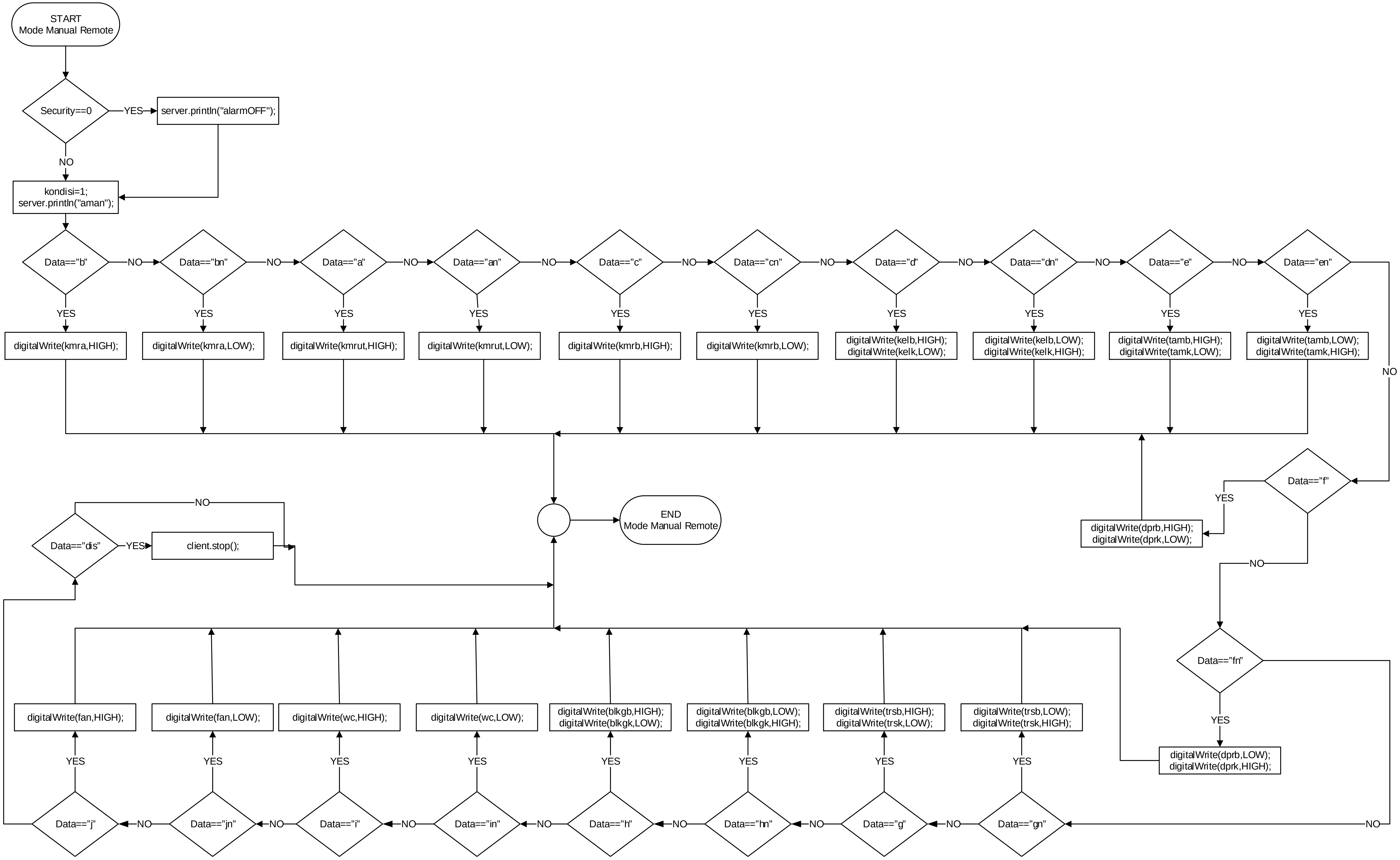
Lampiran A1 : Diagram Alir Mode Monitoring Siang Hari


START
Mode Monitoring Siang Hari
Ambil input dari:
LDRdepan;LDRbelakang;pirkmrut;
pirkmra;pirkmrb;pirwc;
pirtamu;pirkel;pirdpr;
pirtrsdpn;pirtrsblkg;DHT11.
siangsecurity==0;
if (LDRblkg>500)
YES
NO
digitalWrite(kmrb,LOW);
digitalWrite(tiraikmrb,LOW);
server.println("kmr2m");delay(1);
digitalWrite(kelb,LOW);
digitalWrite(kelk,LOW);
digitalWrite(tiraikel,LOW);
server.println("kem");delay(1);
digitalWrite(dprb,LOW);
digitalWrite(dprk,LOW);
digitalWrite(tiraidpr,LOW);
server.println("dpm");delay(1);
digitalWrite(blkgb,LOW);
digitalWrite(blkgk,LOW);
server.println("trsdpm");delay(1);
if (LDRdpn>500)
YES
NO
digitalWrite(kmrut,LOW);
digitalWrite(tiraikmrut,LOW);
server.println("kmrm");delay(1);
digitalWrite(kmra,LOW);
digitalWrite(tiraikmra,LOW);
server.println("mkmr1");delay(1);
digitalWrite(tamb,LOW);
digitalWrite(tamk,LOW);
digitalWrite(tiraitamu,LOW);
server.println("tamm");delay(1);
digitalWrite(trsb,LOW);
digitalWrite(trsk,LOW);
server.println("trsbkm");delay(1);
if ((pirkmrut)==HIGH && LDRdpn<=500)
NO
Else if ((pirkmrut)==LOW && LDRdpn<=500)
YES
YES
NO
digitalWrite(kmrut,HIGH);
digitalWrite(tiraikmrut,HIGH);
server.println("kmrm");delay(1);
digitalWrite(kmrut,LOW);
digitalWrite(tiraikmrut,HIGH);
server.println("kmrm");delay(1);
if (pirkmra)==HIGH && LDRdpn<=500)
NO
Else if ((pirkmra)==LOW) && LDRdpn
YES
YES
NO
digitalWrite(kmra,HIGH);
digitalWrite(tiraikmra,HIGH);
server.println("kmra1");delay(1);
digitalWrite(kmra,LOW);
digitalWrite(tiraikmra,LOW);
server.println("mkmr1");delay(1);
if ((pirkmrb)==HIGH && LDRdpn<=500)
NO
Else if ((pirkmrb)==LOW && LDRdpn<=500)
YES
YES
NO
digitalWrite(kmrb,HIGH);
digitalWrite(tiraikmrb,HIGH);
server.println("kmra2");delay(1);
digitalWrite(kmrb,LOW);
digitalWrite(tiraikmrb,HIGH);
server.println("kmr2m");delay(1);
if ((pirkel)==HIGH && LDRblkg<=500)
NO
Else if ((pirkel)==LOW && LDRblkg<=500)
YES
YES
NO
digitalWrite(kelb,HIGH);
digitalWrite(kelk,LOW);
digitalWrite(tiraikel,HIGH);
server.println("kel");delay(1);
digitalWrite(kelb,LOW);
digitalWrite(kelk,HIGH);
digitalWrite(tiraikel,HIGH);
server.println("kem");delay(1);
if (pirtamu)==HIGH && LDRdpn<=500)
NO
Else if ((pirtamu)==LOW && LDRdpn<=500)
YES
YES
NO
digitalWrite(tamb,HIGH);
digitalWrite(tamk,LOW);
digitalWrite(tiraitamu,LOW);
server.println("tamu");delay(1);
digitalWrite(tamb,LOW);
digitalWrite(tamk,HIGH);
digitalWrite(tiraitamu,LOW);
server.println("tamm");delay(1);
if (pirdpr)==HIGH && LDRblkg<=500)
NO
Else if ((pirdpr)==LOW && LDRdpn<=500)
YES
YES
NO
digitalWrite(dprb,HIGH);
digitalWrite(dprk,LOW);
digitalWrite(tiraidpr,LOW);
server.println("dpr");delay(1);
digitalWrite(dprb,LOW);
digitalWrite(dprk,HIGH);
digitalWrite(tiraidpr,LOW);
server.println("dpm");delay(1);
if ((pirwc)==HIGH && LDRblkg>500)
NO
Else if ((pirwc)==HIGH && LDRblkg<=500)
Else if ((pirwc)==LOW)
YES
YES
YES
digitalWrite(wc,LOW);
server.println("wcof");delay(1);
digitalWrite(wc,HIGH);
server.println("wcon");delay(1);
digitalWrite(wc,LOW);
server.println("wcof");delay(1);
if ((hum-80)>=5)
NO
Else if ((80-hum)>=5)
YES
YES
NO
digitalWrite(fan,HIGH)
Serial.println("lembab")
server.println("fanon");delay(1);
digitalWrite(fan,LOW)
Serial.println("normal")
server.println("fanof");delay(1);
END
Mode Monitoring Siang Hari

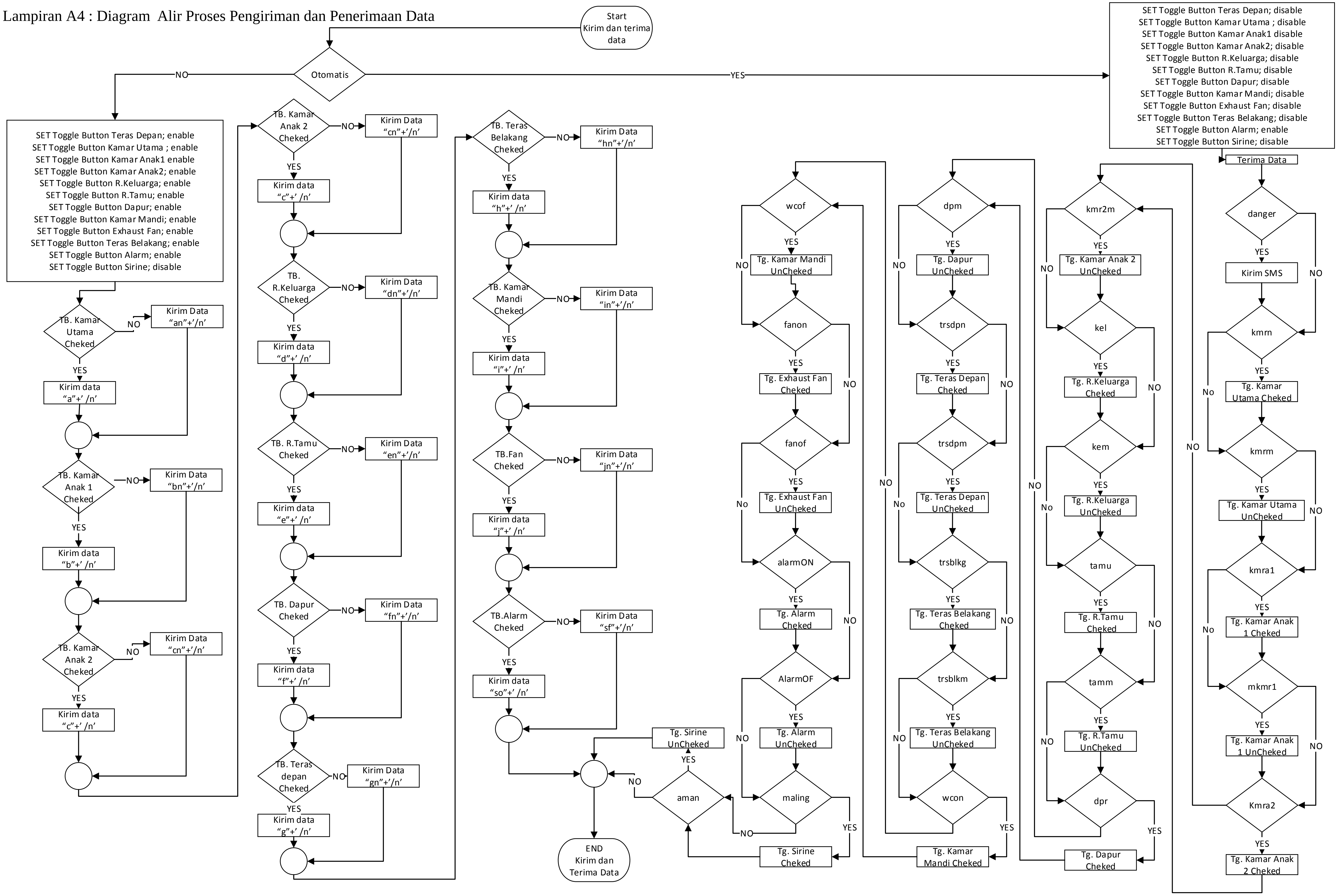
Lampiran A2 : Diagram Alir Mode Monitoring Malam Hari



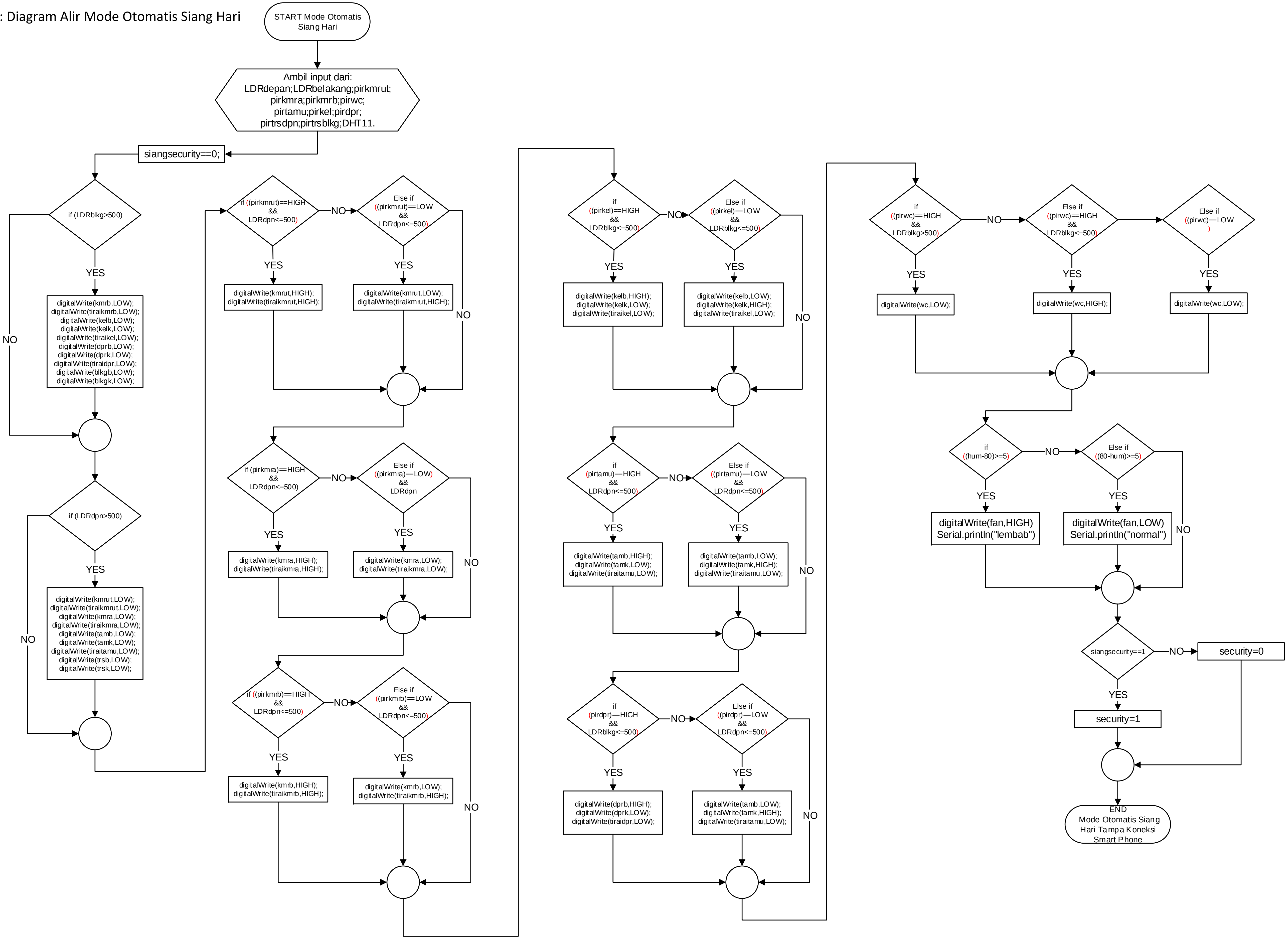
Lampiran A3 : Diagram Alir Mode Manual Remote



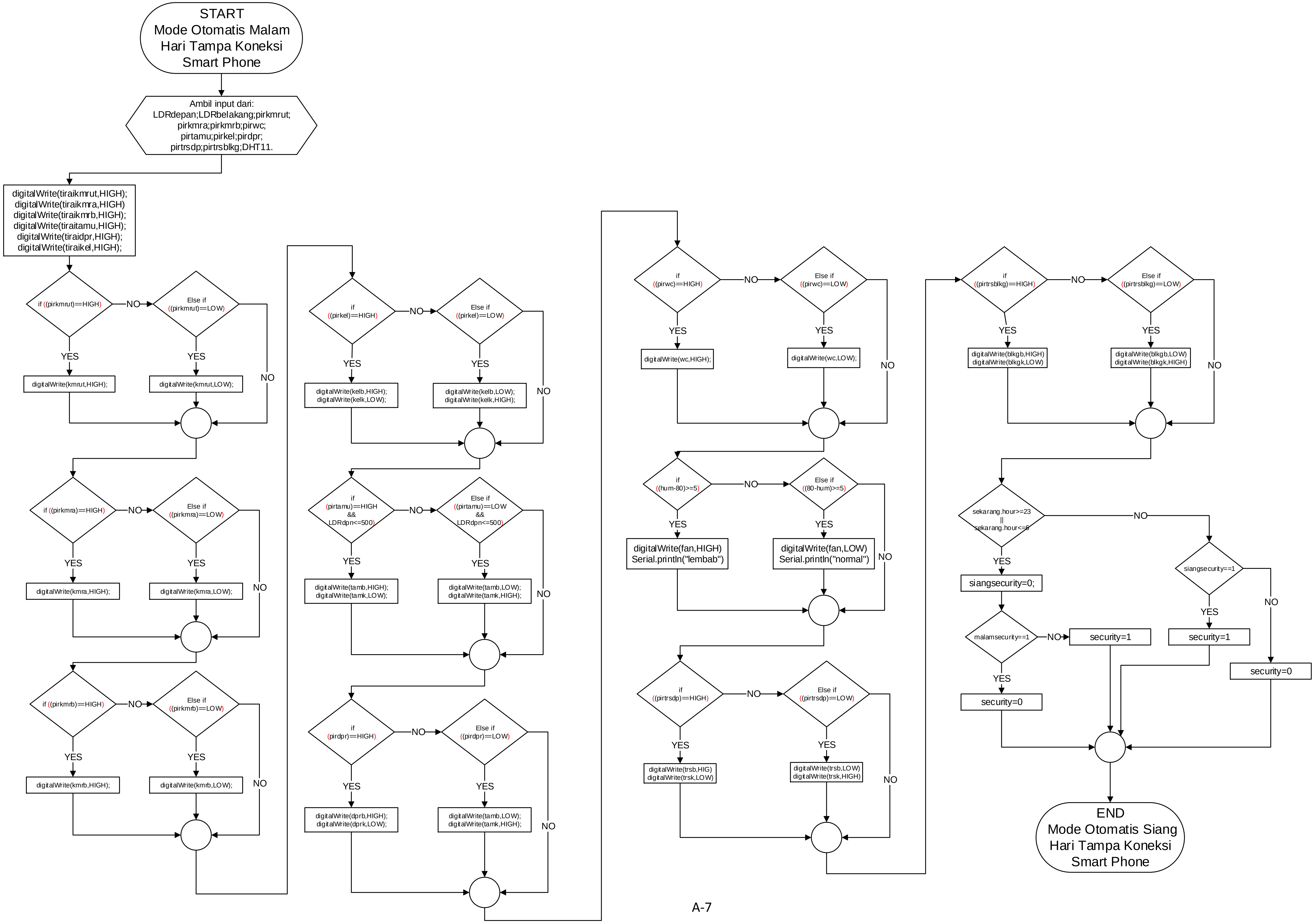
Lampiran A4 : Diagram Alir Proses Pengiriman dan Penerimaan Data



Lampiran A5: Diagram Alir Mode Otomatis Siang Hari


START Mode Otomatis Siang Hari
Ambil input dari: LDRdepan;LDRbelakang;pirkmrut; pirkmra;pirkmrb;pirwc; pirtamu;pirkel;pirdpr; pirtrsdpn;pirtrsblkg;DHT11.
siangsecurity==0;
if (LDRblkg>500)
YES
NO
digitalWrite(kmrb,LOW); digitalWrite(tiraikmrb,LOW); digitalWrite(kelb,LOW); digitalWrite(kelk,LOW); digitalWrite(tiraikel,LOW); digitalWrite(dprb,LOW); digitalWrite(dprk,LOW); digitalWrite(tiraidpr,LOW); digitalWrite(blkgb,LOW); digitalWrite(blkgk,LOW);
if (LDRdpn>500)
YES
NO
digitalWrite(kmrut,LOW); digitalWrite(tiraikmrut,LOW); digitalWrite(kmra,LOW); digitalWrite(tiraikmra,LOW); digitalWrite(tamb,LOW); digitalWrite(tamk,LOW); digitalWrite(tiraitamu,LOW); digitalWrite(trsb,LOW); digitalWrite(trsk,LOW);
if ((pirkmrut)==HIGH && LDRdpn<=500)
NO
Else if ((pirkmrut)==LOW && LDRdpn<=500)
YES
digitalWrite(kmrut,HIGH); digitalWrite(tiraikmrut,HIGH);
YES
digitalWrite(kmrut,LOW); digitalWrite(tiraikmrut,HIGH);
NO
if (pirkmra)==HIGH && LDRdpn<=500)
NO
Else if ((pirkmra)==LOW) && LDRdpn
YES
digitalWrite(kmra,HIGH); digitalWrite(tiraikmra,HIGH);
YES
digitalWrite(kmra,LOW); digitalWrite(tiraikmra,LOW);
NO
if ((pirkmrb)==HIGH && LDRdpn<=500)
NO
Else if ((pirkmrb)==LOW && LDRdpn<=500)
YES
digitalWrite(kmrb,HIGH); digitalWrite(tiraikmrb,HIGH);
YES
digitalWrite(kmrb,LOW); digitalWrite(tiraikmrb,HIGH);
NO
if ((pirkel)==HIGH && LDRblkg<=500)
NO
Else if ((pirkel)==LOW && LDRblkg<=500)
YES
digitalWrite(kelb,HIGH); digitalWrite(kelk,LOW); digitalWrite(tiraikel,LOW);
YES
digitalWrite(kelb,LOW); digitalWrite(kelk,HIGH); digitalWrite(tiraikel,LOW);
NO
if (pirtamu)==HIGH && LDRdpn<=500)
NO
Else if ((pirtamu)==LOW && LDRdpn<=500)
YES
digitalWrite(tamb,HIGH); digitalWrite(tamk,LOW); digitalWrite(tiraitamu,LOW);
YES
digitalWrite(tamb,LOW); digitalWrite(tamk,HIGH); digitalWrite(tiraitamu,LOW);
NO
if (pirdpr)==HIGH && LDRblkg<=500)
NO
Else if ((pirdpr)==LOW && LDRdpn<=500)
YES
digitalWrite(dprb,HIGH); digitalWrite(dprk,LOW); digitalWrite(tiraidpr,LOW);
YES
digitalWrite(tamb,LOW); digitalWrite(tamk,HIGH); digitalWrite(tiraitamu,LOW);
NO
if ((pirwc)==HIGH && LDRblkg>500)
NO
Else if ((pirwc)==HIGH && LDRblkg<=500)
Else if ((pirwc)==LOW)
YES
digitalWrite(wc,LOW);
YES
digitalWrite(wc,HIGH);
YES
digitalWrite(wc,LOW);
if ((hum-80)>=5)
NO
Else if ((80-hum)>=5)
YES
digitalWrite(fan,HIGH) Serial.println("lembab")
YES
digitalWrite(fan,LOW) Serial.println("normal")
NO
siangsecurity==1
NO
security=0
YES
security=1
END Mode Otomatis Siang Hari Tampa Koneksi Smart Phone

Lampiran A6: Diagram Alir Mode Otomatis Malam Hari



LAMPIRAN B
KODE PROGRAM

Lampiran B1 : Kode Pemrograman Arduino Mega2560

```
#include <DHT.h>
#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>
#include "RTClib.h"
#include <Wire.h>
#define DHTPIN 56
#define DHTTYPE DHT11
//mac dan IP
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF,
0xFE, 0xED };
IPAddress serverIP(192,168,1,10);
int serverPort=8888;
int LDR1pin=0;
int LDR2pin=1;
int LDRdpn;
int LDRblkg;
//sensor dht 11
DHT dht(DHTPIN,DHTTYPE);
//lampu
int trsk = 22; int trsb = 23; int tamk=24; int
tamb=25; int kelk=26; int kelb=27; int
blkgk=28; int blkgb = 29;
int dprk= 30; int dprb=31 ; int kmrut = 32;int
kmra= 33; int kmrb=34; int wc= 35;
int fan= 36;
//PIR
int pirwc = 37; int pirtamu =38; int pirkel =
39; int pirdpr =40; int pirtrsdp =41;int
pirtrsblkg=42; int pirkmrut =43;int
pirkmra=44;int pirkmrb=45;
//tirai
int tiraitamu = 46 ; int tiraikmrut = 47; int
tiraikmra = 48; int tiraikmrb=49;int
tiraidpr=50; int tiraikel=7;
int temp=1;
int siang;int sms;
int temp2=1;int security;int siangsecurity;int
malamsecurity;
int remote=0;int kondisi;
int jam;
int buzzer = 52;
RTC_DS1307 RTC;
// set sebagai server
EthernetServer server(serverPort);
void setup()
{
 Wire.begin(); //i2c beegin
 RTC.begin();//rtc setup
 dht.begin();
 //pin mode lampu
 pinMode(trsk,OUTPUT);
pinMode(trsb,OUTPUT);
pinMode(tamk,OUTPUT);
pinMode(tamb,OUTPUT);
pinMode(kelk,OUTPUT);
pinMode(kelb,OUTPUT);
pinMode(blkgk,OUTPUT);
 pinMode(blkgb,OUTPUT);
pinMode(dprk,OUTPUT);
pinMode(dprb,OUTPUT);
pinMode(kmrut,OUTPUT);
pinMode(kmra,OUTPUT);
pinMode(kmrb,OUTPUT);
pinMode(wc,OUTPUT);
pinMode(fan,OUTPUT);

 //pinmode tirai

 pinMode(tiraitamu,OUTPUT);
pinMode(tiraikmrut,OUTPUT);
pinMode(tiraikmra,OUTPUT);
pinMode(tiraikmrb,OUTPUT);
pinMode(tiraidpr,OUTPUT);
pinMode(tiraikel,OUTPUT);

 //pinmodePIR

pinMode(pirwc,INPUT);
pinMode(pirtamu,INPUT)
;pinMode(pirkel,INPUT);
pinMode(pirdpr,INPUT);
pinMode(pirtrsdp,INPUT);
pinMode(pirtrsblkg,INPUT);
pinMode(pirkmrut,INPUT);
pinMode(pirkmra,INPUT);
pinMode(pirkmrb,INPUT);
pinMode (buzzer,OUTPUT);

  Serial.begin(9600);

  // strat server

  Ethernet.begin(mac, serverIP);

  server.begin();

  Serial.println("Server started");//debug

  RTC.adjust(DateTime(__DATE__,
__TIME__));  //set waktu RTC

  attachInterrupt(4, securitysistem,RISING);

}

void securitysistem()

{

temp=2;

sms=1;

}

void loop()

{
```

```
//baca sensor LDR

 DateTime sekarang= RTC.now();

 LDRdpn=analogRead(LDR1pin);

          LDRblkg=analogRead(LDR2pin);

          float hum = dht.readHumidity();

          Serial.println("mode aouto");

          Serial.println(sekarang.hour(),DEC);

         Serial.println(sekarang.minute(),DEC);

         Serial.println(sekarang.second(),DEC);

          jam=sekarang.hour();

           Serial.println(hum);

          Serial.println(LDRdpn);

          delay(2000);

if((temp==0)&&(remote==0))

    {     if(jam<=18&&jam>=6)

           {siang=1;
           digitalWrite(buzzer,LOW);
           Serial.println("pagi");

if (LDRblkg>500)

{
digitalWrite(kmrb,LOW);
digitalWrite(tiraikmrb,LOW);
digitalWrite(kelb,LOW);
digitalWrite(kelk,LOW);
digitalWrite(tiraikel,LOW);
digitalWrite(dprb,LOW);
digitalWrite(dprk,LOW);
digitalWrite(tiraidpr,LOW);
digitalWrite(blkgb,LOW);
digitalWrite(blkgk,LOW);  }

           if(LDRdpn>500)
{digitalWrite(kmrut,LOW);
digitalWrite(tiraikmrut,LOW);
digitalWrite(kmra,LOW);
digitalWrite(tiraikmra,LOW);
digitalWrite(tamb,LOW);
digitalWrite(tamk,LOW);
digitalWrite(tiraitamu,LOW);
digitalWrite(trsb,LOW);
digitalWrite(trsk,LOW);   }


// lampu teras--------------------------------

            if (LDRblkg<=500){

            digitalWrite(blkgb,LOW);
           digitalWrite(blkgk,HIGH);}

            if (LDRdpn<=500){
           digitalWrite(trsb,LOW);
           digitalWrite(trsk,HIGH); }

//lampu teras end-----------------------------
//lamapu kamr utama

if (digitalRead(pirkmrut)==HIGH
&&LDRdpn<=500){
           digitalWrite(kmrut,HIGH);
           digitalWrite(tiraikmrut,HIGH);}

else if
(digitalRead(pirkmrut)==LOW&&LDRdpn<
=500){
           digitalWrite(kmrut,LOW);
digitalWrite(tiraikmrut,HIGH); }

//end lamapu kamr utama

//lampu kamar anaka----------------------------

if (digitalRead(pirkmra)==HIGH
&&LDRdpn<=500){
digitalWrite(kmra,HIGH);
digitalWrite(tiraikmra,HIGH); }

else if (digitalRead(pirkmra)==LOW
&&LDRdpn<=500){
digitalWrite(kmra,LOW);
digitalWrite(tiraikmra,HIGH); }

//end lampu kamar anaka------------------------

//lampu kamar anakb-----------------------------

if (digitalRead(pirkmrb)==HIGH
&&LDRblkg<=500){
digitalWrite(kmrb,HIGH);
digitalWrite(tiraikmrb,HIGH); }
```

```
else if (digitalRead(pirkmrb)==LOW
&&LDRblkg<=500){
digitalWrite(kmrb,LOW);


           digitalWrite(tiraikmrb,HIGH);}

//end lampu kamar anakb------------------------
//ruang keluarga--------------------------------

if (digitalRead(pirkel)==HIGH
&&LDRblkg<=500){
digitalWrite(kelb,HIGH);
digitalWrite(kelk,LOW);
digitalWrite(tiraikel,HIGH); }

else if
(digitalRead(pirkel)==LOW&&LDRblkg<=
500){

 digitalWrite(kelb,LOW);
digitalWrite(kelk,HIGH);
digitalWrite(tiraikel,HIGH); }

//ruang keluarga end----------------------------
//ruang Tamu-----------------------------------

if (digitalRead(pirtamu)==HIGH
&&LDRdpn<=500){

digitalWrite(tamb,HIGH);
digitalWrite(tamk,LOW);
digitalWrite(tiraitamu,HIGH); }

else if
(digitalRead(pirtamu)==LOW&&LDRdpn<
=500){

digitalWrite(kelb,LOW);
digitalWrite(kelk,HIGH);
digitalWrite(tiraikel,LOW); }

//ruang Tamu end---------------------------------
//ruang dpr--------------------------------------

if (digitalRead(pirdpr)==HIGH
&&LDRblkg<=500){

digitalWrite(dprb,HIGH);
digitalWrite(dprk,LOW);
digitalWrite(tiraidpr,LOW); }
```

```
else if (digitalRead(pirdpr)==LOW
&&LDRblkg<=500){
digitalWrite(dprb,LOW);
digitalWrite(dprk,HIGH);
digitalWrite(tiraidpr,LOW); }

//ruang dpr end-----------------------------------
//ruang WC---------------------------------------

if (digitalRead(pirwc)==HIGH){

digitalWrite(wc,HIGH);

}

else if (digitalRead(pirwc)==LOW){

digitalWrite(wc,LOW);

}

if ((hum-80)>=5)

{ digitalWrite(fan,HIGH);
Serial.println("lembab");}

else if ((80-hum)>=5)

{digitalWrite(fan,LOW);
Serial.println("normal");}

//ruang WC end----------------------------------

}//siang end--------------------------------------

else if(jam>=18 || jam<6){

siang=0;

digitalWrite(buzzer,LOW);
Serial.println("malam");
Serial.println(security);
server.println("malam");
digitalWrite(tiraikmrut,HIGH);
digitalWrite(tiraikmra,HIGH);
digitalWrite(tiraikmrb,HIGH);
digitalWrite(tiraitamu,HIGH);
digitalWrite(tiraidpr,HIGH);
digitalWrite(tiraikel,HIGH);
Serial.println("tirai done");

//kamar utama---------------------------

if (digitalRead(pirkmrut)==HIGH ){
digitalWrite(kmrut,HIGH); }
```

```
else if (digitalRead(pirkmrut)==LOW){
digitalWrite(kmrut,LOW);}

//end lamapu kamr utama-----------------------

//lampu kamar anaka---------------------------

if (digitalRead(pirkmra)==HIGH){
digitalWrite(kmra,HIGH);}

else if (digitalRead(pirkmra)==LOW){
digitalWrite(kmra,LOW); }

//end lampu kamar anaka-----------------------

//lampu kamar anakb---------------------------
if (digitalRead(pirkmrb)==HIGH ){
digitalWrite(kmrb,HIGH);}

else if (digitalRead(pirkmrb)==LOW){
digitalWrite(kmrb,LOW); }

//end lampu kamar anakb-----------------------

//ruang keluarga--------------------------------
if (digitalRead(pirkel)==HIGH ){
digitalWrite(kelb,HIGH);
digitalWrite(kelk,LOW); }

else if (digitalRead(pirkel)==LOW){
digitalWrite(kelb,LOW);
digitalWrite(kelk,HIGH); }

//ruang keluarga end---------------------------

//ruang Tamu------------------------------------

if (digitalRead(pirtamu)==HIGH){
digitalWrite(tamb,HIGH);
digitalWrite(tamk,LOW); }

else if (digitalRead(pirtamu)==LOW){
digitalWrite(tamb,LOW);
digitalWrite(tamk,HIGH);  }

//ruang Tamu end--------------------------------

//ruang dpr---------------------------------------

if (digitalRead(pirdpr)==HIGH){
digitalWrite(dprb,HIGH);
digitalWrite(dprk,LOW);
else if (digitalRead(pirdpr)==LOW){
digitalWrite(dprb,LOW);
digitalWrite(dprk,HIGH); }

//ruang dpr end------------------------------------

//ruang WC----------------------------------------

if (digitalRead(pirwc)==HIGH){
digitalWrite(wc,HIGH); }

 else if (digitalRead(pirwc)==LOW){
digitalWrite(wc,LOW); }

if ((hum-80)>=5)

 digitalWrite(fan,HIGH);
Serial.println("lembab");}

else if ((80-hum)>=5)
{digitalWrite(fan,LOW);
Serial.println("normal");}

//ruang WC end-----------------------------------

// teras depan--------------------------------------

if (digitalRead(pirtrsdp)==HIGH){
digitalWrite(trsb,HIGH);
digitalWrite(trsk,LOW); }

else if (digitalRead(pirtrsdp)==LOW){
digitalWrite(trsb,LOW);
digitalWrite(trsk,HIGH); }

//teras depan end----------------------------------

// teras blkg-----------------------------------------

if (digitalRead(pirtrsblkg)==HIGH){
digitalWrite(blkgb,HIGH);
digitalWrite(blkgk,LOW); }

else if (digitalRead(pirtrsblkg)==LOW){
digitalWrite(blkgb,LOW);
digitalWrite(blkgk,HIGH); }

//teras blkg end------------------------------------
}

//malem end---------------------------------------

}

else if((temp==2)&&(remote==0))
```

```
{if (security==1) {

  digitalWrite(buzzer,HIGH);
  digitalWrite(trsb,HIGH);
  digitalWrite(tamb,HIGH);
  digitalWrite(kelb,HIGH);
  digitalWrite(blkgb,HIGH);
  digitalWrite(dprb,HIGH);
  digitalWrite(kmrut,HIGH);
  digitalWrite(kmra,HIGH);
  digitalWrite(kmrb,HIGH);
  digitalWrite(wc,HIGH);
  Serial.println("harus sukses");

  delay(3000);

  digitalWrite(trsb,LOW);
  digitalWrite(tamb,LOW);
  digitalWrite(kelb,LOW);
  digitalWrite(blkgb,LOW);
  digitalWrite(dprb,LOW);
  digitalWrite(kmrut,LOW);
  digitalWrite(kmra,LOW);
  digitalWrite(kmrb,LOW);
  digitalWrite(wc,LOW); }

  else {

  temp=0; }

}


//tunggu client

EthernetClient client = server.available();

 if (client) //jika client konek

{String commandStr ="";//buat penyimpanan perintah

while (client.connected())

{  //selama client konek

remote=1;

if(temp==2)

{Serial.println(jam);

 if (security==1)
 {if (sms==1)
        {
        server.println("danger");delay(1); }

        server.println("maling");delay(1);

digitalWrite(buzzer,HIGH);
digitalWrite(trsb,HIGH);
digitalWrite(tamb,HIGH);
digitalWrite(kelb,HIGH);
digitalWrite(blkgb,HIGH);
digitalWrite(dprb,HIGH);
digitalWrite(kmrut,HIGH);
digitalWrite(kmra,HIGH);
digitalWrite(kmrb,HIGH);
digitalWrite(wc,HIGH);

delay(1000);

digitalWrite(trsb,LOW);
digitalWrite(tamb,LOW);
digitalWrite(kelb,LOW);
digitalWrite(blkgb,LOW);
digitalWrite(dprb,LOW);
digitalWrite(kmrut,LOW);
digitalWrite(kmra,LOW);
digitalWrite(kmrb,LOW);
digitalWrite(wc,LOW);

 delay(1000);

sms=0;

}

else

{   if(kondisi==0){

temp=0;}

 else if(kondisi==1){

temp=1;}

}

}

else if(temp == 0)

 {

kondisi=0;
```

```
DateTime sekarang= RTC.now();
digitalWrite(buzzer,LOW);
LDRdpn=analogRead(LDR1pin);
LDRblkg=analogRead(LDR2pin);
Serial.println("mode aouto");
Serial.println("remote");
Serial.println(sekarang.hour(),DEC);

int jam=sekarang.hour();

float hum = dht.readHumidity();
Serial.println(hum);

Serial.println(LDRdpn);

if(jam<=18 && jam>=6) {

siang=1;

Serial.println("pagi");

if (LDRblkg>500) {
digitalWrite(kmrb,LOW);
digitalWrite(tiraikmrb,LOW);
server.println("kmr2m");delay(1);
digitalWrite(kelb,LOW);
digitalWrite(kelk,LOW);
digitalWrite(tiraikel,LOW);
server.println("kem");delay(1);
digitalWrite(dprb,LOW);
digitalWrite(dprk,LOW);
digitalWrite(tiraidpr,LOW);
server.println("dpm");delay(1);
digitalWrite(blkgb,LOW);
digitalWrite(blkgk,LOW);  }

if(LDRdpn>500)

{digitalWrite(kmrut,LOW);
digitalWrite(tiraikmrut,LOW);
server.println("kmrm");delay(1);
digitalWrite(kmra,LOW);
digitalWrite(tiraikmra,LOW);
server.println("mkmr1");delay(1);
digitalWrite(tamb,LOW);
digitalWrite(tamk,LOW);
digitalWrite(tiraitamu,LOW);
server.println("tamm");delay(1);
digitalWrite(trsb,LOW);
digitalWrite(trsk,LOW); }
```

```
if (digitalRead(pirkmrut)==HIGH
&&LDRdpn<=500){
digitalWrite(kmrut,HIGH);
digitalWrite(tiraikmrut,HIGH);
server.println("kmrn");delay(1); }

else if (digitalRead(pirkmrut)==LOW
&&LDRdpn<=500){
digitalWrite(kmrut,LOW);
digitalWrite(tiraikmrut,HIGH);
server.println("kmrm");delay(1); }

//end lampu kamr utama

//lampu kamar anaka------------------------------

if (digitalRead(pirkmra)==HIGH
&&LDRdpn<=500){
digitalWrite(kmra,HIGH);
digitalWrite(tiraikmra,HIGH);
server.println("kmra1");delay(1); }

else if (digitalRead(pirkmra)==LOW
&&LDRdpn<=500){
digitalWrite(kmra,LOW);
digitalWrite(tiraikmra,HIGH);
server.println("mkmr1");delay(1); }

//end lampu kamar anaka------------------------

//lampu kamar anakb------------------------------

if (digitalRead(pirkmrb)==HIGH
&&LDRblkg<=500){
digitalWrite(kmrb,HIGH);
digitalWrite(tiraikmrb,HIGH);
server.println("kmra2");delay(1);}

else if (digitalRead(pirkmrb)==LOW
&&LDRblkg<=500){
digitalWrite(kmrb,LOW);
digitalWrite(tiraikmrb,HIGH);
server.println("kmr2m");delay(1); }

//end lampu kamar anakb------------------------

//ruang keluarga------------------------------------

if (digitalRead(pirkel)==HIGH
&&LDRblkg<=500){
digitalWrite(kelb,HIGH);
digitalWrite(kelk,LOW);
digitalWrite(tiraikel,HIGH);
server.println("kel");delay(1); }

else if (digitalRead(pirkel)==LOW
&&LDRblkg<=500){
digitalWrite(kelb,LOW);
digitalWrite(kelk,HIGH);
digitalWrite(tiraikel,HIGH);
server.println("kem");delay(1); }

//ruang keluarga end------------------------------

//ruang Tamu--------------------------------------
if (digitalRead(pirtamu)==HIGH
&&LDRdpn<=500){
digitalWrite(tamb,HIGH);
digitalWrite(tamk,LOW);
digitalWrite(tiraitamu,LOW);
server.println("tamu");delay(1); }

else if (digitalRead(pirtamu)==LOW
&&LDRdpn<=500){
digitalWrite(tamb,LOW);
digitalWrite(tamk,HIGH);
digitalWrite(tiraitamu,LOW);
server.println("tamm");delay(1); }

//ruang Tamu end---------------------------------

//ruang dpr------------------------------------------

if (digitalRead(pirdpr)==HIGH
&&LDRblkg<=500){
digitalWrite(dprb,HIGH);
digitalWrite(dprk,LOW);
digitalWrite(tiraidpr,LOW);
server.println("dpr");delay(1); }

else if (digitalRead(pirdpr)==LOW
&&LDRblkg<=500){

digitalWrite(dprb,LOW);
digitalWrite(dprk,HIGH);
digitalWrite(tiraidpr,LOW);
server.println("dpm");delay(1); }

//ruang dpr end------------------------------------

 //ruang WC---------------------------------------

if (digitalRead(pirwc)==HIGH
&&LDRblkg>500){
```

```
digitalWrite(wc,LOW);
server.println("wcof");delay(1); }

else if (digitalRead(pirwc)==HIGH
&&LDRblkg<=500){
digitalWrite(wc,HIGH);
server.println("wcon");delay(1); }

else if (digitalRead(pirwc)==LOW){

digitalWrite(wc,LOW);
server.println("wcof");delay(1); }

if ((hum-80)>=5)

{digitalWrite(fan,HIGH);
Serial.println("lembab");
server.println("fanon");delay(1);  }

 else if ((80-hum)>=5)

{digitalWrite(fan,LOW);
Serial.println("normal");
server.println("fanof");delay(1); }

//ruang WC end-----------------------------------
//teras siang----------------------------------------

if (LDRdpn<=500){

digitalWrite(trsb,LOW);
digitalWrite(trsk,HIGH); }

if (LDRblkg<=500){
digitalWrite(blkgb,LOW);
digitalWrite(blkgk,HIGH);}

//teras siang end----------------------------------

// teras orang---------------------------------------

if (digitalRead(pirtrsblkg)== HIGH){
server.println("trsblkg");delay(1);

}

else if (digitalRead(pirtrsblkg)== LOW){
server.println("trsblkm");delay(1); }

if (digitalRead(pirtrsdp)==HIGH ){
server.println("trsdpn");delay(1); }

else if (digitalRead(pirtrsdp)==LOW){
server.println("trsdpm");delay(1); }
```

```
//teras orang end-----------------------

}

//siang end--------------------------------------

else if(jam>=18 || jam<6)

{siang=0;

digitalWrite(buzzer,LOW);
Serial.println("malam");
server.println("malam");
server.println("aman");delay(1);
digitalWrite(tiraikmrut,HIGH);
digitalWrite(tiraikmra,HIGH);
digitalWrite(tiraikmrb,HIGH);
digitalWrite(tiraitamu,HIGH);
digitalWrite(tiraidpr,HIGH);
digitalWrite(tiraikel,HIGH);
Serial.println("tirai done");
Serial.println(hum);

if (digitalRead(pirkmrut)==HIGH ){
digitalWrite(kmrut,HIGH);
            Serial.println("utama");
server.println("kmrn");delay(1); }

else if (digitalRead(pirkmrut)==LOW){
digitalWrite(kmrut,LOW);
server.println("kmrm");delay(1); }

//end lamapu kamr utama

//lampu kamar anaka-----------------------------

if (digitalRead(pirkmra)==HIGH){

digitalWrite(kmra,HIGH);
server.println("kmra1");delay(1); }

else if (digitalRead(pirkmra)==LOW){

digitalWrite(kmra,LOW);
server.println("mkmr1");delay(1); }

//end lampu kamar anaka------------------------

//lampu kamar anakb-----------------------------

if (digitalRead(pirkmrb)==HIGH ){

digitalWrite(kmrb,HIGH);
server.println("kmra2");delay(1); }
```

```
else if (digitalRead(pirkmrb)==LOW ){
digitalWrite(kmrb,LOW);
server.println("kmr2m");delay(1); }

//end lampu kamar anakb------------------------


//ruang keluarga----------------------------------

if (digitalRead(pirkel)==HIGH ){

digitalWrite(kelb,HIGH);
digitalWrite(kelk,LOW);
server.println("kel");delay(1); }

else if (digitalRead(pirkel)==LOW){

digitalWrite(kelb,LOW);
digitalWrite(kelk,HIGH);
server.println("kem");delay(1); }

//ruang keluarga end------------------------------

//ruang Tamu--------------------------------------

if (digitalRead(pirtamu)==HIGH){

digitalWrite(tamb,HIGH);
digitalWrite(tamk,LOW);
server.println("tamu");delay(1); }

else if (digitalRead(pirtamu)==LOW){

digitalWrite(tamb,LOW);
digitalWrite(tamk,HIGH);
server.println("tamm");delay(1); }

//ruang Tamu end---------------------------------

//ruang dpr-----------------------------------------

if (digitalRead(pirdpr)==HIGH){

digitalWrite(dprb,HIGH);
digitalWrite(dprk,LOW);
server.println("dpr");delay(1); }

else if (digitalRead(pirdpr)==LOW){

digitalWrite(dprb,LOW);
digitalWrite(dprk,HIGH);
server.println("dpm");delay(1); }

//ruang dpr end------------------------------------
```

```
//teras depan--------------------------------------

if (digitalRead(pirtrsdp)==HIGH ){

digitalWrite(trsb,HIGH);
digitalWrite(trsk,LOW);
server.println("trsdpn");delay(1); }

else if (digitalRead(pirtrsdp)==LOW){

digitalWrite(trsb,LOW);
digitalWrite(trsk,HIGH);
server.println("trsdpm");delay(1); }

//teras depan end----------------------------------

// teras blkg----------------------------------------

if (digitalRead(pirtrsblkg)==HIGH){

digitalWrite(blkgb,HIGH);
digitalWrite(blkgk,LOW);
server.println("trsblkg");delay(1); }

else if (digitalRead(pirtrsblkg)==LOW){

digitalWrite(blkgb,LOW);
digitalWrite(blkgk,HIGH);
server.println("trsblkm");delay(1); }

// teras blkg end------------------------------------

//ruang WC------------------------------------------

if (digitalRead(pirwc)==HIGH){

digitalWrite(wc,HIGH);Serial.println("test");
server.println("wcon");delay(1); }

else if (digitalRead(pirwc)==LOW){

digitalWrite(wc,LOW);
server.println("wcof");delay(1); }

if ((hum-80)>=5)

{digitalWrite(fan,HIGH);
Serial.println("lembab");
server.println("fanon");delay(1);}

else if ((80-hum)>=5)
{digitalWrite(fan,LOW);
Serial.println("normal");
server.println("fanof");delay(1);}

//ruang WC end-------------------------------------
```

```
}
//malem end---------------------------------------
}
if (client.available())
{//jika client kirim data
char c = client.read();//baca input dari client
  commandStr+=c;//masukan ke commandstr
   if (c == '\n') //jika ketemu karakter \n maka proses pengiriman selesai
{ Serial.println("Command:"+commandStr);
String data = commandStr;
data.replace("\n","");
Serial.println(data);
if (data==("m")){
        temp = 0;  Serial.println("auto");
        Serial.println(temp);}
else if (data==("o")){
        temp = 1;
        Serial.println("manual"); }
else if (data==("so") ){
        security = 1;Serial.println(security); }
else if (data==("sf") ){
        security = 0; Serial.println(security);}
Serial.println(temp);
//auto end------------------------------------------
//manual start--------------------------------------
if (temp==1) {
 kondisi=1;
if(security==0)
{server.println("alarmOF");}
server.println("aman");delay(1);
         if(data==("b") )
         {digitalWrite(kmra,HIGH);}
         else if(data ==("bn"))
         {digitalWrite(kmra,LOW);}
         else if(data==("a"))
         {digitalWrite(kmrut,HIGH);}
         else if(data==("an")   )
         {digitalWrite(kmrut,LOW);}
         else if(data==("c")   )
         {digitalWrite(kmrb,HIGH);}
         else if(data==("cn")   )
         {digitalWrite(kmrb,LOW);}
         else if(data==("d")   )
         {digitalWrite(kelb,HIGH);
          digitalWrite(kelk,LOW);}
         else if(data==("dn")   )
         {digitalWrite(kelb,LOW);
          digitalWrite(kelk,HIGH);}
         else if(data==("e")   )
         {digitalWrite(tamb,HIGH);
          digitalWrite(tamk,LOW);}
         else if(data==("en")   )
         {digitalWrite(tamb,LOW);
          digitalWrite(tamk,HIGH);}
          else if(data==("f")   )
         {digitalWrite(dprb,HIGH);
          digitalWrite(dprk,LOW);}
          else if(data==("fn")   )
```

```
              {digitalWrite(dprb,LOW);
               digitalWrite(dprk,HIGH);}
               else if(data==("g")   )
              {digitalWrite(trsb,HIGH);
               digitalWrite(trsk,LOW);}
               else if(data==("gn")   )
              {digitalWrite(trsb,LOW);
               digitalWrite(trsk,HIGH);}
               else if(data==("h")   )
              {digitalWrite(blkgb,HIGH);
               digitalWrite(blkgk,LOW);}
               else if(data==("hn")   )
              {digitalWrite(blkgb,LOW);
               digitalWrite(blkgk,HIGH);}
               else if(data==("i")   )
              {digitalWrite(wc,HIGH);}
               else if(data==("in")   )
              {digitalWrite(wc,LOW);}
               else if(data==("j")   )
              {digitalWrite(fan,HIGH);}
              else if(data==("jn")   )
              {digitalWrite(fan,LOW);}
              else if(data==("dis"))
               {client.stop();}}
commandStr="";//reset commandStr
}     } delay(500);//delay untuk terima data
            }
    remote=0;
    delay(1);
    // tutup koneksi

    client.stop();
  }
  }
```

Lampiran B2 : Kode Pemrograman Aplikasi Android

Kode Pemrograman SmartHouse Activity.Java

```
package smart.house;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.SocketAddress;
import android.widget.EditText;
import android.widget.Toast;
import android.app.Activity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.TextView;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ToggleButton;


public class SmarthouseActivity extends Activity {

        EditText alamat;
        TextView textlog;
        Button mode;
        Button konek;
        String temp ="otomatis";
        String alamatip;
        Boolean Connected = false;
        Boolean modemanual = false;
        NetworkTask networktask;
        ToggleButton tgkmrut;
        ToggleButton tgkmrank1;
        ToggleButton tgkmrank2;
        ToggleButton tgkel;
        ToggleButton tgtamu;
        ToggleButton tgtrsdpn;
        ToggleButton tgtrsblkg;
        ToggleButton tgdpr;
        ToggleButton tgkmrmnd;
        ToggleButton tgfan;
        ToggleButton tgalarm;
        ToggleButton tgsirine;


        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.ruang);
                // menghubungkan variabel dengan object
```

```
            textlog = (TextView) findViewById(R.id.progress);
            konek = (Button) findViewById(R.id.konek);
            mode = (Button) findViewById(R.id.mode);
            tgkmrut = (ToggleButton) findViewById(R.id.kmrut);
            tgkmrank1 = (ToggleButton) findViewById(R.id.kmrank1);
            tgkmrank2 = (ToggleButton) findViewById(R.id.kmrank2);
            tgkel = (ToggleButton) findViewById(R.id.keluarga);
            tgtamu = (ToggleButton) findViewById(R.id.tamu);
            tgdpr = (ToggleButton) findViewById(R.id.dpr);
            tgtrsdpn = (ToggleButton) findViewById(R.id.trsdpn);
            tgtrsblkg = (ToggleButton) findViewById(R.id.trsblkg);
            tgkmrmnd = (ToggleButton) findViewById(R.id.kmrmnd);
            tgfan = (ToggleButton) findViewById(R.id.fan);
            tgalarm=(ToggleButton) findViewById(R.id.alarm);
            tgsirine=(ToggleButton) findViewById(R.id.sirine);
            alamat= (EditText) findViewById(R.id.alamat);

             changeConnectionStatus(false);
            //ubahmode(false);

            // event listener
            mode.setOnClickListener(modeclick);
            konek.setOnClickListener(koneklistener);
            mode.setText(temp);
            tgkmrut.setOnClickListener(tgkmrutlst);
            tgkmrank1.setOnClickListener(tgkmrank1lst);
            tgkmrank2.setOnClickListener(tgkmrank2lst);
            tgkel.setOnClickListener(tgkellst);
            tgtamu.setOnClickListener(tgtamulst);
            tgdpr.setOnClickListener(tgdprlst);
            tgtrsdpn.setOnClickListener(tgtrsdpnlst);
            tgtrsblkg.setOnClickListener(tgtrsblkglst);
            tgkmrmnd.setOnClickListener(tgkmrmndlst);
            tgfan.setOnClickListener(tgfanlst);
            tgalarm.setOnClickListener(tgalarmlst);
            tgsirine.setEnabled(false);
            networktask= new NetworkTask();


      }

private OnClickListener tgkmrank1lst = new OnClickListener(){
      public void onClick(View v){
            if((tgkmrank1.isChecked()))
            {     networktask.KirimData("b"+'\n');
      }
      else {
            networktask.KirimData("bn"+'\n');
      }
}};


private OnClickListener tgkmrutlst = new OnClickListener(){
      public void onClick(View v){
            if((tgkmrut.isChecked()))
            {     networktask.KirimData("a"+'\n');
      }
      else {
            networktask.KirimData("an"+'\n');
```

```
    }
}};
private OnClickListener tgkmrank2lst = new OnClickListener(){
    public void onClick(View v){
        if((tgkmrank2.isChecked()))
        {   networktask.KirimData("c"+'\n');
    }
    else {
        networktask.KirimData("cn"+'\n');
    }
}};
private OnClickListener tgkellst = new OnClickListener(){
    public void onClick(View v){
        if((tgkel.isChecked()))
        {   networktask.KirimData("d"+'\n');
    }
    else {
        networktask.KirimData("dn"+'\n');
    }
}};
private OnClickListener tgtamulst = new OnClickListener(){
    public void onClick(View v){
        if((tgtamu.isChecked()))
        {   networktask.KirimData("e"+'\n');
    }
    else {
        networktask.KirimData("en"+'\n');
    }
}};
private OnClickListener tgdprlst = new OnClickListener(){
    public void onClick(View v){
        if((tgdpr.isChecked()))
        {   networktask.KirimData("f"+'\n');
    }
    else {
        networktask.KirimData("fn"+'\n');
    }
}};
private OnClickListener tgtrsdpnlst = new OnClickListener(){
    public void onClick(View v){
        if((tgtrsdpn.isChecked()))
        {   networktask.KirimData("g"+'\n');
    }
    else {
        networktask.KirimData("gn"+'\n');
    }
}};
private OnClickListener tgtrsblkglst = new OnClickListener(){
    public void onClick(View v){
        if((tgtrsblkg.isChecked()))
        {   networktask.KirimData("h"+'\n');
    }
    else {
        networktask.KirimData("hn"+'\n');
    }
}};
private OnClickListener tgkmrmndlst = new OnClickListener(){
    public void onClick(View v){
        if((tgkmrmnd.isChecked()))
        {   networktask.KirimData("i"+'\n');
    }
    else {
        networktask.KirimData("in"+'\n');
    }
}};
private OnClickListener tgfanlst = new OnClickListener(){
    public void onClick(View v){
        if((tgfan.isChecked()))
        {   networktask.KirimData("j"+'\n');
    }
    else {
        networktask.KirimData("jn"+'\n');
    }
}};
private OnClickListener tgalarmlst = new OnClickListener(){
    public void onClick(View v){
        if((tgalarm.isChecked()))
        {   networktask.KirimData("so"+'\n');

    }
    else {
        networktask.KirimData("sf"+'\n');

    }
}};
    // konek button eventlistener
    private OnClickListener koneklistener = new OnClickListener() {
        public void onClick(View v) {
            alamatip =alamat.getText().toString();
            if (!Connected) {

                textproses("connect server");
                networktask = new NetworkTask();
                networktask.execute();


            } else {
                textproses("disconnect server");
                networktask.KirimData("dis"+'\n');
                if (networktask!= null) {
                    networktask.closeSocket();
                    networktask.cancel(true);

                }
            }

        }
    };

    private OnClickListener modeclick = new OnClickListener(){
        public void onClick(View v){
            if (temp == "manual"){
                networktask.KirimData("o"+'\n');
                temp = "otomatis";
                mode.setText("auto");
                tgkmrut.setEnabled(true);
                tgkmrank1.setEnabled(true);
                tgkmrank2.setEnabled(true);
                tgkel.setEnabled(true);
                tgtamu.setEnabled(true);
                tgdpr.setEnabled(true);
                tgtrsdpn.setEnabled(true);
                tgtrsblkg.setEnabled(true);
                tgkmrmnd.setEnabled(true);
                tgfan.setEnabled(true);

            }else{
                networktask.KirimData("m"+'\n');
                temp = "manual";
                mode.setText("manual");
                tgkmrut.setEnabled(false);
                tgkmrank1.setEnabled(false);
                tgkmrank2.setEnabled(false);
                tgkel.setEnabled(false);
                tgtamu.setEnabled(false);
                tgdpr.setEnabled(false);
                tgtrsdpn.setEnabled(false);
                tgtrsblkg.setEnabled(false);
                tgkmrmnd.setEnabled(false);
                tgfan.setEnabled(false);
            }
        }

    };

    // network class
    public class NetworkTask extends AsyncTask<Void, byte[], Boolean> {
        Socket nsocket;
        InputStream nis;
        OutputStream nos;
        BufferedReader inFromServer;

        @Override
        protected void onPreExecute() {
            changeConnectionStatus(true);
        }

        @Override
        protected Boolean doInBackground(Void... params) {
            boolean result = false;

            try {
                SocketAddress sockaddr = new InetSocketAddress(
                        alamatip, 8888);
                nsocket = new Socket();
                nsocket.connect(sockaddr, 5000);
                if (nsocket.isConnected()) {
                    nis = nsocket.getInputStream();
                    nos = nsocket.getOutputStream();
                    inFromServer = new BufferedReader(
                            new InputStreamReader(nis));
                    while (true) {
                        String msgFromServer =
inFromServer.readLine();
                        byte[] theByteArray =
msgFromServer.getBytes();
                        publishProgress(theByteArray);
                    }
                }

            } catch (IOException e) {
                e.printStackTrace();
                result = true;
            } catch (Exception e) {
                e.printStackTrace();
                result = true;
            } finally {
                closeSocket();
            }


            return result;
        }

        // method closesocket
        public void closeSocket() {
            try {
                nis.close();
                nos.close();
                nsocket.close();
            } catch (IOException e) {
                e.printStackTrace();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        public void KirimData(String cmd){
            try {
                if (nsocket.isConnected()){
                    nos.write(cmd.getBytes());
                }else{
                    textproses("tidak dapat mengirim pesan, socket
dalam keadaan tertutup");
                }
            }catch (Exception e){
                textproses("gagal kirimdata");
            }
        }

        //methode terima data


        @Override
        protected void onProgressUpdate(byte[]...values){
            if (values.length > 0) {//jika terima data
                  String command=new String(values[0]);
                  textproses(command);
                  if(command.indexOf("danger")==0)
                  {
                    try{
                        SmsManager smsManager =
SmsManager.getDefault();
                        smsManager.sendTextMessage("085721380200",
null, "ada maling", null, null);
```

```
                                       Toast.makeText(getApplicationContext(),
"SMS Sent!",
       Toast.LENGTH_LONG).show();
                                  }catch (Exception e) {
       Toast.makeText(getApplicationContext(),
                                                     "SMS faild, please try again
later!",
                                                     Toast.LENGTH_LONG).show();
                                              e.printStackTrace();
                          }}
                             else if(command.indexOf("kmrn")==0){
                                 tgkmrut.setChecked(true);
                             }else if(command.indexOf("kmrm")==0){
                                 tgkmrut.setChecked(false);
                             }else if (command.indexOf("kmra1")==0){
                                 tgkmrank1.setChecked(true);
                             }else if (command.indexOf("mkmr1")==0){
                                 tgkmrank1.setChecked(false);
                             }else if (command.indexOf("kmra2")==0){
                                 tgkmrank2.setChecked(true);
                             }else if (command.indexOf("kmr2m")==0){
                                 tgkmrank2.setChecked(false);
                             }else if (command.indexOf("kel")==0){
                                 tgkel.setChecked(true);
                             }else if (command.indexOf("kem")==0){
                                 tgkel.setChecked(false);
                             }else if (command.indexOf("tamu")==0){
                                 tgtamu.setChecked(true);
                             }else if (command.indexOf("tamm")==0){
                                 tgtamu.setChecked(false);
                             }else if (command.indexOf("dpr")==0){
                                 tgdpr.setChecked(true);
                             }else if (command.indexOf("dpm")==0){
                                 tgdpr.setChecked(false);
                             }else if (command.indexOf("trsdpn")==0){
                                 tgtrsdpn.setChecked(true);
                             }else if (command.indexOf("trsdpm")==0){
                                 tgtrsdpn.setChecked(false);
                             }else if (command.indexOf("trsblkg")==0){
                                 tgtrsblkg.setChecked(true);
                             }else if (command.indexOf("trsblkm")==0){
                                 tgtrsblkg.setChecked(false);
                             }else if (command.indexOf("wcon")==0){
                                 tgkmrmnd.setChecked(true);
                             }else if (command.indexOf("wcof")==0){
                                 tgkmrmnd.setChecked(false);
                             }else if (command.indexOf("fanon")==0){
                                 tgfan.setChecked(true);
                             }else if (command.indexOf("fanof")==0){
                                 tgfan.setChecked(false);
                             }else if (command.indexOf("alarmON")==0){
                                 tgalarm.setChecked(true);
                             }else if (command.indexOf("alarmOF")==0){
                                 tgalarm.setChecked(false);
                             }else if (command.indexOf("maling")==0){
                                 tgsirine.setChecked(true);
                             }else if (command.indexOf("aman")==0){
                                 tgsirine.setChecked(false);
                             }
                         }
                    }

             @Override
             protected void onCancelled(){
                    changeConnectionStatus(false);
             }
             //Method is called after taskexecution
             @Override
             protected void onPostExecute(Boolean result){
                    if(result){
                          textproses("onPostExecute: Completed with an Error."
);
                    }else {
                          textproses("onPostExecute: Completed.");
                    }changeConnectionStatus(false);
             }
}

       //network claass abis
public void changeConnectionStatus(Boolean isConnected) {
             Connected=isConnected;//change variable
             if(isConnected){//if connection established
                    textproses("successfully connected to server");//log
                    konek.setText("disconnect");//change Buttontext
             }else{
                    textproses("disconnected from Server!");//log
                    konek.setText("connect");//change Buttontext
             }
       }// text
public void textproses(String msg) {

       textlog.setText(msg+"\n");
}
public void onBackPressed()
{networktask.KirimData("dis"+'\n');
finish();
       }
protected void onPaused(){
       super.onPause();
       networktask.KirimData("dis"+'\n');
       if(networktask!=null){
             networktask.closeSocket();
             networktask.cancel(true);
       }
}
       @Override
protected void onDestroy(){
             super.onDestroy();
             if(networktask!=null){
                    networktask.closeSocket();
                    networktask.cancel(true);
             }
       }
}
```

Kode Pemrograman Ruang.xml (layout)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.c
om/apk/res/android"
    android:layout_width="match_parent"

android:layout_height="match_parent"
    android:orientation="vertical" >

    <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"
        android:orientation="vertical"
>

        <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content" >

            <EditText

android:id="@+id/alamat"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_weight="1"
                android:ems="10"

android:imeOptions="actionDone"
                >

                <requestFocus />
            </EditText>

            <Button
                android:id="@+id/konek"

android:layout_width="132dp"

android:layout_height="wrap_content"
                android:text="Button"
/>

        </LinearLayout>

        <LinearLayout

android:layout_width="match_parent"

android:layout_height="38dp"
```

```
android:orientation="vertical" >

            <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/progress"

android:layout_width="180dp"

android:layout_height="wrap_content"

android:text="Medium Text"

android:textAppearance="?android:attr/t
extAppearanceMedium" />

            </LinearLayout>
        </LinearLayout>

        <Button
            android:id="@+id/mode"

android:layout_width="match_parent"

android:layout_height="wrap_content"
            android:text="Button" />

        <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content" >

            <ToggleButton
                android:id="@+id/alarm"

android:layout_width="wrap_content"

android:layout_height="match_parent"

android:layout_weight="0.15"

android:textOff="@string/security_off"

android:textOn="@string/security_on" />

            <ToggleButton

android:id="@+id/sirine"
```

```
android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_weight="0.09"

android:textOn="@string/DANGER"

android:textOff="@string/aman" />

        </LinearLayout>

    </LinearLayout>

    <ScrollView
        android:id="@+id/scrollView1"

android:layout_width="match_parent"

android:layout_height="wrap_content" >

        <TableLayout

android:layout_width="match_parent"

android:layout_height="wrap_content" >

            <TableRow

android:id="@+id/tableRow1"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView11"

android:layout_width="176dp"

android:layout_height="match_parent"

android:layout_weight="0.00"

android:text="@string/kamar_utama"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/kmrut"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>

            <TableRow

android:id="@+id/tableRow2"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView1"

android:layout_width="176dp"

android:layout_height="wrap_content"

android:layout_weight="0.00"

android:text="@string/kamar_anak1"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/kmrank1"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>

            <TableRow

android:id="@+id/tableRow3"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView2"

android:layout_width="176dp"

android:layout_height="wrap_content"

android:text="@string/kamar_anak2"
```

```
android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/kmrank2"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:textOff="off"
                    android:textOn="On"
/>
            </TableRow>

            <TableRow

android:id="@+id/tableRow4"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView5"

android:layout_width="176dp"

android:layout_height="wrap_content"

android:text="@string/ruang_tamu"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/tamu"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>

            <TableRow

android:id="@+id/tableRow5"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >
```

```
                <TextView

android:id="@+id/textView3"

android:layout_width="176dp"

android:layout_height="match_parent"

android:text="@string/ruang_keluarga"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/keluarga"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>

            <TableRow

android:id="@+id/tableRow6"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView4"

android:layout_width="176dp"

android:layout_height="wrap_content"

android:text="@string/teras_depan"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/trsdpn"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>
            <TableRow

android:id="@+id/tableRow7"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView6"

android:layout_width="176dp"

android:layout_height="wrap_content"

android:text="@string/teras_belakang"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/trsblkg"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>

            <TableRow

android:id="@+id/tableRow8"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView7"

android:layout_width="176dp"

android:layout_height="wrap_content"

android:text="@string/dapur"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/dpr"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>

            <TableRow

android:id="@+id/tableRow9"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView8"

android:layout_width="176dp"

android:layout_height="wrap_content"

android:text="@string/wc"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/kmrmnd"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>

            <TableRow

android:id="@+id/tableRow10"

android:layout_width="wrap_content"

android:layout_height="wrap_content" >

                <TextView

android:id="@+id/textView9"

android:layout_width="176dp"

android:layout_height="wrap_content"
```

```
android:text="@string/kipas_kamar"

android:textAppearance="?android:attr/t
extAppearanceLarge" />

                <ToggleButton

android:id="@+id/fan"

android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"

android:textOff="@string/off"

android:textOn="@string/on" />
            </TableRow>
        </TableLayout>
    </ScrollView>

</LinearLayout>
```

Kode Pemrograman Android Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="smart.house"
    android:versionCode="1"
    android:versionName="1.0" >
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.SEND_SMS"  />
    <uses-sdk android:minSdkVersion="8" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".SmarthouseActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>


    </application>

</manifest>
```

LAMPIRAN C

DATA SHEET

8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash

ATmega640/V
ATmega1280/V
ATmega1281/V
ATmega2560/V
ATmega2561/V

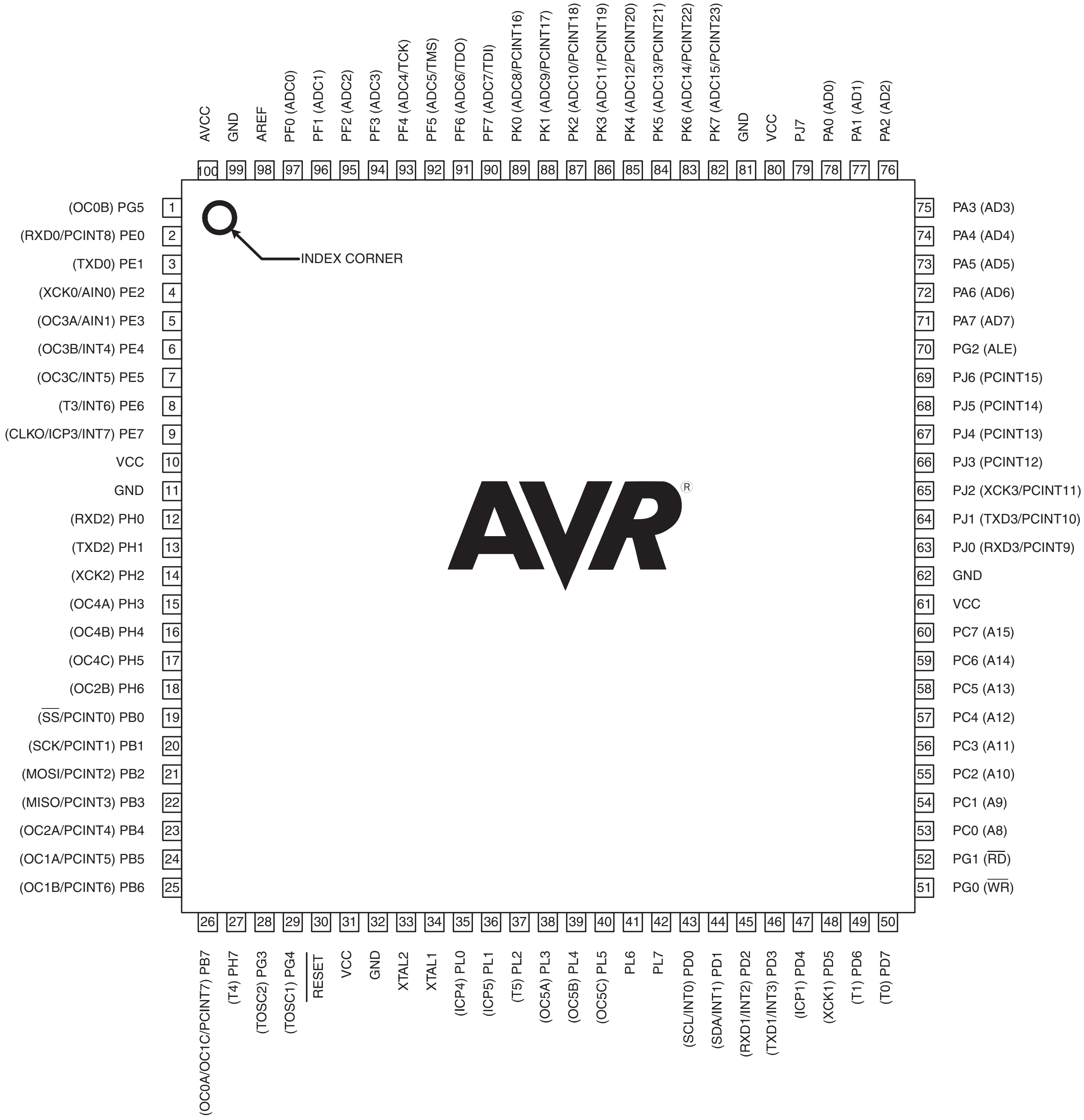
Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 × 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 64K/128K/256KBytes of In-System Self-Programmable Flash
  - 4Kbytes EEPROM
  - 8Kbytes Internal SRAM
  - Write/Erase Cycles:10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/ 100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
    - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix® acquisition
  - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
  - Output Compare Modulator
  - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
  - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
  - Master/Slave SPI Serial Interface
  - Byte Oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
  - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
  - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
  - RoHS/Fully Green
- Temperature Range:
  - -40°C to 85°C Industrial
- Ultra-Low Power Consumption
  - Active Mode: 1MHz, 1.8V: 500µA
  - Power-down Mode: 0.1µA at 1.8V
- Speed Grade:
  - ATmega640V/ATmega1280V/ATmega1281V:
    - 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega2560V/ATmega2561V:
    - 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
  - ATmega640/ATmega1280/ATmega1281:
    - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
  - ATmega2560/ATmega2561:
    - 0 - 16MHz @ 4.5V - 5.5V

2549O–AVR–05/12

1. Pin Configurations

Figure 1-1. TQFP-pinout ATmega640/1280/2560



31. Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to V_{CC}+0.5V
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0mA
DC Current V_{CC} and GND Pins	200.0mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

31.1 DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage,Except XTAL1 and Reset pin	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	-0.5 -0.5		$0.2V_{CC}$[1] $0.3V_{CC}$[1]	V
V_{IL1}	Input Low Voltage, XTAL1 pin	V_{CC} = 1.8V - 5.5V	-0.5		$0.1V_{CC}$[1]	
V_{IL2}	Input Low Voltage, RESET pin	V_{CC} = 1.8V - 5.5V	-0.5		$0.1V_{CC}$[1]	
V_{IH}	Input High Voltage, Except XTAL1 and RESET pins	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	$0.7V_{CC}$[2] $0.6V_{CC}$[2]		V_{CC} + 0.5 V_{CC} + 0.5	
V_{IH1}	Input High Voltage, XTAL1 pin	V_{CC} = 1.8V - 2.4V V_{CC} = 2.4V - 5.5V	$0.8V_{CC}$[2] $0.7V_{CC}$[2]		V_{CC} + 0.5 V_{CC} + 0.5	
V_{IH2}	Input High Voltage, RESET pin	V_{CC} = 1.8V - 5.5V	$0.9V_{CC}$[2]		V_{CC} + 0.5	
V_{OL}	Output Low Voltage[3], Except RESET pin	I_{OL} = 20mA, V_{CC} = 5V I_{OL} = 10mA, V_{CC} = 3V			0.9 0.6	
V_{OH}	Output High Voltage[4], Except RESET pin	I_{OH} = -20mA, V_{CC} = 5V I_{OH} = -10mA, V_{CC} = 3V	4.2 2.3			
I_{IL}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin low (absolute value)			1	µA
I_{IH}	Input Leakage Current I/O Pin	V_{CC} = 5.5V, pin high (absolute value)			1	
R_{RST}	Reset Pull-up Resistor		30		60	kΩ
R_{PU}	I/O Pin Pull-up Resistor		20		50	

T_A = -40°C to 85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted) (Continued)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
I_{CC}	Power Supply Current[5]	Active 1MHz, V_{CC} = 2V (ATmega640/1280/2560/1V)		0.5	0.8	mA
		Active 4MHz, V_{CC} = 3V (ATmega640/1280/2560/1L)		3.2	5	
		Active 8MHz, V_{CC} = 5V (ATmega640/1280/1281/2560/2561)		10	14	
		Idle 1MHz, V_{CC} = 2V (ATmega640/1280/2560/1V)		0.14	0.22	
		Idle 4MHz, V_{CC} = 3V (ATmega640/1280/2560/1L)		0.7	1.1	
		Idle 8MHz, V_{CC} = 5V (ATmega640/1280/1281/2560/2561)		2.7	4	
	Power-down mode	WDT enabled, V_{CC} = 3V		<5	15	µA
		WDT disabled, V_{CC} = 3V		<1	7.5	
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V V_{in} = $V_{CC}/2$		<10	40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V V_{in} = $V_{CC}/2$	-50		50	nA
t_{ACID}	Analog Comparator Propagation Delay	V_{CC} = 2.7V V_{CC} = 4.0V		750 500		ns

Notes:
1. "Max" means the highest value where the pin is guaranteed to be read as low.
2. "Min" means the lowest value where the pin is guaranteed to be read as high.
3. Although each I/O port can sink more than the test conditions (20mA at VCC = 5V, 10mA at VCC = 3V) under steady state conditions (non-transient), the following must be observed:
   ATmega1281/2561:
   1.)The sum of all IOL, for ports A0-A7, G2, C4-C7 should not exceed 100mA.
   2.)The sum of all IOL, for ports C0-C3, G0-G1, D0-D7 should not exceed 100mA.
   3.)The sum of all IOL, for ports G3-G5, B0-B7, E0-E7 should not exceed 100mA.
   4.)The sum of all IOL, for ports F0-F7 should not exceed 100mA.
   ATmega640/1280/2560:
   1.)The sum of all IOL, for ports J0-J7, A0-A7, G2 should not exceed 200mA.
   2.)The sum of all IOL, for ports C0-C7, G0-G1, D0-D7, L0-L7 should not exceed 200mA.
   3.)The sum of all IOL, for ports G3-G4, B0-B7, H0-B7 should not exceed 200mA.
   4.)The sum of all IOL, for ports E0-E7, G5 should not exceed 100mA.
   5.)The sum of all IOL, for ports F0-F7, K0-K7 should not exceed 100mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
4. Although each I/O port can source more than the test conditions (20mA at VCC = 5V, 10mA at VCC = 3V) under steady state conditions (non-transient), the following must be observed:
   ATmega1281/2561:
   1)The sum of all IOH, for ports A0-A7, G2, C4-C7 should not exceed 100mA.
   2)The sum of all IOH, for ports C0-C3, G0-G1, D0-D7 should not exceed 100mA.
   3)The sum of all IOH, for ports G3-G5, B0-B7, E0-E7 should not exceed 100mA.
   4)The sum of all IOH, for ports F0-F7 should not exceed 100mA.
   ATmega640/1280/2560:
   1)The sum of all IOH, for ports J0-J7, G2, A0-A7 should not exceed 200mA.
   2)The sum of all IOH, for ports C0-C7, G0-G1, D0-D7, L0-L7 should not exceed 200mA.
   3)The sum of all IOH, for ports G3-G4, B0-B7, H0-H7 should not exceed 200mA.
   4)The sum of all IOH, for ports E0-E7, G5 should not exceed 100mA.

5)The sum of all IOH, for ports F0-F7, K0-K7 should not exceed 100mA.
If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.

5. Values with "PRR1 – Power Reduction Register 1" enabled (0xFF).

31.2 Speed Grades

Maximum frequency is depending on $V_{CC.}$ As shown in Figure 31-1 trough Figure 31-4 on page 370, the Maximum Frequency vs. V_{CC} curve is linear between 1.8V < V_{CC} < 2.7V and between 2.7V < V_{CC} < 4.5V.

31.2.1 8MHz

Figure 31-1. Maximum Frequency vs. V_{CC}, ATmega640V/1280V/1281V/2560V/2561V

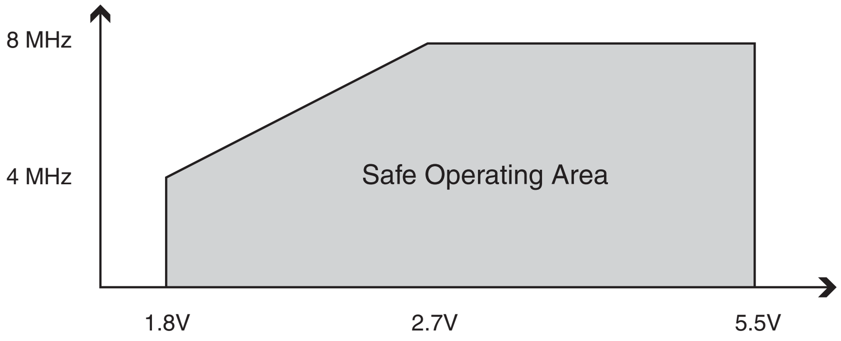
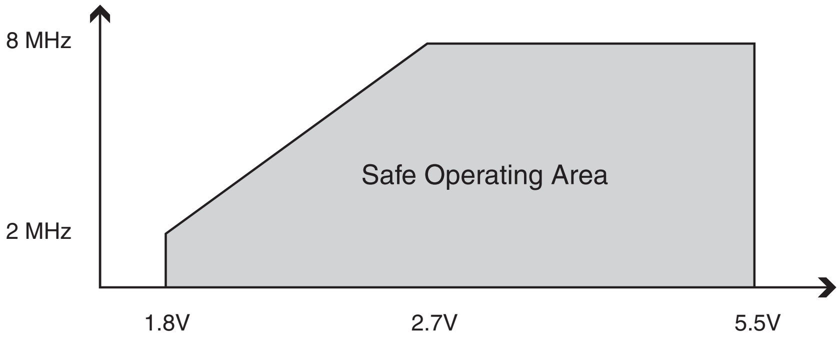


Figure 31-2. Maximum Frequency vs. V_{CC} when also No-Read-While-Write Section[(1)], ATmega2560V/ATmega2561V, is used



Note: 1. When only using the Read-While-Write Section of the program memory, a higher speed can be achieved at low voltage, see "Read-While-Write and No Read-While-Write Flash Sections" on page 317 for addresses.

DHT11 Humidity & Temperature Sensor

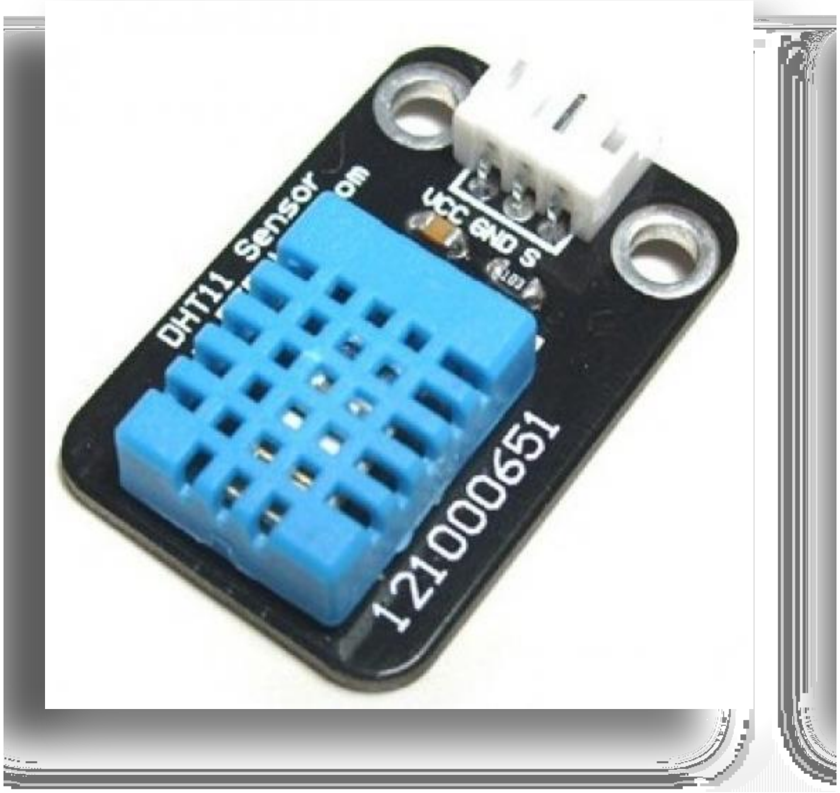
D-Robotics UK (www.droboticsonline.com)

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output.

D-Robotics

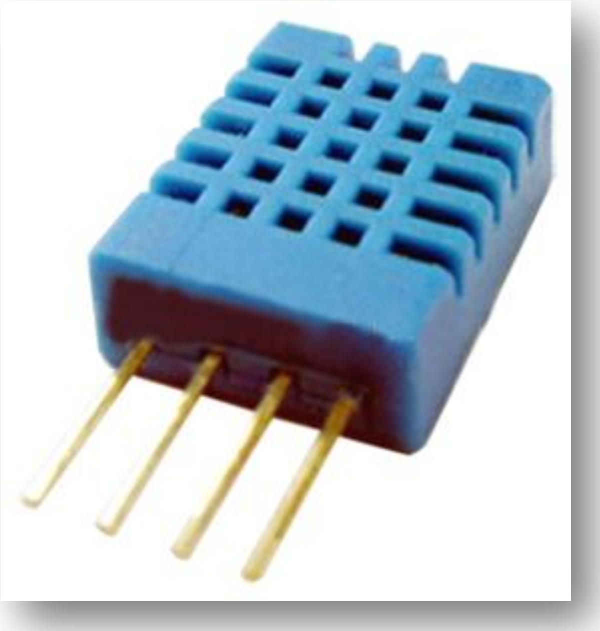
7/30/2010

DHT 11 Humidity & Temperature Sensor



1. Introduction

This DFRobot DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.



Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

2. Technical Specifications:

Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5％RH	±2℃	1	4 Pin Single Row

Detailed Specifications:

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃，1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

3. Typical Application (Figure 1)

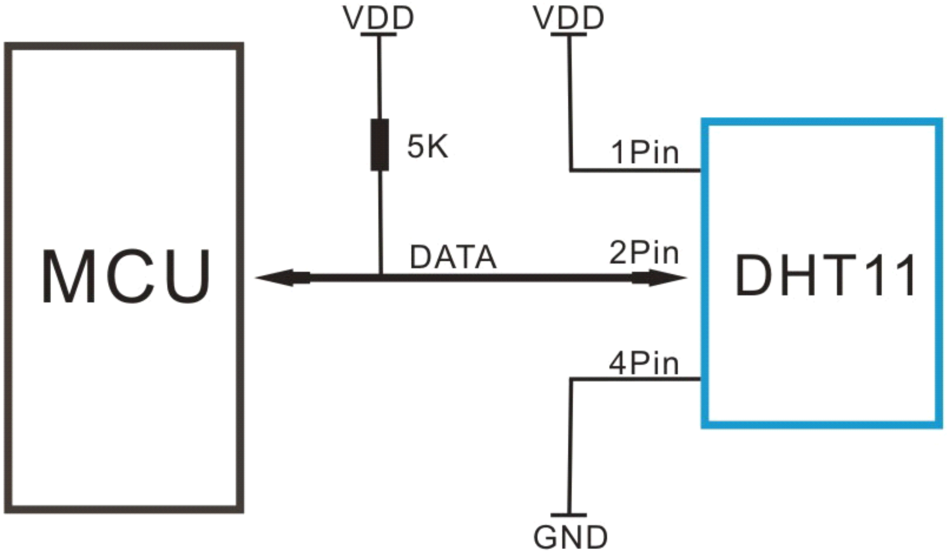


Figure 1 Typical Application

Note: 3Pin – Null; MCU = Micro-computer Unite or single chip Computer

When the connecting cable is shorter than 20 metres, a 5K pull-up resistor is recommended; when the connecting cable is longer than 20 metres, choose a appropriate pull-up resistor as needed.

4. Power and Pin

DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

5. Communication Process: Serial Interface (Single-Wire Two-Way)

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.

Data consists of decimal and integral parts. A complete data transmission is **40bit**, and the sensor sends **higher data bit** first.

Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

5.1 Overall Communication Process (Figure 2, below)

When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again.

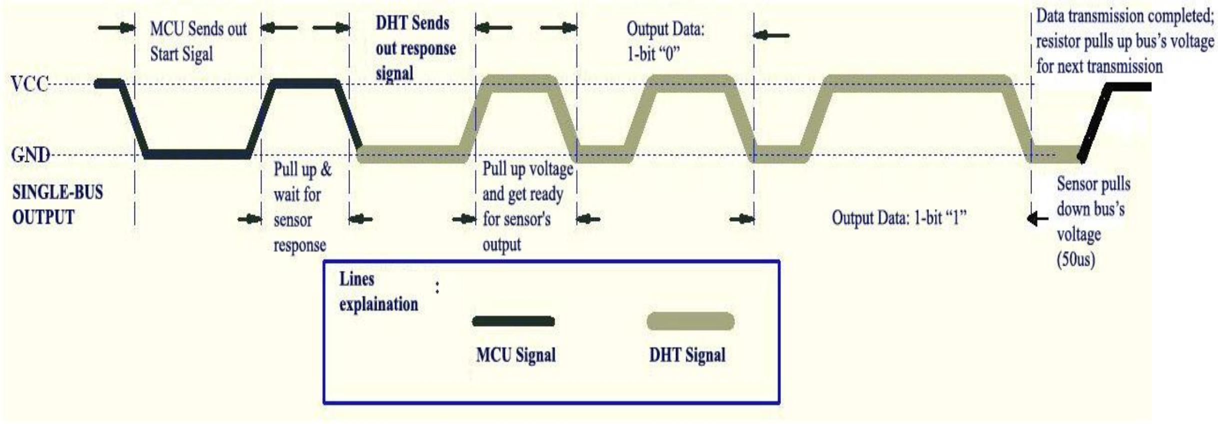

MCU Sends out Start Sigal
DHT Sends out response signal
Output Data: 1-bit "0"
Data transmission completed; resistor pulls up bus's voltage for next transmission
VCC
GND
SINGLE-BUS OUTPUT
Pull up & wait for sensor response
Pull up voltage and get ready for sensor's output
Output Data: 1-bit "1"
Sensor pulls down bus's voltage (50us)
Lines explaination :
MCU Signal
DHT Signal

Figure 2 Overall Communication Process

5.2 MCU Sends out Start Signal to DHT (Figure 3, below)

Data Single-bus free status is at high voltage level. When the communication between MCU and DHT11 begins, the programme of MCU will set Data Single-bus voltage level from high to low and this process must take at least 18ms to ensure DHT's detection of MCU's signal, then MCU will pull up voltage and wait 20-40us for DHT's response.

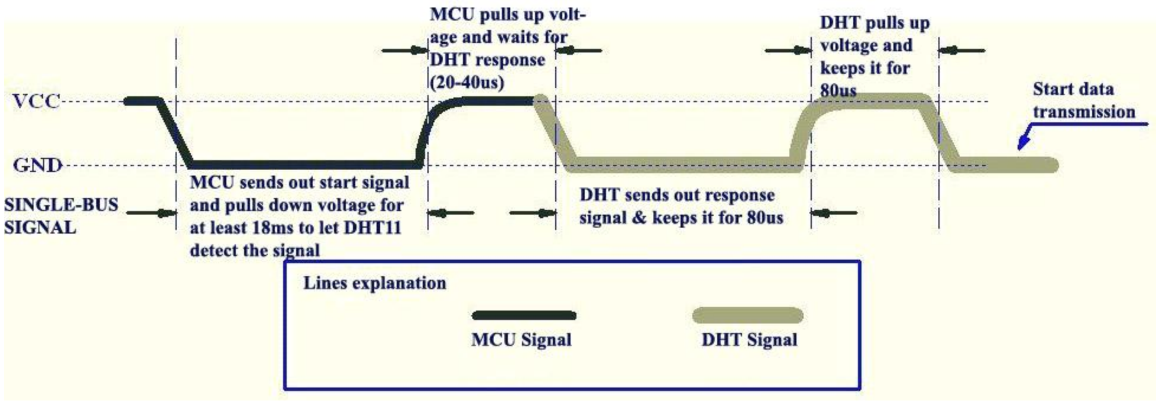

MCU pulls up voltage and waits for DHT response (20-40us)
DHT pulls up voltage and keeps it for 80us
VCC
GND
Start data transmission
SINGLE-BUS SIGNAL
MCU sends out start signal and pulls down voltage for at least 18ms to let DHT11 detect the signal
DHT sends out response signal & keeps it for 80us
Lines explanation
MCU Signal
DHT Signal

Figure 3 MCU Sends out Start Signal & DHT Responses

5.3 DHT Responses to MCU (Figure 3, above)

Once DHT detects the start signal, it will send out a low-voltage-level response signal, which lasts 80us. Then the programme of DHT sets Data Single-bus voltage level from low to high and keeps it for 80us for DHT's preparation for sending data.

When DATA Single-Bus is at the low voltage level, this means that DHT is sending the response signal. Once DHT sent out the response signal, it pulls up voltage and keeps it for 80us and prepares for data transmission.

When DHT is sending data to MCU, every bit of data begins with the 50us low-voltage-level and the length of the following high-voltage-level signal determines whether data bit is "0" or "1" (see Figures 4 and 5 below).

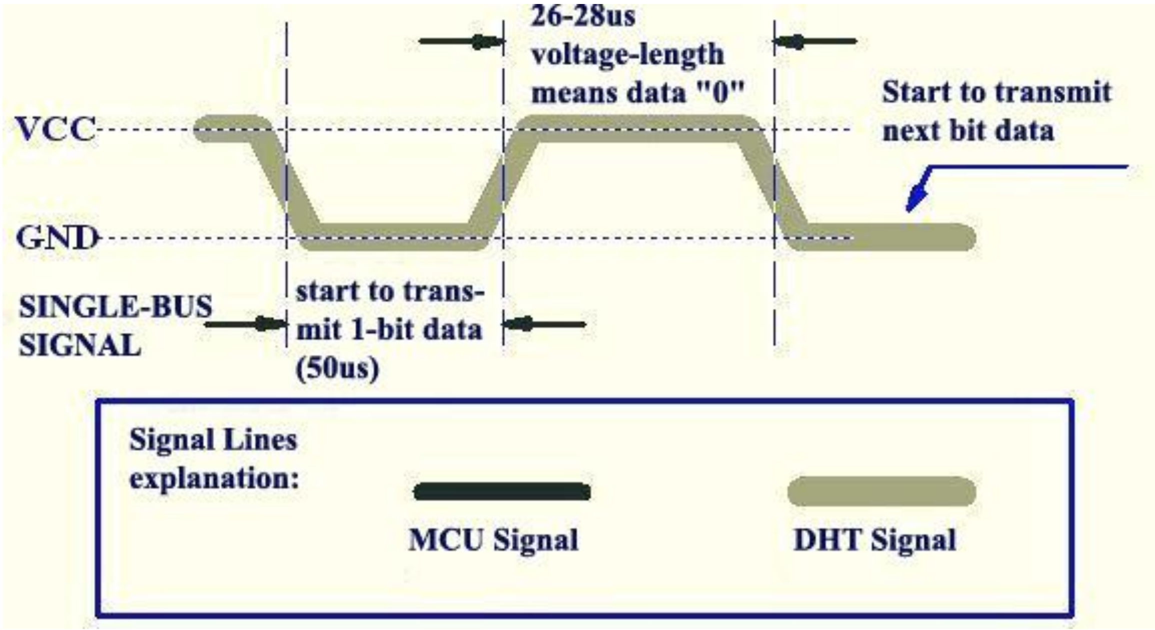


Figure 4 Data "0" Indication

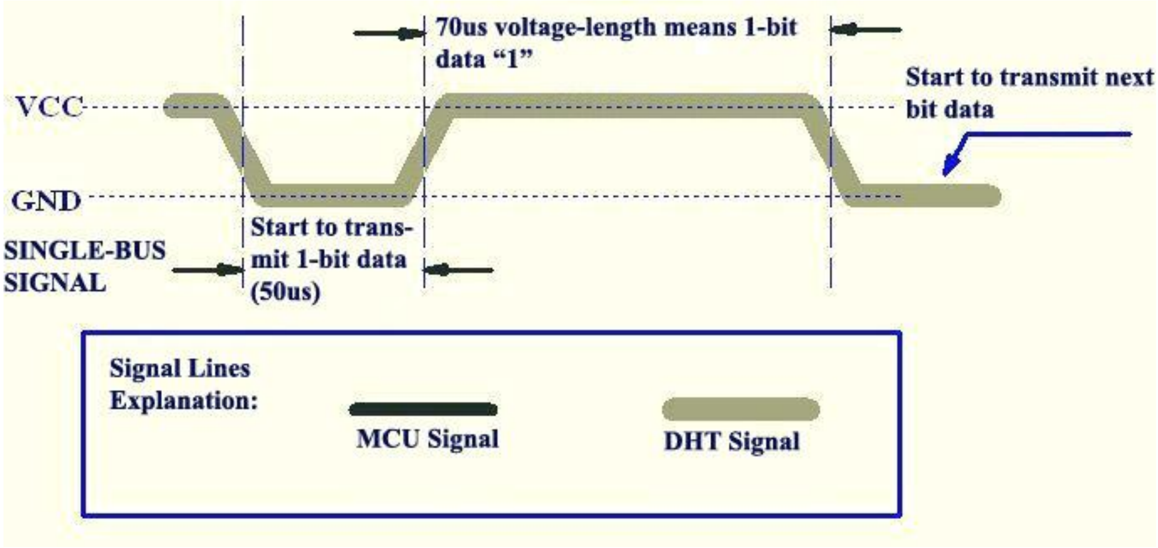

70us voltage-length means 1-bit data "1"
Start to transmit next bit data
VCC
GND
SINGLE-BUS SIGNAL
Start to trans- mit 1-bit data (50us)
Signal Lines Explanation:
MCU Signal
DHT Signal

Figure 5 Data "1" Indication

If the response signal from DHT is always at high-voltage-level, it suggests that DHT is not responding properly and please check the connection. When the last bit data is transmitted, DHT11 pulls down the voltage level and keeps it for 50us. Then the Single-Bus voltage will be pulled up by the resistor to set it back to the free status.

6. Electrical Characteristics

VDD=5V, T = 25℃ (unless otherwise stated)

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

Note: Sampling period at intervals should be no less than 1 second.

7. Attentions of application

(1) Operating conditions

Applying the DHT11 sensor beyond its working range stated in this datasheet can result in 3%RH signal shift/discrepancy. The DHT11 sensor can recover to the calibrated status gradually when it gets back to the normal operating condition and works within its range. Please refer to (3) of this section to accelerate its recovery. Please be aware that operating the DHT11 sensor in the non-normal working conditions will accelerate sensor's aging process.

(2) Attention to chemical materials

Vapor from chemical materials may interfere with DHT's sensitive-elements and debase its sensitivity. A high degree of chemical contamination can permanently damage the sensor.

(3) Restoration process when (1) & (2) happen

Step one: Keep the DHT sensor at the condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;
Step two:K keep the DHT sensor at the condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Temperature Affect

Relative humidity largely depends on temperature. Although temperature compensation technology is used to ensure accurate measurement of RH, it is still strongly advised to keep the humidity and temperature sensors working under the same temperature. DHT11 should be mounted at the place as far as possible from parts that may generate heat.

(5) Light Affect

Long time exposure to strong sunlight and ultraviolet may debase DHT's performance.

(6) Connection wires

The quality of connection wires will affect the quality and distance of communication and high quality shielding-wire is recommended.

(7) Other attentions

* Welding temperature should be bellow 260Celsius and contact should take less than 10 seconds.
* Avoid using the sensor under dew condition.
* Do not use this product in safety or emergency stop devices or any other occasion that failure of DHT11 may cause personal injury.
* Storage: Keep the sensor at temperature 10-40℃, humidity <60%RH.

Declaim:

This datasheet is a translated version of the manufacturer's datasheet. Although the due care has been taken during the translation, D-Robotics is not responsible for the accuracy of the information contained in this document. Copyright © D-Robotics.

D-Robotics: www.droboticsonline.com

Email contact: d_robotics@hotmail.co.uk

DATA SHEET

NEC

NPN SILICON TRANSISTOR
2SC945

NPN SILICON TRANSISTOR

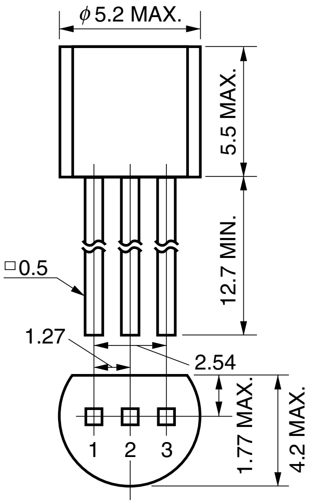
DESCRIPTION

The 2SC945 is designed for use in driver stage of AF amplifier and low speed switching.

FEATURES

- High voltage
  LV_{CEO} = 50 V MIN.
- Excellent h_{FE} linearity
  h_{FE1} = (0.1 mA)/h_{FE2} (1.0 mA) = 0.92 TYP.

★ PACKAGE DRAWING (Unit: mm)



1. Emitter	EIAJ:	SC43B
2. Collector	JEDEC:	TO92
3. Base	IEC:	PA33

ABSOLUTE MAXIMUM RATINGS

Maximum Temperature		
Storage Temperature		–55 to +150°C
Junction Temperature		+150°C Maximum
Maximum Power Dissipation (T_A = 25°C)		
Total Power Dissipation		250 mW
Maximum Voltages and Currents (T_A = 25°C)		
V_{CBO}	Collector to Base Voltage	60 V
V_{CEO}	Collector to Emitter Voltage	50 V
V_{EBO}	Emitter to Base Voltage	5.0 V
I_C	Collector Current	100 mA
I_B	Base Current	20 mA

ELECTRICAL CHARACTERISTICS (T_A = 25°C)

CHARACTERISTIC	SYMBOL	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
DC Current Gain	h_{FE1}	V_{CE} = 6.0 V, I_C = 0.1 mA	50	185		
DC Current Gain	h_{FE2}	V_{CE} = 6.0 V, I_C = 1.0 mA	90	200	600	
Gain Bandwidth Product	f_T	V_{CE} = 6.0 V, I_E = –10 mA		250		MHz
Collector to Base Capacitance	C_{ob}	V_{CB} = 6.0 V, I_E = 0, f = 1.0 MHz		3.0		pF
Collector Cutoff Current	I_{CBO}	V_{CB} = 60 V, I_E = 0 A			100	nA
Emitter Cutoff Current	I_{EBO}	V_{EB} = 5.0 V, I_C = 0 A			100	nA
Base to Emitter Voltage	V_{BE}	V_{CE} = 6.0 V, I_C = 1.0 mA	0.55	0.62	0.65	V
Collector Saturation Voltage	$V_{CE(sat)}$	I_C = 100 mA, I_B = 10 mA		0.15	0.3	V
Base Saturation Voltage	$V_{BE(sat)}$	I_C = 100 mA, I_B = 10 mA		0.86	1.0	V

CLASSIFICATION OF h_{FE2}

Rank	R	Q	P	K
Range	90 to 180	135 to 270	200 to 400	300 to 600

Remark h_{FE2} Test Conditions: V_{CE} = 6.0 V, I_C = 1.0 mA

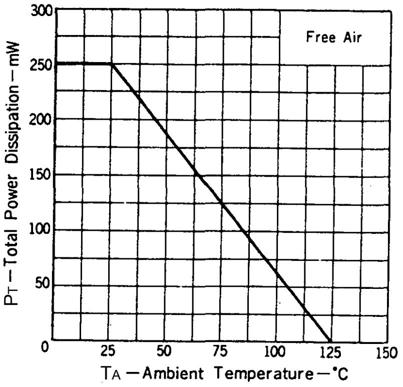
The information in this document is subject to change without notice. Before using this document, please confirm that this is the latest version.
Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.

Document No. D17117EJ2V0DS00 (2nd edition)
(Previous No. TC-3005C)
Date Published March 2004 N CP(K)
Printed in Japan

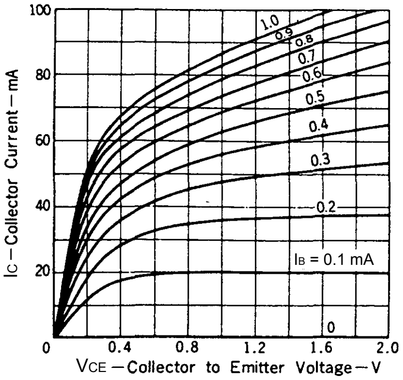
The mark ★ shows major revised points.

TYPICAL CHARACTERISTICS (T_A = 25°C, unless otherwise noted.)

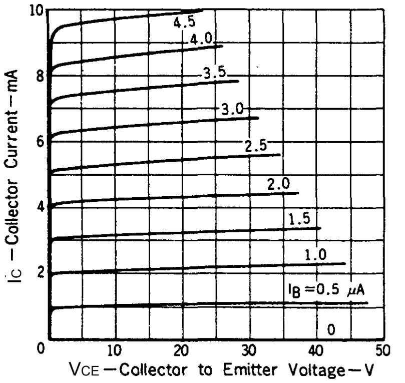
TOTAL POWER DISSIPATION vs. AMBIENT TEMPERATURE



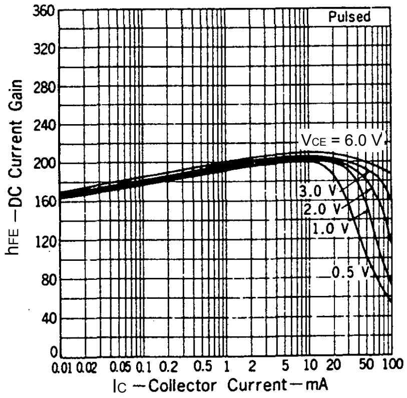
COLLECTOR CURRENT vs. COLLECTOR TO EMITTER VOLTAGE



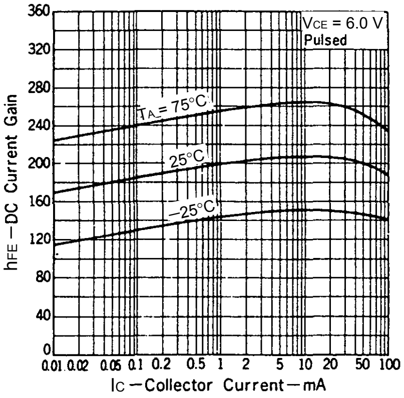
COLLECTOR CURRENT vs. COLLECTOR TO EMITTER VOLTAGE



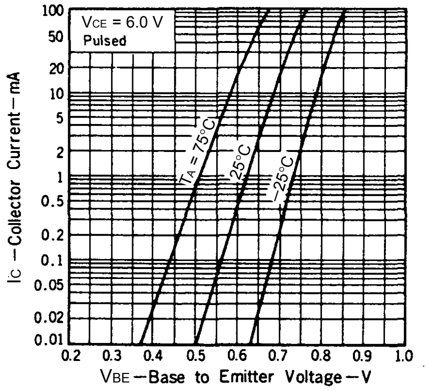
DC CURRENT GAIN vs. COLLECTOR CURRENT



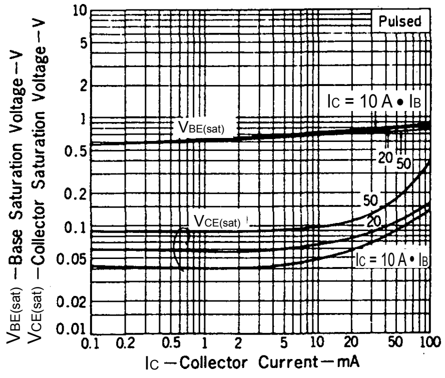
DC CURRENT GAIN vs. COLLECTOR CURRENT



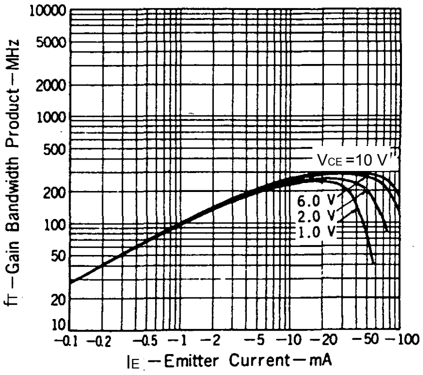
COLLECTOR CURRENT vs. BASE TO EMITTER VOLTAGE



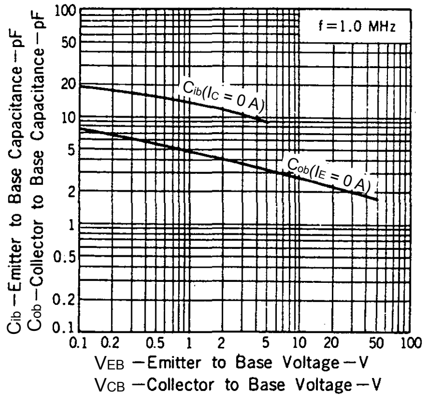
COLLECTOR AND BASE SATURATION VOLTAGE vs. COLLECTOR CURRENT



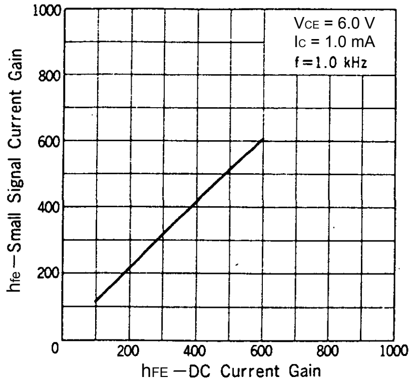
GAIN BANDWIDTH PRODUCT vs. EMITTER CURRENT



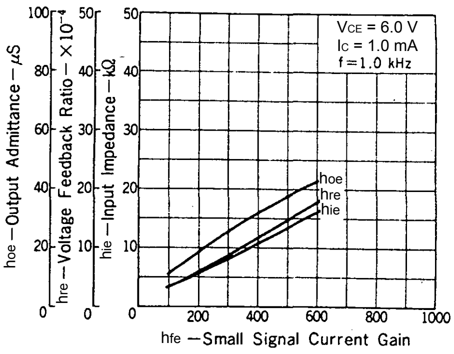
EMITTER TO BASE AND COLLECTOR TO BASE CAPACITANCE vs. REVERSE VOLTAGE



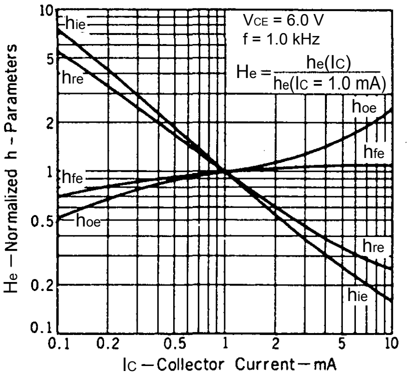
SMALL SIGNAL CURRENT GAIN vs. DC CURRENT GAIN



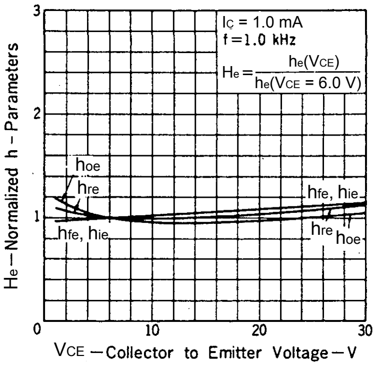
INPUT IMPEDANCE, VOLTAGE FEEDBACK RATIO AND OUTPUT ADMITTANCE vs. SMALL SIGNAL CURRENT GAIN



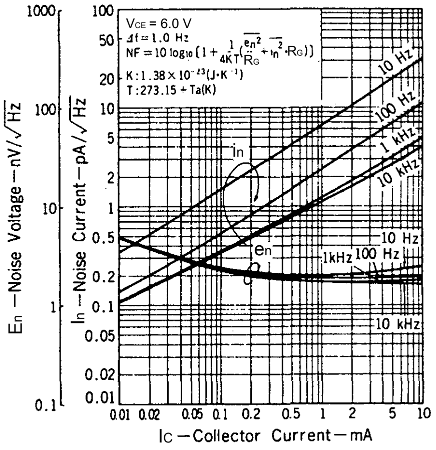
NORMALIZED h-PARAMETERS vs. COLLECTOR CURRENT



NORMALIZED h-PARAMETERS vs. COLLECTOR TO EMITTER VOLTAGE



E_n AND I_n vs. COLLECTOR CURRENT



- **The information in this document is current as of March, 2004. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

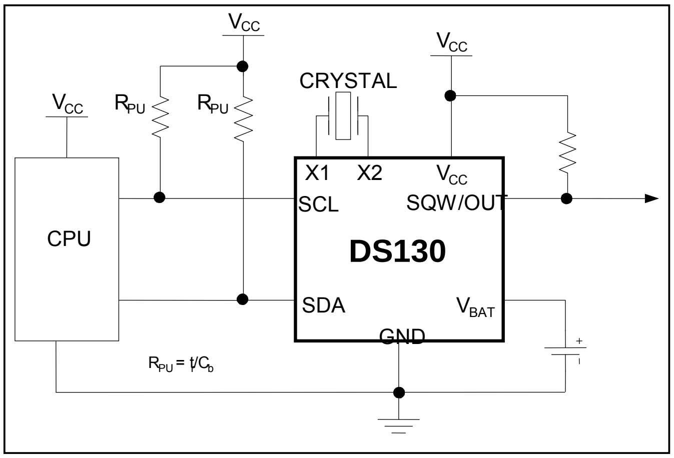
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

DS1307
64 x 8, Serial, I^{2}C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I^{2}C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

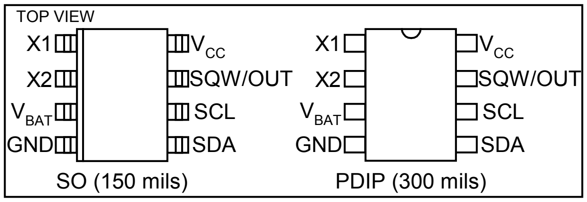
TYPICAL OPERATING CIRCUIT



FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I^{2}C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+Denotes a lead-free/RoHS-compliant package.
**A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.*

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial	0°C to +70°C
Industrial	-40°C to +85°C
Storage Temperature Range	-55°C to +125°C
Soldering Temperature (DIP, leads)	+260°C for 10 seconds
Soldering Temperature (surface mount)	Refer to the JPC/JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS

(T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Logic 1 Input	V_{IH}		2.2		V_{CC} + 0.3	V
Logic 0 Input	V_{IL}		-0.3		+0.8	V
V_{BAT} Battery Voltage	V_{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I_{LI}		-1		1	µA
I/O Leakage (SDA, SQW/OUT)	I_{LO}		-1		1	µA
Logic 0 Output (I_{OL} = 5mA)	V_{OL}				0.4	V
Active Supply Current (f_{SCL} = 100kHz)	I_{CCA}				1.5	mA
Standby Current	I_{CCS}	(Note 3)			200	µA
V_{BAT} Leakage Current	I_{BATLKG}			5	50	nA
Power-Fail Voltage (V_{BAT} = 3.0V)	V_{PF}		1.216 x V_{BAT}	1.25 x V_{BAT}	1.284 x V_{BAT}	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{BAT} Current (OSC ON); SQW/OUT OFF	I_{BAT1}			300	500	nA
V_{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I_{BAT2}			480	800	nA
V_{BAT} Data-Retention Current (Oscillator Off)	I_{BATDR}			10	100	nA

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}		0		100	kHz
Bus Free Time Between a STOP and START Condition	t_{BUF}		4.7			µs
Hold Time (Repeated) START Condition	$t_{HD:STA}$	(Note 4)	4.0			µs
LOW Period of SCL Clock	t_{LOW}		4.7			µs
HIGH Period of SCL Clock	t_{HIGH}		4.0			µs
Setup Time for a Repeated START Condition	$t_{SU:STA}$		4.7			µs
Data Hold Time	$t_{HD:DAT}$		0			µs
Data Setup Time	$t_{SU:DAT}$	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t_R				1000	ns
Fall Time of Both SDA and SCL Signals	t_F				300	ns
Setup Time for STOP Condition	$t_{SU:STO}$		4.7			µs

CAPACITANCE

(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	$C_{I/O}$				10	pF
Capacitance Load for Each Bus Line	C_B	(Note 7)			400	pF

Note 1: All voltages are referenced to ground.

Note 2: Limits at -40°C are guaranteed by design and are not production tested.

Note 3: I_{CCS} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.

Note 4: After this period, the first clock pulse is generated.

Note 5: A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the $V_{IH(MIN)}$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.

Note 6: The maximum $t_{HD:DAT}$ only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.

Note 7: C_B—total capacitance of one bus line in pF.

TIMING DIAGRAM

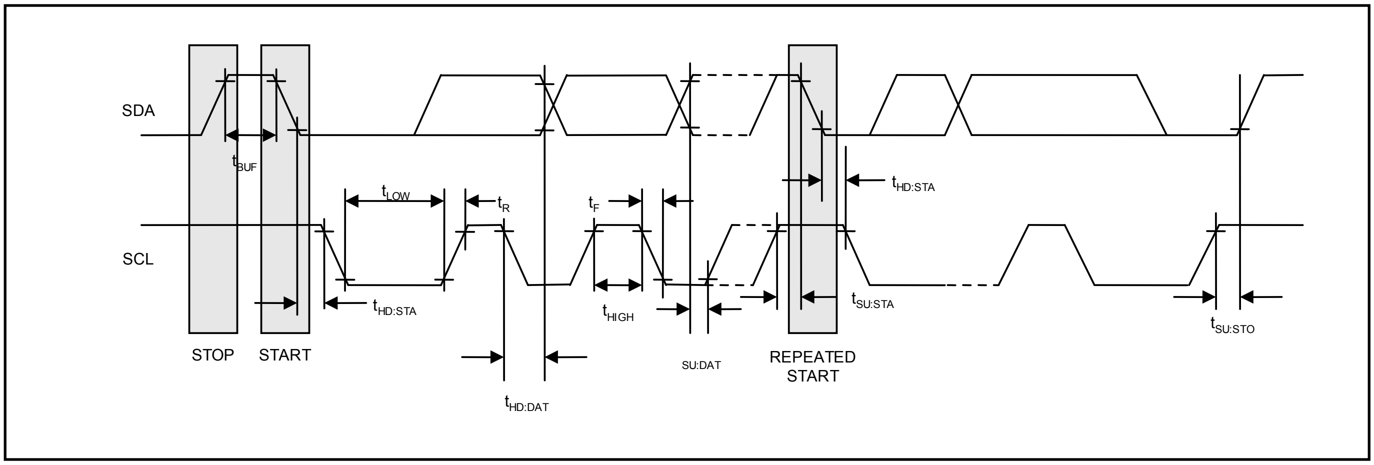
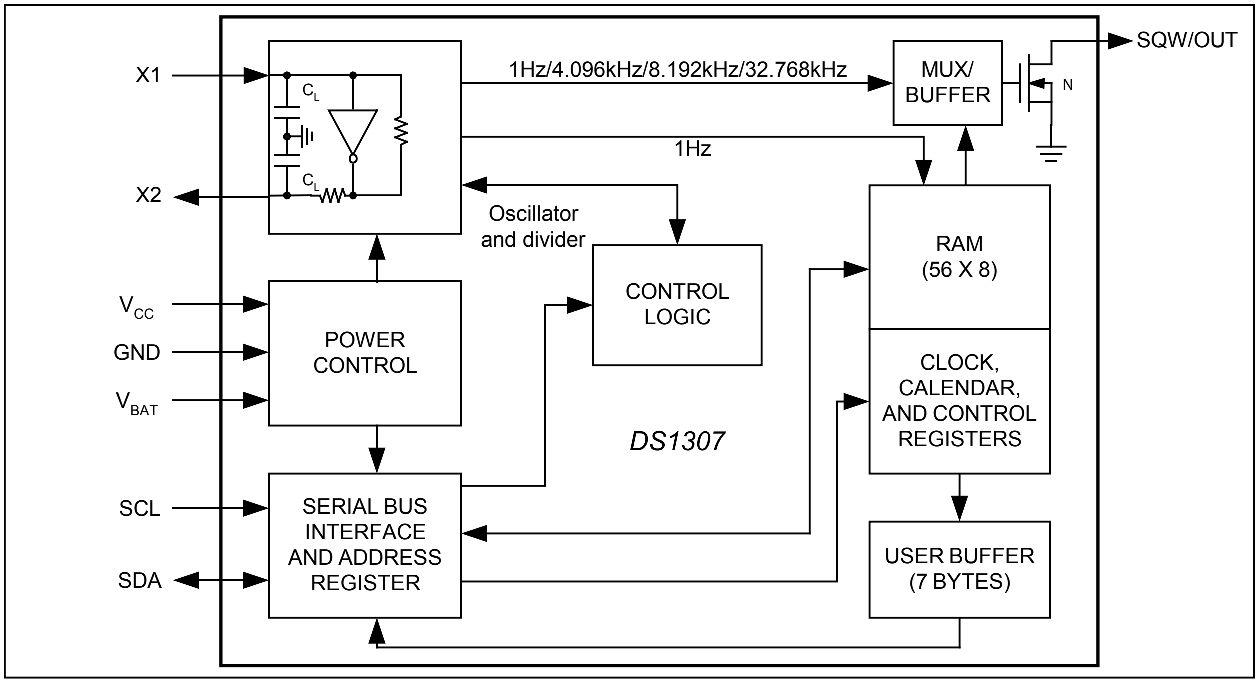
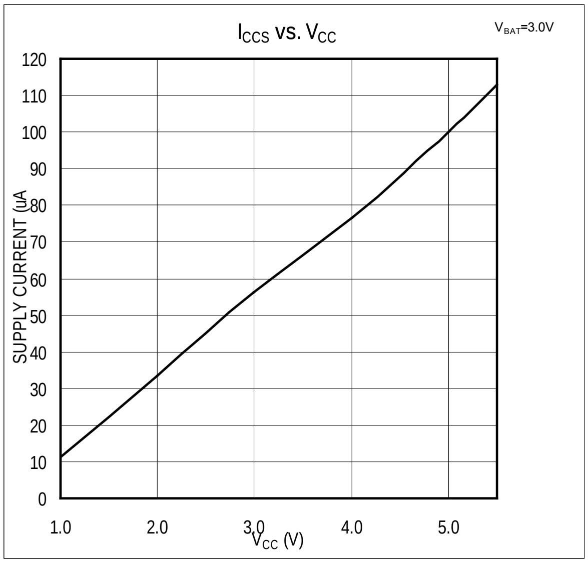


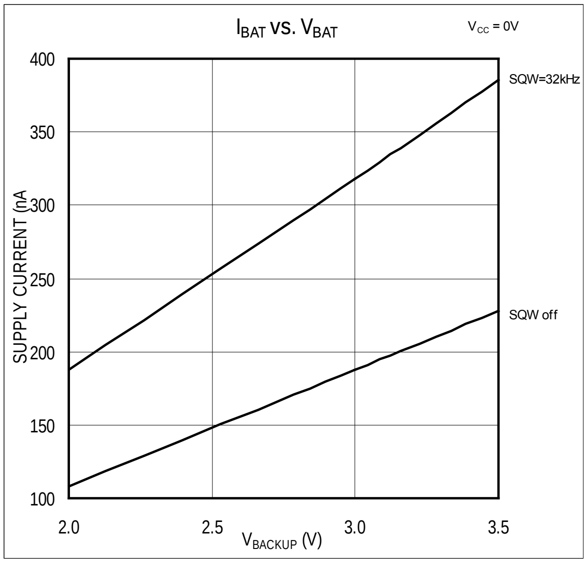
Figure 1. Block Diagram

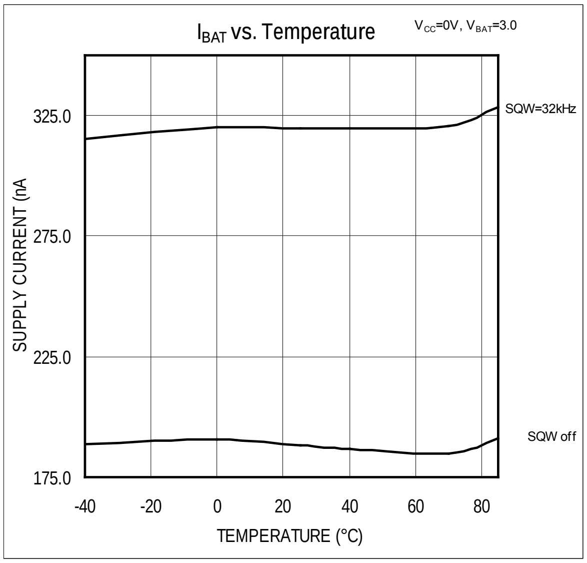


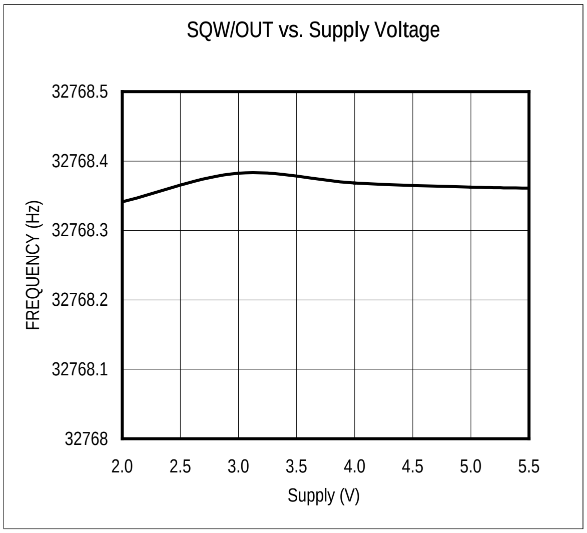
TYPICAL OPERATING CHARACTERISTICS

(V_{CC} = 5.0V, T_A = +25°C, unless otherwise noted.)









PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1. **Note:** For more information on crystal selection and crystal layout considerations, refer to *Application Note 58: Crystal Considerations with Dallas Real-Time Clocks*.
2	X2	
3	V_{BAT}	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V_{BAT} pin may prevent proper operation. If a backup supply is not required, V_{BAT} must be grounded. The nominal power-fail trip point (V_{PF}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/qa/info/ul/.
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I^{2}C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}.
6	SCL	Serial Clock Input. SCL is the clock input for the I^{2}C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}.
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V_{CC} or V_{BAT} applied. The pullup voltage can be up to 5.5V regardless of the voltage on V_{CC}. If not used, this pin can be left floating.
8	V_{CC}	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V_{CC} is below V_{TP}, read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I^{2}C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} +0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the serial RTC.

OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1 shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

CLOCK ACCURACY

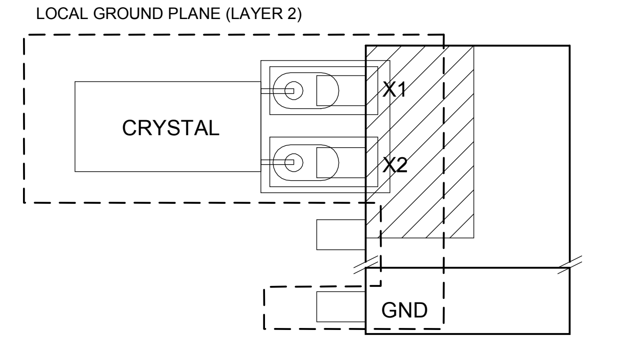
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to Application Note 58: *Crystal Considerations with Dallas Real-Time Clocks* for detailed information.

Table 1. Crystal Specifications*

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	f_O		32.768		kHz
Series Resistance	ESR			45	kΩ
Load Capacitance	C_L		12.5		pF

*_The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to_ Application Note 58: Crystal Considerations for Dallas Real-Time Clocks _for additional specifications._

Figure 2. Recommended Layout for Crystal



NOTE: AVOID ROUTING SIGNAL LINES IN THE CROSSHATCHED AREA (UPPER LEFT QUADRANT) OF THE PACKAGE UNLESS THERE IS A GROUND PLANE BETWEEN THE SIGNAL LINE AND THE DEVICE PACKAGE.

RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on.) Illogical time and date entries result in undefined operation. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled. On first application of power to the device the time and date registers are typically reset to 01/01/00 01 00:00:00 (MM/DD/YY DOW HH:MM:SS). The CH bit in the seconds register will be set to a 1. The clock can be halted whenever the timekeeping functions are not required, which minimizes current (I_{BATDR}).

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12-hour or 24-hour mode-select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20 to 23 hours). The hours value must be re-entered whenever the 12/24-hour mode bit is changed.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any I^{2}C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I^{2}C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

Table 2. Timekeeper Registers

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

0 = Always reads back as 0.

CONTROL REGISTER

The DS1307 control register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Bit 7: Output Control (OUT). This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0. On initial application of power to the device, this bit is typically set to a 0.

Bit 4: Square-Wave Enable (SQWE). This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave. On initial application of power to the device, this bit is typically set to a 0.

Bits 1 and 0: Rate Select (RS[1:0]). These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits. On initial application of power to the device, these bits are typically set to a 1.

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

I^{2}C DATA BUS

The DS1307 supports the I^{2}C protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the I^{2}C bus.

Figures 3, 4, and 5 detail how data is transferred on the I^{2}C bus.

- Data transfer can be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

START data transfer: A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

STOP data transfer: A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

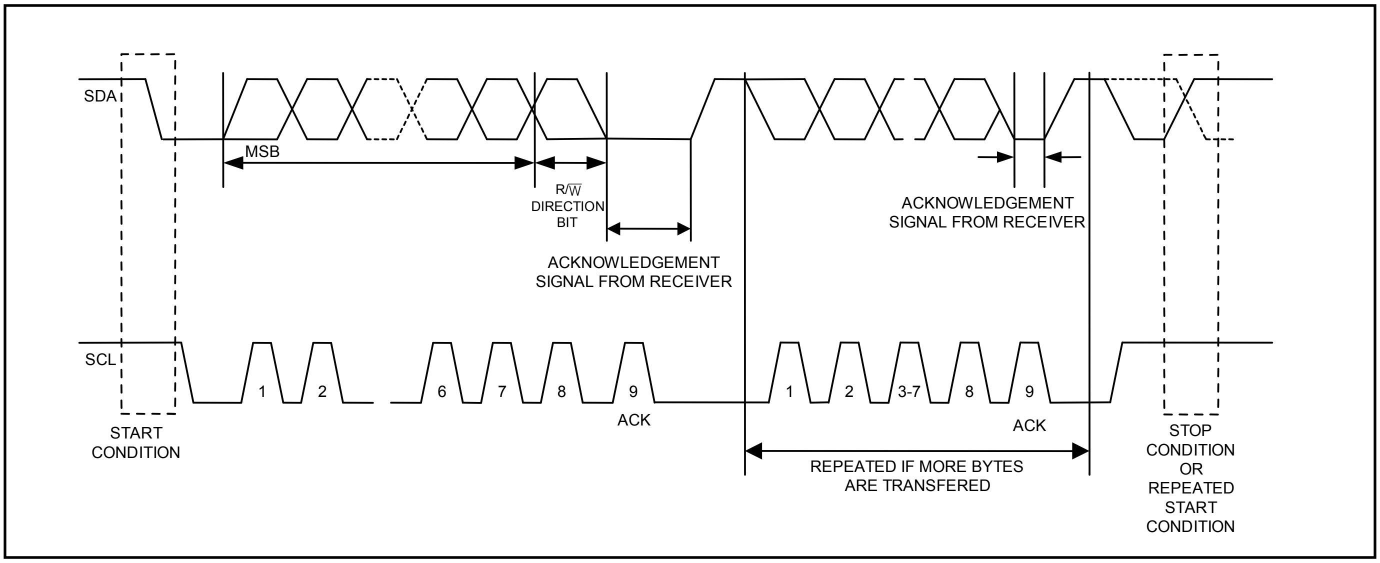
Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the I^{2}C bus specifications a standard mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the standard mode (100kHz) only.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Figure 3. Data Transfer on I^{2}C Serial Bus



Depending upon the state of the R/W bit, two types of data transfer are possible:

1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.

2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a “not acknowledge” is returned.

   The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307 can operate in the following two modes:

1. **Slave Receiver Mode (Write Mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Hardware performs address recognition after reception of the slave address and direction bit (see Figure 4). The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit ($R/\overline{W}$), which for a write is 0. After receiving and decoding the slave address byte, the DS1307 outputs an acknowledge on SDA. After the DS1307 acknowledges the slave address + write bit, the master transmits a word address to the DS1307. This sets the register pointer on the DS1307, with the DS1307 acknowledging the transfer. The master can then transmit zero or more bytes of data with the DS1307 acknowledging each byte received. The register pointer automatically increments after each data byte are written. The master will generate a STOP condition to terminate the data write.

2. **Slave Transmitter Mode (Read Mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. The DS1307 transmits serial data on SDA while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (see Figure 5). The slave address byte is the first byte received after the START condition is generated by the master. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit ($R/\overline{W}$), which is 1 for a read. After receiving and decoding the slave address the DS1307 outputs an acknowledge on SDA. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The register pointer automatically increments after each byte are read. The DS1307 must receive a Not Acknowledge to end a read.

Figure 4. Data Write—Slave Receiver Mode

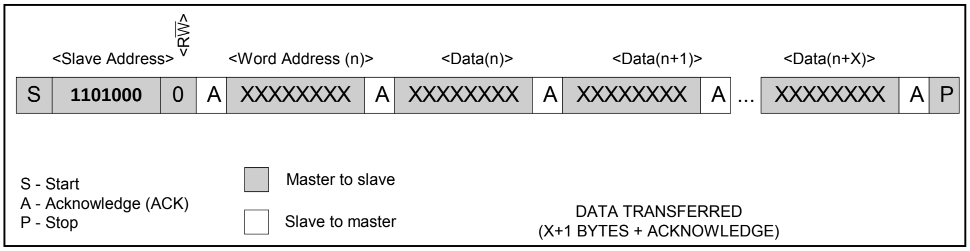


Figure 5. Data Read—Slave Transmitter Mode

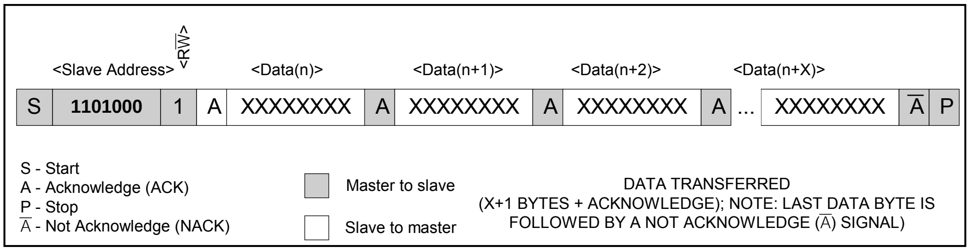
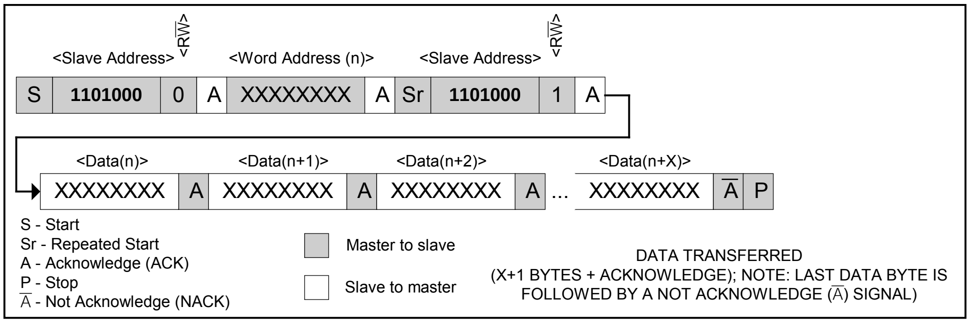


Figure 6. Data Read (Write Pointer, Then Read)—Slave Receive and Transmit



PACKAGE INFORMATION

For the latest package outline information and land patterns, go to www.maxim-ic.com/packages.

PACKAGE TYPE	PACKAGE CODE	DOCUMENT NO.
8 PDIP	—	21-0043
8 SO	—	21-0041

REVISION HISTORY

REVISION DATE	DESCRIPTION	PAGES CHANGED
100208	Moved the *Typical Operating Circuit* and *Pin Configurations* to first page.	1
	Removed the leaded part numbers from the *Ordering Information* table.	1
	Added an open-drain transistor to SQW/OUT in the block diagram (Figure 1).	4
	Added the pullup voltage range for SDA, SCL, and SQW/OUT to the *Pin Description* table and noted that SQW/OUT can be left open if not used.	6
	Added default time and date values on first application of power to the *Clock and Calendar* section and deleted the note that initial power-on state is not defined.	8
	Added default on initial application of power to bit info in the *Control Register* section.	9
	Updated the *Package Information* section to reflect new package outline drawing numbers.	13

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.