

DAFTAR PUSTAKA

- [1]. Waltenegus Dargie and Christian Poellabauer. "Fundamental of Wireless Sensor Networks", Wiley & Sons, 2010
- [2] Nazeran H. "Wavelet-based segmentation and feature extraction of heart sounds for Intelligent PDA-based Phonocardiography", Methods of Information in Medicine Journal, Special Issue on Biomedical Signal Interpretation, Vol 46, No 2, pp 135-141, 2007
- [3]. Sohraby, K., Minoli, D., and Znati, T. "Wireless sensor networks: technology, protocols, and applications, John Wiley and Sons", 2007
- [4]. Arduino Uno, <http://www.arduino.cc/> diunduh [10 Desember 2012]
- [5]. Pulse sensor, <http://www.pulsesensor.com/> diunduh [10 Januari 2013]
- [6]. "Temperatur, Body. "World Book Encyclopedia. Chicago : Field Enterprises, 1996.
- [7]. <http://www.id.scribd.com/doc/57181049/Suhu-Tubuh-Manusia>
diunduh [12 Agustus 2013]
- [8]. <http://www.e-belajarelektronika.com/?s=sensor+suhu+tubuh+manusia>
diunduh [12 Agustus 2013]
- [9]. E. Palantei, D. Utami, A.E.A. Febriani, M. Agus, U. Umar, M. Baharuddin and Andani A. , "A 2.5 GHz Wireless ECG System for Remotely Monitoring Heart Pulses" accepted for presentation at IEEE Antenna and Propagation Society (APS) Symposium/URSI/USNC Meeting 2012, July 8-14, Chicago, IL, USA.
- [10] Zhang, dkk., "Bluetooth-based sensor network for remotely

monitoring the physiological signals of a patient”, WPAN Transactions on Information Technology in Biomedicine, vol. 13, no. 6, pp. 10401048, 2009.

[11]. Alat - alat sensor kesehatan dan info alat - alat kesehatan

http://www.wsntech.net/KR/index.php?option=com_content&task=view&id=168&Itemid=101 diunduh [10 Desember 2012]

[12]. W. B. Saunders, “Oxygen Saturation Monitoring by Pulse Oxymetry”

In : AACN Procedure Manual For Critical Care, Fourth Edition, 2001

[13]. Jan Axelson. “Serial Port Complete Second Edition”, Lakeview Research LLC Madison, 2007.

[14]. Rusmin Nuryadin dan Iqra aswad, Tugas Akhir “Implementasi dan Evaluasi Kinerja Aplikasi Jaringan Sensor Nirkabel”, Fakultas Teknik Universitas Hasanuddin, Makassar, 2012.

[15]. Puput Dani Prasetyo Adi, Tesis “Analisis Kinerja Jaringan Sensor Nirkabel IEEE 802.15.4 Dengan Data Denyut Nadi Pasien”, Fakultas Teknik Universitas Hasanuddin, Makassar, 2013.

LAMPIRAN 1

Source Code Pulse dan Suhu Untuk Arduino

```
/*
>> Pulse Sensor Amped 1.1 <<
This code is for Pulse Sensor Amped by Joel Murphy and Yury Gitman
www.pulsesensor.com
>>> Pulse Sensor purple wire goes to Analog Pin 0 <<<
Pulse Sensor sample aquisition and processing happens in the background via Timer 2
interrupt. 2mS sample rate.

PWM on pins 3 and 11 will not work when using this code, because we are using Timer
2!

The following variables are automatically updated:
Signal : int that holds the analog signal data straight from the sensor. updated every
2mS.

IBI : int that holds the time interval between beats. 2mS resolution.

BPM : int that holds the heart rate value, derived every beat, from averaging
previous 10 IBI values.

QS : boolean that is made true whenever Pulse is found and BPM is updated. User
must reset.

Pulse : boolean that is true when a heartbeat is sensed then false in time with pin13
LED going out.

This code is designed with output serial data to Processing sketch
"PulseSensorAmped_Processing-xx"

The Processing sketch is a simple data visualizer.

All the work to find the heartbeat and determine the heartrate happens in the code
below.

Pin 13 LED will blink with heartbeat.

If you want to use pin 13 for something else, adjust the interrupt handler

It will also fade an LED on pin fadePin with every beat. Put an LED and series resistor
from fadePin to GND.

Check here for detailed code walkthrough:
http://pulsesensor.myshopify.com/pages/pulse-sensor-amped-arduino-v1dot1

Code Version 02 by Joel Murphy & Yury Gitman Fall 2012
```

This update changes the HRV variable name to IBI, which stands for Inter-Beat Interval, for clarity.

Switched the interrupt to Timer2. 500Hz sample rate, 2mS resolution IBI value.

Fade LED pin moved to pin 5 (use of Timer2 disables PWM on pins 3 & 11).

Tidied up inefficiencies since the last version.

*/

```
#include <SoftwareSerial.h> //Software Serial Port
#include <LiquidCrystal.h>
#define RxD 6
#define TxD 7
#define DEBUG_ENABLED 1
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
SoftwareSerial blueToothSerial(RxD,TxD);
// VARIABLES
float tempC;
int count=0;
int total=0;
int inf;
//int suhu=0;
int tempPin = 2;
int pulsePin = 0;      // Pulse Sensor purple wire connected to analog pin 0
int blinkPin = 13;     // pin to blink led at each beat
int fadePin = 5;       // pin to do fancy classy fading blink at each beat
int fadeRate = 0;      // used to fade LED on with PWM on fadePin
// these variables are volatile because they are used during the interrupt service
routine!
volatile int BPM;      // used to hold the pulse rate
volatile int Signal;    // holds the incoming raw data
volatile int IBI = 600;  // holds the time between beats, the Inter-Beat Interval
volatile boolean Pulse = false; // true when pulse wave is high, false when it's low
volatile boolean QS = false; // becomes true when Arduino finds a beat.
void setup(){
```

```

pinMode(blinkPin,OUTPUT);      // pin that will blink to your heartbeat!
pinMode(fadePin,OUTPUT);      // pin that will fade to your heartbeat!
Serial.begin(57600);
pinMode(RxD, INPUT);
pinMode(TxD, OUTPUT);
setupBlueToothConnection();
// we agree to talk fast!
interruptSetup();           // sets up to read Pulse Sensor signal every 2mS
// UN-COMMENT THE NEXT LINE IF YOU ARE POWERING The Pulse Sensor AT LOW
VOLTAGE,
// AND APPLY THAT VOLTAGE TO THE A-REF PIN
//analogReference(EXTERNAL);
}

void loop(){
char recvChar;
while(1){
if(blueToothSerial.available()){//check if there's any data sent from the remote
bluetooth shield
recvChar = blueToothSerial.read();
Serial.print(recvChar);
}
if(Serial.available()){//check if there's any data sent from the local serial terminal, you
can add the other applications here
recvChar = Serial.read();
blueToothSerial.print(recvChar);
}
//sendDataToProcessing('S', Signal); // send Processing the raw Pulse Sensor data
if (QS == true){                // Quantified Self flag is true when arduino finds a
heartbeat
fadeRate = 255;                // Set 'fadeRate' Variable to 255 to fade LED with pulse
sendDataToProcessing('B',BPM);  // send heart rate with a 'B' prefix
// sendDataToProcessing('Q',IBI); // send time between beats with a 'Q' prefix
}
}

```

```

lcd.begin(4, 2);
lcd.print("BPM = ");
lcd.print(BPM);
lcd.setCursor(0, 1);
lcd.print("Suhu = ");
lcd.print((byte)tempC);
lcd.print(" ");
lcd.print(millis()/1000);
for(inf=15;inf<=90;inf+=25){
  if(millis()/1000==inf) //awalnya 20
  {
    //total=total/3;
    //blueToothSerial.print("Nilai BPM Anda: "); //view "Suhu Tubuh Anda"
    //blueToothSerial.println(total);
    //total=0;
    delay(10000);
  }
  else if(millis()/1000>90)
  {
    delay(100000);
  }
}
//count=count+1;
//total=total+BPM;
//total+=BPM;
QS = false;           // reset the Quantified Self flag for next time
}
ledFadeToBeat();          // wait one second before sending new data
delay(20);             // take a break
}

```

```

//}

void setupBlueToothConnection()
{
    blueToothSerial.begin(38400); //Set BluetoothBee BaudRate to default baud rate
38400

    blueToothSerial.print("\r\n+STWMOD=0\r\n"); //set the bluetooth work in slave mode
    blueToothSerial.print("\r\n+STNA=BTSlave\r\n"); //set the bluetooth name as
    "SeeedBTSlave"

    blueToothSerial.print("\r\n+STOAUT=1\r\n"); // Permit Paired device to connect me
    blueToothSerial.print("\r\n+STAUTO=0\r\n"); // Auto-connection should be forbidden
here

    delay(2000); // This delay is required.

    blueToothSerial.print("\r\n+INQ=1\r\n"); //make the slave bluetooth inquirable
    Serial.println("The slave bluetooth is inquirable!");
    delay(2000); // This delay is required.

    blueToothSerial.flush();
}

void ledFadeToBeat(){

    fadeRate -= 15;           // set LED fade value
    fadeRate = constrain(fadeRate,0,255); // keep LED fade value from going into
negative numbers!

    analogWrite(fadePin,fadeRate); // fade LED
}

void sendDataToProcessing(char symbol, int data ){

    Serial.print(symbol); // symbol prefix tells Processing what type of data is
coming

    Serial.println(data); // the data to send culminating in a carriage return
    blueToothSerial.print("Denyut Nadi Anda : "); //view "Suhu Tubuh Anda"
    blueToothSerial.print(" "); //view "Suhu Tubuh Anda"
    blueToothSerial.print(symbol);
    blueToothSerial.println(data);
    // blueToothSerial.print("Suhu Tubuh Anda: "); //view "Suhu Tubuh Anda"
}

```

```
tempC = analogRead(tempPin);      //read the value from the sensor
tempC = (5.0 * tempC * 100.0)/1024.0; //convert the analog data to temperature
blueToothSerial.print("Suhu Tubuh Anda: "); //view "Suhu Tubuh Anda"
blueToothSerial.print((byte)tempC);      //send the data to the computer
blueToothSerial.print(" C ");          //view "C"
blueToothSerial.print(" & ");
delay(20);
;
//delay(2000);                      //wait one second before sending new data
}
```

INTERRUPT.INO

```
volatile int rate[10];           // used to hold last ten IBI values
volatile unsigned long sampleCounter = 0;    // used to determine pulse timing
volatile unsigned long lastBeatTime = 0;    // used to find the inter beat interval
volatile int P =512;           // used to find peak in pulse wave
volatile int T = 512;          // used to find trough in pulse wave
volatile int thresh = 512;     // used to find instant moment of heart beat
volatile int amp = 100;        // used to hold amplitude of pulse waveform
volatile boolean firstBeat = true; // used to seed rate array so we startup with
reasonable BPM
volatile boolean secondBeat = true; // used to seed rate array so we startup with
reasonable BPM
void interruptSetup(){
    // Initializes Timer2 to throw an interrupt every 2mS.
    TCCR2A = 0x02; // DISABLE PWM ON DIGITAL PINS 3 AND 11, AND GO INTO CTC
    MODE
    TCCR2B = 0x06; // DON'T FORCE COMPARE, 256 PRESCALER
    OCR2A = 0X7C; // SET THE TOP OF THE COUNT TO 124 FOR 500Hz SAMPLE RATE
    TIMSK2 = 0x02; // ENABLE INTERRUPT ON MATCH BETWEEN TIMER2 AND OCR2A
    sei(); // MAKE SURE GLOBAL INTERRUPTS ARE ENABLED
}
// THIS IS THE TIMER 2 INTERRUPT SERVICE ROUTINE.
// Timer 2 makes sure that we take a reading every 2 miliseconds
ISR(TIMER2_COMPA_vect){ // triggered when Timer2 counts to 124
    cli(); // disable interrupts while we do this
    Signal = analogRead(pulsePin); // read the Pulse Sensor
    sampleCounter += 2; // keep track of the time in mS with this variable
    int N = sampleCounter - lastBeatTime; // monitor the time since the last beat to
    avoid noise
    // find the peak and trough of the pulse wave
    if(Signal < thresh && N > (IBI/5)*3){ // avoid dichrotic noise by waiting 3/5 of last
    IBI
```

```

if (Signal < T){           // T is the trough
    T = Signal;           // keep track of lowest point in pulse wave
}
}

if(Signal > thresh && Signal > P){      // thresh condition helps avoid noise
    P = Signal;           // P is the peak
}
// keep track of highest point in pulse wave

// NOW IT'S TIME TO LOOK FOR THE HEART BEAT
// signal surges up in value every time there is a pulse

if (N > 250){           // avoid high frequency noise
    if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3 ) {
        Pulse = true;       // set the Pulse flag when we think there is a pulse
        digitalWrite(blinkPin,HIGH); // turn on pin 13 LED
        IBI = sampleCounter - lastBeatTime; // measure time between beats in mS
        lastBeatTime = sampleCounter; // keep track of time for next pulse
        if(firstBeat){          // if it's the first time we found a beat, if firstBeat == TRUE
            firstBeat = false; // clear firstBeat flag
            return;             // IBI value is unreliable so discard it
        }
        if(secondBeat){         // if this is the second beat, if secondBeat == TRUE
            secondBeat = false; // clear secondBeat flag
            for(int i=0; i<=9; i++){ // seed the running total to get a realistic BPM at
                startup
                    rate[i] = IBI;
                }
            }
        // keep a running total of the last 10 IBI values
        word runningTotal = 0; // clear the runningTotal variable
        for(int i=0; i<=8; i++){ // shift data in the rate array
            rate[i] = rate[i+1]; // and drop the oldest IBI value
            runningTotal += rate[i]; // add up the 9 oldest IBI values
        }
    }
}

```

```

rate[9] = IBI;           // add the latest IBI to the rate array
runningTotal += rate[9]; // add the latest IBI to runningTotal
runningTotal /= 10;      // average the last 10 IBI values
BPM = 60000/runningTotal; // how many beats can fit into a minute? that's
BPM!

QS = true;              // set Quantified Self flag
// QS FLAG IS NOT CLEARED INSIDE THIS ISR
}

}

if (Signal < thresh && Pulse == true){ // when the values are going down, the beat is
over

digitalWrite(blinkPin,LOW); // turn off pin 13 LED
Pulse = false;           // reset the Pulse flag so we can do it again
amp = P - T;             // get amplitude of the pulse wave
thresh = amp/2 + T;      // set thresh at 50% of the amplitude
P = thresh;              // reset these for next time
T = thresh;
}

if (N > 2500){          // if 2.5 seconds go by without a beat
thresh = 512;            // set thresh default
P = 512;                // set P default
T = 512;                // set T default
lastBeatTime = sampleCounter; // bring the lastBeatTime up to date
firstBeat = true;         // set these to avoid noise
secondBeat = true;        // when we get the heartbeat back
}

sei();                  // enable interrupts when you're done!
}// end isr

```

LAMPIRAN 2

Source Code Pulse dan Suhu Tanpa Bluetooth

SOURCE CODE SUHU DAN PULSE TANPA PAKAI BLUETOOTH.INO

```
/*
>> Pulse Sensor Amped <<
*/
// VARIABLES
float tempC;
int tempPin = 0;
int pulsePin = 2;      // pulse sensor purple wire connected to analog pin 0
int fadeRate = 0;      // used to fade LED on PWM pin 11
//int analogPin = 0;    //temperature
//int readValue = 0;    //temperature
//float temperature = 0; //temperature
//float temperatureF = 0; //temperature
                           // these are volatile because they are used during the interrupt!
volatile int BPM;        // used to hold the pulse rate
volatile int Signal;     // holds the incoming raw data
volatile int HRV;        // holds the time between beats
volatile boolean Pulse = false; // true when pulse wave is high, false when it's low
volatile boolean QS = false; // becomes true when pulse rate is determined. every 20
                            pulses
void setup(){
  pinMode(13,OUTPUT); // pin 13 will blink to your heartbeat!
  pinMode(11,OUTPUT); // pin 11 will fade to your heartbeat!
  Serial.begin(115200); // we agree to talk fast!
  interruptSetup(); // sets up to read Pulse Sensor signal every 1mS
  // UN-COMMENT THE NEXT LINE IF YOU ARE POWERING THE PulseSensor AT LOW
  VOLTAGE,
  // AND APPLY THAT VOLTAGE TO THE A-REF PIN
  //analogReference(EXTERNAL);
}
void loop(){
  sendDataToProcessing('S', Signal); // send Processing the raw Pulse Sensor data
```

```

if (QS == true){           // Quantified Self flag is true when arduino finds a
heartbeat

    fadeRate = 255;         // Set 'fadeRate' Variable to 255 to fade LED with pulse
    sendDataToProcessing('B',BPM); // send the time between beats with a 'B' prefix
    sendDataToProcessing('Q',HRV); // send heart rate with a 'Q' prefix
    QS = false;            // reset the Quantified Self flag for next time
}

ledFadeToBeat();

//delay(2000);           // take a break

Serial.print("Detak Nadi Pasien : ");
Serial.print(Signal);
Serial.println(" (kali)");

//readValue = analogRead(analogPin); //Serial.println(readValue);
//temperature = (readValue * 0.0049); //temperatur
//temperature = temperature * 100; //temperatur
//Serial.print("Suhu Tubuh Anda: "); //temperatur
//Serial.print(temperature); //temperatur
//Serial.print(" C "); //temperatur
//delay(1000);

tempC = analogRead(tempPin); //read the value from the sensor
tempC = (5.0 * tempC * 100.0)/1024.0; //convert the analog data to temperature
Serial.print("Suhu Tubuh Pasien: "); //view "Suhu Tubuh Anda"
Serial.print((byte)tempC); //send the data to the computer
Serial.print(" C "); //view "C"
Serial.print("dan ");

delay(2000);           //wait one second before sending new data
}

void ledFadeToBeat(){

    fadeRate -= 15;        // set LED fade value
    fadeRate = constrain(fadeRate,0,255); // keep LED fade value from going into
negative numbers!

    analogWrite(11,fadeRate); // fade LED
}

```

```
}

void sendDataToProcessing(char symbol, int data ){
    //Serial.print(symbol);    // symbol prefix tells Processing what type of data is coming
    //Serial.println(data);    // the data to send
}
```

LAMPIRAN 3

Source Code Slave Bluetooth

```
/*
For more details about the product please check http://www.seeedstudio.com/depot/
*/
/* Upload this sketch into Seeeduino and press reset*/
#include <SoftwareSerial.h> //Software Serial Port

#define RxD 6
#define TxD 7
#define DEBUG_ENABLED 1

SoftwareSerial blueToothSerial(RxD,TxD);

void setup()
{
    Serial.begin(9600);
    pinMode(RxD, INPUT);
    pinMode(TxD, OUTPUT);
    setupBlueToothConnection();
}

void loop()
{
    char recvChar;
    while(1){
        if(blueToothSerial.available()){//check if there's any data sent from the remote
            bluetooth shield
        recvChar = blueToothSerial.read();
        Serial.print(recvChar);
    }
}
```

```
if(Serial.available()){//check if there's any data sent from the local serial terminal, you  
can add the other applications here
```

```
    recvChar = Serial.read();  
  
    blueToothSerial.print(recvChar);  
  
}  
  
}  
  
}  
  
void setupBlueToothConnection()  
{  
  
    blueToothSerial.begin(38400); //Set BluetoothBee BaudRate to default baud rate  
38400  
  
    blueToothSerial.print("\r\n+STWMOD=0\r\n"); //set the bluetooth work in slave mode  
  
    blueToothSerial.print("\r\n+STNA=SeeedBTSlave\r\n"); //set the bluetooth name as  
"SeeedBTSlave"  
  
    blueToothSerial.print("\r\n+STOAUT=1\r\n"); // Permit Paired device to connect me  
  
    blueToothSerial.print("\r\n+STAUTO=0\r\n"); // Auto-connection should be forbidden  
here  
  
    delay(2000); // This delay is required.  
  
    blueToothSerial.print("\r\n+INQ=1\r\n"); //make the slave bluetooth inquirable  
  
    Serial.println("The slave bluetooth is inquirable!");  
  
    delay(2000); // This delay is required.  
  
    blueToothSerial.flush();  
  
}
```

LAMPIRAN 4

Data Sheet LM35DZ

LM35

Precision Centigrade Temperature Sensors

General Description

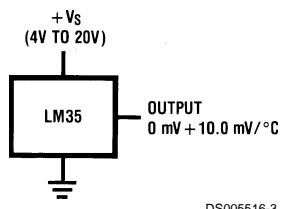
The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of $\pm 1/4^\circ\text{C}$ at room temperature and $\pm 3/4^\circ\text{C}$ over a full -55 to $+150^\circ\text{C}$ temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55 to $+150^\circ\text{C}$ temperature range, while the LM35C is rated for a -40 to $+110^\circ\text{C}$ range (-10° with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

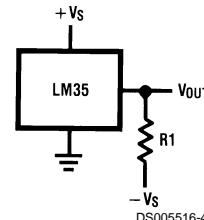
Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at $+25^\circ\text{C}$)
- Rated for full -55 to $+150^\circ\text{C}$ range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 μA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm 1/4^\circ\text{C}$ typical
- Low impedance output, 0.1Ω for 1 mA load

Typical Applications



**FIGURE 1. Basic Centigrade Temperature Sensor
($+2^\circ\text{C}$ to $+150^\circ\text{C}$)**

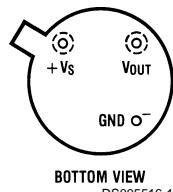


Choose $R_1 = -V_S/50 \mu\text{A}$
 $V_{OUT} = +1,500 \text{ mV at } +150^\circ\text{C}$
 $= +250 \text{ mV at } +25^\circ\text{C}$
 $= -550 \text{ mV at } -55^\circ\text{C}$

FIGURE 2. Full-Range Centigrade Temperature Sensor

Connection Diagrams

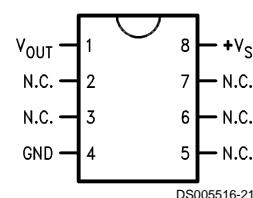
TO-46
Metal Can Package*



*Case is connected to negative pin (GND)

**Order Number LM35H, LM35AH, LM35CH, LM35CAH or
LM35DH**
See NS Package Number H03H

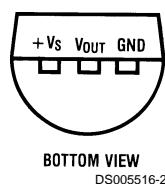
SO-8
Small Outline Molded Package



N.C. = No Connection

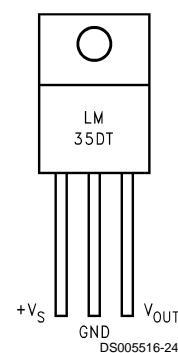
Top View
Order Number LM35DM
See NS Package Number M08A

TO-92
Plastic Package



**Order Number LM35CZ,
LM35CAZ or LM35DZ**
See NS Package Number Z03A

TO-220
Plastic Package*



*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|---|-----------------|
| Supply Voltage | +35V to -0.2V |
| Output Voltage | +6V to -1.0V |
| Output Current | 10 mA |
| Storage Temp.: | |
| TO-46 Package, | -60°C to +180°C |
| TO-92 Package, | -60°C to +150°C |
| SO-8 Package, | -65°C to +150°C |
| TO-220 Package, | -65°C to +150°C |
| Lead Temp.: | |
| TO-46 Package, (Soldering, 10 seconds) | 300°C |

| | |
|---|-----------------|
| TO-92 and TO-220 Package, (Soldering, 10 seconds) | 260°C |
| SO Package (Note 12) | |
| Vapor Phase (60 seconds) | 215°C |
| Infrared (15 seconds) | 220°C |
| ESD Susceptibility (Note 11) | 2500V |
| Specified Operating Temperature Range: T _{MIN} to T _{MAX} (Note 2) | |
| LM35, LM35A | -55°C to +150°C |
| LM35C, LM35CA | -40°C to +110°C |
| LM35D | 0°C to +100°C |

Electrical Characteristics

(Notes 1, 6)

| Parameter | Conditions | LM35A | | | LM35CA | | | Units (Max.) |
|--|--|---------|--------------------------|--------------------------|---------|--------------------------|--------------------------|-----------------|
| | | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | |
| Accuracy (Note 7) | T _A =+25°C | ±0.2 | ±0.5 | | ±0.2 | ±0.5 | | °C |
| | T _A =-10°C | ±0.3 | | | ±0.3 | | ±1.0 | °C |
| | T _A =T _{MAX} | ±0.4 | ±1.0 | | ±0.4 | ±1.0 | | °C |
| | T _A =T _{MIN} | ±0.4 | ±1.0 | | ±0.4 | | ±1.5 | °C |
| Nonlinearity (Note 8) | T _{MIN} ≤T _A ≤T _{MAX} | ±0.18 | | ±0.35 | ±0.15 | | ±0.3 | °C |
| Sensor Gain (Average Slope) | T _{MIN} ≤T _A ≤T _{MAX} | +10.0 | +9.9, +10.1 | | +10.0 | | +9.9, +10.1 | mV/°C |
| Load Regulation (Note 3) 0≤I _L ≤1 mA | T _A =+25°C | ±0.4 | ±1.0 | | ±0.4 | ±1.0 | | mV/mA |
| | T _{MIN} ≤T _A ≤T _{MAX} | ±0.5 | | ±3.0 | ±0.5 | | ±3.0 | mV/mA |
| Line Regulation (Note 3) | T _A =+25°C | ±0.01 | ±0.05 | | ±0.01 | ±0.05 | | mV/V |
| | 4V≤V _S ≤30V | ±0.02 | | ±0.1 | ±0.02 | | ±0.1 | mV/V |
| Quiescent Current (Note 9) | V _S =+5V, +25°C | 56 | 67 | | 56 | 67 | | µA |
| | V _S =+5V | 105 | | 131 | 91 | | 114 | µA |
| | V _S =+30V, +25°C | 56.2 | 68 | | 56.2 | 68 | | µA |
| | V _S =+30V | 105.5 | | 133 | 91.5 | | 116 | µA |
| Change of Quiescent Current (Note 3) | 4V≤V _S ≤30V, +25°C | 0.2 | 1.0 | | 0.2 | 1.0 | | µA |
| | 4V≤V _S ≤30V | 0.5 | | 2.0 | 0.5 | | 2.0 | µA |
| Temperature Coefficient of Quiescent Current | | +0.39 | | +0.5 | +0.39 | | +0.5 | µA/°C |
| Minimum Temperature for Rated Accuracy | In circuit of Figure 1, I _L =0 | +1.5 | | +2.0 | +1.5 | | +2.0 | °C |
| Long Term Stability | T _J =T _{MAX} , for 1000 hours | ±0.08 | | | ±0.08 | | | °C |

Electrical Characteristics

(Notes 1, 6)

| Parameter | Conditions | LM35 | | | LM35C, LM35D | | | Units (Max.) |
|--|---|------------------------------|--------------------------|--------------------------|------------------------------|-----------------------------|--------------------------|--|
| | | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | Typical | Tested Limit (Note 4) | Design Limit (Note 5) | |
| Accuracy, LM35, LM35C (Note 7) | $T_A=+25^\circ\text{C}$ | ± 0.4 | ± 1.0 | ± 1.5 | ± 0.4 | ± 1.0 | ± 1.5 | $^\circ\text{C}$ |
| | $T_A=-10^\circ\text{C}$ | ± 0.5 | | | ± 0.5 | | | $^\circ\text{C}$ |
| | $T_A=T_{\text{MAX}}$ | ± 0.8 | ± 1.5 | | ± 0.8 | | | $^\circ\text{C}$ |
| | $T_A=T_{\text{MIN}}$ | ± 0.8 | | | ± 0.8 | | | $^\circ\text{C}$ |
| Accuracy, LM35D (Note 7) | $T_A=+25^\circ\text{C}$ | | | | ± 0.6 | ± 1.5 | $^\circ\text{C}$ | $^\circ\text{C}$ |
| | $T_A=T_{\text{MAX}}$ | | | | ± 0.9 | | | $^\circ\text{C}$ |
| | $T_A=T_{\text{MIN}}$ | | | | ± 0.9 | | | $^\circ\text{C}$ |
| Nonlinearity (Note 8) | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | ± 0.3 | | | ± 0.5 | ± 0.2 | | ± 0.5 $^\circ\text{C}$ |
| Sensor Gain (Average Slope) | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | +10.0 | +9.8, +10.2 | | +10.0 | | | +9.8, +10.2 mV/ $^\circ\text{C}$ |
| Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA | $T_A=+25^\circ\text{C}$ | ± 0.4 | ± 2.0 | ± 5.0 | ± 0.4 | ± 2.0 | ± 5.0 | mV/mA |
| | $T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$ | ± 0.5 | | | ± 0.5 | | | mV/mA |
| Line Regulation (Note 3) | $T_A=+25^\circ\text{C}$ | ± 0.01 | ± 0.1 | ± 0.2 | ± 0.01 | ± 0.1 | ± 0.2 | mV/V |
| | $4V \leq V_S \leq 30V$ | ± 0.02 | | | ± 0.02 | | | mV/V |
| Quiescent Current (Note 9) | $V_S=+5V, +25^\circ\text{C}$ | 56 | 80 | 158 | 56 | 80 | 138 | μA |
| | $V_S=+5V$ | 105 | | | 91 | | | μA |
| | $V_S=+30V, +25^\circ\text{C}$ | 56.2 | 82 | | 56.2 | 82 | | μA |
| | $V_S=+30V$ | 105.5 | | | 161 | 91.5 | | 141 μA |
| Change of Quiescent Current (Note 3) | $4V \leq V_S \leq 30V, +25^\circ\text{C}$ | 0.2 | 2.0 | 3.0 | 0.2 | 2.0 | 3.0 | μA |
| | $4V \leq V_S \leq 30V$ | 0.5 | | | 0.5 | | | μA |
| Temperature Coefficient of Quiescent Current | | +0.39 | | +0.7 | +0.39 | | +0.7 | $\mu\text{A}/^\circ\text{C}$ |
| Minimum Temperature for Rated Accuracy | In circuit of <i>Figure 1</i> , $I_L=0$ | +1.5 | | +2.0 | +1.5 | | +2.0 | $^\circ\text{C}$ |
| Long Term Stability | $T_J=T_{\text{MAX}}$, for 1000 hours | ± 0.08 | | | ± 0.08 | | | $^\circ\text{C}$ |

Note 1: Unless otherwise noted, these specifications apply: $-55^\circ\text{C} \leq T_J \leq +150^\circ\text{C}$ for the LM35 and LM35A; $-40^\circ\text{C} \leq T_J \leq +110^\circ\text{C}$ for the LM35C and LM35CA; and $0^\circ\text{C} \leq T_J \leq +100^\circ\text{C}$ for the LM35D. $V_S=+5\text{Vdc}$ and $I_{\text{LOAD}}=50\text{ }\mu\text{A}$, in the circuit of *Figure 2*. These specifications also apply from $+2^\circ\text{C}$ to T_{MAX} in the circuit of *Figure 1*. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is 400°C/W , junction to ambient, and 24°C/W junction to case. Thermal resistance of the TO-92 package is 180°C/W junction to ambient. Thermal resistance of the small outline molded package is 220°C/W junction to ambient. Thermal resistance of the TO-220 package is 90°C/W junction to ambient. For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and $10\text{mv}/^\circ\text{C}$ times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in $^\circ\text{C}$).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of *Figure 1*.

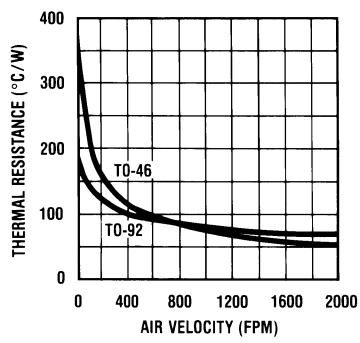
Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a $1.5\text{ k}\Omega$ resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

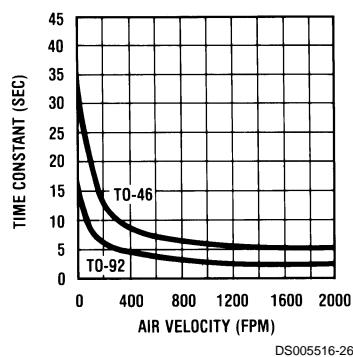
Typical Performance Characteristics

Thermal Resistance
Junction to Air



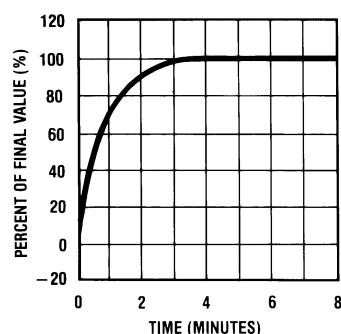
DS005516-25

Thermal Time Constant



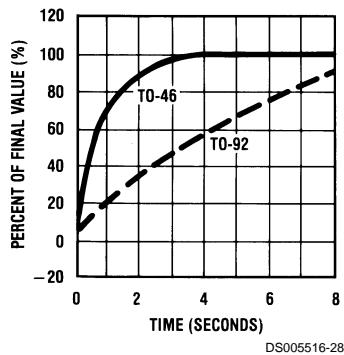
DS005516-26

Thermal Response
in Still Air



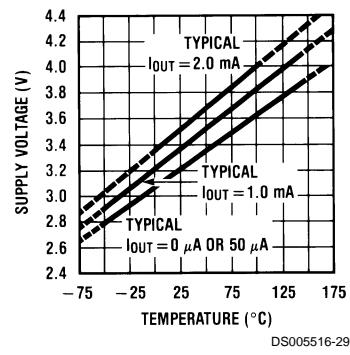
DS005516-27

Thermal Response in
Stirred Oil Bath



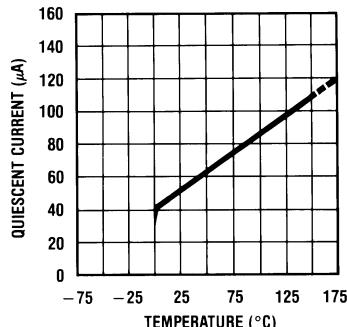
DS005516-28

Minimum Supply
Voltage vs. Temperature



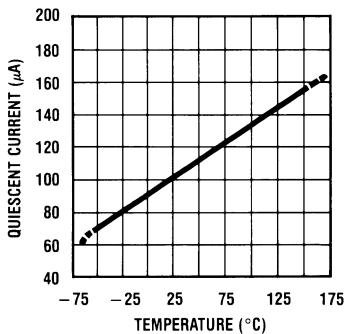
DS005516-29

Quiescent Current
vs. Temperature
(In Circuit of Figure 1.)



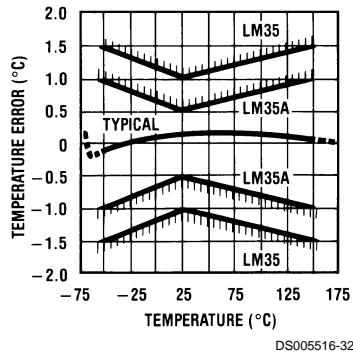
DS005516-30

Quiescent Current
vs. Temperature
(In Circuit of Figure 2.)



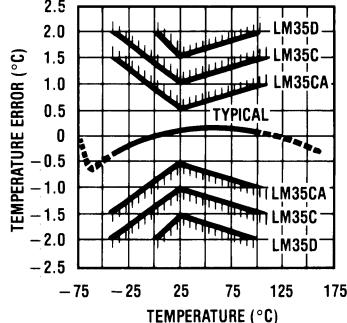
DS005516-31

Accuracy vs. Temperature
(Guaranteed)



DS005516-32

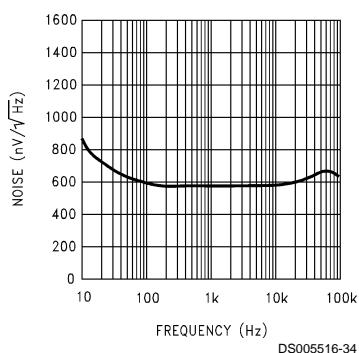
Accuracy vs. Temperature
(Guaranteed)



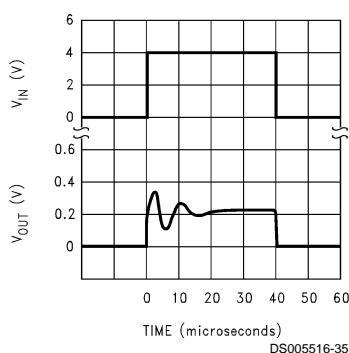
DS005516-33

Typical Performance Characteristics (Continued)

Noise Voltage



Start-Up Response



Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V₋ terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

Temperature Rise of LM35 Due To Self-heating (Thermal Resistance, θ_{JA})

| | TO-46, no heat sink | TO-46*, small heat fin | TO-92, no heat sink | TO-92**, small heat fin | SO-8 no heat sink | SO-8** small heat fin | TO-220 no heat sink |
|---|---------------------------|---------------------------|---------------------------|----------------------------|-------------------------|--------------------------|---------------------------|
| Still air | 400°C/W | 100°C/W | 180°C/W | 140°C/W | 220°C/W | 110°C/W | 90°C/W |
| Moving air | 100°C/W | 40°C/W | 90°C/W | 70°C/W | 105°C/W | 90°C/W | 26°C/W |
| Still oil | 100°C/W | 40°C/W | 90°C/W | 70°C/W | | | |
| Stirred oil | 50°C/W | 30°C/W | 45°C/W | 40°C/W | | | |
| (Clamped to metal, Infinite heat sink) | | (24°C/W) | | | | (55°C/W) | |

*Wakefield type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.

**TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

Typical Applications

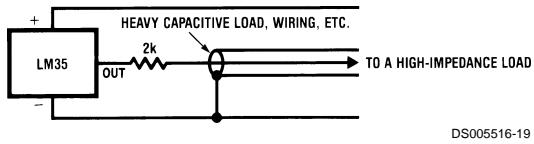


FIGURE 3. LM35 with Decoupling from Capacitive Load

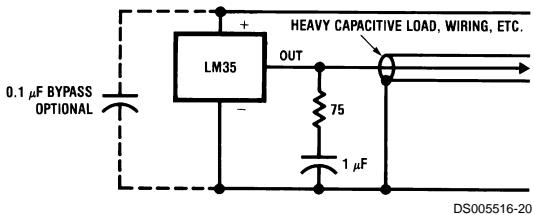


FIGURE 4. LM35 with R-C Damper

CAPACITIVE LOADS

Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pF without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor; see *Figure 3*. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground; see *Figure 4*.

When the LM35 is applied with a 200Ω load resistor as shown in *Figure 5*, *Figure 6* or *Figure 8* it is relatively immune to wiring capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients, etc, as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from V_{IN} to ground and a series R-C damper such as 75Ω in series with 0.2 or 1 μF from output to ground are often useful. These are shown in *Figure 13*, *Figure 14*, and *Figure 16*.

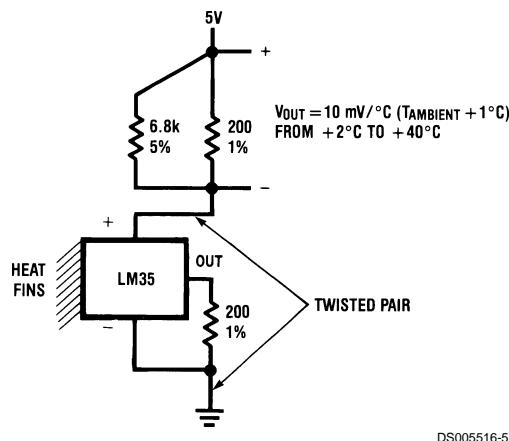


FIGURE 5. Two-Wire Remote Temperature Sensor (Grounded Sensor)

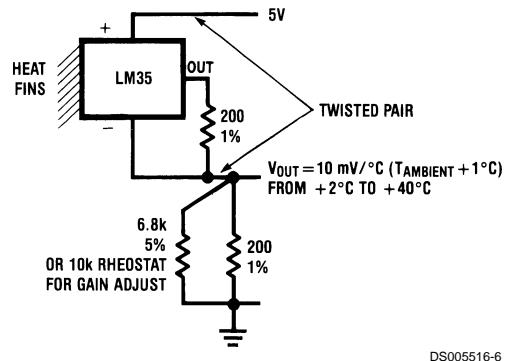


FIGURE 6. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

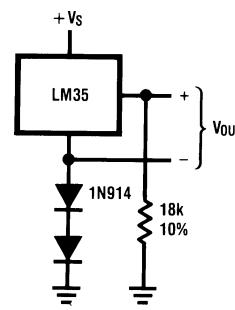


FIGURE 7. Temperature Sensor, Single Supply, -55° to +150°C

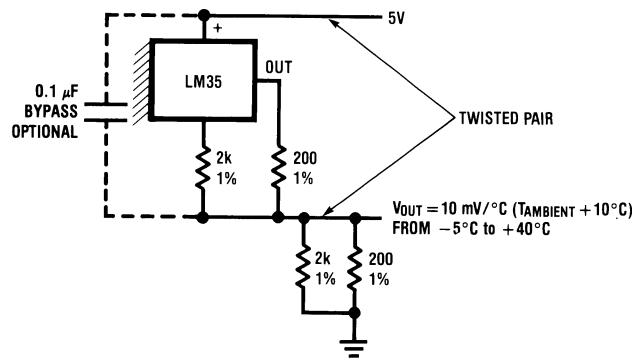


FIGURE 8. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

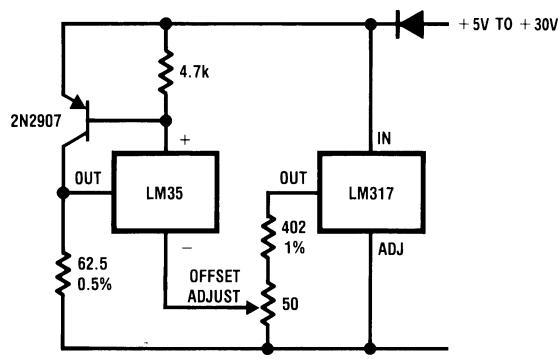


FIGURE 9. 4-To-20 mA Current Source (0°C to +100°C)

Typical Applications (Continued)

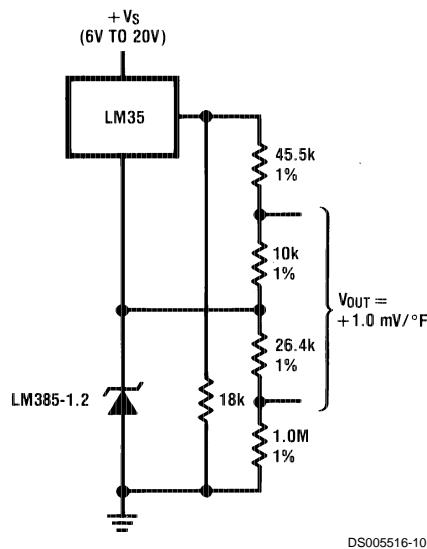


FIGURE 10. Fahrenheit Thermometer

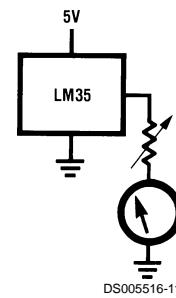


FIGURE 11. Centigrade Thermometer (Analog Meter)

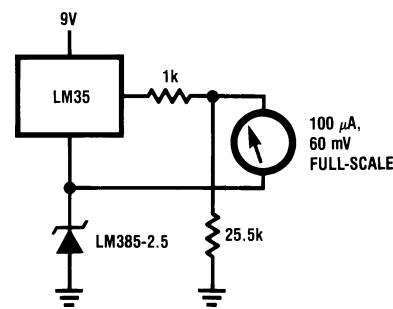
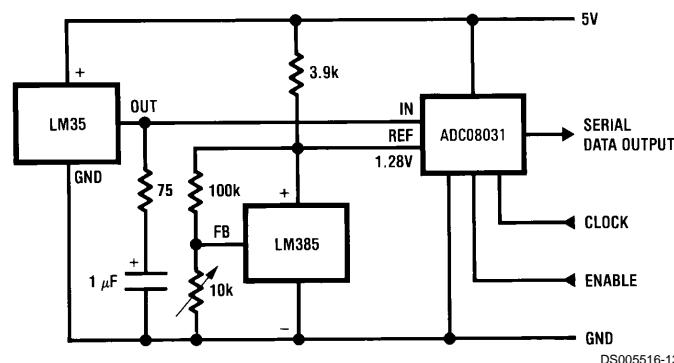
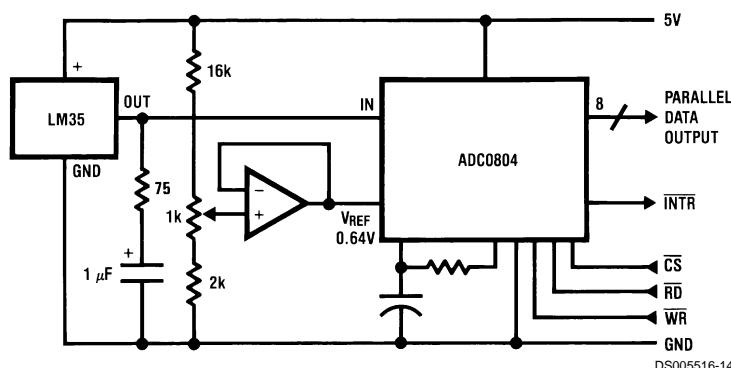
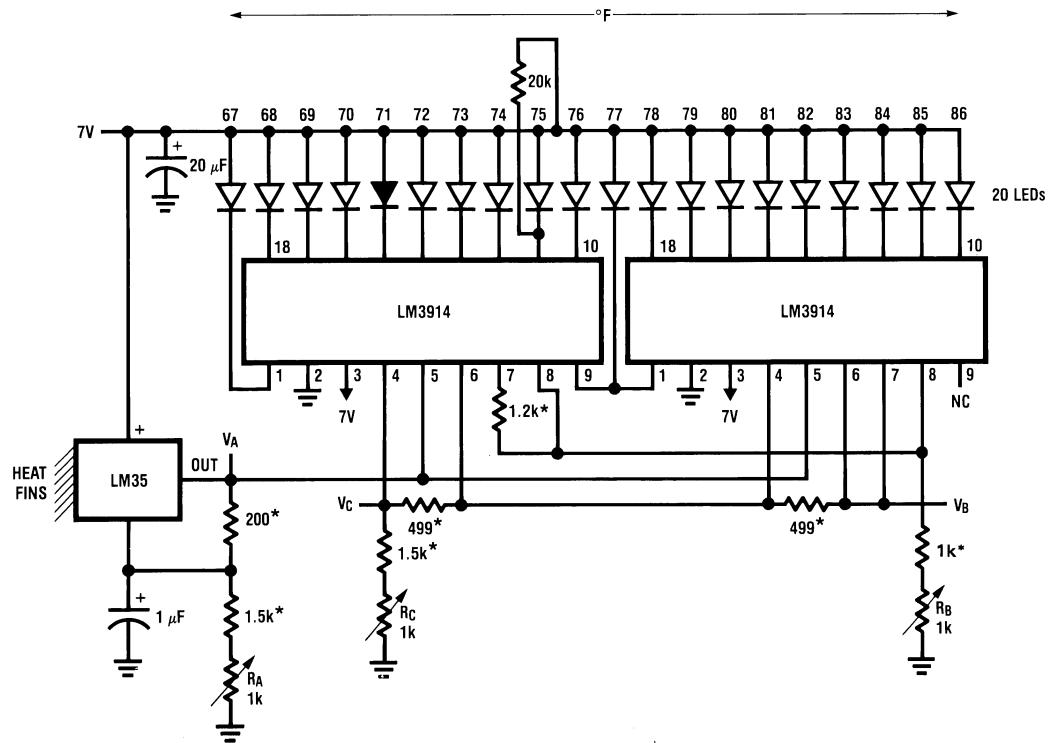
FIGURE 12. Fahrenheit Thermometer Expanded Scale Thermometer
(50° to 80° Fahrenheit, for Example Shown)

FIGURE 13. Temperature To Digital Converter (Serial Output) (+128°C Full Scale)

FIGURE 14. Temperature To Digital Converter (Parallel TRI-STATE™ Outputs for Standard Data Bus to μP Interface) (128°C Full Scale)

Typical Applications (Continued)

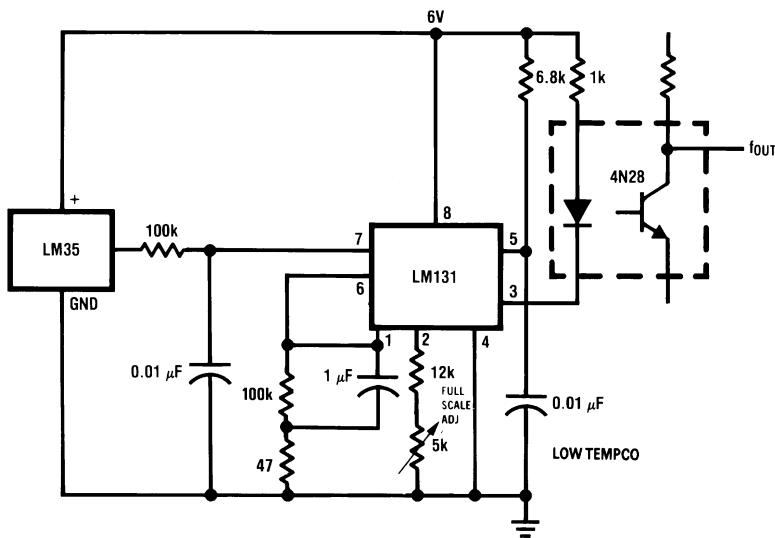


DS005516-16

*=1% or 2% film resistor

Trim R_B for V_B=3.075VTrim R_C for V_C=1.955VTrim R_A for V_A=0.075V + 100mV/C × T_{ambient}Example, V_A=2.275V at 22°C

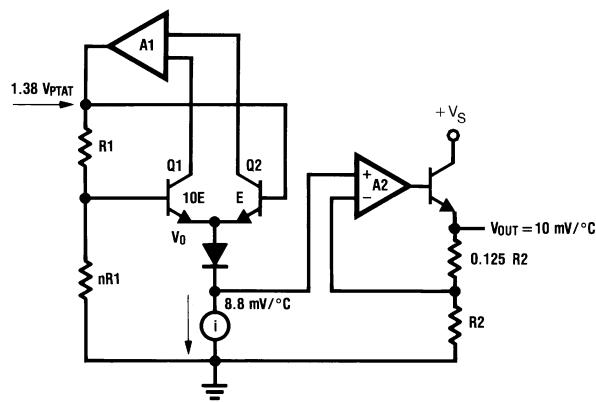
FIGURE 15. Bar-Graph Temperature Display (Dot Mode)



DS005516-15

**FIGURE 16. LM35 With Voltage-To-Frequency Converter And Isolated Output
(2°C to +150°C; 20 Hz to 1500 Hz)**

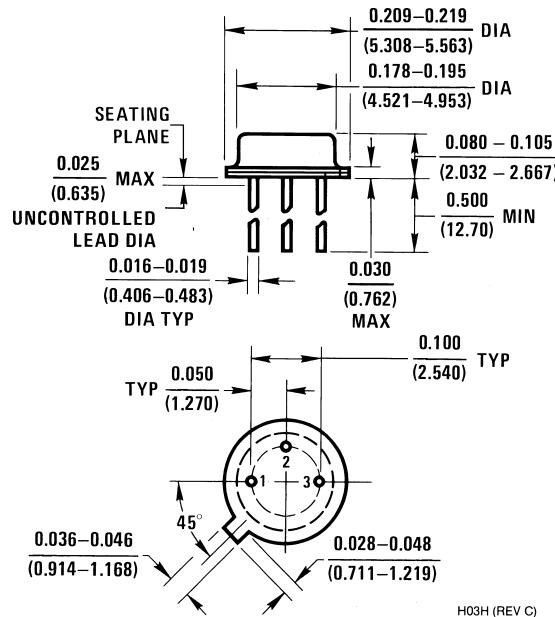
Block Diagram



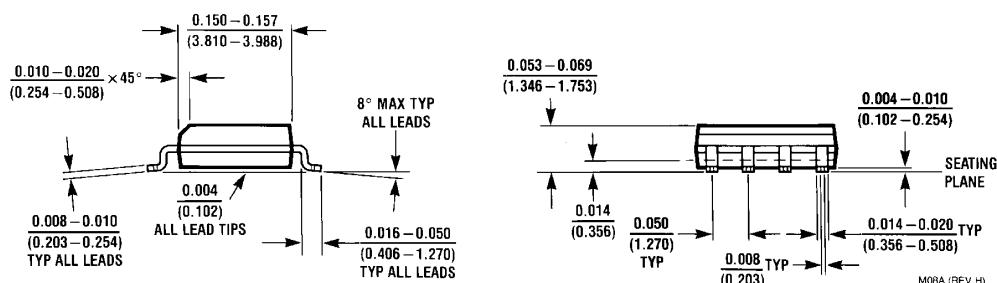
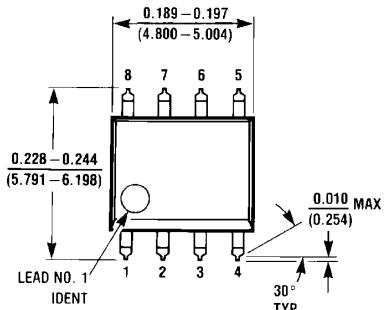
DS005516-23

Physical Dimensions

inches (millimeters) unless otherwise noted



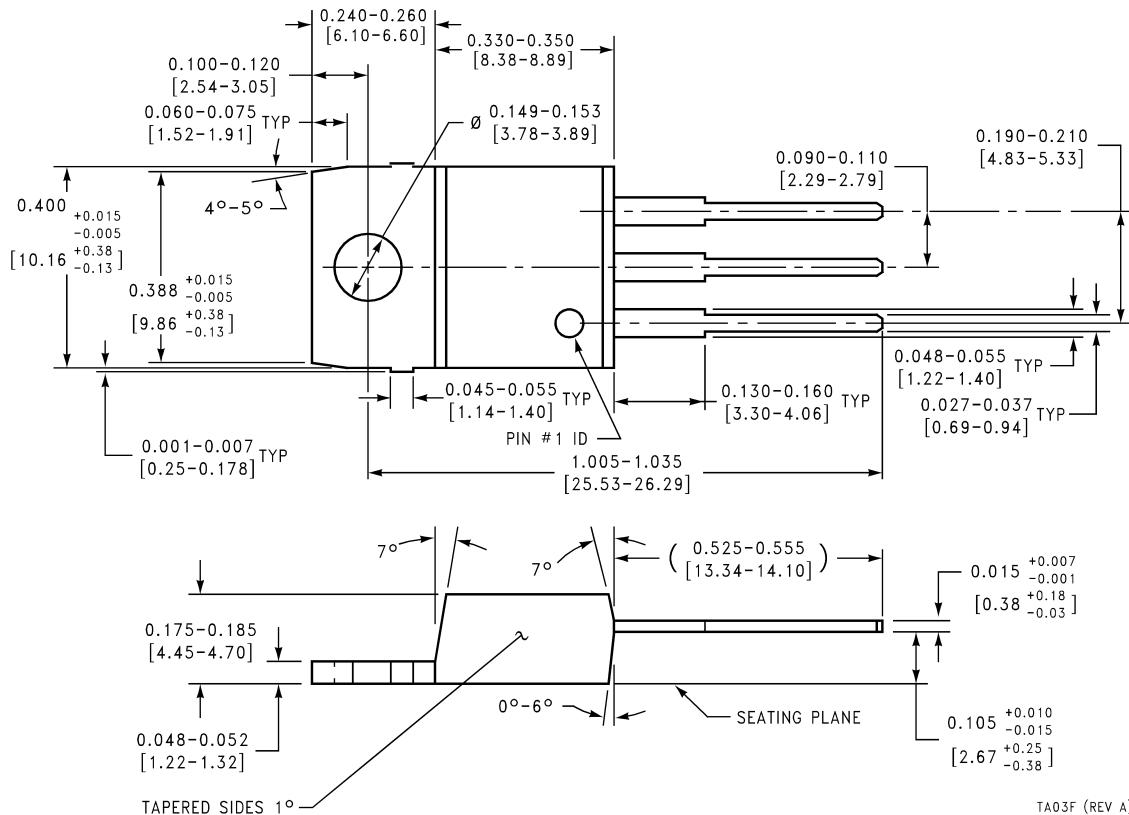
TO-46 Metal Can Package (H)
Order Number LM35H, LM35AH, LM35CH,
LM35CAH, or LM35DH
NS Package Number H03H



SO-8 Molded Small Outline Package (M)
Order Number LM35DM
NS Package Number M08A

Physical Dimensions

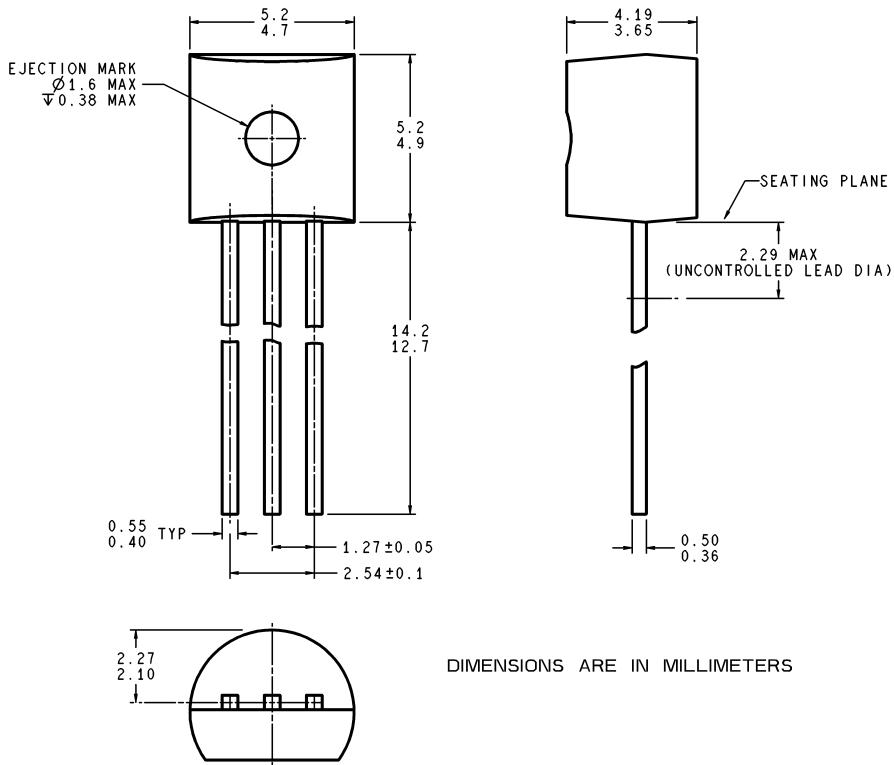
inches (millimeters) unless otherwise noted (Continued)



Power Package TO-220 (T)
Order Number LM35DT
NS Package Number TA03F

TA03F (REV A)

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Z03A (Rev G)

TO-92 Plastic Package (Z)
Order Number LM35CZ, LM35CAZ or LM35DZ
NS Package Number Z03A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

 **National Semiconductor Corporation**
 Americas
 Tel: 1-800-272-9959
 Fax: 1-800-737-7018
 Email: support@nsc.com
www.national.com

National Semiconductor Europe
 Fax: +49 (0) 180-530 85 86
 Email: europe.support@nsc.com
 Deutsch Tel: +49 (0) 69 9508 6208
 English Tel: +44 (0) 870 24 0 2171
 Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
 Tel: 65-2544466
 Fax: 65-2504466
 Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
 Tel: 81-3-5639-7560
 Fax: 81-3-5639-7507

This datasheet has been download from:

www.datasheetcatalog.com

Datasheets for electronics components.

LAMPIRAN 5

Data Sheet Bluetooth Shield

BT Shield

-Bluetooth to Serial Port Module Shield

Overview



BT Shield V2.1 is a Serial port Bluetooth module (Slave) breakout board, and it's compatible with Arduino and IFlat-32, it can directly plug in with Arduino/IFlat-32 board, use the UART port for communicating to Arduino/IFlat-32 or FT232.

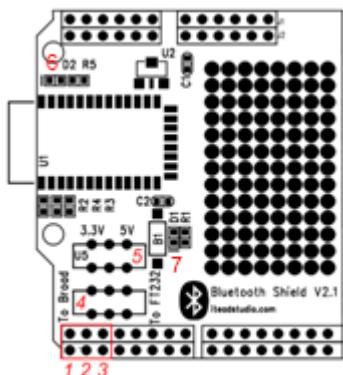
Specifications

| | |
|------------------------|-----------------------|
| Microprocessor | CSR BC417 |
| PCB size | 53.3mm X 47mm X 1.6mm |
| Indicators | PWR ,State |
| Power supply | 5V DC |
| IO | 3 |
| Communication Protocol | UART/Bluetooth 2.0 |
| RoSH | Yes |

Electrical Characteristics

| Specification | | Min | Type | Max | Unit |
|---------------------|-----------------------|------|------|-----|------|
| Power Voltage | | 4.5 | 5 | 5.5 | VDC |
| Input Voltage VH | Target Voltage = 3.3V | 3 | 3.3 | 3.6 | V |
| | Target Voltage = 5V | 4.5 | 5 | 5.5 | |
| Input Voltage VL: | | -0.3 | 0 | 0.5 | V |
| Current Consumption | | - | 20 | 40 | mA |

Hardware



| Pin | Pad Name | Type | Description |
|-----|----------|------|--|
| 1 | RX/TX | I/O | UART communication Port (Depend on switcher 4) |
| 2 | TX/RX | I/O | UART communication Port (Depend on switcher 4) |
| 3 | State | O | State Direction |

| Switcher | Name | Description |
|----------|------------------------------|---------------------------|
| 4 | UART Communication Switch | Connect to broad or FT232 |
| 5 | Communication Voltage Switch | Set the interface voltage |

| LED | Name | Description |
|-----|-------|--|
| 6 | PWR | When power on, the PWR LED light. |
| 7 | State | When the module in standby mode, the State LED will alternating light off. When the serial port open, the State LED light. |

AT command

Default:

Slave, 9600 baud rate, N, 8, 1. Pincode 1234

AT command:

1. Communications Test :

Sent : AT

receive : OK

2. Change baud rate :

Sent : AT+BAUD1

receive : OK1200

Sent : AT+BAUD2

receive : OK2400

1-----1200

2-----2400

3-----4800

4-----9600

5-----19200

6-----38400

7-----57600

8-----115200

Baud rate setting can be save even power down.

3. Change Bluetooth device name:

Sent : AT+NAMEdevicename

receive : OKname

(devicename is the name you want the device to be , and it will be searched with this name)

Name setting can be save even power down.

4. Change Pincode:

Sent : AT+PINxxxx

receive : OKsetpin

(xxxx is the pin code you set)

Pin code can be save even power down.

Demo Code

```
unsigned int timeout=0;
unsigned char state=0;

ISR(TIMER2_OVF_vect)          //Timer2 Service
{
    TCNT2 = 0;
    timeout++;
    if (timeout>61)
    {
        state=1;
        timeout=0;
    }
}

void init_timer2(void)
{
    TCCR2A |= (1 << WGM21) | (1 << WGM20);
    TCCR2B |= 0x07;    // by clk/1024
    ASSR |= (0<<AS2);    // Use internal clock – external clock not used in Arduino
    TIMSK2 |= 0x01;    //Timer2 Overflow Interrupt Enable
    TCNT2 = 0;
    sei();
}

void setup()
{
    Serial.begin(9600);
    pinMode(2,INPUT);
    pinMode(13,OUTPUT);
    attachInterrupt(0,cleanTime,FALLING);
    init_timer2();
}

void loop()
{
    switch(state)
    {
        case 0:
            digitalWrite(13,LOW);
            break;
    }
}
```

```

case 1:
    digitalWrite(13,HIGH);
    Serial.print("Hellow BT");
    break;
}

}

void cleantime()
{
    timeout=0;
    state=0;
}

```

License



Attribution Non-Commercial Share Alike
cc by-nc-sa

Revision History

| Rev. | Description | Release date |
|------|-----------------|--------------|
| v1.0 | Initial version | 1/15/2011 |
| | | |
| | | |

BT Shield 2.2

-Bluetooth to Serial Port Module Shield

Overview



BT shield V2.2 is a serial port Bluetooth module (with master and slave mode) breakout board, it's compatible with Arduino and IFlat-32, it can directly plug on Arduino/IFlat-32 board, use UART port for communication with Arduino/IFlat-32 or PC.

Specifications

| | |
|------------------------|-----------------------|
| Microprocessor | CSR BC417 |
| PCB size | 53.3mm X 47mm X 1.6mm |
| Indicators | PWR State |
| Power supply | 5V DC |
| IO | 6 |
| Communication Protocol | UART/Bluetooth 2.0 |
| RoHS | Yes |

Electrical Characteristics

| Specification | | Min | Type | Max | Unit |
|---------------------|-----------------------|------|------|-----|------|
| Power Voltage | | 4.5 | 5 | 5.5 | VDC |
| Input Voltage VH | Target Voltage = 3.3V | 3 | 3.3 | 3.6 | V |
| | Target Voltage = 5V | 4.5 | 5 | 5.5 | |
| Input Voltage VL: | | -0.3 | 0 | 0.5 | V |
| Current Consumption | | - | 20 | 40 | mA |

Hardware

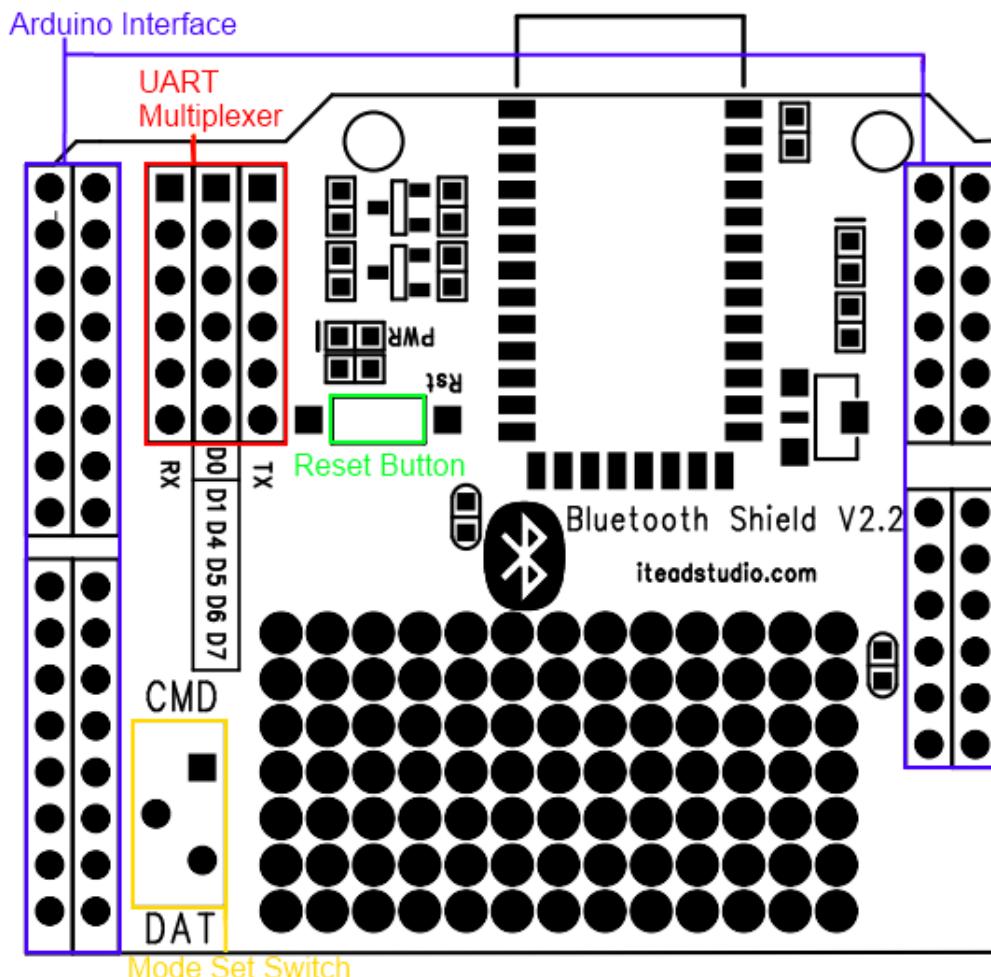


Figure 1 Top view of BT shield V2.2

UART Multiplexer (For free UART connection setting)

You can use the jumper to connect the TXD and RXD pins of HC-05 to D0, D1, D4~D7 pin of Arduino.

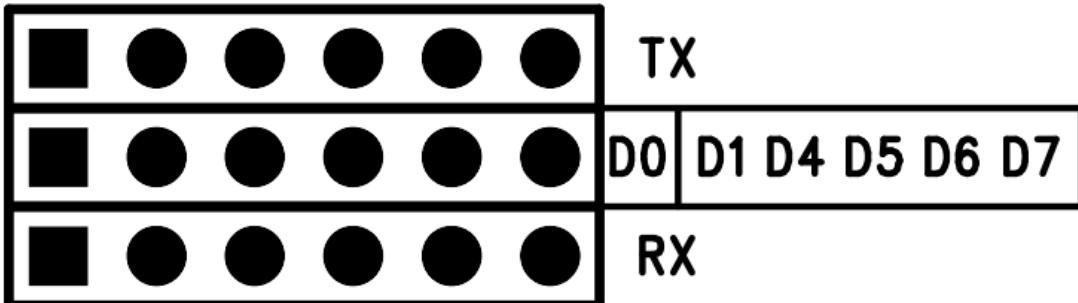


Figure 2 UART Multiplexer

When using the connection as Figure 3, the BT shield connects to the ATMega328 chip on board.

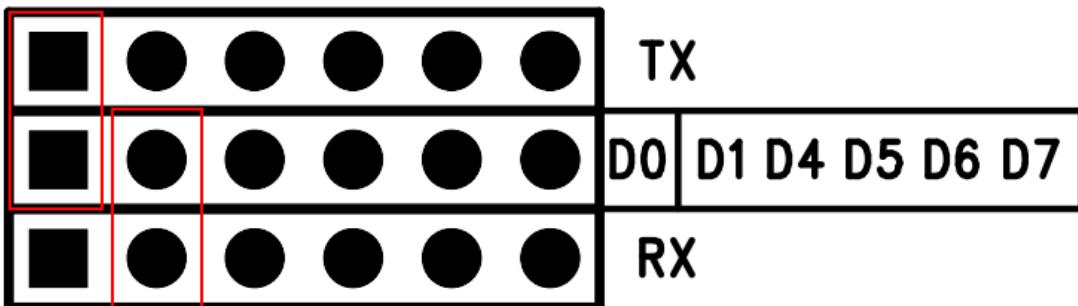


Figure 3 Connect the Arduino board

When using the connection as Figure 4, the HC-05 connects with the FT232RL chip, and the FT232RL connect to PC by USB. With this configuration you can use the serial software on PC to control or configure the HC-05 module.

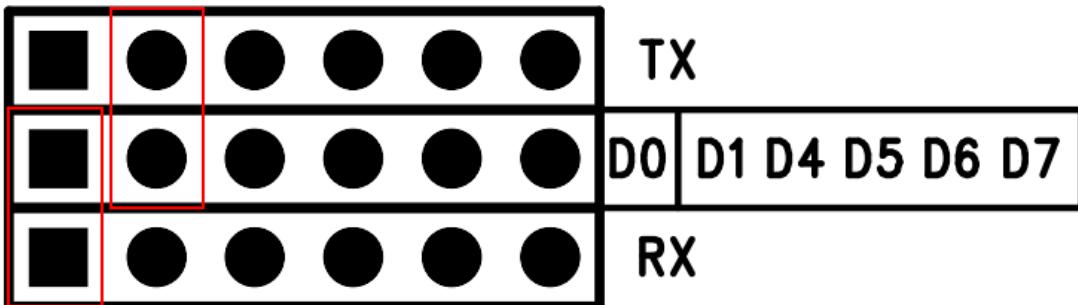


Figure 4 Connect the UART Interface as FT232

Except the 2 configurations above, you can connect the TXD and RXD to any other pins from D4-D7, and using the software-serial library to control the HC-05 module.

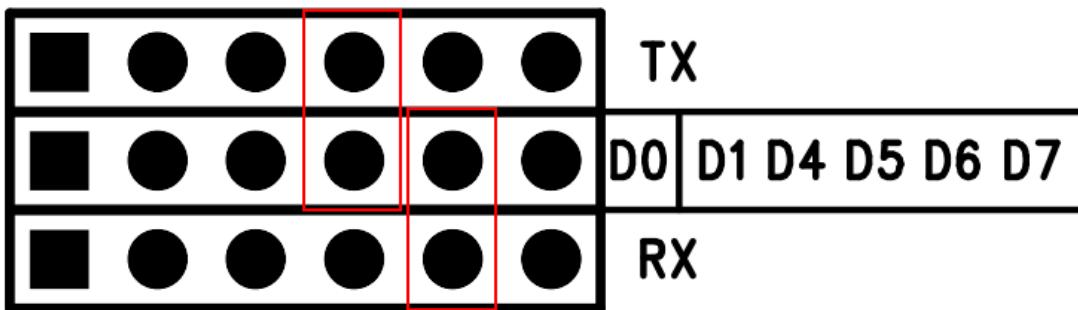


Figure 5 an example for software-serial connection

Mode Switch

The Mode Set Switch is connected to PIO11, when it is pushed to CMD and powered on, the HC-05 enter command mode, HC-05 module can receive and response AT command in this mode. When it is pushed to DATA and powered on. The HC-05 enters data mode and will not accept AT command.

States LED

When power on and disconnect the port, states LED blinks 1time/2s ; when the module connect and open the serial port, states LED blinks 2times/s.

AT COMMAND

1. Test command:

| Command | Respond | Parameter |
|---------|---------|-----------|
| AT | OK | - |

2. Reset

| Command | Respond | Parameter |
|----------|---------|-----------|
| AT+RESET | OK | - |

3. Get firmware version

| Command | Respond | Parameter |
|-------------|------------------------|-----------------------------|
| AT+VERSION? | +VERSION:<Param> OK | Param : firmware version |

Example:

```
AT+VERSION?\r\n
+VERSION:2.0-20100601
OK
```

4. Restore default

| Command | Respond | Parameter |
|---------|---------|-----------|
| AT+ORGL | OK | - |

Default state:

Slave mode, pin code :1234, device name: H-C-2010-06-01 ,Baud 38400bits/s.

5. Get module address

| Command | Respond | Parameter |
|----------|---------------------|------------------------------------|
| AT+ADDR? | +ADDR:<Param> OK | Param: address of Bluetooth module |

Bluetooth address: NAP: UAP : LAP

Example:

```
AT+ADDR?\r\n
+ADDR:1234:56:abcdef
OK
```

6. Set/Check module name:

| Command | Respond | Parameter |
|-----------------|-----------------------------|---|
| AT+NAME=<Param> | OK | Param: Bluetooth module name (Default :HC-05) |
| AT+NAME? | +NAME:<Param> OK (/FAIL) | |

Example:

```
AT+NAME=HC-05\r\n      set the module name to "HC-05"
OK
AT+NAME=ITeadStudio\r\n
OK
AT+NAME?\r\n
+NAME: ITeadStudio
OK
```

7. Get the Bluetooth device name:

| Command | Respond | Parameter |
|-------------------|---------------------------------------|--|
| AT+RNAME?<Param1> | 1. +NAME:<Param2> OK 2. FAIL | Param1,Param 2 : the address of Bluetooth device |

Example: (Device address 00:02:72:0d:22:24, name: ITead)
 AT+RNAME? 0002, 72, od2224\r\n
 +RNAME:ITead
 OK

8. Set/Check module mode:

| Command | Respond | Parameter |
|---------|---------|-----------|
|---------|---------|-----------|

| | | |
|-----------------|---------------------|--|
| AT+ROLE=<Param> | OK | Param: 0- Slave 1-Master 2-Slave-Loop |
| AT+ ROLE? | +ROLE:<Param> OK | |

9. Set/Check device class

| Command | Respond | Parameter |
|------------------|------------------------------------|---------------------|
| AT+CLASS=<Param> | OK | Param: Device Class |
| AT+ CLASS? | 1. +CLASS:<Param> OK 2. FAIL | |

10. Set/Check GIAC (General Inquire Access Code)

| Command | Respond | Parameter |
|----------------|--------------------|-----------------------------------|
| AT+IAC=<Param> | 1.OK 2. FAIL | Param: GIAC (Default : 9e8b33) |
| AT+IAC | +IAC:<Param> OK | |

Example:

```
AT+IAC=9e8b3f\r\n
OK
AT+IAC?\r\n
+IAC: 9e8b3f
OK
```

11. Set/Check -- Query access patterns

| Command | Respond | Parameter |
|-----------------------------------|--|---|
| AT+INQM=<Param>,<Param2>,<Param3> | 1.OK 2. FAIL | Param: 0— inquiry_mode_st andard 1— inquiry_mode_rs si Param2: Maximum number of Bluetooth devices to respond to Param3: Timeout (1-48 : 1.28s to 61.44s) |
| AT+ INQM? | +INQM :<Param>,<Param2>,<Param3> OK | |

Example:

```
AT+INQM=1,9,48\r\n
OK
AT+INQM\r\n
```

+INQM:1, 9, 48

OK

12. Set/Check PIN code:

| Command | Respond | Parameter |
|-----------------|-----------------------|-----------------|
| AT+PSWD=<Param> | OK | Param: PIN code |
| AT+ PSWD? | + PWD : <Param> OK | (Default 1234) |

13. Set/Check serial parameter:

| Command | Respond | Parameter |
|-------------------------------------|---|--|
| AT+UART=<Param>, <Param2>, <Param3> | OK | Param1: Baud Param2: Stop bit Param3: Parity |
| AT+ UART? | +UART=<Param>, <Param2>, <Param3> OK | |

Example:

AT+UART=115200, 1,2,\r\n

OK

AT+UART?

+UART:115200,1,2

OK

14. Set/Check connect mode:

| Command | Respond | Parameter |
|-------------------|------------------------|--|
| AT+C MODE=<Param> | OK | Param: 0 - connect fixed address 1 - connect any address 2 - slave-Loop |
| AT+ C MODE? | + C MODE:<Param> OK | |

15. Set/Check fixed address:

| Command | Respond | Parameter |
|-----------------|----------------------|-----------------------------|
| AT+BIND=<Param> | OK | Param: Fixed address |
| AT+ BIND? | + BIND:<Param> OK | (Default 00:00:00:00:00:00) |

Example:

AT+BIND=1234, 56, abcdef\r\n

OK

AT+BIND?\r\n

+BIND:1234:56:abcdef

OK

16. Set/Check LED I/O

| Command | Respond | Parameter |
|---------------------------|------------------------------------|--|
| AT+POLAR=<Param1,<Param2> | OK | Param1: 0- PIO8 low drive LED 1- PIO8 high drive LED Param2: 0- PIO9 low drive LED 1- PIO9 high drive LED |
| AT+ POLAR? | + POLAR=<Param1>,<Param2> OK | |

17. Set PIO output

| Command | Respond | Parameter |
|--------------------------|---------|--|
| AT+PIO=<Param1>,<Param2> | OK | Param1: PIO number Param2: PIO level 0- low 1- high |

Example:

1. PIO10 output high level

AT+PIO=10, 1\r\n

OK

18. Set/Check – scan parameter

| Command | Respond | Parameter |
|---|---|--|
| AT+IPSCAN=<Param1>,<Param2>,<Param3>,<Param4> | OK | Param1: Query time interval Param2 : Query duration Param3 : Paging interval Param4 : Call duration |
| AT+IPSCAN? | +IPSCAN:<Param1>,<Param2>,<Param3>,<Param4> OK | |

Example:

AT+IPSCAN =1234,500,1200,250\r\n

OK

AT+IPSCAN?

+IPSCAN:1234,500,1200,250

19. Set/Check – SHIFF parameter

| Command | Respond | Parameter |
|--|--|--|
| AT+SNIFF=<Param1>,<Param2>,<Param3>,<Param4> | OK | Param1: Max time Param2: Min time Param3: Retry time Param4: Time out |
| AT+ SNIFF? | +SNIFF:<Param1>,<Param2>,<Param3>,<Param4> OK | |

20. Set/Check security mode

| Command | Respond | Parameter |
|---------------------------|-----------------------------------|---|
| AT+SENM=<Param1>,<Param2> | 1. OK 2. FAIL | Param1: 0—sec_mode0+off 1—sec_mode1+non_secure 2—sec_mode2_service 3—sec_mode3_link 4—sec_mode_unknown Param2: 0—hci_enc_mod_e_off 1—hci_enc_mod_e_pt_to_pt 2—hci_enc_mod_e_pt_to_pt_and_broadcast |
| AT+ SENM? | + SENM:<Param1>,<Param2> OK | |

21. Delete Authenticated Device

| Command | Respond | Parameter |
|------------------|---------|--|
| AT+PMSAD=<Param> | OK | Param: Authenticated Device Address |

Example:

AT+PMSAD =1234,56,abcdef\r\n
OK

22. Delete All Authenticated Device

| Command | Respond | Parameter |
|---------|---------|-----------|
|---------|---------|-----------|

| | | |
|-----------|----|---|
| AT+ RMAAD | OK | - |
|-----------|----|---|

23. Search Authenticated Device

| Command | Respond | Parameter |
|-----------------|------------------|-----------------------|
| AT+FSAD=<Param> | 1. OK 2. FAIL | Param: Device address |

24. Get Authenticated Device Count

| Command | Respond | Parameter |
|----------|----------------------|---------------------|
| AT+ADCN? | +ADCN: <Param> OK | Param: Device Count |

25. Most Recently Used Authenticated Device

| Command | Respond | Parameter |
|----------|-----------------------|--|
| AT+MRAD? | + MRAD: <Param> OK | Param: Recently Authenticated Device Address |

26. Get the module working state

| Command | Respond | Parameter |
|------------|------------------------|--|
| AT+ STATE? | + STATE: <Param> OK | Param: "INITIALIZED" "READY" "PAIRABLE" "PAIRED" "INQUIRING" "CONNECTING" "CONNECTED" "DISCONNECTED" "NUKNOW" |

27. Initialize the SPP profile lib

| Command | Respond | Parameter |
|---------|------------------|-----------|
| AT+INIT | 1. OK 2. FAIL | - |

28. Inquiry Bluetooth Device

| Command | Respond | Parameter |
|---------|--|---|
| AT+INQ | +INQ: <Param1> , <Param2>, <Param3> OK | Param1: Address Param2 : Device Class Param3 : RSSI Signal strength |

Example:

```
AT+INIT\r\n
OK
AT+IAC=9e8b33\r\n
OK
AT+CLASS=0\r\n
AT+INQM=1,9,48\r\n
At+INQ\r\n
+INQ:2:72:D2224,3E0104,FFBC
+INQ:1234:56:0,1F1F,FFC1
+INQ:1234:56:0,1F1F,FFC0
+INQ:1234:56:0,1F1F,FFC1
+INQ:2:72:D2224,3F0104,FFAD
+INQ:1234:56:0,1F1F,FFBE
+INQ:1234:56:0,1F1F,FFC2
+INQ:1234:56:0,1F1F,FFBE
+INQ:2:72:D2224,3F0104,FFBC
OK
```

28. Cancel Inquiring Bluetooth Device

| Command | Respond | Parameter |
|----------|---------|-----------|
| AT+ INQC | OK | - |

29. Equipment Matching

| Command | Respond | Parameter |
|---------------------------|------------------|---|
| AT+PAIR=<Param1>,<Param2> | 1. OK 2. FAIL | Param1 : Device Address Param2: Time out |

30. Connect Device

| Command | Respond | Parameter |
|-----------------|------------------|------------------------|
| AT+LINK=<Param> | 1. OK 2. FAIL | Param : Device Address |

Example:

```
AT+FSAD=1234,56,abcdef\r\n
OK
AT+LINK=1234,56,abcdef\r\n
OK
```

31. Disconnect

| Command | Respond | Parameter |
|---------|------------------|----------------|
| AT+DISC | 1. +DISC:SUCCESS | Param : Device |

| | | |
|--|---|---------|
| | OK 2. +DISC:LINK_LOSS OK 3. +DISC:NO_SLC OK 4. +DISC:TIMEOUT OK 5. +DISC:ERROR OK | Address |
|--|---|---------|

32. Energy-saving mode

| Command | Respond | Parameter |
|--------------------|---------|------------------------|
| AT+ENSNIFF=<Param> | OK | Param : Device Address |

33. Exerts Energy-saving mode

| Command | Respond | Parameter |
|--------------------|---------|------------------------|
| AT+EXSNIFF=<Param> | OK | Param : Device Address |

Application Example

This is a demo that HC-05 is a master device and communicates to hc-06.

Step 1. Push the mode switch to CMD

Step 2. Power on, module enter command state

Step 3. Using baud rate 38400, send the "AT+ROLE=1\r\n" to module, with "OK\r\n" means setting successes.

Step 4. Send "AT+CMODE=1\r\n", set HC-05 connect to any address, with "OK\r\n" means setting successes.

Revision History

| Rev. | Description | Release date |
|------|-----------------|--------------|
| v1.0 | Initial version | 2011-7-22 |

LAMPIRAN 6

Data Sheet Arduino

LAMPIRAN 7

Data Sheet Dioda 1N914

Fast Switching Diode

Features

- Fast switching speed
- High reliability
- High conductance
- For general purpose switching applications

Mechanical Data

Case: DO-35 Glass Case



94 9367

Weight: approx. 130 mg

Packaging Codes/Options:

TR / 10 k per 13 " reel (52 mm tape), 50 k/box

TAP / 10 k per Ammopack (52 mm tape), 50 k/box

Parts Table

| Part | Type differentiation | Ordering code | Remarks |
|-------|--------------------------|----------------------|--------------------------|
| 1N914 | $V_{RRM} = 75 \text{ V}$ | 1N914-TAP / 1N914-TR | Ammopack / Tape and Reel |

Absolute Maximum Ratings

$T_{amb} = 25 \text{ }^{\circ}\text{C}$, unless otherwise specified

| Parameter | Test condition | Symbol | Value | Unit |
|---|--|--------------|-------|------|
| Non repetitive peak reverse voltage | | V_{RM} | 100 | V |
| Repetitive peak reverse voltage | | V_{RRM} | 75 | V |
| Working peak reverse voltage | | V_{RWM} | 75 | V |
| DC blocking voltage | | V_R | 75 | V |
| RMS Reverse voltage | | $V_{R(RMS)}$ | 53 | V |
| Forward current | | I_F | 300 | mA |
| Average rectified current | half wave rectification with resistive load and $f > 50 \text{ MHz}$ | I_{FAV} | 200 | mA |
| Non repetitive peak forward surge current | $t = 1 \text{ s}$ | I_{FSM} | 1 | A |
| | $t = 1 \mu\text{s}$ | I_{FSM} | 4 | A |
| Power dissipation | $I = 4 \text{ mm}, T_L = 25 \text{ }^{\circ}\text{C}$ | P_d | 500 | mW |

Thermal Characteristics

$T_{amb} = 25 \text{ }^{\circ}\text{C}$, unless otherwise specified

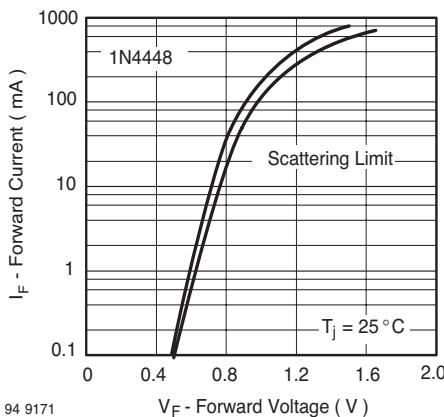
| Parameter | Test condition | Symbol | Value | Unit |
|---|---|----------------|-------------|------|
| Junction ambient | $I = 4 \text{ mm}, T_L = \text{constant}$ | R_{thJA} | 300 | K/W |
| Operating and storage temperature range | | T_j, T_{stg} | -65 to +175 | °C |

Electrical Characteristics

$T_{amb} = 25^{\circ}\text{C}$, unless otherwise specified

| Parameter | Test condition | Symbol | Min | Typ. | Max | Unit |
|-----------------------|--|----------|-----|------|-----|---------------|
| Forward voltage | $I_F = 10 \text{ mA}$ | V_F | | | 1 | V |
| Breakdown Voltage | $I_R = 100 \mu\text{A}$ | V_R | 100 | | | V |
| Peak reverse current | $V_R = 75 \text{ V}$ | I_R | | | 5.0 | μA |
| | $V_R = 20 \text{ V}, T_j = 150^{\circ}\text{C}$ | I_R | | | 50 | μA |
| | $V_R = 20 \text{ V}$ | I_R | | | 25 | nA |
| Diode capacitance | $V_R = 0, f = 1 \text{ MHz}$ | C_D | | | 4 | pF |
| Reverse recovery time | $I_F = 10 \text{ mA}$ to $I_R = 1 \text{ mA}$, $V_R = 6 \text{ V}, R_L = 100 \Omega$ | t_{rr} | | | 4 | ns |

Typical Characteristics ($T_{amb} = 25^{\circ}\text{C}$ unless otherwise specified)



94 9171

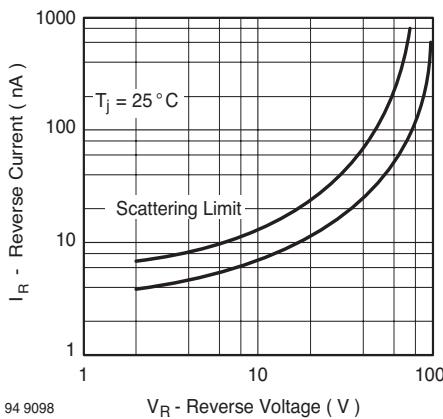
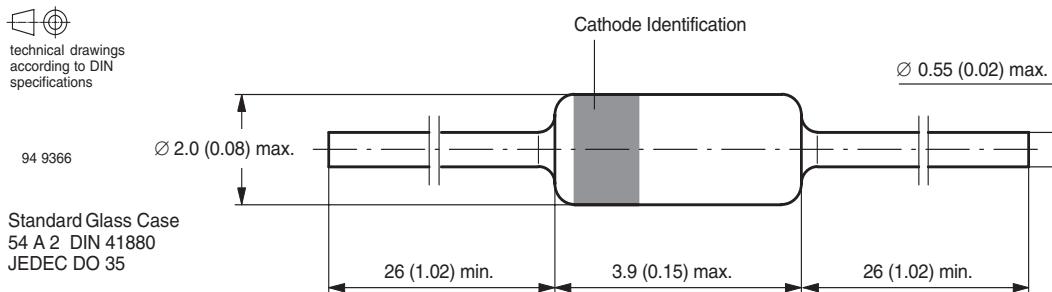


Figure 2. Reverse Current vs. Reverse Voltage

Figure 1. Forward Current vs. Forward Voltage

Package Dimensions in mm (Inches)



Ozone Depleting Substances Policy Statement

It is the policy of **Vishay Semiconductor GmbH** to

1. Meet all present and future national and international statutory requirements.
2. Regularly and continuously improve the performance of our products, processes, distribution and operating systems with respect to their impact on the health and safety of our employees and the public, as well as their impact on the environment.

It is particular concern to control or eliminate releases of those substances into the atmosphere which are known as ozone depleting substances (ODSs).

The Montreal Protocol (1987) and its London Amendments (1990) intend to severely restrict the use of ODSs and forbid their use within the next ten years. Various national and international initiatives are pressing for an earlier ban on these substances.

Vishay Semiconductor GmbH has been able to use its policy of continuous improvements to eliminate the use of ODSs listed in the following documents.

1. Annex A, B and list of transitional substances of the Montreal Protocol and the London Amendments respectively
2. Class I and II ozone depleting substances in the Clean Air Act Amendments of 1990 by the Environmental Protection Agency (EPA) in the USA
3. Council Decision 88/540/EEC and 91/690/EEC Annex A, B and C (transitional substances) respectively.

Vishay Semiconductor GmbH can certify that our semiconductors are not manufactured with ozone depleting substances and do not contain such substances.

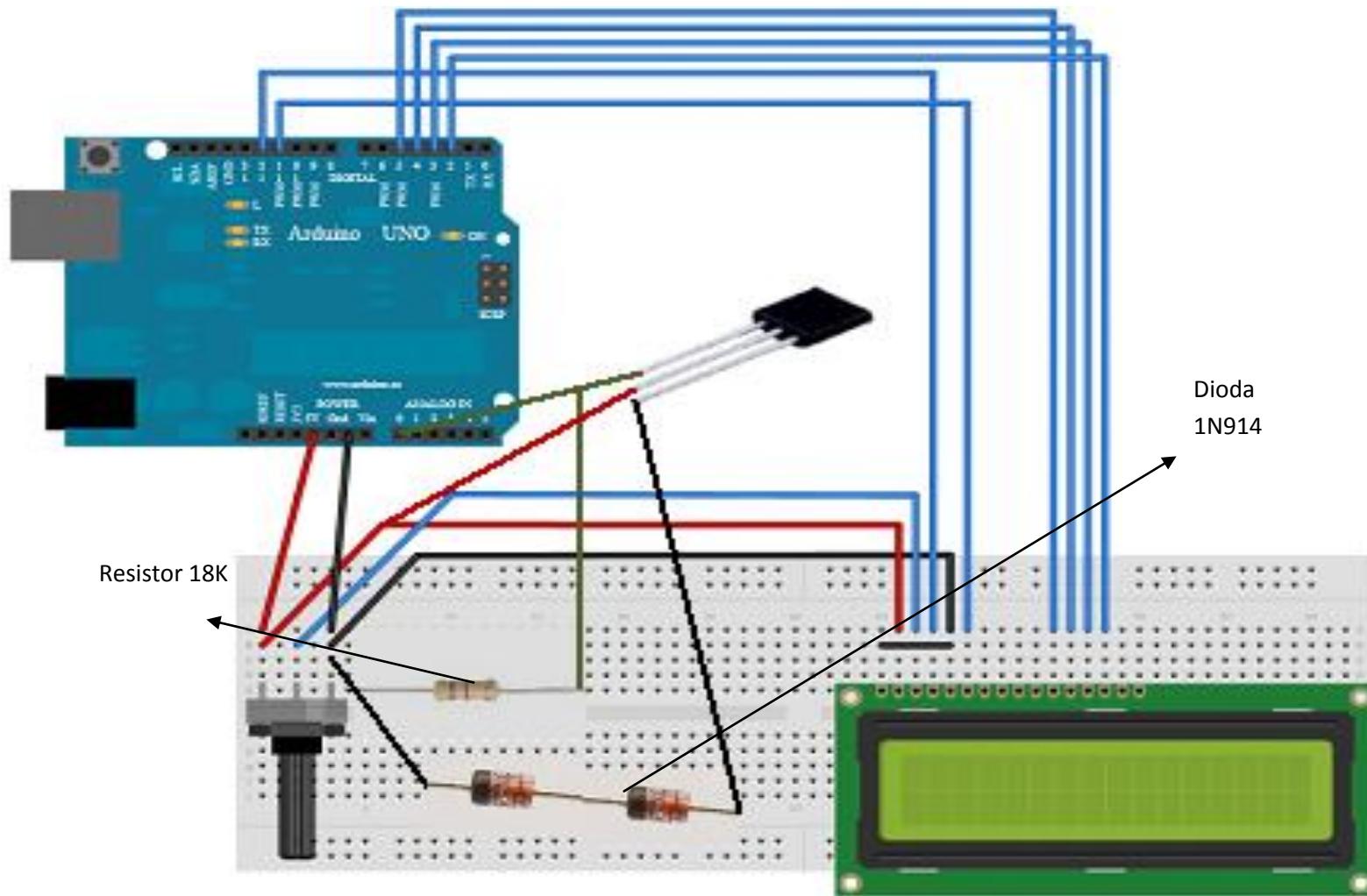
We reserve the right to make changes to improve technical design and may do so without further notice.

Parameters can vary in different applications. All operating parameters must be validated for each customer application by the customer. Should the buyer use Vishay Semiconductors products for any unintended or unauthorized application, the buyer shall indemnify Vishay Semiconductors against all claims, costs, damages, and expenses, arising out of, directly or indirectly, any claim of personal damage, injury or death associated with such unintended or unauthorized use.

Vishay Semiconductor GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany
Telephone: 49 (0)7131 67 2831, Fax number: 49 (0)7131 67 2423

LAMPIRAN 8

Rangkaian LM35DZ Yang Dimodifikasi



LAMPIRAN 9

Rangkaian Skematik Sistem Multi Sensor

