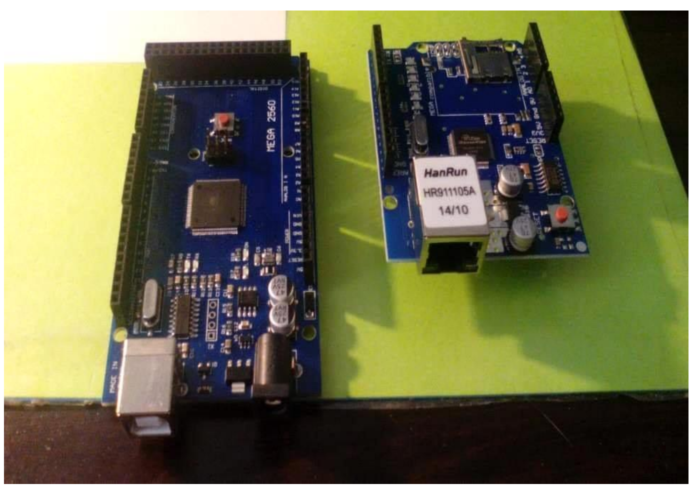
Curso - Técnico de Eletrónica e Telecomunicações

Anos letivos 2013-2016

Relatório da Prova de Aptidão Profissional - P.A.P

Eficiência prática – Alternativa económica

Arduino Placa Input



Formando:Pedro Pacheco de Sousa

Formadores e orientadores: Daniel Oliveira, Fábio Amaral e Paulo Martinho

19/2/2016

Agradecimentos

Em primeiro lugar, quero expressar aqui o meu profundo agradecimento à Escola de Novas Tecnologias - ENTA- que me acolheu, e ao Dr. João Lima pela oportunidade que me deu ao ingressar nos estudos, através do curso - Técnico de Eletrónica e Telecomunicações.

Durante a realização desta Prova de Aptidão Profissional pude contar com o apoio de diversas pessoas e com o recurso a várias fontes de informação, que contribuíram de diversas formas para o resultado obtido.

Agradeço aos meus Formadores, pelo tempo disponibilizado e pelo empenho com que me auxiliaram.

Ao professor Daniel Oliveira, por ter partilhado comigo a sua sabedoria, o que me permitiu limar determinadas arestas do meu trabalho;

Ao professor Fábio Amaral, por me ter orientado e por me ter facultado os materiais necessários para que nada faltasse;

Ao professor Paulo Martinho, por me orientar para a área da minha vocação e na qual pretendo focar toda a minha aprendizagem. Teria sido muito diferente sem a sua ajuda.

De igual modo, agradeço aos restantes formadores, Sílvia Gouveia, Anabela Ferreira, Cecília Dowling, Pedro Fonseca, Duarte Cota, Pedro Rosa, André Rodrigues.

Agradeço em especial aos meus pais, António e Délia Sousa, por todo o apoio recebido e por me proporcionarem as melhores condições de trabalho.

"If you really want to do something, you'll find a way. If you don't, you'll find an excuse."

- Jim Rohn

Resumo:

O projeto do meu trabalho final, na ENTA, despertou em mim um interesse adormecido pela área da programação de componentes eletrónicos com microcontroladores. Iniciei este projeto com o objetivo de melhorar, em termos de transmissão de dados, uma das placas de INPUT da GlobeStar Systems, que demonstrou ser também uma alternativa open source económica, sendo bem-vinda face às situações de crise dos dias de hoje. Enquanto desenvolvi uma placa que substituísse a placa Input da GlobeStar Systems que utiliza um PIC, este projeto proporcionou-me um ponto de vista mais amplo sobre a programação em C++ com microcontroladores Atmega.

Desde princípios de Agosto, dei início ao meu trabalho, tendo como primeira etapa sentir o ambiente da GlobeStar Systems, frequentando o espaço, de modo a tirar quaisquer dúvidas sobre ligação entre servidor cliente, tanto com o pessoal da empresa, bem como com o meu orientador. Utilizei a placa Input, o meu arduino mega+ Ethernet shield, cabo RJ45 como materiais e dei início às pesquisas sobre servidores e clientes em geral, salvando os resultados todos em blocos de notas, documentados após testes. Pesquisei sobre o mesmo tema com arduinos e comparei os princípios encontrados, focando-me mais na base TELNET CHAT SERVER do arduino pela curiosidade que me despertou.

Concluída esta parte, continuei a pesquisar os componentes na placa INPUT, com os dados fornecidos pela GlobeStar Systems e também pesquisei os sensores que ia usar mais futuramente para efeitos demonstrativos. De acordo com a minha pesquisa, a ligação ao servidor com a placa arduino foi um sucesso. Em paralelo, os testes individuais dos sensores-botões correram como esperado e foram documentados para futura integração. Após comparar as diferenças entre as minhas tentativas com os trabalhos base, pude também observar vários métodos de tornar a minha placa mais autónoma (e prática), de forma a que ao ser conetada à rede ficasse constantemente a fazer “reset” a si mesma até se ligar ao servidor.

Concluindo, ficou verificado que é possível estabelecer uma ligação ao programa Connexall da GlobeStar, atravez do Standard Input Clien,t usando a interface TCP Client, utilizando um cabo RJ45 conectado do Arduino Ethernet Shield ao modem.

Em simultâneo, descobri durante as pesquisas outras maneiras de simplificar o meu projeto, de forma a tornar a placa mais "amigável" para o utilizador ao implementar a função "auto ip", que atribui um IP automaticamente à placa e ficando o mesmo visível para o utilizador, bem como a função "Auto reset Ethernet" em que basta ligar a placa e colocar o cabo de seguida. Aprendi imenso com este projeto, que servirá de base para outros projetos, na medida em que posso fazer ligações entre servidores e enviar os dados que mais forem convenientes.

Palavras-Chave

Económico; RJ45; Open Source; Microcontroladores; Placa Input; Placa Arduino; Arduino Ethernet Shield; "Auto IP; "Auto reset Ethernet" "Amigável ao utilizador";

ÍNDICE Pág.

INTRODUÇÃO

A presente Prova de Aptidão Profissional - PAP - integrada no Curso de Técnico de Eletrónica e Telecomunicações na Escola de Novas Tecnologias dos Açores foi me sugerida para demonstrar as minhas capacidades a partir dos conhecimentos adquiridos ao longo dos 3 anos do curso, permitindo-me concluir com aproveitamento a formação profissional. Dediquei-me à projeção e utilização de microcontroladores para as ligações cliente-servidor, servidor-cliente, melhorando as capacidades atuais de processamento, transmissão e custo monetário.

Este projeto foi sugerido pelo formador Fábio Amaral, no sentido de me aproximar mais da programação devido ao meu interesse nesta área, o que me beneficiou como pessoa e como estudante.

Observei que a placa Input da Globestar Systems utiliza basicamente um microcontrolador das séries PIC, imensos optocopuladores para a receção de dados e por fim a parte mais interessante, a velha entrada I/O RS232. Constatei que estava ao meu alcance, como formando, o desafio de atualizar esta placa com um Arduino Mega 2560 e um adaptador Ethernet SHIELD, de modo a que as comunicações usufruissem dos benefícios da rede local via RJ45, que atualmente se encontra na maioria dos locais de trabalho.

FASE 1

1. DECRIÇÃO DO PROJETO / Funcionamento geral

Decidi dedicar-me à projeção e programação de uma placa Cliente que estabelecesse ligação a servidores, neste caso a um servidor da GlobeStar Systems, que já tem a sua placa atual de input, baseada num PIC 17C752, com uma série de Optocopuladores em circuito com a função de detectar sinais emitidos pelos aparelhos.

Apliquei os meus métodos, utilizando por base a placa Arduino mega 2560, de maneira que este funcionasse de igual modo com a receção de dados em relação à placa baseada no PIC 17C752, com o extra de os dados serem transmitidos via internet.

Este projeto foi motivado por uma conversa com o meu diretor de turma, professor Paulo Martinho, que ao notar o meu interesse pela programação não hesitou em direcionar-me para o Professor Fábio Amaral, que por sua vez orientou-me no esqueleto da ideia. A partir de então comecei a ganhar mais interesse pela minha área de estudo.

1. Programação – Ligação do arduíno ao servidor

Comecei por me organizar com rascunhos do compilador arduino em cada pasta, com um bloco de notas a acompanhar. Efetuei as minhas pesquisas sobre cada sensor adquirido, e correspondendo a cada sensor, guardei um programa que me servisse de exemplo para ver o sensor a trabalhar individualmente, permitindo-me testar o funcionamento ao alterar o código e introduzir as minhas variáveis ajustadas.

Primeiramente, concentrei-me somente em estabelecer uma ligação ao servidor, tomei por base o servidor TELNET nos exemplos do arduino. Uma vez funcionando, organizei o código, ficando cada sensor individualmente na sua função, para ser mais fácil detetar erros e facilitar o acesso e manutenção de variáveis.

Tendo por base somente o arduino mega e o Ethernet Shield, defeni as bibliotecas que ia usar, seguido do respetivo mac address, o IP do servidor a ligar e por fim para evitar introduzir o IP manualmente(pelo menos do SHIELD). Antes do setup, loop, chamei as livrarias necessárias "#include <Ethernet.h>, #include <SPI.h>" eAtribuí um mac address e também introduzi manualmente o IP do servidor de Input.

“Byte mac[] = {   0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };”

// IP do servidor SIC

char server[] = "192.168.1.69"; //Este tem tendência a mudar de rede para rede;

Criei no “void setup(){}” um conjunto de funções, entre elas o conjunto de instruções, condições e variáveis que denominei em comentário“Auto Ethernet”, que visa atribuir um IP à placa de rede, fazer a ligação automaticamente à rede, iniciando a ligação à internet por “Ethernet.begin(mac,Ethernet.localIP());”tornando a obtenção do IP mais acessível. “postData();”é chamada e indica o estado da ligação do arduino. Caso falhe a ligação em vez de “("Ligado")” obtemos “("Ligação falhou")” e automaticamente chama a função “resetEthernetShield();” que faz com que o programa fique repetidamente a tentar conectar-se ao servidor com os dados previamente introduzidos.

A partir daqui comentei o código a cores por ser conveniente destacar:

```
“#include <Ethernet.h>
#include <SPI.h>”

“Byte mac[] = {   0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };”

char server[] = "192.168.1.69"; //IP  do SIC

void setup()

{

  Serial.begin(115200); //Monitor de série é iniciado

//Auto Ethernet - iniciação da ligação à internet:
  if (Ethernet.begin(mac) == 0) {//Se o mac address não for estabelecido
    Serial.println("Verifique o cabo de RJ45");//notifica por m.série
// Não faz mais nada se o problema for físico. Aqui isto somente acontece quando o mac address entra em conflito, ou seja, quando não houver internet ou falha do cabo.
    resetEthernetShield();//Chama a função e repete a ligação;
        for(;;)
      ;  }
// Imprime o endereço de IP local atribuido à placa arduino
```

```
  Serial.println("IP adquirido");//m.série notifica
  Serial.println(Ethernet.localIP());//m.série imprime o IP
//Fim auto Ethernet*******************************

   Ethernet.begin(mac,Ethernet.localIP());Inicia a ligação à internet.
 delay(1000);//delay para estabilidade
Serial.println("Ligando...");//Fim auto Ethernet

  postData();// postData é chamada aqui para enviar dados sobre o estado da ligação;


void postData() {// postData, para enviar os dados sobre a ligação cliente-servidor


  if (client.connect(server, 22000)) {// Se a ligação for bem sucedida
    Serial.println("Ligado");//O m.série notifica

    client.println("Arduino ligado");//O cliente notifica

  }
  else {//Caso contrário se a ligação falhar
    Serial.println("Ligação falhou");//O m.série notifica
    Serial.println("Deslligando");//O m.série notifica
    resetEthernetShield();//Chama a função e repete a ligação;

//em suma notifica e chama a função reset. Isto funciona quando o servidor está desligado, o arduino insiste indefenidamente pela ligação IP que foi configurada pelo utilizador.
    //é ótimo para não termos que fazer reset nós mesmos em instalação, sendo mais prático.
  }
}
```

void resetEthernetShield() // Esta função no final do código é uma mera repetição dos parâmetros que se encontram no void setup(){}, sendo somente chamada em condicionais, caso a ligação à internet falhe.

```
{
  delay(100);//delay para estabilidade
  Serial.println("Fazendo reset à Ethernet");//m.série notifica
  client.stop();//cliente pára.
//Auto Ethernet
  // iniciação da ligação à internet:
  if (Ethernet.begin(mac) == 0) {//Se o mac address não for estabelecido
    Serial.println("Verifique o cabo de RJ45");//m.série sugere solução
// Não faz mais nada se o problema for físico
    for(;;)
      ;
  }
// Imprime o endereço de IP local atribuido à placa arduino
  Serial.println("IP adquirido");//O m.série notifica
Serial.println(Ethernet.localIP());//m.série imprime o IP
//Fim auto Ethernet

   Ethernet.begin(mac,Ethernet.localIP()); //Inicia a ligação à internet.
   delay(1000);//delay para estabilidade
Serial.println("Ligando..."); //m.série notifica
  delay(100);//delay para estabilidade
    postData();();// postData é chamada aqui para enviar dados sobre o estado da ligação;

}
```

1.1 – Integração de sensores

Antes do "void setup(){}", além dos parâmetros de ligação à internet para efeitos de demonstração de receção de dados ao servidor, declarei as variáveis onde seriam guardados valores como "long", constantes numéricas, "float" para aproximar os dados analógicos em valores mais contínuos e a declaração dos pinos dos sensores bem como declarações de estado iniciais, valores de tolerância. Abaixo, iremos encontrar a azul as respetivas declarações de botões, sensores devidamente comentadas a verde, por questões de conveniência e iremos encontrá-las ao longo do programa até ao final.

```
long ct=0; //guarda 4 bytes, servirá para contagens tendo em conta a
long ct2=0; //frequência do micro controlador

// variáveis para o botão e led

//estados dos leds
const int ledvermelho = 0; //Estado lógico inicial
const int ledverde = 1; //Estado lógico inicial

const int gasPin = A1; // Input para o sensor de gás(A0)
float GAS;//variável float para valores mais precisos
//variáveis dos pinos
const int buttonPin = 12; // pino do 1º botão
const int buttonPin2 = 9;// pino do 2º botão
const int buttonPin3 = 8;// pino do 3º botão
const int buttonPin4 = 7; //pino do 4º botão
const int ledPin =  10;
int led2 = 11; //led vermelho
int buttonState = 0; //Estado lógico inicial do 1º Botão
int buttonState2 =0; //Estado lógico inicial do 2º Botão
int buttonState3 =0; //Estado lógico inicial do 3º Botão
int buttonState4= 0; //Estado lógico inicial do 4º Botão
```

//**Variáveis do sensor de temperatura** (TMP36)

float temp; //variável float para valores mais precisos

int sensorPin = A0; //O PIN ANALÓGICO DO tmp36 Vout, tem uma resolução de 10mV por grau centígrado com limite de 500mV para permitir temperaturas negativas.

int tempMin = 22; // temperatura mínima

int tempMax = 35; //temperatura máxima

int fan = 6; // o pin onde o ventilador esta ligado

int fanSpeed = 0;//defenição da velocidade inicial do ventilador

int tolerancia = 6; //Tolerância em graus, para evitar o ligar e desligarconstante sempre que a temperatura máxima é atingida, sendo agora a tempMax = tempMin+tolerancia = 22+6 = 28 ºc

//Graças a isto, o ventilador só para de trabalhar quando a temperatura chega a 22 ºc

//Sem isto, o ventilador começava e parava constantemente entre valores próximos de 28 ºc para manter a temperaturaabaixo dos 28 ºc

int GASMAX =750;// defenição do valor máximo de gás permitido

int GASMIN =35;// defenição do valor mínimo de gás permitido

int TOLERANCIA=200;// para evitar ligar e desligar os ventiladores nos limites

int FAN =3;// ventilador de gás 1

int FAN2 =4;// ventilador de gás 2

int Led = 13 ; // define LED Interface

int SENSOR = 5 ; // Valor defenido para o sensor magnético

int val ; // defenição da variável val

int estado1 = 0;// Estado inicialdo sensor magnético

Nota: Decidi comentar(EXCLUIR) o LCD e o vetor de leds devido a problemas de interferência.

Contudo o contador Geiger faz a sua função principal, enviando as contações por minuto para o servidor.

//Geiger começa aqui

```
// livraria para o lcd
//#include <LiquidCrystal.h>

//Iniciação da livraria com os pinos
//LiquidCrystal lcd(3,4,5,6,7,8);


// Limites para a barra led
#define TH1 45 //Thresholds, são limites
#define TH2 95
#define TH3 200
#define TH4 400
#define TH5 600

// Conversão de CPM para Usv/h

#define CONV_FACTOR 0.00812

// Variáveis
//int ledArray [] = {10,11,12,13,9};
int geiger_input = 2;
long count = 0; //guarda 4 bytes, servirá para contagens tendo em conta a frequência do microcontrolador.
long countPerMinute = 0;
long timePrevious = 0;
long timePreviousMeassure = 0;
long time = 0;
long countPrevious = 0;
float radiationValue = 0.0;
// Fim Geiger
```

1.1.1 Void setup(){}

Para iniciar as variáveis, pinos, inciei-as como OUTPUT, INPUT tal como iniciei o monitor de série para ajudar a visualizara atividade enquanto tudo decorre.

```
// o led a ser ativado
  pinMode(ledPin, OUTPUT);// ler verde quando envia dados dos botões
   pinMode(led2, OUTPUT); //led vermelho quando não envia dados dos botões
  // o butão de este para premir
  pinMode(buttonPin, INPUT);

//Pinos do ventilador e sensor
 pinMode(fan, OUTPUT);//Pinos do ventilador
 pinMode(sensorPin, INPUT);// Pino declarado como input

   pinMode (Led, OUTPUT) ;    // defenição de LED como interface
pinMode (SENSOR, INPUT) ;  // defenição do sensor magnético como input

analogReference(EXTERNAL);//referência para valores analógicos estáveis

//Geiger
  pinMode(geiger_input, INPUT);
  digitalWrite(geiger_input,HIGH);
for (int i=0;i<5;i++){
//    pinMode(ledArray[i],OUTPUT); //As variáveis para o vetor em desuso
  }


  attachInterrupt(0,countPulse,FALLING); // pino interrupt 2, que facilita operações em
simutâneo sem interferências.
//Fim Geiger
}
}
```

1.2 – Organização dos sensores

Aqui decidi organizar no void loop(){} as funções de forma individual para ser mais fácil encontrar erros. Aqui no loop, o programa corre para verificar qual destas funções é chamada de forma repetida. Isto também impede o programa correr funções que desnecessáriamente usam processamento, quando estas não são chamadas, como o caso dos botões que só funcionam uma vez pressionados e no caso o sensor magnético.

```
void loop() {

carregarBotao1();//chama a função Botão1
  delay(1);       // delay para estabilidade
carregarBotao2();//chama a função Botão2
  delay(1);       // delay para estabilidade

carregarBotao3();//chama a função Botão3
  delay(1);       // delay para estabilidade

  carregarBotao4();//chama a função Botão4
  delay(1);       //delay para estabilidade

//os botões foram postos separadamente para melhor segurança e organização
ventilador1();//chama a função ventilador2, somente ativado pelo senso de temperatura
  delay(1); // delay para estabilidade

gas();//chama a função gas, ativa dois ventiladores ao atingir a tolerância de gás máxima
delay(1); // delay para estabilidade

doorsensor();//chama a função sensor magnético
delay(1); // delay para estabilidade

GEIGER(); //chama a função geiger
delay(1);  // delay para estabilidade

}
```

2.3 - As funções dos sensores

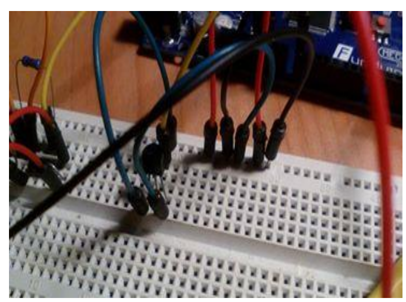
Aqui encontraremos os sensores que vão enviar os respetivos dados para o servidor a partir das funções estabelecidas, onde poderemos observar maioria das variáveis declaradas acima.

```
void carregarBotao1() {//Quando a função é chamada executa

if((debounceButton(buttonPin)==1) && (buttonState==0))//
{
  buttonState=1;//o uso de estados aqui serve para "bloquear" o botão no estado inicial, limitando o funcionamento a uma única mensagem por pressionamento.
Serial.println("Nurse Call");//m.série notifica
   client.println("Nurse Call");//cliente notifica
digitalWrite(ledPin, HIGH);//Acende led verde quando envia dados
    digitalWrite(led2, LOW);//Apaga led vermelho quando envia dados
//aqui teremos a menságem dependente do estado do botão pressionado
}

if((debounceButton(buttonPin)==0) && (buttonState==1))
{
  buttonState=0;
digitalWrite(ledPin, LOW);//Apaga led verde quando nada é enviado
digitalWrite(led2, HIGH);//Acende led vermelho quando nada é enviado
}  //

}
// Os seguintes três botões têm a mesma configuração e princípio, mudando somente a númeração para 2,3,4 respetivamente.
```

```
void carregarBotao2() {

if((debounceButton2(buttonPin2)==1) && (buttonState2==0))
{
  buttonState2=1;
   Serial.println("Next Patient");
client.println("Next Patient");
digitalWrite(ledPin, HIGH); //Acende led verde quando envia dados
    digitalWrite(led2, LOW); //Apaga led vermelho quando envia dados
//aqui teremos mais uma mensagem dependente do estado do botão pressionado
}

if((debounceButton2(buttonPin2)==0) && (buttonState2==1))
{
  buttonState2=0;
digitalWrite(ledPin, LOW); //Apaga led verde quando nada é enviado
    digitalWrite(led2, HIGH); //Acende led vermelho quando nada é enviado
}  //
}


void carregarBotao3() {
if((debounceButton3(buttonPin3)==1) && (buttonState3==0))
{
  buttonState3=1;
   Serial.println("X Ray room");
      client.println("X Ray room");
digitalWrite(ledPin, HIGH); //Acende led verde quando envia dados
    digitalWrite(led2, LOW); //Apaga led vermelho quando envia dados
 //aqui teremos mais outra mensagem dependente do estado do botão pressionado
}

if((debounceButton3(buttonPin3)==0) && (buttonState3==1))
{
```

```
  buttonState3=0;
digitalWrite(ledPin, LOW); //Apaga led verde quando nada é enviado
    digitalWrite(led2, HIGH); //Acende led vermelho quando nada é enviado
}
}

void carregarBotao4() {
if((debounceButton4(buttonPin4)==1) && (buttonState4==0))
{
  buttonState4=1;
   Serial.println("Maternity Ward");
client.println("Maternity Ward");
digitalWrite(ledPin, HIGH); //Acende led verde quando envia dados
    digitalWrite(led2, LOW); //Apaga led vermelho quando envia dados
 //e a menságem do último botão
}

if((debounceButton4(buttonPin4)==0) && (buttonState4==1))
{
  buttonState4=0;
digitalWrite(ledPin, LOW); //Apaga led verde quando nada é enviado
    digitalWrite(led2, HIGH); //Acende led vermelho quando nada é enviado
}  //
}
```

```
void ventilador1() {//função ventilador é executada

int reading = analogRead(sensorPin); //lê o sensor de temperature em volts

loat voltage = reading * 5.0;// Conversão da leitura para voltagem
voltage /= 1024.0;

// Conversão da voltage para temperatura
float temperatureC = (voltage - 0.5) * 100 ;  //cpnversão de 10 mv por grau com 500 mV de "offset"
//para graus ((voltagem - 500mV) vezes 100)

// se quisermos em fahrenheit
 float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
// Deixei fahrenheit comentado porque é pouco usado nestes lados da europa.
 if (ct==100)// Aqui temos o micro controlador a contar 100 vezes antes de ler e emitir o valor da voltagem e temperatura.
Útil se quisermos ter várias tarefaz a serem executadas pelo mesmo micro controlador, sem interrupções.
 {
   Serial.print(voltage); //monitor de série notifica o utilizador
   Serial.println(" volts");//monitor de série notifica o utilizador
   Serial.print(temperatureC);//monitor de série notifica o utilizador
   Serial.print(" Celsius ");//monitor de série notifica o utilizador
client.println(temperatureC) && client.print(" Celsius =");
// Cliente notifica servidor com dados da temperatura em Celsius
//Serial.print(temperatureF); Serial.println(" degrees F");
//Serial.println();

 ct=0;// Aqui o contador faz reset e volta a zero
 }
```



```
 delayMicroseconds(1); //espera de microsegundo
ct++;//Adiciona +1 para iniciar a contagem, contando de novo até 100 para efetuar a leutura e envio novamente.

if(temperatureC < tempMin) {     //se a temperatura for mais baixa que o valor mínimo+tolerancia
      fanSpeed = 0;      // O ventilador fica imóvel
      digitalWrite(fan, LOW);
  }
  if((temperatureC >= tempMin+tolerancia) && (temperatureC <= tempMax))   //se a temperatura for mais alta que o valor mínimo+tolerancia
  {
      fanSpeed = map(temp, tempMin, tempMax, 240, 255); // A velocidade atual do ventilador é máxima
      analogWrite(fan, fanSpeed);  //  Rodar o ventilador
// client.println("Ventilador ativo");
       Serial.print(" Ventilador ativo ");//m.série notifica
}
//
 }

void gas(){ //função gás é executada

//fan function
//getting the voltage reading from the temperature sensor
int reading2 = analogRead(gasPin);

     if (ct2=10000000) )// Aqui temos o micro controlador a contar 10000000 vezes antes de ler e emitir o valor da voltagem e temperatura.
Útil, mais uma vez se quisermos ter várias tarefaz a serem executadas pelo mesmo micro controlador, sem interrupções entre si.

{
```

```
Serial.println(analogRead(gasPin));
  Serial.print("GAS ");
ct2=0;// Aqui o contador faz reset e volta a zero
 }
 delayMicroseconds(1);
 ct2++;//Adiciona +1 para iniciar a contagem, contando de novo até 10000000 para efetuar a leutura e envio novamente.

if(reading2 < GASMIN) {      // se o valor do gás for mais baixo que o valor mínimo+tolerancia
fanSpeed = 0;      // ventiladores ficam imóveis
     digitalWrite(FAN, LOW);
     digitalWrite(FAN2, LOW);
}
  if((reading2 >= GASMIN+TOLERANCIA) && (reading2 <= GASMAX))  //se o valor do gás for mais alto que o valor mínimo+tolerancia
  {
     fanSpeed = map(GAS, GASMIN, GASMAX, 240, 255); // A velocidade atual dos ventiladors é máxima
     analogWrite(FAN, fanSpeed);  //Rodar o ventilador
      analogWrite(FAN2, fanSpeed);  //Rodar o ventilador
      client.println("LEAK ALERT");//cliente notifica o servidor
//  Serial.print(" Ventilador ativo ");
  }
   }
```

```
void doorsensor(){//função sensor magnético é executada
      val = digitalRead (SENSOR) ; // leitura do sensor

if((val==1) && (estado1==0))
  {
   estado1=1;
     client.println("Door Open");//cliente notifica
     Serial.print(" Door Open ");// m.série notifica
}
  if((digitalRead(SENSOR)==0) && (estado1==1))
  {;//o uso de estados aqui serve para "bloquear" o botão no estado inicial, limitando o funcionamento a uma única mensagem por pressionamento. É o mesmo princípio usado nos botões, por detetar ou não um campo magnético.

estado1=0;
     client.println("Door Closed");//Cliente notifica
     Serial.print(" Door Closed");//m.série notifica
}
      }
```

//aqui estão as funções debounce que serão chamadas para evitar contagens erradas devido às oscilações metálicas dos botões, pois sendo constituídos por partes que oscilam, um botão apesar de bloqueado pode enviar duas mensagens,embora do ponto de vista virtual só devesse enviar uma.

DebounceButton, limita o tempo de contagem do microprocessador, para que este apenas tenha tempo de ouvir a primeira oscilação de todas, reduzindo a probabilidade de ler mais do que uma oscilação ao premir um botão, para próximo de zero.

```
boolean debounceButton(boolean state)
{
  boolean stateNow = digitalRead(buttonPin);
  if(state!=stateNow)
  {
    delay(10);
    stateNow = digitalRead(buttonPin);
}
  return stateNow;//Devolve o estado

}
```

// As seguintes três funções debounce têm a mesma configuração e princípio, mudando somente a númeração para 2,3,4 respetivamente.

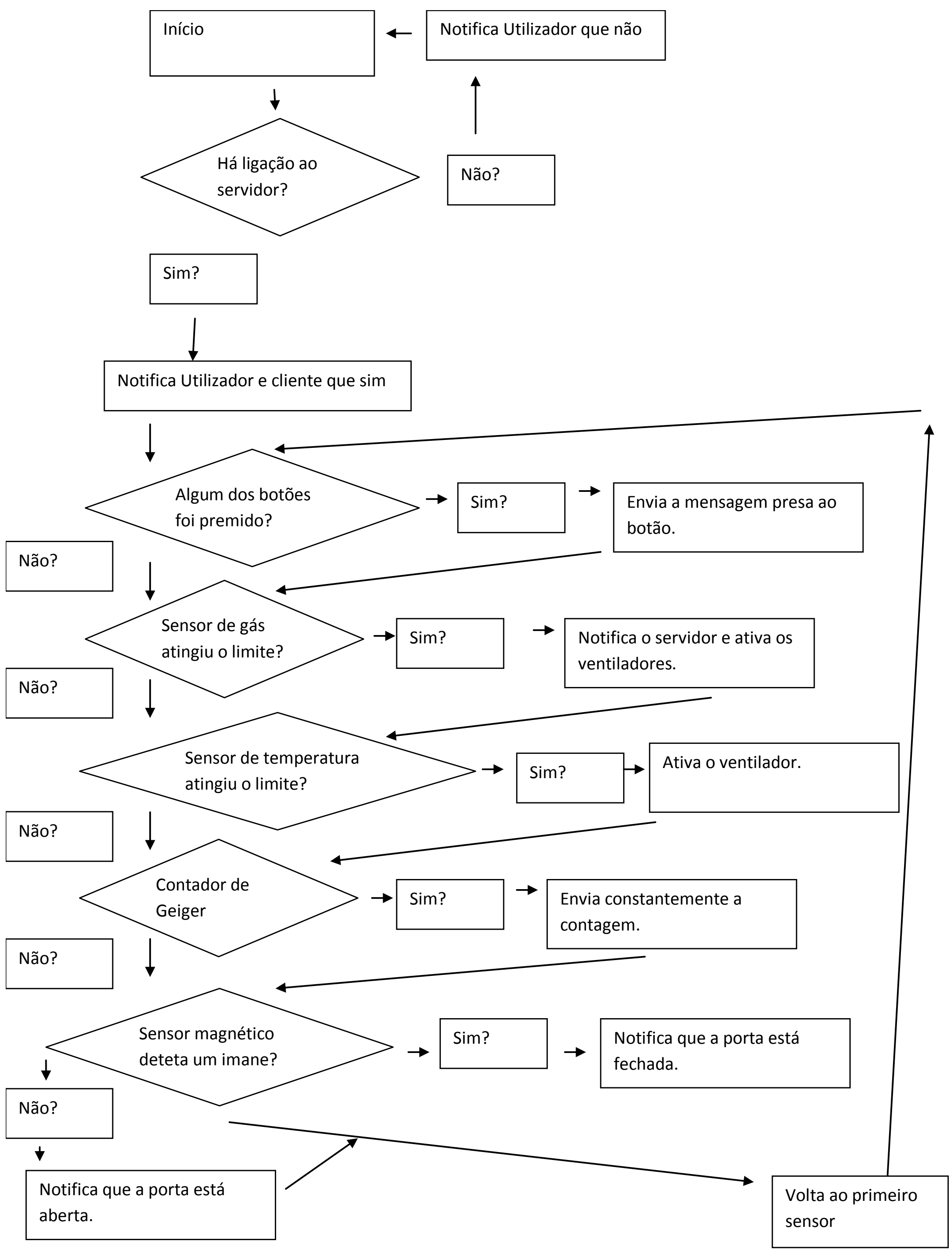
```
boolean debounceButton2(boolean state)
{
  boolean stateNow = digitalRead(buttonPin2);
  if(state!=stateNow)
  {
    delay(10);
    stateNow = digitalRead(buttonPin2);
  }
  return stateNow;
```

```
}

boolean debounceButton3(boolean state)
{
  boolean stateNow = digitalRead(buttonPin3);
  if(state!=stateNow)
  {
    delay(10);
    stateNow = digitalRead(buttonPin3);
}
  return stateNow;

}

boolean debounceButton4(boolean state)
{
  boolean stateNow = digitalRead(buttonPin4);
  if(state!=stateNow)
  {
    delay(10);
    stateNow = digitalRead(buttonPin4);
}
  return stateNow;

}
```

```
void GEIGER(){//função Geiger é exectada
   if (millis()-timePreviousMeassure > 10000){//Aqui com a estrutura millis() o micro controlador devolve o número de milisegundos desde que oo arduino começou a corer o programa. O número volta a zero após 48 dias.

   countPerMinute = 6*count;//contagens por minute em conversão
radiationValue = countPerMinute * CONV_FACTOR;
   timePreviousMeassure = millis();
Serial.print("cpm = "); //Notificção em m.série
   Serial.print(countPerMinute,DEC); //Notificção em m.série
   Serial.print(" - ");//Notificção em m.série
   Serial.print("uSv/h = ");//Notificção em m.série
   Serial.println(radiationValue,4); //Notificção em m.série
      client.println("        cpm = "); //Notificção pelo cliente
   client.println(countPerMinute,DEC); //Notificção pelo cliente

//como referi acima, o LCD encontra-se em desuso devido a conflitos com o resto to programa
//lcd.clear();
  //lcd.setCursor(0, 0);
  //lcd.print("CPM=");
  //lcd.setCursor(4,0);
  //lcd.print(countPerMinute);  //  lcd.setCursor(0,1);
  //lcd.print(radiationValue,4);
  //lcd.setCursor(6,1);
//lcd.print(" uSv/h");

// O vetor também se encontra em desuso devido a conflitos com o programa
//led var setting
   if(countPerMinute <= TH1) ledVar(0);
   if((countPerMinute <= TH2)&&(countPerMinute>TH1)) ledVar(1);
   if((countPerMinute <= TH3)&&(countPerMinute>TH2)) ledVar(2);
   if((countPerMinute <= TH4)&&(countPerMinute>TH3)) ledVar(3);
```

```
    if((countPerMinute <= TH5)&&(countPerMinute>TH4)) ledVar(4);
    if(countPerMinute>TH5) ledVar(5);

    count = 0;

}
    }

//esta função lida com o pino attachIterrupt, complementando a função GEIGER
void countPulse(){
  detachInterrupt(0);
  count++;
  while(digitalRead(2)==0){
}
  attachInterrupt(0,countPulse,FALLING);
}
// Esta função bem podia estar toda comentada, pois desativei os leds como mencionei
previamente, mas tenciono pô-los a funcionar em breve, por isso apenas comentei o
essencial para permitir que o programa corra sem interferências.
void ledVar(int value){
  if (value > 0){
    for(int i=0;i<=value;i++){
//      digitalWrite(ledArray[i],HIGH);
    }
    for(int i=5;i>value;i--){
   //   digitalWrite(ledArray[i],LOW);
}
  }
  else {
    for(int i=5;i>=0;i--){
//   digitalWrite(ledArray[i],LOW);
}
  }
}
```

FLUXOGRAMA – REPRESENTAÇÃO DA TOTALIDADE DO CÓDIGO



Decidi alimentar o projeto em questão com uma fonte de computador, excluindo o sensor de temperatura por questões técnicas.

Botões de pressão:

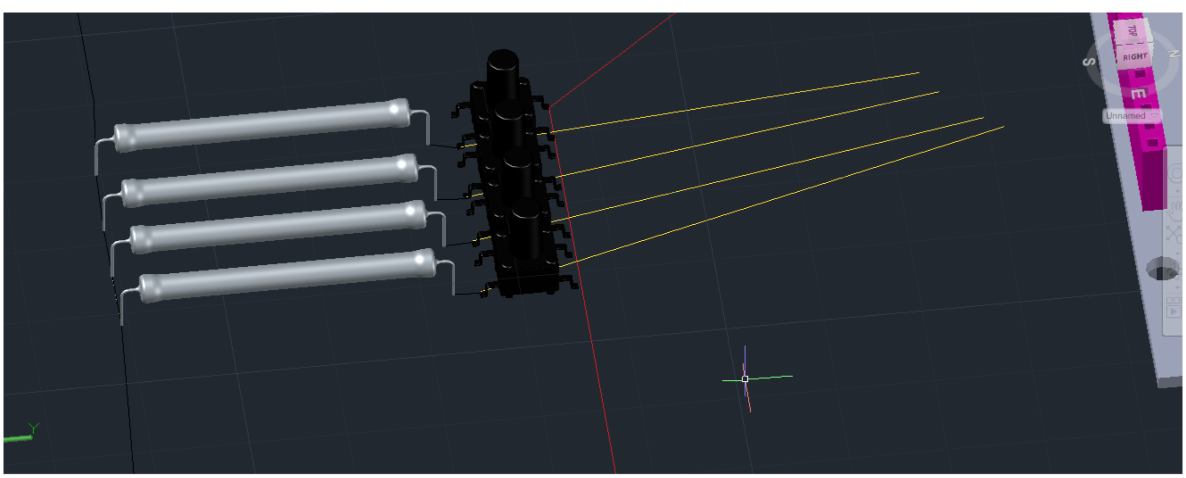
Estes em si requerem uma ligação à corrente protegida por uma resistência de 1k, outra à massa e outra a uma porta lógica, que irá detetar a passagem de corrente à massa ao premir o botão.

Os ventiladotres:

Requerem um transístor2n222 para controlar a frequência das rotações por portas que suportam PWM, protegidas por uma resistência na base e um diódo ligado ao coletor para evitar s inversões de carga das espiras existentes nos ventiladores.

O sensor de temperatura, teve de ser ligado únicamente aos 5V e massa analógica do arduíno usando a porta AREF, de modo a manter os valores de temperatura mais estáveis usando 5V como tensão de referência.

Todos os restantes sensores, gás, contador geiger,sensor magnético, puderam ser ligados diretamente à massa e corrente da fonte de alimentação, devido ao facto de já estarem embutidos nos seus próprios circuitos integrados, devidamente preparados para 5v.



FASE 2

PLANEAMENTO DO PROJETO

Planeei construir uma maquete em esferovite para representar o funcionamento do meu Projeto baseado na placa Arduino, numa situação problema que visa representar um local de trabalho hospitalar/laboratorial.

Recursos materiais:

Arduino Mega 2560;
Ethernet Shield;
Fonte de alimentação 300 W aproveitadapara alimentar o projeto;
Cabos fêmea de pc aproveitados;
Três placas de esferovite prensada(por dimensões aqui(Comprimento*Largura*Altura)) para simular a infra-estrutura 3D;
Três ventiladores de pc aproveitadas;
X fios jumper para ligações;
Cabo RJ45;
Cabo USB-RS232;
Sensor de Temperatura;
Sensor de GÁS;
Sensor de campo magético;
Contador de Geiger;
Quatro botões;
Três transistores 2n222;
Três díodos IN4007;
Quatro resistências de 1K Ohm;
Uma breadboard;
Cartolina cinzenta;

Cortei a esferovite para servir de base com 100cm x 60cm, tendo as paredes por sua vez 14cm de altura.

Dividi a secção num total de seis salas, quatro de um lado, duas do outro, separadas entre si por paredes com 4cm de espessura e divididas por um corredor ao meio, perfazendo um rectângulo. Criei pelo corredor afora entradas, de modo,a esconder os cabos de forma subterrânea, perfurei uma das paredes para embutir dois dos ventiladores numa sala onde se encontra o sensor de gás.

Na segunda sala, coloquei um sensor magnético, que utilizei para nos indicar o estado da porta de entrada entre "Aberto" e Fechado", correspondendo estes aos valores lógico "1" e "0" respetivamente.

Na terceira sala, coloquei o contador de Geiger, passando unicamente o "tubo Geiger" para o centro da sala.

A sala principal, onde se encontra a fonte de alimentação, Arduino, Ethernet Shield, breadboard, quatro botões de pressão, sensor de temperatura, acabou por ocupar duas salas devido ao tamanho da fonte em si, o espaço para dar folga aos cabos e prática manual. Nesta Sala, o sensor de temperatura está encaixado no microchip do EthernetShield, sendo este arrefecido pelo terceiro ventilador aos 26ºc. Os botões de pressão simulam os callpoints na vida real do programa Connexall da Globestar Systems.

Chegando aqui, tendo improvisado o espaço e usando duas salas a mais, para a sala principal, acabamos na mesma por ficar com duas salas disponíveis para a PARTE II, para uma extensão futura deste projeto igualmente com demonstração, denominada PARTE II – Output.

FASE 3

REFLEXÃO CRÍTICA

Tendo finalizada a PAP, concluí ter atingido os objetivos assim que foi estabelecida uma ligação ao servidor por parte do meu cliente arduíno, sendo a proposta inicial escrita deste projeto uma solução mais rápida e económica para placa de Input de dados.

Da minha perspetiva, fazendo baixar o custo das placas Input e tirando proveito das redes locais para transferência de dados, a GlobeStar Systems ficaria a ganhar tanto na eficácia como em questões financeiras.

Na situação problema, ao lidar com sensores e envios de dados e sinal, pude constatar que a placa tem agilidade para uma variedade de situações em que é necessário lidar com valores analógicos, além dos valores digitais. A parte mais desafiante deste projeto revelou-se na situação problem, ao introduzir o Contador Geiger, que sendo um Tubo metálico com a sua placa específica para seu funcionamento mais complexa na programação, causou algumas interferências durante os testes. Mencionando testes, durante as pesquisas encontrei o Notepad++, que facilitou a comparação de diferentes rascunhos de código para ver as diferenças previamente editadas a cores, permitindo-me trabalhar mais depressa. Entre pesquisas, adquiri o hábito de salvar constantemente o rascunho atual de trabalho, numa pasta de forma organizada, comprimir e enviar para várias drives para garantir a segurança, após um acidente.

Em suma, este trabalho acabou por se tornar numa realização para a minha pessoa, ao constatar que fui além dos objetivos atingidos após inúmeros testes e pesquisas feitas, representando uma boa experiência para a minha formação e para os meus futuros estudos\profissão.

Agora sinto-me mais perto da área a que aspiro dedicar as minhas horas laborais, podendo aplicar estes conhecimentos numa diversidade de situações.

NETGRAFIA

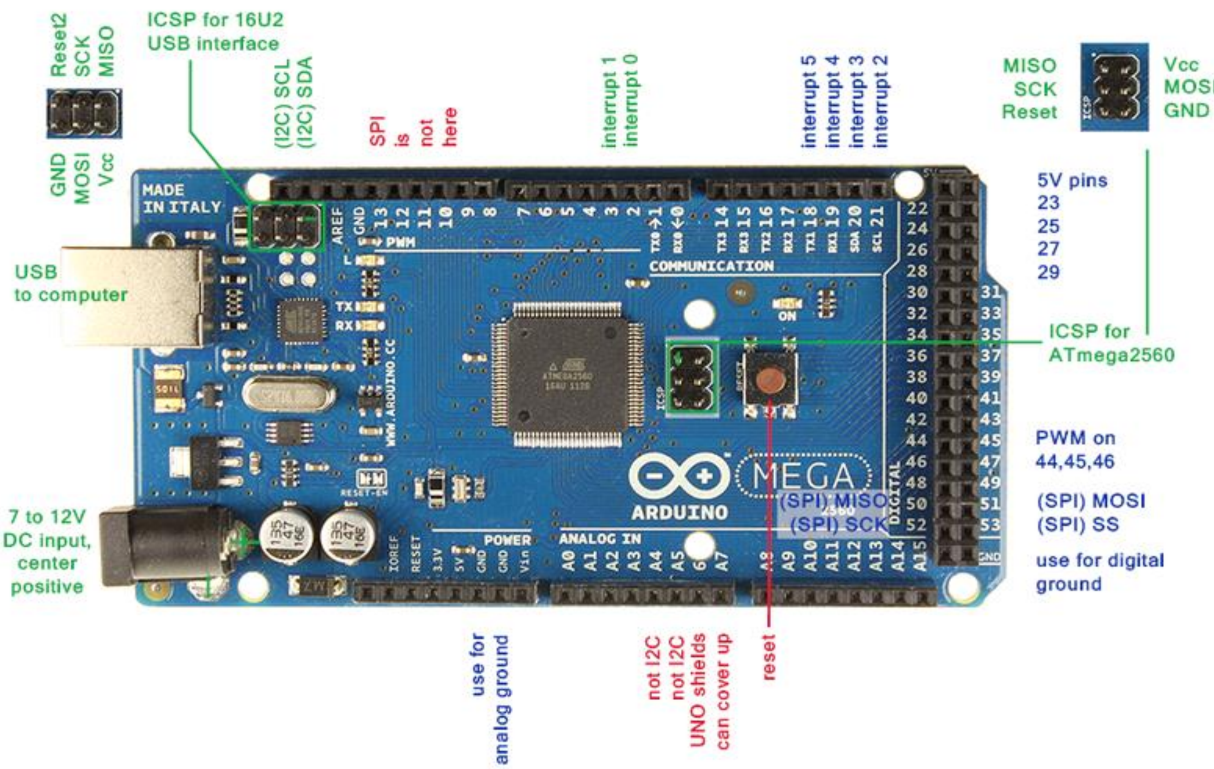
https://www.arduino.cc/en/Reference/HomePage

http://www.instructables.com/

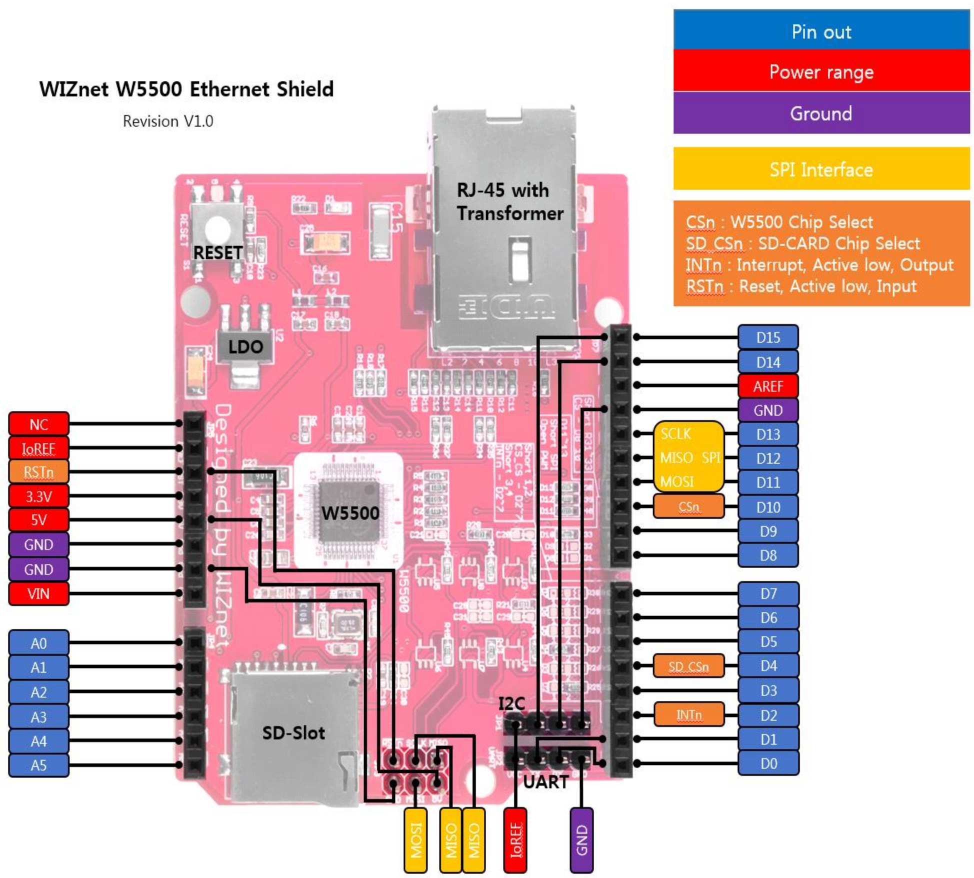
ANEXO I

Esquema das placas Arduino

Na Figura 1 podemos observar a A. Mega 2560, com todas as entradas descriminadas.



ANEXO II



•

https://www.arduino.cc/en/Main/ArduinoBoardEthernet

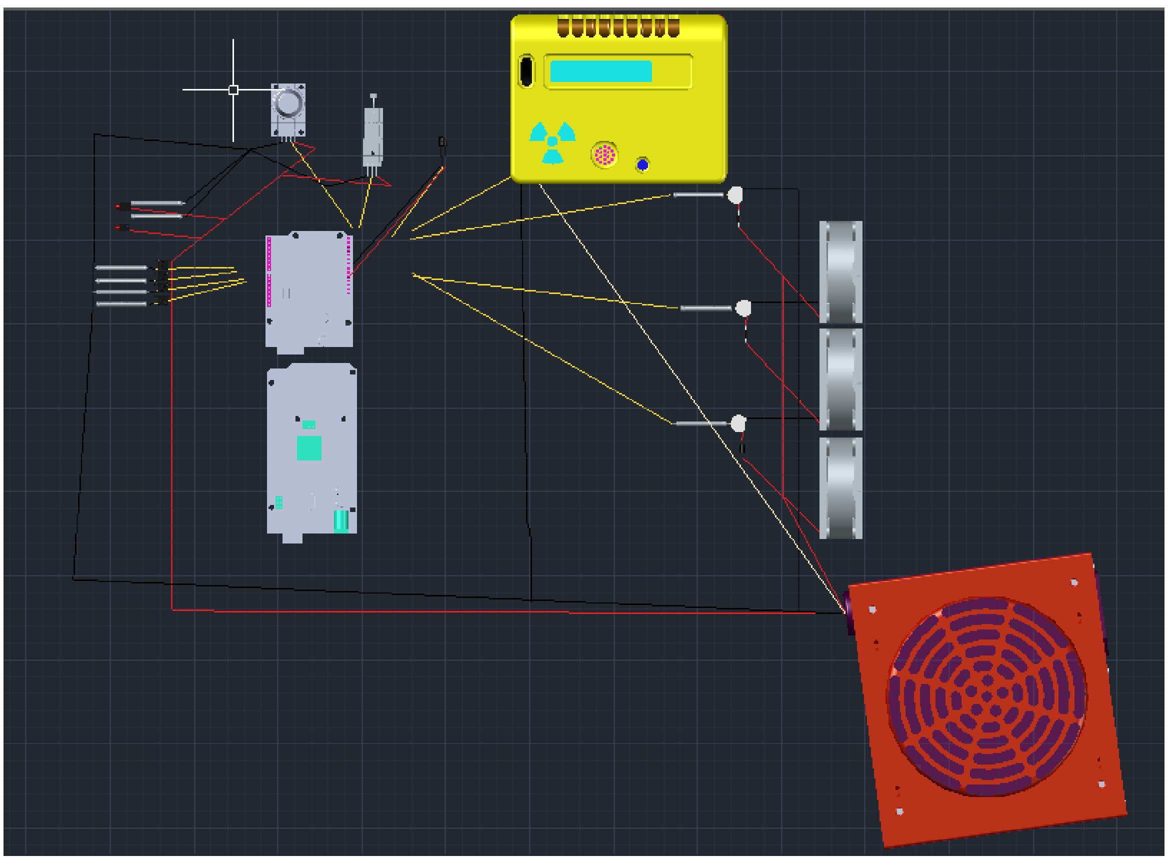
ANEXO III

Contador Geiger:

https://www.cooking-hacks.com/documentation/tutorials/geiger-counter-radiation-sensor-board-arduino-raspberry-pi-tutorial/

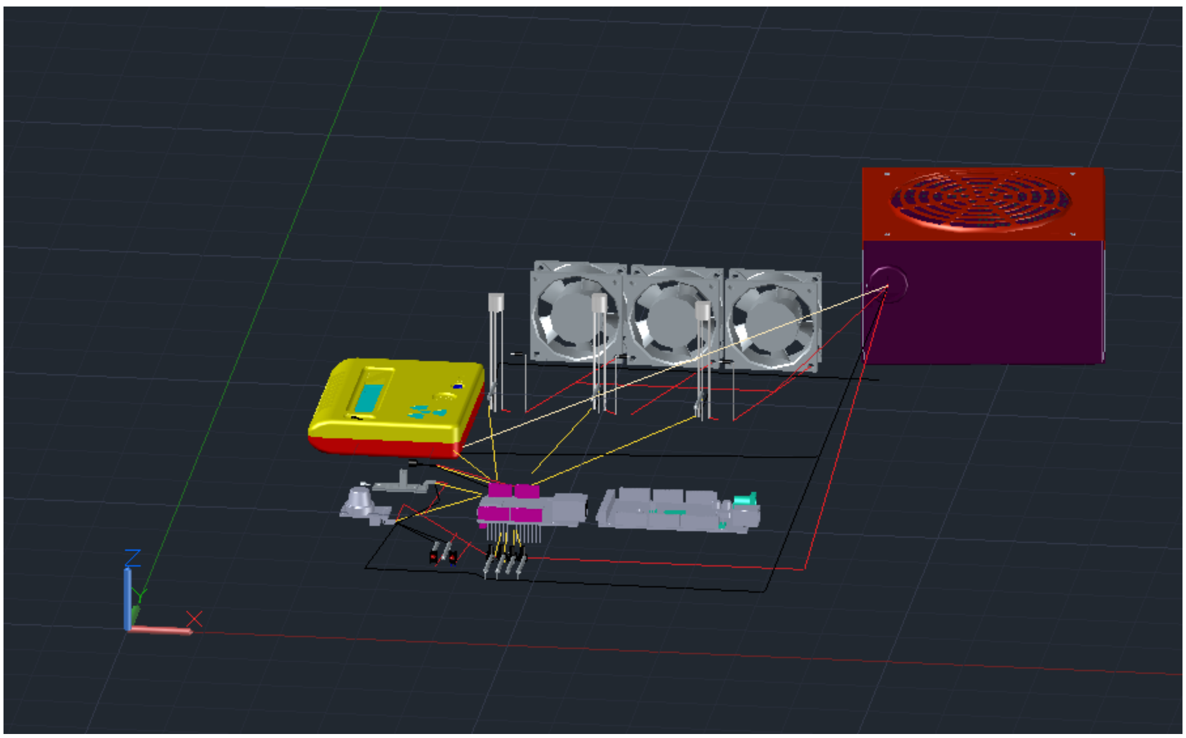
ANEXO IV

Projecto visto de cima



ANEXO V

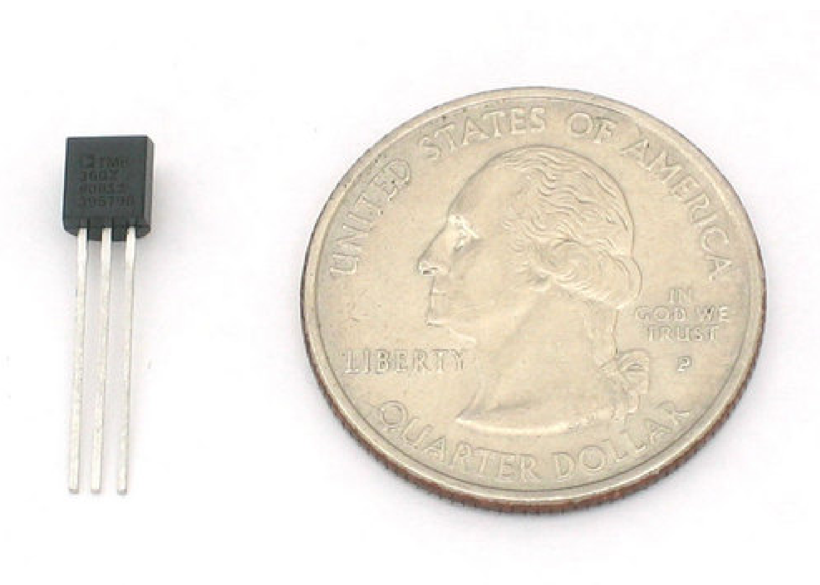
Projecto visto de outra perspectiva



adafruit learning system

TMP36 Temperature Sensor

Created by Ladyada

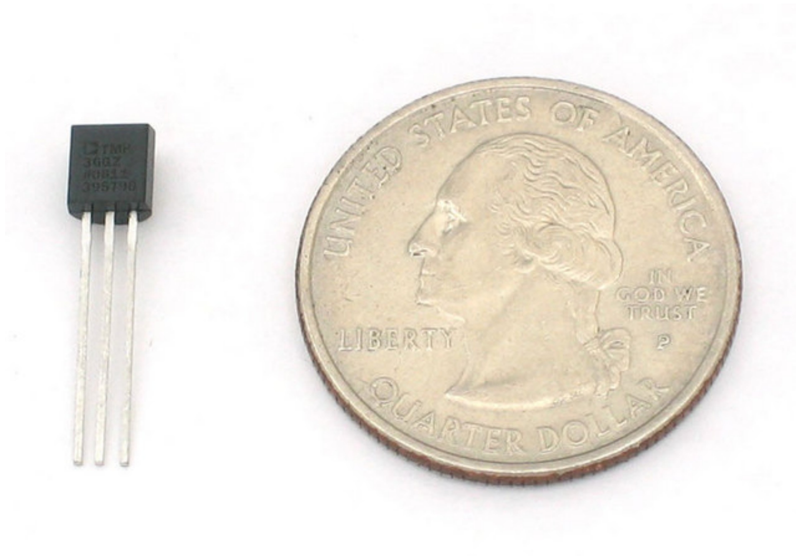


Last updated on 2013-09-11 02:16:00 PM EDT

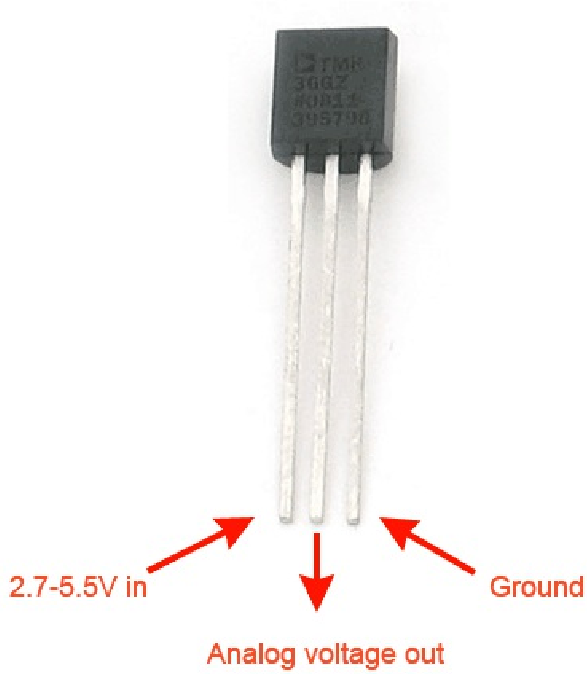
Guide Contents

Overview

An analog temperature sensor is pretty easy to explain, its a chip that tells you what the ambient temperature is!



These sensors use a solid-state technique to determine the temperature. That is to say, they don't use mercury (like old thermometers), bimetalic strips (http://adafru.it/aKJ) (like in some home thermometers or stoves), nor do they use thermistors (http://adafru.it/aK6)(temperature sensitive resistors). Instead, they use the fact as temperature increases, the voltage across a diode increases at a known rate. (Technically, this is actually the voltage drop between the base and emitter - the Vbe - of a transistor.) By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature. There have been some improvements on the technique but, essentially that is how temperature is measured.



Because these sensors have no moving parts, they are precise, never wear out, don't need calibration, work under many environmental conditions, and are consistant between sensors and readings. Moreover they are very inexpensive and quite easy to use.

Some Basic Stats

These stats are for the temperature sensor in the Adafruit shop, the Analog Devices TMP36 (http://adafru.it/clW) (-40 to 150C). Its very similar to the LM35/TMP35 (Celsius output) and LM34/TMP34 (Farenheit output). The reason we went with the '36 instead of the '35 or '34 is that this sensor has a very wide range and doesn't require a negative voltage to read sub-zero temperatures. Otherwise, the functionality is basically the same.

- **Size:** TO-92 package (about 0.2" x 0.2" x 0.2") with three leads
- **Price:** $2.00 at the Adafruit shop (http://adafru.it/alH)
- **Temperature range:** -40°C to 150°C / -40°F to 302°F
- **Output range:** 0.1V (-40°C) to 2.0V (150°C) but accuracy decreases after 125°C
- **Power supply:** 2.7V to 5.5V only, 0.05 mA current draw
- **Datasheet (http://adafru.it/clW)**

How to Measure Temperature

Using the TMP36 is easy, simply connect the left pin to power (2.7-5.5V) and the right pin to ground. Then the middle pin will have an analog voltage that is directly proportional (linear) to the temperature. The analog voltage is independant of the power supply.

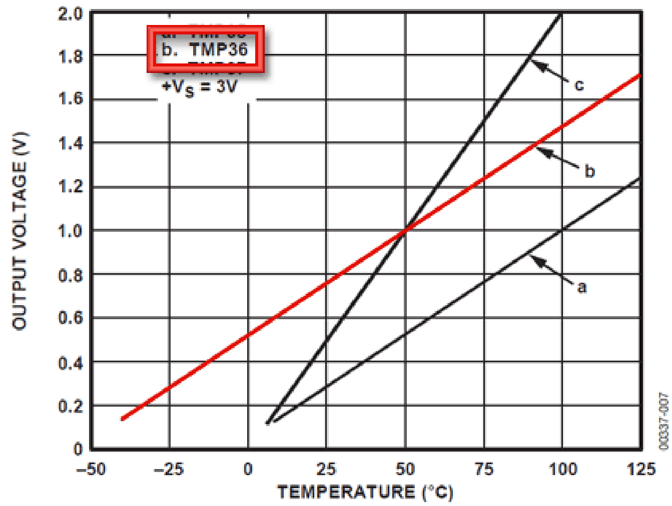


Figure 6. Output Voltage vs. Temperature

To convert the voltage to temperature, simply use the basic formula:

Temp in °C = [(Vout in mV) - 500] / 10

So for example, if the voltage out is 1V that means that the temperature is **((1000 mV - 500) / 10) = 50 °C**

If you're using a LM35 or similar, use line 'a' in the image above and the formula: **Temp in °C = (Vout in mV) / 10**

Problems you may encounter with multiple sensors:

If, when adding more sensors, you find that the temperature is inconsistant, this indicates that the sensors are interfering with each other when switching the analog reading circuit from one pin to the other. You can fix this by doing two delayed readings and tossing out the first one

[See this post for more information (http://adafru.it/aKL)](http://adafru.it/aKL)

Testing a Temp Sensor

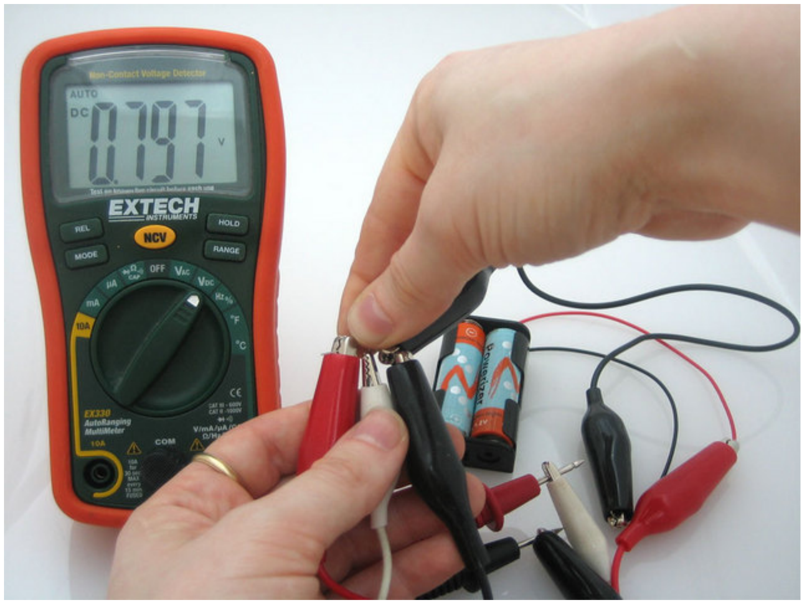
Testing these sensors is pretty easy but you'll need a battery pack or power supply.

Connect a 2.7-5.5V power supply (2-4 AA batteries work fantastic) so that ground is connected to pin 3 (right pin), and power is connected to pin 1 (left pin)

Then connect your multimeter in DC voltage mode to ground and the remaining pin 2 (middle). If you've got a TMP36 and its about room temperature (25°C), the voltage should be about 0.75V. Note that if you're using a LM35, the voltage will be 0.25V

The sensor is indicating that the temperature is 26.3°C also known as 79.3°F

You can change the voltage range by pressing the plastic case of the sensor with your fingers, you will see the temperature/voltage rise.



With my fingers on the sensor, heating it up a little, the temperature reading is now 29.7°C / 85.5°F

Or you can touch the sensor with an ice cube, perferrably in a plastic bag so it doesn't get water on your circuit, and see the temperature/voltage drop.

I pressed an ice-cube against the sensor, to bring the temperature down to 18.6°C / 65.5°F

Using a Temp Sensor

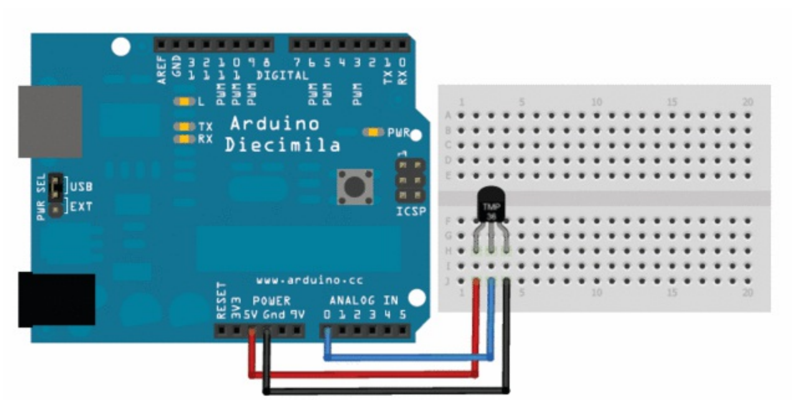
Connecting to a Temperature Sensor

These sensors have little chips in them and while they're not that delicate, they do need to be handled properly. Be careful of static electricity when handling them and make sure the power supply is connected up correctly and is between 2.7 and 5.5V DC - so don't try to use a 9V battery!

They come in a "TO-92" package which means the chip is housed in a plastic hemi-cylinder with three legs. The legs can be bent easily to allow the sensor to be plugged into a breadboard. You can also solder to the pins to connect long wires. If you need to waterproof the sensor, you can see below for an Instructable for how to make an excellent case.

Reading the Analog Temperature Data

Unlike the FSR or photocell sensors we have looked at, the TMP36 and friends doesn't act like a resistor. Because of that, there is really only one way to read the temperature value from the sensor, and that is plugging the output pin directly into an Analog (ADC) input.



Remember that you can use anywhere between 2.7V and 5.5V as the power supply. For this example I'm showing it with a 5V supply but note that you can use this with a 3.3v supply just as easily. No matter what supply you use, the analog voltage reading will range from about 0V (ground) to about 1.75V.

If you're using a 5V Arduino, and connecting the sensor directly into an Analog pin, you can use these formulas to turn the 10-bit analog reading into a temperature:

Voltage at pin in milliVolts = (*reading from ADC*) * (5000/1024)
This formula converts the number 0-1023 from the ADC into 0-5000mV (= 5V)

If you're using a 3.3V Arduino, you'll want to use this:

Voltage at pin in milliVolts = (*reading from ADC*) * (3300/1024)
This formula converts the number 0-1023 from the ADC into 0-3300mV (= 3.3V)

Then, to convert millivolts into temperature, use this formula:

Centigrade temperature = [(analog voltage in mV) - 500] / 10

Simple Thermometer

This example code for Arduino shows a quick way to create a temperature sensor, it simply prints to the serial port what the current temperature is in both Celsius and Fahrenheit.

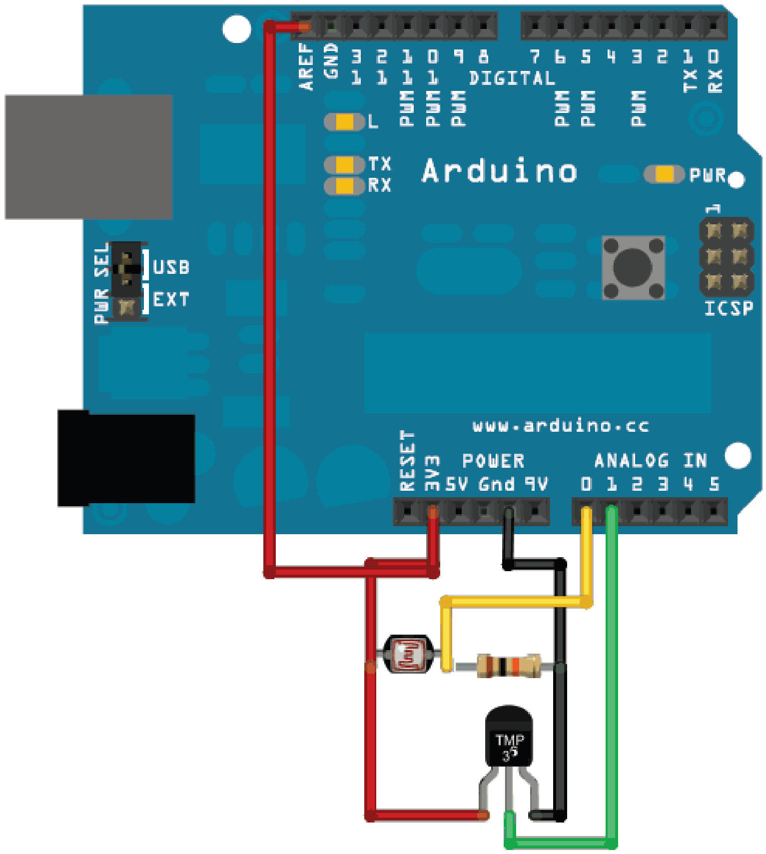
```
//TMP36 Pin Variables
int sensorPin = 0; //the analog pin the TMP36's Vout (sense) pin is connected to
                        //the resolution is 10 mV / degree centigrade with a
                        //500 mV offset to allow for negative temperatures

/*
 * setup() - this function runs once when you turn your Arduino on
 * We initialize the serial connection with the computer
 */
void setup()
{
  Serial.begin(9600);  //Start the serial connection with the computer
                       //to view the result open the serial monitor
}

void loop()                     // run over and over again
{
 //getting the voltage reading from the temperature sensor
 int reading = analogRead(sensorPin);

 // converting that reading to voltage, for 3.3v arduino use 3.3
 float voltage = reading * 5.0;
 voltage /= 1024.0;

 // print out the voltage
 Serial.print(voltage); Serial.println(" volts");

 // now print out the temperature
 float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per degree wit 500 mV offse
                                               //to degrees ((voltage - 500mV) times 100)
 Serial.print(temperatureC); Serial.println(" degrees C");

 // now convert to Fahrenheit
 float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
 Serial.print(temperatureF); Serial.println(" degrees F");

 delay(1000);                                     //waiting a second
}
```

Getting Better Precision

For better results, using the 3.3v reference voltage as ARef instead of the 5V will be more precise and less noisy

This example from the light&temp datalogging tutorial has a photocell but you can ignore it

Note we've changed the TMP36 to A1



> To use the 3.3v pin as your analog reference, don't forget to specify "analogReference(EXTERNAL)" in your setup as in the code below:

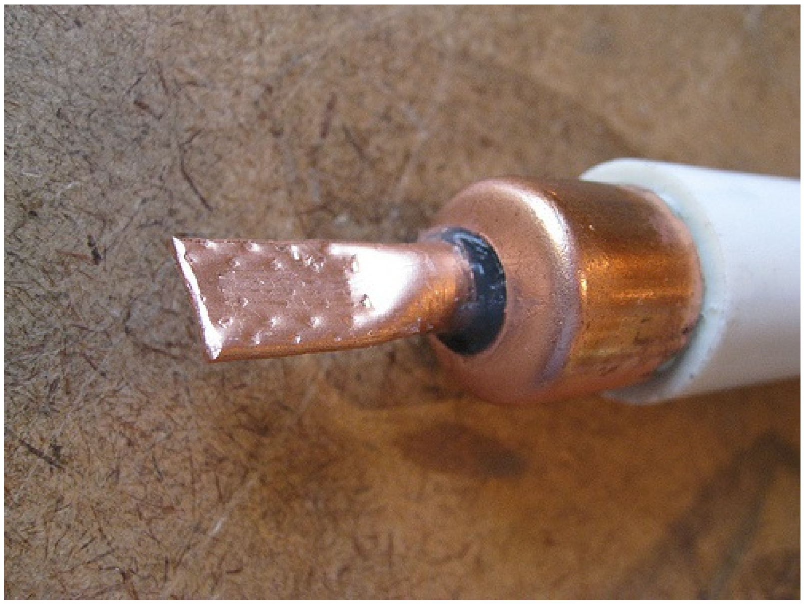
```
/* Sensor test sketch
  for more information see http://www.ladyada.net/make/logshield/lighttemp.html
  */

#define aref_voltage 3.3         // we tie 3.3V to ARef and measure it with a multimeter!




//TMP36 Pin Variables
int tempPin = 1;        //the analog pin the TMP36's Vout (sense) pin is connected to
                        //the resolution is 10 mV / degree centigrade with a
                        //500 mV offset to allow for negative temperatures
int tempReading;        // the analog reading from the sensor

void setup(void) {
  // We'll send debugging information via the Serial monitor
  Serial.begin(9600);
```

```
  // If you want to set the aref to something other than 5v
  analogReference(EXTERNAL);
}


void loop(void) {

  tempReading = analogRead(tempPin);

  Serial.print("Temp reading = ");
  Serial.print(tempReading);     // the raw analog reading

  // converting that reading to voltage, which is based off the reference voltage
  float voltage = tempReading * aref_voltage;
  voltage /= 1024.0;

  // print out the voltage
  Serial.print(" - ");
  Serial.print(voltage); Serial.println(" volts");

  // now print out the temperature
  float temperatureC = (voltage - 0.5) * 100 ;  //converting from 10 mv per degree wit 500 mV offse
                                               //to degrees ((volatge - 500mV) times 100)
  Serial.print(temperatureC); Serial.println(" degrees C");

  // now convert to Fahrenheight
  float temperatureF = (temperatureC * 9.0 / 5.0) + 32.0;
  Serial.print(temperatureF); Serial.println(" degrees F");

  delay(1000);
}
```

Example Projects

Remote temperature sensor

Video editor that uses biofeedback (body temperature)



How to waterproof a LM35 sensor for use in a Remotely Operated Vehicle (robot submarine) (http://adafru.it/aKM)

A "smart coaster" lets you know when your coffee/tea is safe to drink (http://adafru.it/aKN) Some of these projects use thermistors (resistors that change their resistance based on temperature), but can very easily be adapted to to a solid state sensor like the TMP36.

Buy a Temperature Sensor

Buy a Temperature Sensor (http://adafru.it/165)

TECHNICAL DATA MQ-2 GAS SENSOR

FEATURES

Wide detecting scope　　Fast response and High sensitivity
Stable and long life　　Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, i-butane, propane, methane ,alcohol, Hydrogen, smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
V_H	Heating voltage	5V±0.1	ACOR DC
R_L	Load resistance	can adjust	
R_H	Heater resistance	33Ω ±5%	Room Tem
P_H	Heating consumption	less than 800mw	

B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-20℃-50℃	
Tas	Storage Tem	-20℃-70℃	
R_H	Related humidity	less than 95%Rh	
O_2	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	3KΩ-30KΩ (1000ppm iso-butane)	Detecting concentration scope: 200ppm-5000ppm LPG and propane 300ppm-5000ppm butane 5000ppm-20000ppm methane 300ppm-5000ppm H_2 100ppm-2000ppm Alcohol
α (3000/1000) isobutane	Concentration Slope rate	≤0.6	
Standard Detecting Condition	Temp: 20℃±2℃ Vc:5V±0.1 Humidity: 65%±5% Vh: 5V±0.1		
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

	Parts	Materials
1	Gas sensing layer	SnO_2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al_2O_3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

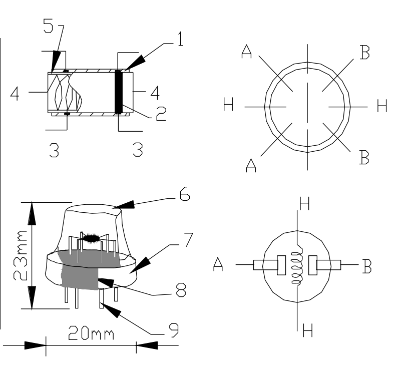


Fig. 1

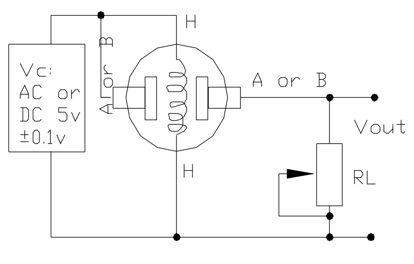
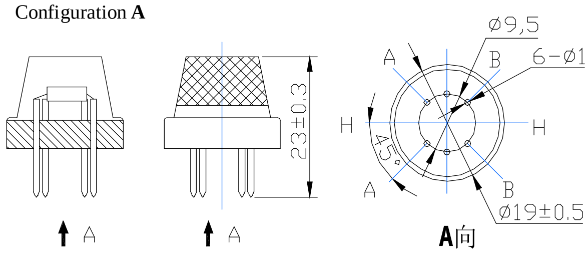
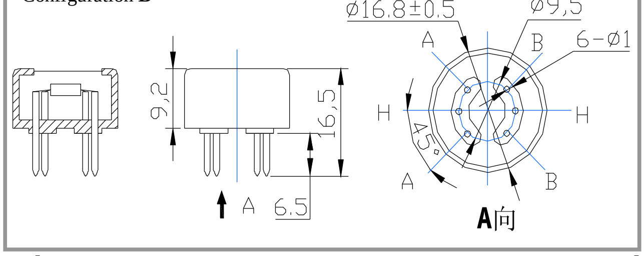


Fig.2

Configuration **A**



Configuration **B**



Structure and configuration of MQ-2 gas sensor is shown as Fig. 1 (Configuration **A** or **B**), sensor composed by micro AL_2O_3 ceramic tube, Tin Dioxide (SnO_2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

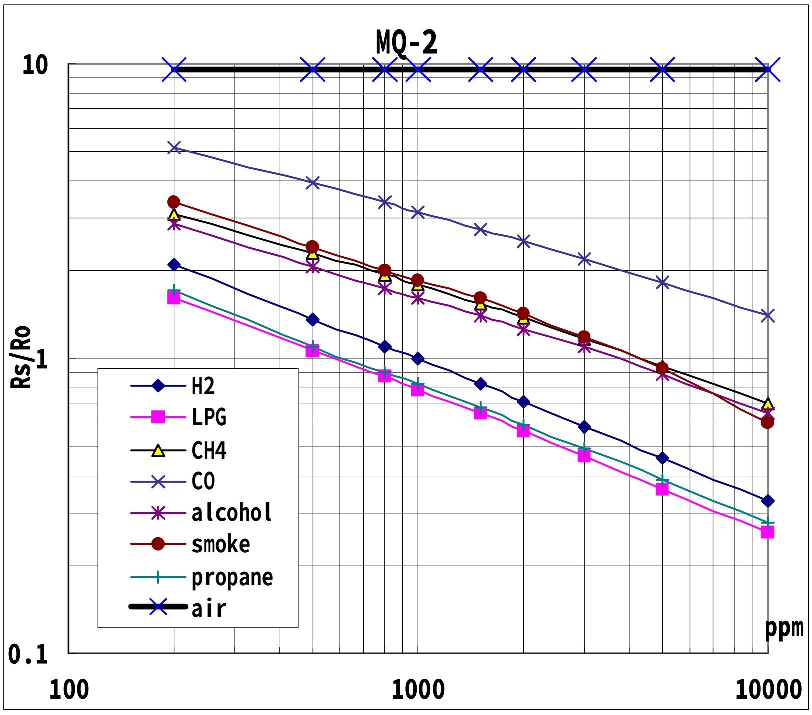


Fig.2 sensitivity characteristics of the MQ-2

Fig.3 is shows the typical sensitivity characteristics of the MQ-2 for several gases.

in their: Temp: 20℃、
Humidity: 65%、
O_2 concentration 21%
RL=5kΩ

Ro: sensor resistance at 1000ppm of H_2 in the clean air.

Rs:sensor resistance at various concentrations of gases.

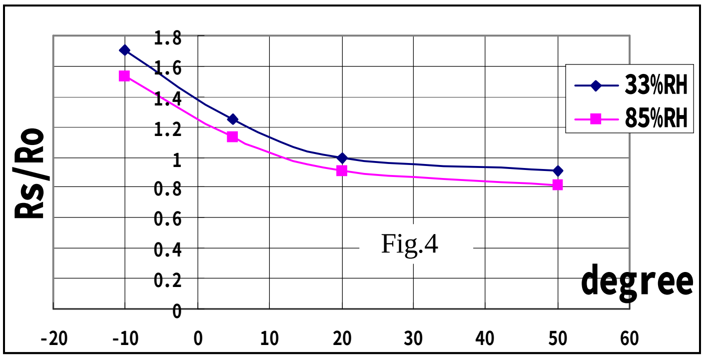


Fig.4 is shows the typical dependence of the MQ-2 on temperature and humidity.

Ro: sensor resistance at 1000ppm of H_2 in air at 33%RH and 20 degree.

Rs: sensor resistance at 1000ppm of H_2 at different temperatures and humidities.

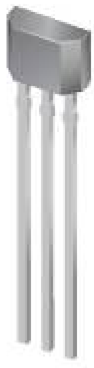
SENSITVITY ADJUSTMENT

Resistance value of MQ-2 is difference to various kinds and various concentration gases. So,When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 1000ppm liquified petroleum gas<LPG>,or 1000ppm iso-butane<i-C_4H_{10}>concentration in air and use value of Load resistance that(R_L) about 20 KΩ(5KΩ to 47 KΩ).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

Features

Miniature construction
Low-Noise Output
4.5 V to 6 V Operation
Magnetically Optimized Package
Linear output for circuit design flexibility
Temperature range of -40 °C to 85 °C



3 pin SIP (suffix UA)

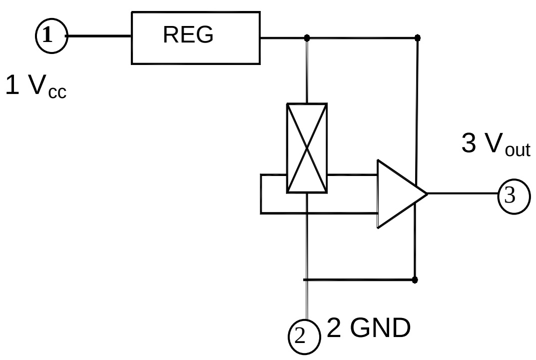
Description

SS49E Linear Hall-effect sensor is small, versatile linear Hall-effect device that is operated by the magnetic field from a permanent magnet or an electromagnet. The linear sourcing output voltage is set by the supply voltage and varies in proportion to the strength of the magnetic field. The integrated circuitry features low noise output, which makes it unnecessary to use external filtering. It also includes thin film resistors to provide increased temperature stability and accuracy. The linear Hall sensor has an operating temperature range of -40 °C to 85 °C appropriate for commercial, consumer and industrial environments.

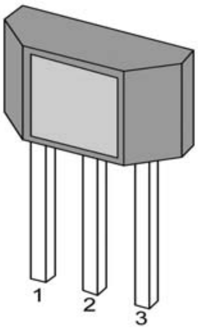
Typical Applications

Motor control
Magnetic code reading
Ferrous metal detector
Current sensing
Position sensing

Functional Block Diagram



Pinning



Pin Description

Name	No	Status	Description
Vdd	1	P	Power Supply
Gnd	2	P	IC Ground
Output	3	O	Output

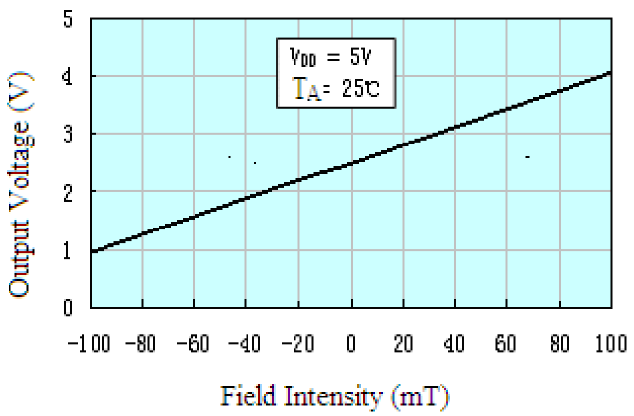
Absolute Maximum Ratings

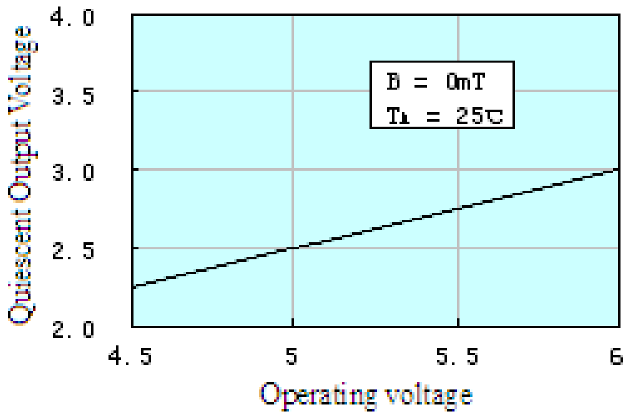
Parameter	Symbol	Value	Units
Supply Voltage (operating)	V_{CC}	8.0	V
Output Current	I_{OUT}	20	mA
Operating Temperature Range	T_A	-40~85	℃
Storage Temperature Rang	T_S	-65~150	℃

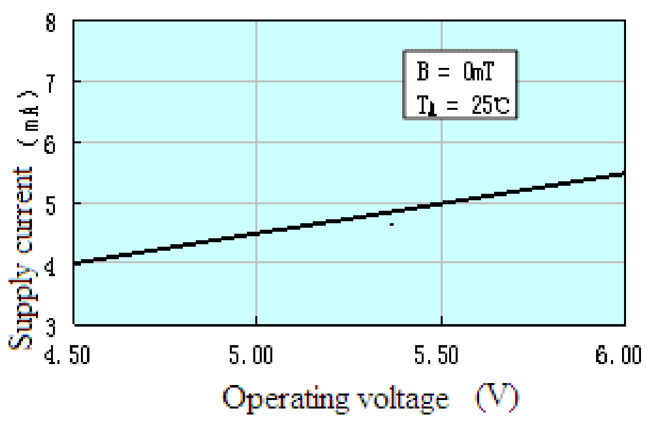
Electrical Characteristics（TA =25℃， VCC =5.0V）

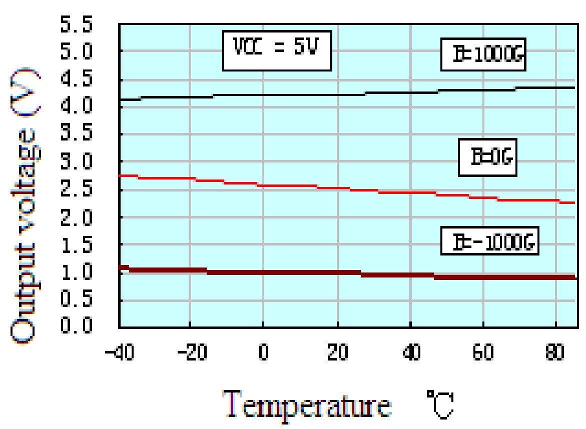
Parameter	Symbol	Test Conditions	Min	Typ	Max	Units
Operating voltage	V_{CC}	Operating	3.0		6.5	V
Supply current	I_{CC}	Average		4.2	8.0	mA
Output Current	I_{OUT}		1.0	1.5		mA
Response Time	Tack			3		uS
Quiescent Output Voltage	Vo	B=0G	2.25	2.5	2.75	V
Sensitivity	△Vout	TA=25℃	1.6	1.8	2.0	mV/G
Min Output Voltage		B=-1500G		0.86		V
Max Output Voltage		B=1500G		4.21		V

Performance Characteristics



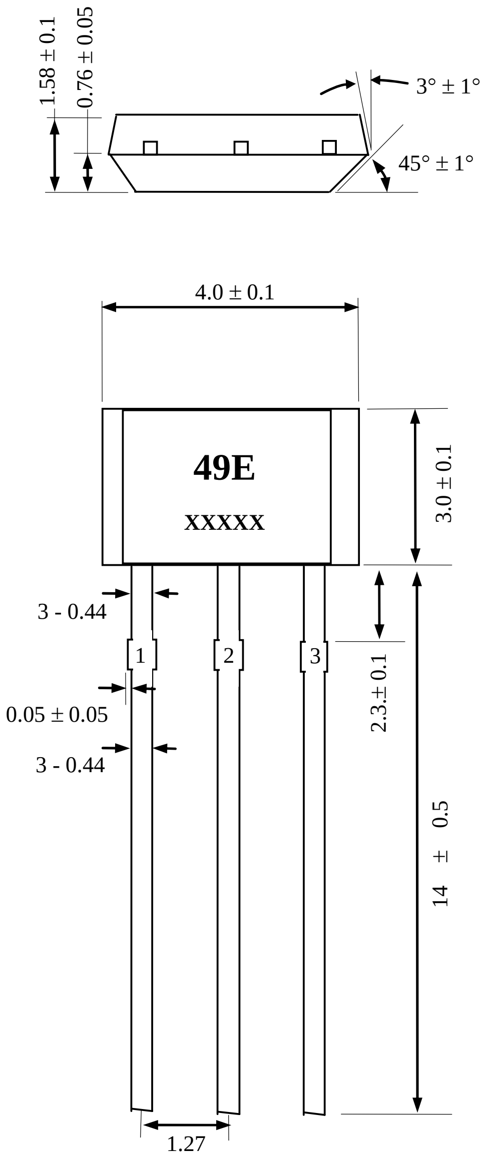


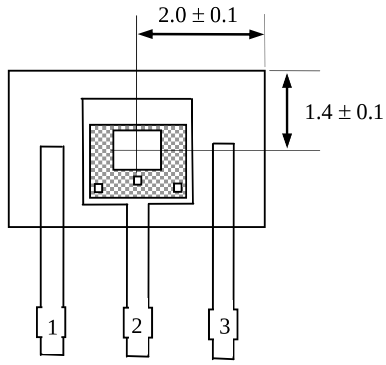




Package Information

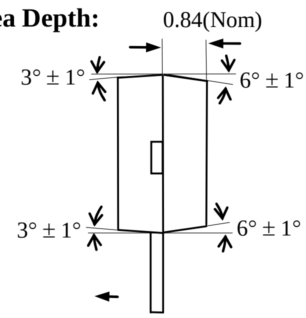
Package UA, 3-Pin SIP:





Hall plate Location

Active Area Depth:



Notes:
1). Controlling dimension : mm ;
2). Leads must be free of flash and plating voids ;
3). Do not bend leads within 1 mm of lead to package interface ;
4). PINOUT: Pin 1 VDD
Pin 2 GND
Pin 3 Output

Marking:
49E --- Code of Device;
XXXXX -- Production Lot ;

Ordering Information

Part No.	Pb-free	Temperature Code	Package Code	Packing
SS49E	YES	-40°C to 85°C	TO-92	Bulk, 1000 pieces/bag

ATX12V
Power Supply Design Guide

Version 2.01

IMPORTANT INFORMATION AND DISCLAIMERS

Revision History

Version	Release Date	Notes
1.0	Feb, 2000	• Public release
1.1	Aug, 2000	• Increase 3.3 V current; add more explanation for power sharing; do minor edits and format fixes
1.2	Jan, 2002	• Typical Power Distribution. Change +5V loading on all power supplies distribution tables defined in DG to 0.3A • PS_ON# Add text "The power supply should not latch into a shutdown state when PS_ON# is driven active by pulses between 10ms to 100ms during the decay of the power rails." • Remove –5V from all power distribution tables.
1.3	April, 2003	• Update Power and Current guidance • Added efficiency guidance at typical and light load • Increased min efficiency at full load from 68% to 70% • Serial ATA* connector definition added • Acoustic levels added for low noise power supply design • Reformat and update revision table • Update Disclaimers • Remove guidelines for ATX • Remove guidance for –5V rail • Updated guidance for Energy Star and stand by efficiency
2.0	February, 2003	• Added Terminology section • Updated power and current guidance • Includes 250W, 300W, 350W, and 400W guidance • Updated cross regulation graphs • Updated load tables • Updated required efficiency targets. Added recommended efficiency targets. • Increased required minimum efficiency at typical and light load. • Main Power Connector changes to 2x12. • Aux power connector removed. • Required Serial ATA Connector. • Isolated current limit on 2x2 connector for 12V2 rail.
2.01	June, 2004	• Updated 3.3 V remote sense pin # on the main power connector • Updated 12V2 DC Output Noise/Ripple information • Removed -5V reference • Updated 5Vsb maximum current step

Contents

Figures

Tables

1. Introduction

1.1. Scope

This document provides design suggestions and reference specifications for a family of power supplies that comply with the *ATX Specification, Version 2.03*[†] for motherboards and chassis. It includes supplementary information not expressly detailed in the *ATX Specification*, such as information about the physical form factor of the power supply, cooling requirements, connector configuration, and pertinent electrical and signal timing specifications.

This document is provided as a convenience only and is not intended to replace the user's independent design and validation activity. It should not be inferred that all ATX12V power supplies must conform exactly to the content of this document. The design specifics described herein are not intended to support all possible system configurations. System power supply needs vary widely depending on factors such as the application (that is, for desktop, workstation, or server), intended ambient environment (temperature, line voltage), or motherboard power requirements.

1.2. Key Changes for ATX12V Version 2.0 and later as Compared with ATX and Pervious Versions of ATX12V Power Supply

This section briefly summarizes the major changes made to this document that now defines ATX12V power supply. With the move to 12V voltage regulators for the processor, ATX guidelines for 5V as main power are no longer provided.

1.2.1. Increased +12 VDC output capability

System components that use 12V are continuing to increase in power. In cases where expected current requirements is greater than 18A a second 12 V rail should be made available. ATX12V power supplies should be designed to accommodate these increased +12 VDC currents.

1.2.2. Minimum Efficiency

Minimum measured efficiency is required to be 70% at full and typical (~50%) load and 60% at light (~20%) load. New recommended guidance has been added to provide direction for future requirements.

[†] 2.03 is the current version of the *ATX Specification* as of this writing. Future references to the *ATX Specification* in this document imply version 2.03 or later, as applicable.

1.2.3. Main Power Connector:

The 2 x 10 main power connector has been replaced by a 2 x 12 connector. This was made to support 75 watt PCI Express*requirements. Pinout assignments are based on the SSI recommendation.

With the added 12V, 5V, and 3.3V pins the need for an Aux Power connector is no longer needed and the guidance for this connector has been removed.

1.2.4. Separate current limit for 12V2 on the 2x2 connector:

The 12V rail on the 2 x 2 power connector should be a separate current limited output to meet the requirements of UL and EN 60950.

1.3 Terminology

The following terms are used in this document:

Term	Description
Required	The status given to items within this design guide, which are required to meet design guide and a large majority of system applications.
Recommended	The status given to items within this design guide, which are not required to meet design guide, however, are required by many system applications.
Optional	The status given to items within this design guide, which are not required to meet design guide, however, some system applications may optionally use these features.
BA	Declared sound power, LwAd. The declared sound power level shall be measured according to ISO* 7779 for the power supply and reported according to ISO 9296.
CFM	Cubic Feet per Minute (airflow).
Monotonically	A waveform changes from one level to another in a steady fashion, without intermediate retracement or oscillation.
Noise	The periodic or random signals over frequency band of 0 Hz to 20 MHz.

2. Applicable Documents

The following documents support this design guide as additional reference material.

Document Title	Description
FCC Rules Part 15, Class B	Title 47, Code of Federal Regulations, Part 15
ICES-003: 1997, Class B	Interference-Causing Equipment Standard – Digital Apparatus
EN 55022: 1998 + Amendment A1:2000 Class B	Information Technology Equipment – Radio disturbance characteristics – Limits and methods of measurement
CISPR 22: 1997, Class B	Information Technology Equipment – Radio disturbance characteristics – Limits and methods of measurement
AS/NZS 3548:1995, Class B	Information Technology Equipment – Radio disturbance characteristics – Limits and methods of measurement
EN 55024:1998	Information Technology Equipment – Immunity Characteristics – Limits and methods of measurement
IEC 60950, 3rd ed., 1999	Safety of Information Technology Equipment
EN 60950: 2000	Safety of Information Technology Equipment
UL 60950, 3rd ed., 2000	Safety of Information Technology Equipment
CSA 22.2 No. 60950-00	Safety of Information Technology Equipment

3. Electrical

The electrical requirements that follow are to be met over the environmental ranges specified in Section 5 unless otherwise noted.

3.1. AC Input

Table 1 lists AC input voltage and frequency requirements for continuous operation. The power supply shall be capable of supplying full-rated output power over two input voltage ranges rated 100-127 VAC and 200-240 VAC RMS nominal. The correct input range for use in a given environment may be either switch-selectable or auto-ranging. The power supply shall automatically recover from AC power loss. The power supply must be able to start up under peak loading at 90 VAC.

Table 1. AC Input Line Requirements

Parameter	Minimum	Nominal+	Maximum	Unit
V_{in} (115 VAC)	90	115	135	VAC $_{rms}$
V_{in} (230 VAC)	180	230	265	VAC $_{rms}$
V_{in} Frequency	47	--	63	Hz

+Note: Nominal voltages for test purposes are considered to be within ±1.0 V of nominal.

3.1.1. Input Over-current Protection

The power supply shall incorporate primary fusing for input over-current protection to prevent damage to the power supply and meet product safety requirements. Fuses should be slow-blow–type or equivalent to prevent nuisance trips[‡].

3.1.2. Inrush Current Limiting

Maximum inrush current from power-on (with power on at any point on the AC sine) and including, but not limited to, three line cycles, shall be limited to a level below the surge rating of the input line cord, AC switch if present, bridge rectifier, fuse, and EMI filter components. Repetitive ON/OFF cycling of the AC input voltage should not damage the power supply or cause the input fuse to blow.

[‡]. For Denmark and Switzerland international safety requirements, if the internal over-current protective devices exceed 8A for Denmark and 10A for Switzerland, then the power supply must pass international safety testing to EN 60950 using a maximum 16A over-current protected branch circuit, and this 16A (time delay fuse) branch circuit protector must not open during power supply abnormal operation (output short circuit and component fault) testing.

3.1.3. Input Under-voltage

The power supply shall contain protection circuitry such that the application of an input voltage below the minimum specified in Section 3.1, Table 1, shall not cause damage to the power supply.

3.1.4. Regulatory

The power supply is required to be tested and comply with the most current version of the following regulatory specification requirements and/or standards

PRODUCT SAFETY

UL* 60950, 3rd Edition –CAN/CSA-C22.2-60950-00,

EN*60 950, 3rd Edition

IEC*60 950, 3rd Edition (CB Report to include all national deviations)

EU* Low Voltage Directive (73/23/EEC) (CE Compliance)

GB4943-90 CCIB* (China)

ELECTROMAGNETIC CAMPATIBILITY

FCC*, Class B, Part 15 (Radiated & Conducted Emissions)

CISPR* 22 / EN55022, 3rd Edition (Radiated & Conducted Emissions)

EN55024 (ITE Specific Immunity)

EN 61000-4-2 – Electrostatic Discharge

EN 61000-4-3– Radiated RFI Immunity

EN 61000-4-4– Electrical Fast Transients.

EN 61000-4-5 – Electrical Surge

EN 61000-4-6 – RF Conducted

EN 61000-4-8 – Power Frequency Magnetic Fields

EN 61000-4-11 – Voltage Dips, Short Interrupts and Fluctuations

EN61000-3-2 (Harmonics)

EN61000-3-3 (Voltage Flicker)

EU EMC Directive ((8/9/336/EEC) (CE Compliance)

Other Certifications and/or Declarations

GB925 (China/CCC*), CNS13438 (Taiwan/BSMI*),

AS/NZ3548 (Australia/C-tick* based on CISPR22)

3.1.5. Catastrophic Failure Protection

Should a component failure occur, the power supply should not exhibit any of the following:

- Flame
- Excessive smoke
- Charred PCB
- Fused PCB conductor
- Startling noise
- Emission of molten material

3.2. DC Output

3.2.1. DC Voltage Regulation

The DC output voltages shall remain within the regulation ranges shown in Table 2 when measured at the load end of the output connectors under all line, load, and environmental conditions. The voltage regulation limits shall be maintained under continuous operation for any steady state temperature and operating conditions specified in Section 5.

Table 2. DC Output Voltage Regulation

Output	Range	Min.	Nom.	Max.	Unit
+12V1DC	±5%	+11.40	+12.00	+12.60	Volts
+12V2DC (1)	±5%	+11.40	+12.00	+12.60	Volts
+5VDC	±5%	+4.75	+5.00	+5.25	Volts
+3.3VDC (2)	±5%	+3.14	+3.30	+3.47	Volts
-12VDC	±10%	-10.80	-12.00	-13.20	Volts
+5VSB	±5%	+4.75	+5.00	+5.25	Volts

(1) At +12 VDC peak loading, regulation at the +12 VDC output can go to ± 10%.

(2) Voltage tolerance is required at main connector and S-ATA connector (if used).

3.2.2. Remote Sensing

The +3.3 VDC output should have provisions for remote sensing to compensate for excessive cable drops. The default sense should be connected to pin 13 of the main power connector. The power supply should draw no more than 10 mA through the remote sense line to keep DC offset voltages to a minimum.

3.2.3. Typical Power Distribution

DC output power requirements and distributions will vary based on specific system options and implementation. Significant dependencies include the quantity and types of processors, memory, add-in card slots, and peripheral bays, as well as support for advanced graphics or other features. *It is ultimately the responsibility of the designer to derive a power budget for a given target product and market.*

Table 3 through Table 5 and Figure 1 through Figure 3 provide sample power distributions and a graphical recommendation for cross loading. It should not be inferred that all power supplies must conform to these tables, nor that a power supply designed to meet the information in the tables will work in all system configurations.

3.2.3.1. ATX12V Configurations

Table 3. Typical Power Distribution for a 250 W ATX12V Configuration

Output	Min. Current (amps)	Max. Current (amps)	Peak Current (amps)
+12 V1DC(1, 2)	1	8	10
+12 V2DC(1, 2)	1	14	
+5 VDC	0.3	18	
+3.3 VDC	0.5	17	
-12 VDC	0	0.3	
+5 VSB	0	2	2.5

Note: Total combined output of 3.3 V and 5 V is ≤ 115W
Peak currents may last up to 17 seconds with not more than one occurrence per minute
(1)12V1DC and 12V2DC should have separate current limit circuits to meet 240VA safety requirements.
(2)12V2 supports processor power requirements and must have a separate current limit

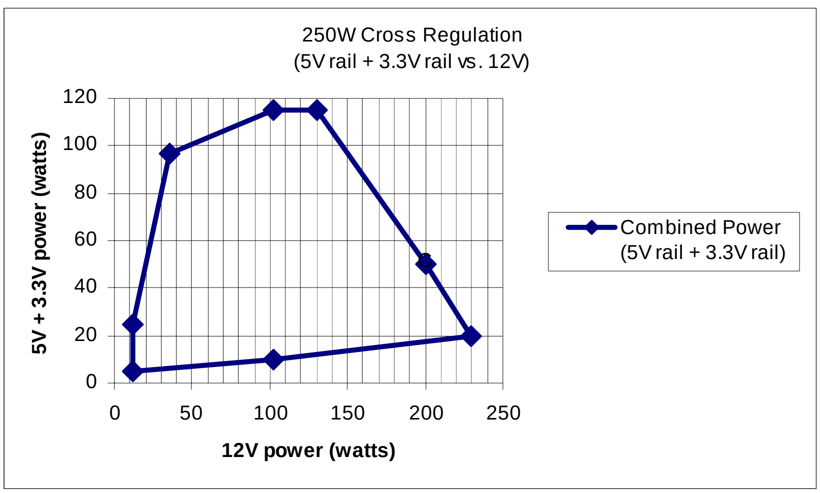


Figure 1. Cross Loading Graph for 250W Configuration

Table 4. Typical Power Distribution for a 300 W ATX12V Configuration

Output	Min. Current (amps)	Max. Current (amps)	Peak Current (amps)
+12 V1DC(1, 2)	1.0	8.0	10.0
+12 V2DC(1, 2)	1.0	14.0	
+5 VDC	0.3	20.0	
+3.3 VDC	0.5	20.0	
-12 VDC	0.0	0.3	
+5 VSB	0.0	2.0	2.5

Note: Total combined output of 3.3 V and 5 V is ≤ 120 W
Peak currents may last up to 17 seconds with not more than one occurrence per minute
(1)12V1DC and 12V2DC should have separate current limit circuits to meet 240VA safety requirements.
(2)12V2 supports processor power requirements and must have a separate current limit

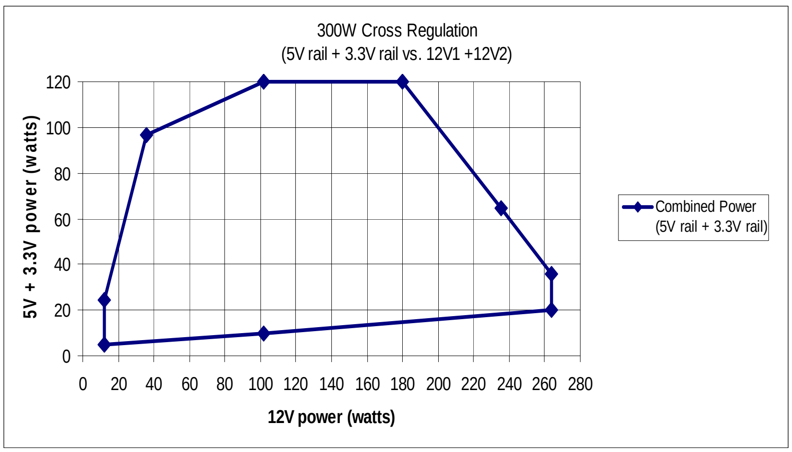


Figure 2. Cross Loading Graph for 300W Configuration

Table 5. Typical Power Distribution for a 350 W ATX12V Configuration

Output	Min. Current (amps)	Max. Current (amps)	Peak Current (amps)
+12 V1DC[(1, 2)]	1	10	12
+12 V2DC[(1, 2)]	1	15	
+5 VDC	0.3	21	
+3.3 VDC	0.5	22	
-12 VDC	0.0	0.3	
+5 VSB	0.0	2.0	2.5

Note: Total combined output of 3.3 V and 5 V is ≤ 130 W
Peak currents may last up to 17 seconds with not more than one occurrence per minute
[(1)]12V1DC and 12V2DC should have separate current limit circuits to meet 240VA safety requirements.
[(2)]12V2 supports processor power requirements and must have a separate current limit

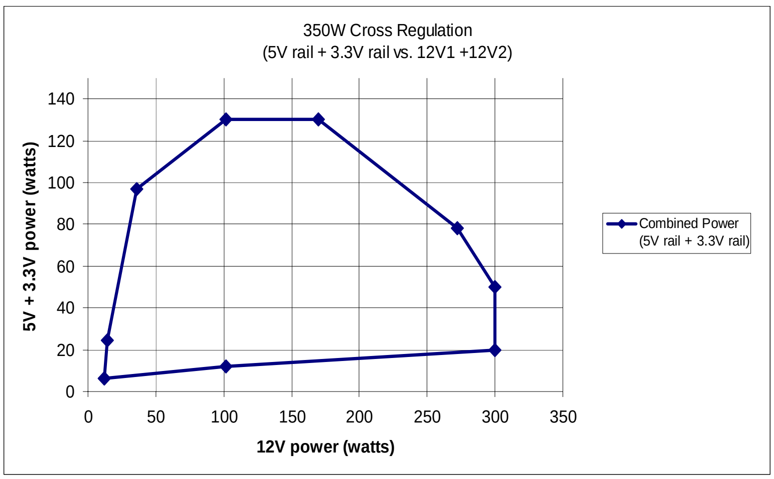


Figure 3. Cross Loading Graph for 350W Configuration

Table 6. Typical Power Distribution for a 400 W ATX12V Configuration

Output	Min. Current (amps)	Max. Current (amps)	Peak Current (amps)
+12 V1DC	1	14	16
+12 V2DC	1	15	
+5 VDC	0.3	28	
+3.3 VDC	0.5	30	
-12 VDC	0	0.3	
+5 VSB	0	2	2.5

Note: Total combined output of 3.3 V and 5 V is ≤ 130 W
Peak currents may last up to 17 seconds with not more than one occurrence per minute
12V1DC and 12V2DC should have separate current limit circuits to meet 240VA safety requirements.

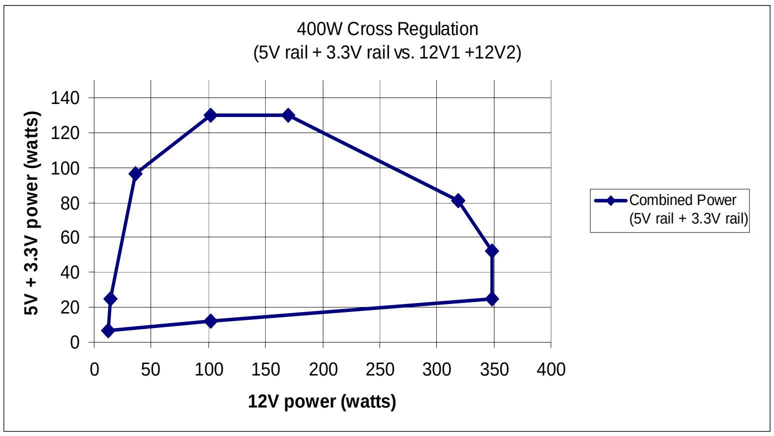


Figure 4. Cross Loading Graph for 400W Configuration

3.2.4. Power Limit / Hazardous Energy Levels

Under normal or overload conditions, no output shall continuously provide more than 240 VA under any conditions of load including output short circuit, per the requirement of UL 1950/CSA 950 / EN 60950/IEC 950.

3.2.5. Efficiency

3.2.5.1. General

The power supply required minimum is 70% efficient under “Full” load, 70% under “typical” load, and 60% in a “light” load or idle condition. The efficiency of the power supply should be tested at nominal input voltage of 115VAC input and/or 230VAC input, under the load conditions defined in Table 7 and Table 8, and under the temperature and operating conditions defined in Section 5. The loading condition for testing efficiency shown in Table 8 represents a fully loaded system, a ~50% (typical) loaded system, and a ~20% (light) loaded system.

Table 7. Minimum Efficiency Vs Load

Loading	Full load	Typical load	Light load
Required Minimum Efficiency	70%	70%	60%
Recommended Minimum Efficiency	75%	80%	68%

Table 8. Loading Table for Efficiency Measurements

250W (loading shown in Amps)						
Loading	**+12V1**	**+12V2**	**+5V**	**+3.3V**	**-12V**	**+5Vsb**
Full	4	11.5	6.8	6.5	0.3	1.0
Typical	3	5	3	4	0.1	1.0
Light	2	2.4	0.3	0.5	0.0	1.0
300W (loading shown in Amps)						
Loading	**+12V1**	**+12V2**	**+5V**	**+3.3V**	**-12V**	**+5Vsb**
Full	7	12	8	7.5	0.2	1.0
Typical	4	8	3	4	0.1	1.0
Light	2	2	0.5	1.5	0.0	1.0
350W (loading shown in Amps)						
Loading	**+12V1**	**+12V2**	**+5V**	**+3.3V**	**-12V**	**+5Vsb**
Full	10	13	9	10	0.3	1.0
Typical	5	9	3	5	0.1	1.0
Light	3	3	1.0	2.0	0.0	1.0
400W (loading shown in Amps)						
Loading	**+12V1**	**+12V2**	**+5V**	**+3.3V**	**-12V**	**+5Vsb**
Full	12	14	9	11	0.3	1.0
Typical	5	9	3	5	0.1	1.0
Light	3	3	1	3	0.0	1.0

3.2.5.2. Energy Star*

The "Energy Star" efficiency requirements of the power supply depend on the intended system configuration. In the low-power / sleep state (S1 or S3) the system should consume power in accordance with the values listed in Table 9.

Table 9. Energy Star Input Power Consumption

Maximum Continuous Power Rating of Power Supply	RMS Watts from the AC line in sleep/low-power mode
≤ 200 W	≤ 15 W
> 200 W ≤ 300 W	≤ 20 W
> 300 W ≤ 350 W	≤ 25 W
> 350 W ≤ 400 W	≤ 30 W
> 400 W	10% of the maximum continuous output rating

Note: To help meet the "Energy Star" system requirements, it is recommended that the power supply have > 50% efficiency in standby mode.

3.2.5.3. Other Low Power System Requirements

For power supplies designed for low standby power, the following provides some general guidance. Requirements will vary with geographic region and target end user market.

To help meet the Blue Angel*, RAL-UZ 78, US Presidential executive order 13221, future EPA requirements, and other low Power system requirements, it is recommended that the +5 VSB standby supply should be as efficient as possible. Standby efficiency is measured with the main outputs off (PS_ON# high state). Standby efficiency should be greater than 50% with a minimum loading of 100mA.

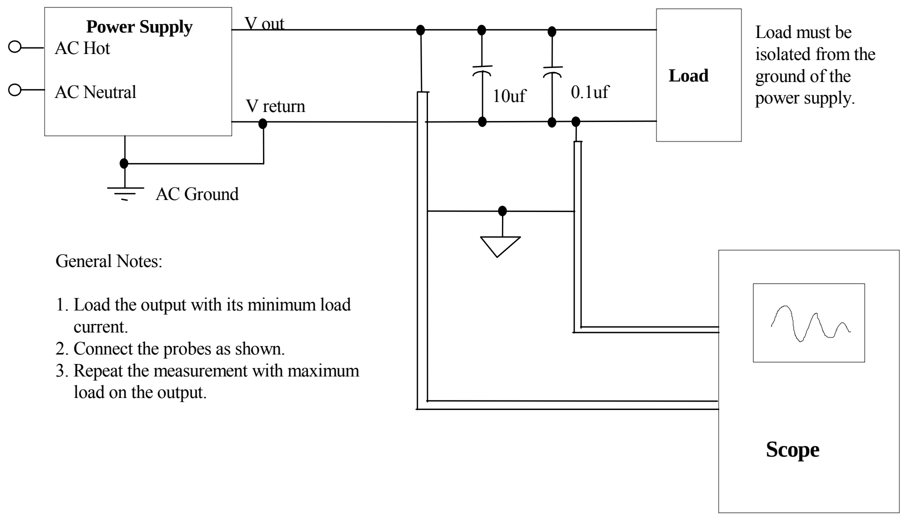
3.2.6. Output Ripple/Noise

The output ripple/noise requirements listed in Table 10 should be met throughout the load ranges specified in Section 3.2.3 and under all input voltage conditions as specified in Section 3.1.

Ripple and noise are defined as periodic or random signals over a frequency band of 10 Hz to 20 MHz. Measurements shall be made with an oscilloscope with 20 MHz bandwidth. Outputs should be bypassed at the connector with a 0.1 µF ceramic disk capacitor and a 10 µF electrolytic capacitor to simulate system loading. See Figure 5.

Table 10. DC Output Noise/Ripple

Output	Max. Ripple & Noise (mV_{pp})
+12 V1DC	120
+12 V2DC	120
+5 VDC	50
+3.3 VDC	50
-12 VDC	120
+5 VSB	50



General Notes:

1. Load the output with its minimum load current.
2. Connect the probes as shown.
3. Repeat the measurement with maximum load on the output.

Filter Note:

0.1uf - Kemet, C1206C104K5RAC or equivalent
10uf - United Chemi-con, 293D106X0025D2T or equivalent

Scope Note:

Use Tektronix TDS460 Oscilloscope or equivalent and a P6046 probe or equivalent.

Figure 5. Differential Noise Test Setup

3.2.7. Output Transient Response

Table 11 summarizes the expected output transient step sizes for each output. The transient load slew rate is = 1.0 A/µs.

Table 11. DC Output Transient Step Sizes

Output	Max. step size (% of rated output amps per Sec 3.2.3) [1]	Max. step size (amps)
+12 V1DC	40%	
+12 V2DC	60%	
+5 VDC	30%	
+3.3 VDC	30%	
-12 VDC		0.1 A
+5 VSB		0.5 A

[1] For example, for a rated +5 VDC output of 18 A, the transient step would be 30% × 18 A = 5.4 A

Comment: Different thean SFX or TFX which state 0.5

Output voltages should remain within the regulation limits of Section 3.2.1, and the power supply should be stable when subjected to load transients per Table 11 from any steady state load, including any or all of the following conditions:

- Simultaneous load steps on the +12 VDC, +5 VDC, and +3.3 VDC outputs (all steps occurring in the same direction)
- Load-changing repetition rate of 50 Hz to 10 kHz
- AC input range per Section 3.1
- Capacitive loading per Table 12.

3.2.8. Capacitive Load

The power supply should be able to power up and operate normally with the following capacitances simultaneously present on the DC outputs. This capacitive loading should be used to check stability and should not be included for noise testing.

Table 12. Output Capacitive Loads

Output	ATX12V Capacitive load (µF)
+12 V1DC	5,000
+12 V2DC	3,000
+5 VDC	6,000
+3.3 VDC	6,000
-12 VDC	350
+5 VSB	350

3.2.9. Closed-loop Stability

The power supply shall be unconditionally stable under all line/load/transient load conditions including capacitive loads specified in Section 3.2.8. A minimum of 45 degrees phase margin and 10 dB gain margin is recommended at both the maximum and minimum loads.

3.2.10. +5 VDC / +3.3 VDC Power Sequencing

The +12 VDC and +5 VDC output levels must be equal to or greater than the +3.3 VDC output at all times during power-up and normal operation. The time between the +12 VDC or +5 VDC output reaching its minimum in-regulation level and +3.3 VDC reaching its minimum in-regulation level must be ≤ 20 ms.

3.2.11. Voltage Hold-up Time

The power supply should maintain output regulation per Section 3.2.1 despite a loss of input power at the low-end nominal range—115 VAC / 47 Hz or 230 VAC / 47 Hz—at maximum continuous output load as applicable for a minimum of 17 ms.

3.3. Timing / Housekeeping / Control

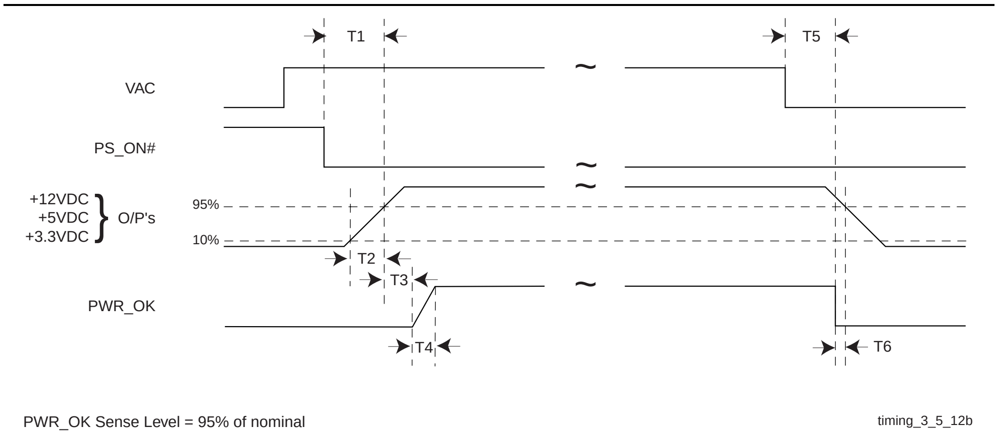


Figure 6. Power Supply Timing

Notes: T1 is defined in Section 3.3.4. T2 is defined in Section 3.3.5. T3, T4, T5, and T6 are defined in Table 13

3.3.1. PWR_OK

PWR_OK is a "power good" signal. It should be asserted high by the power supply to indicate that the +12 VDC, +5VDC, and +3.3VDC outputs are above the under-voltage thresholds listed in Section 3.2.1 and that sufficient mains energy is stored by the converter to guarantee continuous power operation within specification for at least the duration specified in Section 3.2.11, "Voltage Hold-up Time." Conversely, PWR_OK should be de-asserted to a low state when any of the +12 VDC, +5 VDC, or +3.3 VDC output voltages falls below its under-voltage threshold, or when mains power has been removed for a time sufficiently long such that power supply operation cannot be guaranteed beyond the power-down warning time. The electrical and timing characteristics of the PWR_OK signal are given in Table 13 and in Figure 6.

Table 13. PWR_OK Signal Characteristics

Signal Type	+5 V TTL compatible
Logic level low	< 0.4 V while sinking 4 mA
Logic level high	Between 2.4 V and 5 V output while sourcing 200 µA
High-state output impedance	1 kΩ from output to common
PWR_OK delay	100 ms < T_3 < 500 ms
PWR_OK risetime	$T_4 \leq 10$ ms
AC loss to PWR_OK hold-up time	$T_5 \geq 16$ ms
Power-down warning	$T_6 \geq 1$ ms

3.3.2. PS_ON#

PS_ON# is an active-low, TTL-compatible signal that allows a motherboard to remotely control the power supply in conjunction with features such as soft on/off, Wake on LAN*, or wake-on-modem. When PS_ON# is pulled to TTL low, the power supply should turn on the four main DC output rails: +12VDC, +5VDC, +3.3VDC and -12VDC. When PS_ON# is pulled to TTL high or open-circuited, the DC output rails should not deliver current and should be held at zero potential with respect to ground. PS_ON# has no effect on the +5VSB output, which is always enabled whenever the AC power is present. Table 14 lists PS_ON# signal characteristics.

The power supply shall provide an internal pull-up to TTL high. The power supply shall also provide de-bounce circuitry on PS_ON# to prevent it from oscillating on/off at startup when activated by a mechanical switch. The DC output enable circuitry must be SELV-compliant.

The power supply shall not latch into a shutdown state when PS_ON# is driven active by pulses between 10ms to 100ms during the decay of the power rails.

Table 14. PS_ON# Signal Characteristics

	Min.	Max.
V_{IL}, Input Low Voltage	0.0 V	0.8 V
I_{IL}, Input Low Current (Vin = 0.4 V)		-1.6 mA
V_{IH}, Input High Voltage (Iin = -200 µA)	2.0 V	
V_{IH} open circuit, Iin = 0		5.25 V

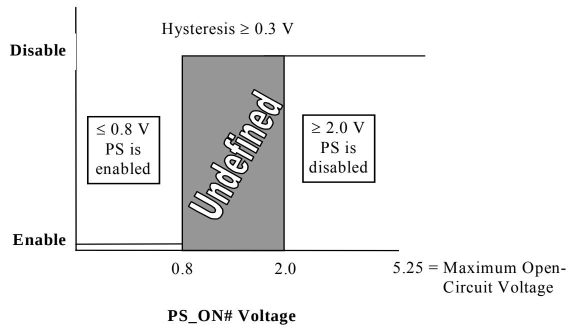


Figure 7. PS_ON# Signal Characteristics

3.3.3. +5 VSB

+5 VSB is a standby supply output that is active whenever the AC power is present. It provides a power source for circuits that must remain operational when the five main DC output rails are in a disabled state. Example uses include soft power control, Wake on LAN, wake-on-modem, intrusion detection, or suspend state activities.

The +5 VSB output should be capable of delivering a minimum of 2.0 A at +5 V ± 5% to external circuits. The power supply must be able to provide the required power during a "wake up" event. If an external USB device generates the event, there may be peak currents as high as 2.5A lasting no more than 500mS.

Overcurrent protection is required on the +5 VSB output regardless of the output current rating. This ensures the power supply will not be damaged if external circuits draw more current than the supply can provide.

3.3.4. Power-on Time

The power-on time is defined as the time from when PS_ON# is pulled low to when the +12 VDC, +5 VDC, and +3.3 VDC outputs are within the regulation ranges specified in Section 3.2.1. The power-on time shall be less than 500 ms ($T_1 < 500$ ms).

+5 VSB shall have a power-on time of two seconds maximum after application of valid AC voltages.

3.3.5. Risetime

The output voltages shall rise from ≤10% of nominal to within the regulation ranges specified in Section 3.2.1 within 0.1 ms to 20 ms ($0.1 \text{ ms} \leq T_2 \leq 20 \text{ ms}$).

There must be a smooth and continuous ramp of each DC output voltage from 10% to 90% of its final set-point within the regulation band, while loaded as specified in Section 3.2.3. The smooth turn-on requires that, during the 10% to 90% portion of the rise time, the slope of the turn-on waveform must be positive and have a value of between 0 V/ms and [Vout,nominal / 0.1] V/ms. Also, for any 5 ms segment of the 10% to 90% risetime waveform, a straight line drawn between the end points of the waveform segment must have a slope ≥ [Vout,nominal / 20] V/ms.

3.3.6. Overshoot at Turn-on / Turn-off

The output voltage overshoot upon the application or removal of the input voltage, or the assertion/deassertion of PS_ON#, under the conditions specified in Section 3.1, shall be less than 10% above the nominal voltage. No voltage of opposite polarity shall be present on any output during turn-on or turn-off.

3.3.7. Reset after Shutdown

If the power supply latches into a shutdown state because of a fault condition on its outputs, the power supply shall return to normal operation only after the fault has been removed and the PS_ON# (or AC input) has been cycled OFF/ON with a minimum OFF time of 1 second.

3.3.8. +5 VSB at AC Power-down

After AC power is removed, the +5 VSB standby voltage output should remain at its steady state value for the minimum hold-up time specified in Section 3.2.11 until the output begins to decrease in voltage. The decrease shall be monotonic in nature, dropping to 0.0 V. There shall be no other perturbations of this voltage at or following removal of AC power.

3.4. Output Protection

3.4.1. Over-voltage Protection

The over-voltage sense circuitry and reference shall reside in packages that are separate and distinct from the regulator control circuitry and reference. No single point fault shall be able to cause a sustained over-voltage condition on any or all outputs. The supply shall provide latch-mode over-voltage protection as defined in Table 15.

Table 15. Overvoltage Protection

Output	Min.	Nom.	Max.	Unit
+12 V1DC & +12V2DC	13.4	15.0	15.6	Volts
+5 VDC	5.74	6.3	7.0	Volts
+3.3 VDC	3.76	4.2	4.3	Volts

3.4.2. Short-circuit Protection

An output short circuit is defined as any output impedance of less than 0.1 ohms. The power supply shall shut down and latch off for shorting the +3.3 VDC, +5 VDC, or +12 VDC rails to return or any other rail. The +12 V1DC and +12V2DC should have separate short circuit and overload protection. Shorts between main output rails and +5 VSB shall not cause any damage to the power supply. The power supply shall either shut down and latch off or fold back for shorting the negative rails. +5 VSB must be capable of being shorted indefinitely, but when the short is removed, the power supply shall recover automatically or by cycling PS_ON#. The power supply shall be capable of withstanding a continuous short-circuit to the output without damage or overstress to the unit (for example, to components, PCB traces, connectors) under the input conditions specified in Section 3.1. The maximum short-circuit energy in any output shall not exceed 240 VA, per IEC 60950 requirements.

3.4.3. No-load Operation

No damage or hazardous condition should occur with all the DC output connectors disconnected from the load. The power supply may latch into the shutdown state.

3.4.4. Over-current Protection

Overload currents applied to each tested output rail will cause the output to trip before reaching or exceeding 240 VA. For testing purposes, the overload currents should be ramped at a minimum rate of 10 A/s starting from full load.

3.4.5. Over-temperature Protection

The power supply may include an over-temperature protection sensor, which can trip and shut down the power supply at a preset temperature point. Such an overheated condition is typically the result of internal current overloading or a cooling fan failure. If the protection circuit is nonlatching, then it should have hysteresis built in to avoid intermittent tripping.

3.4.6. Output Bypass

The output return may be connected to the power supply chassis. The return will be connected to the system chassis by the system components.

4. Mechanical

4.1. Labeling / Marking

The following is a non-inclusive list of suggested markings for each power supply unit. Product regulation stipulations for sale into various geographies may impose additional labeling requirements.

- Manufacturer information: manufacturer's name, part number, and lot date code, etc., in human-readable text and/or bar code formats
- Nominal AC input operating voltages (100-127 VAC and 200-240 VAC) and current rating certified by all applicable safety agencies (Section 8)
- DC output voltages and current ratings
- Access warning text ("Do not remove this cover. Trained service personnel only. No user serviceable components inside.") in English, German, Spanish, French, Chinese, and Japanese with universal warning markings

4.2. Physical Dimensions

The supply shall be enclosed and meet the physical outline shown in either Figure 8 or 9, as applicable.

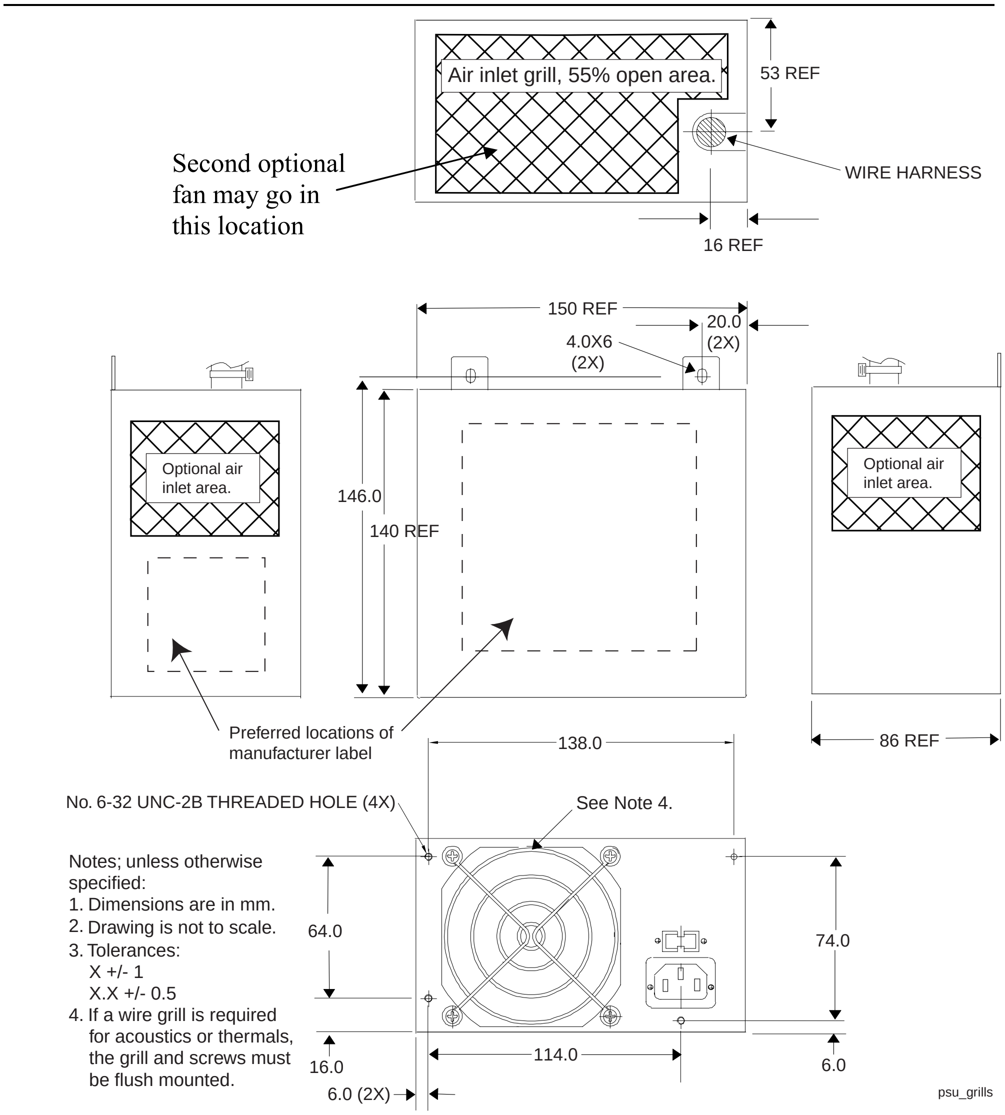


Figure 8. Power Supply Dimensions for Chassis That Does Not Require Top Venting

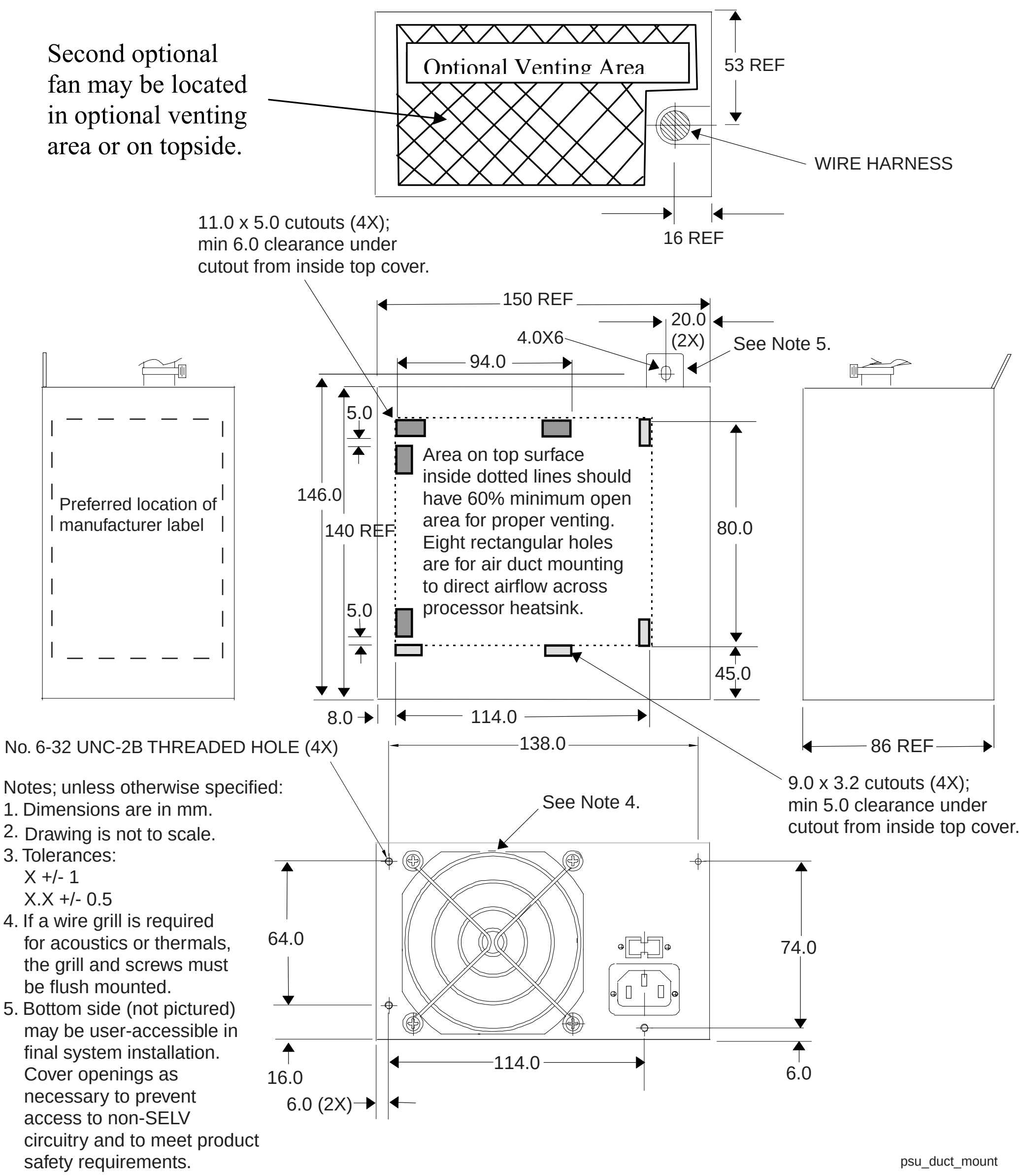


Figure 9. Power Supply Dimensions for Chassis That Require Top Venting

4.3. Airflow / Fan

The *ATX Specification* allows for numerous (and often confusing) possibilities for power supply fan location, direction, speed, and venting. The designer's choice of a power supply cooling solution depends in part on the targeted end-use system application(s). At a minimum, the power supply design must ensure its own reliable and safe operation.

Fan location/direction. In general, exhausting air from the system chassis enclosure via a power supply fan at the rear panel is the preferred, most common, and most widely applicable system-level airflow solution. Other solutions are permitted, including fans on the topside of figure 5 and the Wire harness side of figure 4 or 5. Some system/chassis designers may choose to use other solutions to meet specific system cooling requirements.

Fan size/speed. An 80 mm or larger axial fan is typically needed to provide enough cooling airflow through an average ATX system. Exact CFM requirements vary by application and end-use environment, but 25-35 CFM is typical for the fan itself.

For consumer or other noise-sensitive applications, it is recommended that a thermally sensitive fan speed control circuit be used to balance system-level thermal and acoustic performance. The circuit typically senses the temperature of an internal heatsink and/or incoming ambient air and adjusts the fan speed as necessary to keep power supply and system component temperatures within specification. Both the power supply and system designers should be aware of the dependencies of the power supply and system temperatures on the control circuit response curve and fan size and should specify them very carefully.

The power supply fan should be turned off when PS_ON# is de-asserted (high). In this state, any remaining active power supply circuitry must rely only on passive convection for cooling.

Venting. In general, more venting in a power supply case yields reduced airflow impedance and improved cooling performance. Intake and exhaust vents should be as large, open, and unobstructed as possible so as not to impede airflow or generate excessive acoustic noise. In particular, avoid placing objects within 0.5 inches of the intake or exhaust of the fan itself. A flush-mount wire fan grill can be used instead of a stamped metal vent for improved airflow and reduced acoustic noise.

There are three caveats to the venting guidelines above:

- Openings must be sufficiently designed to meet the safety requirements described in Section 8.
- Larger openings yield decreased EMI-shielding performance (see Section 6).
- Venting in inappropriate locations can detrimentally allow airflow to bypass those areas where it is needed.

The *ATX Specification* offers two options for venting between the power supply and the system interior:

- The venting shown in Figure 8 provides the most effective channeled airflow for the power supply itself, with little regard for directly cooling any system components. This venting method is nearly always used in conjunction with a fan that exhausts out the rear of the power supply.
- The venting shown in Figure 9 allows designers to more directly couple the power supply airflow to system components such as the processor or motherboard core, potentially cooling all critical components with a single fan. Both the power supply fan location and direction may vary in this case. The trade-off is usually one of reduced system cost versus narrower design applicability.

4.4. AC Connector

The AC input receptacle should be an IEC 320 type or equivalent. In lieu of a dedicated switch, the IEC 320 receptacle may be considered the mains disconnect.

4.5. DC Connectors

Figure 10 shows pinouts and profiles for typical ATX power supply DC harness connectors.

Listed or recognized component appliance wiring material (AVLV2), CN, rated min 85 °C, 300 VDC shall be used for all output wiring.

There are no specific requirements for output wire harness lengths, as these are largely a function of the intended end-use chassis, motherboard, and peripherals. Ideally, wires should be short to minimize electrical/airflow impedance and simplify manufacturing, yet they should be long enough to make all necessary connections without any wire tension (which can cause disconnections during shipping and handling). Recommended minimum harness lengths for general-use power supplies are 280 mm for the +12 V power connector and 250 mm for all other wire harnesses. Measurements are made from the exit port of the power supply case to the wire side of the first connector on the harness.

NOTE

Details of the 2x3 “Optional Power Connector” mentioned in the ATX 2.03 Specification are omitted from this design guide until such time as the signals on that connector are more rigidly defined.

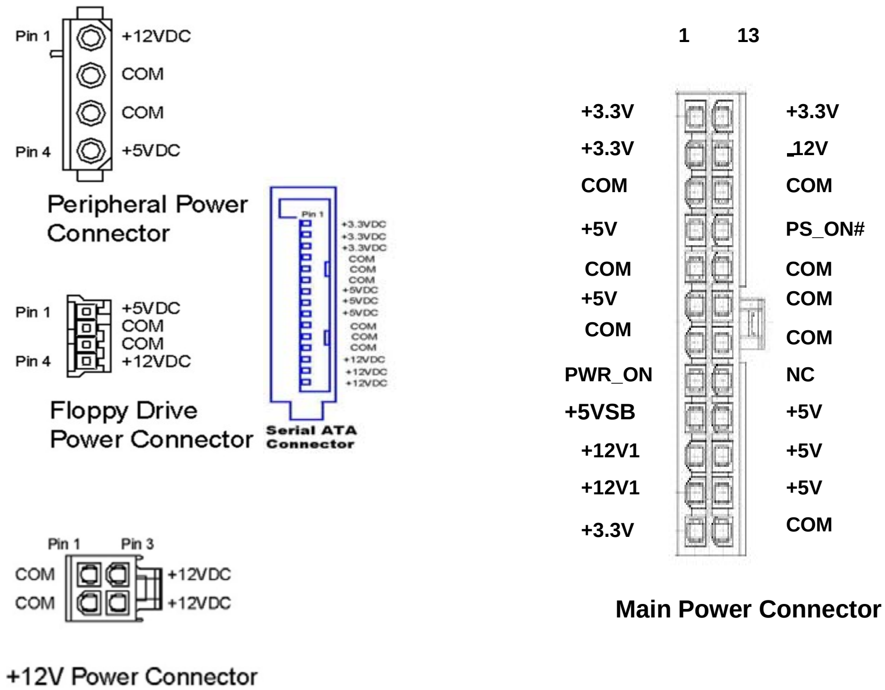


Figure 10. ATX12V Power Supply Connectors
(Pin-side view, not to scale)

4.5.1. ATX Main Power Connector

Connector: MOLEX* housing: 24 Pin Molex Mini-Fit Jr. PN# 39-01-2240 or equivalent
(Mating motherboard connector is Molex 44206-0007 or equivalent)

18 AWG is suggested for all wires except for the +3.3 V sense return wire, pin 13 (22 AWG). For 300 W configurations, 16 AWG is recommended for all +12 VDC, +5 VDC, +3.3 VDC, and COM.

Pin	Signal	Color	Pin	Signal	Color
1	+3.3VDC	Orange	13 [13]	+3.3VDC [+3.3 V default sense]	Orange [Brown]
2	+3.3VDC	Orange	14	-12VDC	Blue
3	COM	Black	15	COM	Black
4	+5VDC	Red	16	PS_ON#	Green
5	COM	Black	17	COM	Black
6	+5VDC	Red	18	COM	Black
7	COM	Black	19	COM	Black
8	PWR_OK	Gray	20	Reserved	N/C
9	+5VSB	Purple	21	+5VDC	Red
10	+12 V1DC	Yellow	22	+5VDC	Red
11	+12 V1DC	Yellow	23	+5 VDC	Red
12	+3.3 VDC	Orange	24	COM	Black

4.5.2. +12 V Power Connector

Connector: MOLEX 39-01-2040 or equivalent
(Mating motherboard connector is Molex 39-29-9042 or equivalent)

Pin	Signal	18 AWG Wire	Pin	Signal	18 AWG Wire
1	COM	Black	3	+12V2DC	Yellow /Black Stripe
2	COM	Black	4	+12V2DC	Yellow/ Black Stripe

4.5.3. Peripheral Connector(s)

Connector: AMP 1-480424-0 or MOLEX 8981-04P or equivalent.

Contacts: AMP 61314-1 or equivalent.

Pin	Signal	18 AWG Wire
1	+12V1DC	Yellow
2	COM	Black
3	COM	Black
4	+5VDC	Red

4.5.4. Serial ATA Power Connector

This is a required connector for systems with Serial ATA* devices.

The detailed requirements for the Serial ATA Power Connector can be found in the "Serial ATA: High Speed Serialized AT Attachment" specification, Section 6.3 "Cables and connector specification". http://www.serialata.org/

Assembly: Molex 88751 or equivalent

Wire	Signal	18 AWG Wire
5	+3.3 VDC	Orange
4	COM	Black
3	+5 VDC	Red
2	COM	Black
1	+12 V1DC	Yellow

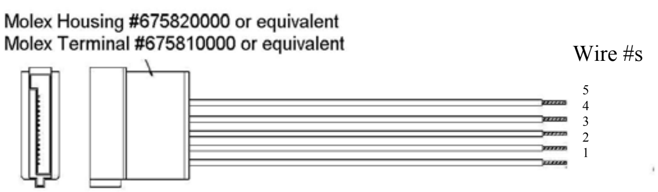


Figure 11. Serial ATA Connector

4.5.5. Floppy Drive Connector

Connector: AMP 171822-4 or equivalent		
Pin	**Signal**	**20 AWG Wire**
1	+5VDC	Red
2	COM	Black
3	COM	Black
4	+12V1DC	Yellow

5. Environmental

The following subsections define recommended environmental specifications and test parameters, based on the typical conditions to which an ATX12V power supply may be subjected during operation or shipment.

5.1. Temperature

Operating ambient	+10 °C to +50 °C
	(At full load, with a maximum temperature rate of change of 5 °C/10 minutes, but no more than 10 °C/hr.)
Non-operating ambient	-40 °C to +70 °C
	(Maximum temperature rate of change of 20 °C/hr.)

5.2. Thermal Shock (Shipping)

Non-operating	-40 °C to +70 °C
	15 °C/min ≤ dT/dt ≤ 30 °C/min Tested for 50 cycles; Duration of exposure to temperature extremes for each half cycle shall be 30 minutes.

5.3. Humidity

Operating	To 85% relative humidity (non-condensing)
Non-operating	To 95% relative humidity (non-condensing)
	Note: 95% RH is achieved with a dry bulb temperature of 55 °C and a wet bulb temperature of 54 °C.

5.4. Altitude

Operating	To 10,000 ft
Non-operating	To 50,000 ft

5.5. Mechanical Shock

Non-operating	50 g, trapezoidal input; velocity change ≥ 170 in/s
	Three drops on each of six faces are applied to each sample.

5.6. Random Vibration

Non-operating	0.01 g^2/Hz at 5 Hz, sloping to 0.02 g^2/Hz at 20 Hz, and maintaining 0.02 g^2/Hz from 20 Hz to 500 Hz. The area under the PSD curve is 3.13 gRMS. The duration shall be 10 minutes per axis for all three axes on all samples.

5.7. Acoustics

For power supplies designed for low noise, the following provides some general guidance.

Guidelines Sound Power: The power supply assembly shall not produce a declared sound power level greater than 4.0 BA. Sound power determination is to be performed at 43C, 50% of maximum rated load, at sea level. This test point is chosen to represent the environment seen inside a typical system at the idle acoustic test condition, with the 43C being derived from the standard ambient assumption of 23C, with 20C added for the temperature rise within the system (what is typically seen by the inlet fan). The declared sound power level shall be measured according to ISO 7779 and reported according to ISO 9296.

Pure Tones: The power supply assembly shall not produce any prominent discrete tone determined according to ISO 7779, Annex D.

6. Electromagnetic Compatibility

The following subsections outline sample product regulations requirements for a typical power supply. Actual requirements will depend on the design, product end use, target geography, and other variables. Consult your company's Product Safety and Regulations department for more details.

6.1. Emissions

The power supply shall comply with FCC Part 15, EN55022: 1998 and CISPR 22: 1997, meeting Class B for both conducted and radiated emissions with a 4 dB margin. Tests shall be conducted using a shielded DC output cable to a shielded load. The load shall be adjusted as follows for three tests: No load on each output; 50% load on each output; 100% load on each output. Tests will be performed at 100 VAC 50Hz, 120 VAC 60 Hz, and 230 VAC 50 Hz power.

6.2. Immunity

The power supply shall comply with EN 55024:1998.

6.3. Input Line Current Harmonic Content and Line Flicker

For sales in EU (European Union) or Japan the power supply shall meet the requirements of EN61000-3-2 Class D and the Guidelines for the Suppression of Harmonics in Appliances and General Use Equipment Class D for harmonic line current content at full rated power. See Table 16 for the harmonic limits.

Table 16. Harmonic Limits, Class D Equipment

	Per: EN 61000-3-2		*Per: JEIDA MITI*
Harmonic Order n	**Maximum permissible Harmonic current at 230 VAC / 50 Hz in Amps**		**Maximum permissible Harmonic current at 100VAC / 50 Hz in Amps**
	Odd harmonics		
3	2.3		5.29
5	1.14		2.622
7	0.77		1.771
9	0.4		0.92
11	0.33		0.759
13	0.21		0.483
15≤ n ≤39	0.15 x (15/n)		0.345 x (15/n)

6.4. Magnetic Leakage Fields

A PFC choke magnetic leakage field should not cause any interference with a high-resolution computer monitor placed next to or on top of the end-use chassis.

7. Reliability

7.1. Component De-rating

The de-rating process promotes quality and high reliability. All electronic components should be designed with conservative device de-ratings for use in commercial and industrial environments.

8. Safety

The following subsections outline sample product regulations requirements for a typical power supply. Actual requirements will depend on the design, product end use, target geography, and other variables. Consult your company's Product Safety and Regulations department for more details.

8.1. North America

The power supply must be certified by an NRTL (Nationally Recognized Testing Laboratory) for use in the USA and Canada under the following conditions:

- The supply must be recognized for use in Information Technology Equipment including Electrical Business Equipment per UL 60950, 3rd edition, 2000. The certification must include external enclosure testing for the AC receptacle side of the power supply. (see Figures 8 and 9).
- The supply must have a full complement of tests conducted as part of the certification, such as input current, leakage current, hi-pot, temperature, energy discharge test, transformer output characterization test (open-circuit voltage, short-circuit current, and maximum VA output), and abnormal testing (to include stalled-fan tests and voltage-select–switch mismatch).
- The enclosure must meet fire enclosure mechanical test requirements per clauses 2.9.1 and 4.2 of the above-mentioned standard.

Production hi-pot testing must be included as a part of the certification and indicated as such in the certification report.

There must not be unusual or difficult conditions of acceptability such as mandatory additional cooling or power de-rating. The insulation system shall not have temperatures exceeding their rating when tested in the end product.

The certification mark shall be marked on each power supply.

The power supply must be evaluated for operator-accessible secondary outputs (reinforced insulation) that meet the requirements for SELV and do not exceed 240 VA under any condition of loading.

The proper polarity between the AC input receptacle and any printed wiring boards connections must be maintained (that is, brown=line, blue=neutral, green or green/yellow=earth/chassis).

Failure of any single component in the fan-speed control circuit shall not cause the internal component temperatures to exceed the abnormal fault condition temperatures per IEC 60950 3rd ed., 1999.

8.2. International

The vendor must provide a complete CB certificate and test report to IEC 60950: 3rd ed., 1999. The CB report must include ALL CB member country national deviations. CB report must include evaluation to EN 60950: 2000. All evaluations and certifications must be for reinforced insulation between primary and secondary circuits.

8.3. Proscribed Materials

Cadmium should not be used in painting or plating.

No quaternary salt electrolytic capacitors shall be used.

Mercury shall not be used.

The use of CFCs or HFCs shall not be used in the design or manufacturing process.