ANEXOS

ANEXO I

En el anexo I se adjunta el código completo del programa desarrollado. Se han incluido comentarios para facilitar su comprensión.

```
/**Programa Completo**/

#include <LiquidCrystal.h> //Librería control LCD
#include <Wire.h>          //Libreria uso de la comunicaión I2C

int pinEncoderA=19;    // Fase A en interrupción 4
int pinEncoderB=17;    // Fase B para lectura digital
int pinBoton=18;       // Pulsador selector
int numChips=20;       // Numero de chips disponibles en tabla
int indiceOld=0;
volatile int indice=1,indiceTest=0;

//A continuación se escriben las referencias y descripciones de cada modelo de chip, lo que veremos en el display
char* ref[]={"               ","74HC00        ","74HC02        ","74HC04-4069UB  ","74HC08         ","74HC10         ",
"74HC11         ",
  "74HC27         ","74HC32         ","74HC4511       ","74HC74         ","74HC85         ","74HC86         ","74HC93
     ",
  "74HC107        ","74HC151        ","74HC163        ","74HC191        ","74HC194        ","74HC283        ","74HC390
     "};

char* descrip[]={"                    ","2-i NAND Gates (x4)","2-i NOR Gates (x4)  ","Hex inverters      ","2-i AND
Gates (x4) ","3-i NAND Gates (x3)",
  "3-i AND Gates (x3)  ","3-i NOR Gates (x3)  ","2-i OR Gates (x4)   ","BCD to 7 seg Decoder","Dual D Flip-Flops   ","4
bit Comparator    ",
  "2-i XOR Gates (x4)  ","Binary Asyn. Counter","Dual JK Flip-Flops  ","8-i Multiplexer     ","Binary Sync Counter ",
"Up/Down Bin. Counter","Bidir Shift Register",
  "Binary Full Adder   ","Dual Decade Counter "};

byte indiceVerif[]={-1,0,7,8,1,4,5,6,2,9,12,11,3,-1,13,10,14,-1,15,-1,-1};

int intensidad[3], IMax=47; //Vector de tres componentes para medir la diferencia de consumos y límite de corriente que
queremos detectar. Deja pasar hasta 7,5-10 mA

//Declaración de vectores "Test" y de vectores "Respuestas esperadas" para comprobar la serie de chips combinacionales.
//Para el puerto C
byte vectorC[][16]={{B00000000, B00010010, B00001001, B00011011},                                    /*74HC00*/

                    {B00000000, B00010010, B00001001, B00011011},                                    /*74HC08*/
```

```
                    {B00000000, B00010010, B00001001, B00011011},                                                   /*74HC32*/
                    {B00000000, B00010010, B00001001, B00011011},                                                   /*74HC86*/
                    {B00000000, B00000100, B00001001, B00001101, B00010010, B00010110, B00011011, B00011111}, /*74HC10*/
                    {B00000000, B00000100, B00001001, B00001101, B00010010, B00010110, B00011011, B00011111}, /*74HC11*/
                    {B00000000, B00000100, B00001001, B00001101, B00010010, B00010110, B00011011, B00011111}, /*74HC27*/
                    {B00000000, B00010010, B00100100, B00110110},                                                   /*74HC02*/
                    {B00000000, B00010101},                                                                         /*74HC04-
4069UB*/
                    {B00001100, B01001100, B00001101, B01001101, B00001110, B01001110, B00001111, B01001111, B00101100,
B01101100}, /*74HC4511*/
                    {0, B00001000, 0, B00000100, 0, B00000010, 0, B00000001, 0, 0, 0, 0, 0, 0, 0, 0}};         /*74HC151*/
//Para el Puerto L
byte vectorL[][16]={{B00000001, B00100101, B00010011, B00110111},
                    {B00000001, B00100101, B00010011, B00110111},
                    {B00000001, B00100101, B00010011, B00110111},
                    {B00000001, B00100101, B00010011, B00110111},
                    {B00000001, B00001011, B00010001, B00011011, B00100001, B00101011, B00110001, B00111011},
                    {B00000001, B00001011, B00010001, B00011011, B00100001, B00101011, B00110001, B00111011},
                    {B00000001, B00001011, B00010001, B00011011, B00100001, B00101011, B00110001, B00111011},
                    {B00000001, B00100101, B01001001, B01101101},
                    {B01010101, B00101011},
                    {0},
                    {B00011111, B00011111, B00111111, B00111111, B01011111, B01011111, B01111111, B01111111, B10011101,
B10011111, B10111011, B10111111, B11010111, B11011111, B11101111, B11111111}};

//Generación de las respuestas esperadas.
byte respuestaC[][16]={{B00100100, B00110110, B00101101, B00011011},
                       {B00000000, B00010010, B00001001, B00111111},
                       {B00000000, B00110110, B00101101, B00111111},
                       {B00000000, B00110110, B00101101, B00011011},
                       {B00100000, B00100100, B00101001, B00101101, B00110010, B00110110, B00111011, B00011111},
                       {B00000000, B00000100, B00001001, B00001101, B00010010, B00010110, B00011011, B00111111},
                       {B00100000, B00000100, B00001001, B00001101, B00010010, B00010110, B00011011, B00011111},
                       {B00001001, B00010010, B00100100, B00110110},
                       {B00101010, B00010101},
                       {0},
                       {B00100000, B00011000, B00100000, B00010100, B00100000, B00010010, B00100000, B00010001, B00100000,
 B00010000, B00100000, B00010000, B00100000, B00010000, B00100000, B00010000}};
```

```
byte respuestaL[][16]={{B01001001, B01101101, B01011011, B00110111},
                       {B00000001, B00100101, B00010011, B01111111},
                       {B00000001, B01101101, B01011011, B01111111},
                       {B00000001, B01101101, B01011011, B00110111},
                       {B01000101, B01001111, B01010101, B01011111, B01100101, B01101111, B01110101, B00111011},
                       {B00000001, B00001011, B00010001, B00011011, B00100001, B00101011, B00110001, B01111111},
                       {B01000101, B00001011, B00010001, B00011011, B00100001, B00101011, B00110001, B00111011},
                       {B00010011, B00100101, B01001001, B01101101},
                       {B01010101, B00101011},
                       {B11111011, B00110001, B11011101, B01111101, B00110111, B01101111, B11100111, B00111001, B11111111,
 B00111111},
                       {0}};

LiquidCrystal lcd(7,6,5,4,3,2);  //Declaración pines LCD (RS,E,DB4,DB5,DB6,DB7)


void setup()
{
 //Inicialización Display LCD
   lcd.begin(20, 4); // Nº de columnas y filas de la pantalla
   lcd.clear();

 //Configuración líneas de entrada para fases A y B del Encoder (activa resistencias de pull-up)
   pinMode(pinEncoderA, INPUT_PULLUP);
   pinMode(pinEncoderB, INPUT_PULLUP);
   pinMode(pinBoton, INPUT_PULLUP);

 //Declaración de las interrupciones
   attachInterrupt(4, SelPosicion, CHANGE);  // Encoder pin A: interrupción 4 (pin 19)
   attachInterrupt(5, StartTest, FALLING);   // Interruptor Botón : interrupción 5(pin 18)
   indice=1; //Se supone que al declararlo ya se iguala a 1, probar a quitar esto a ver si funciona

   //Configuración de los registros del ADC para medir consumo de corrientes.
   ADCSRA=0b10000111; //Activa el ADC, escribe 1 en el bit ADEN (7) de ADSRA. Los tres 1's del final indican que la
frecuencia que entra al conversor es la de reloj/128.
   ADCSRB=0b00000000; //Inicializa todos a 0 porque no se usa el trigger. El bit 3 de este registro es el bit más
significativo de las patas seleccionadas.
   ADMUX=0b10010010;  //Selecciona entradas ADC2 y ADC1 y ganancia x1 junto con el bit 3 del ADCSRB. Los dos bits más
significativos en 10 para que la referencia sea 1,1V.
```

```
   Wire.begin(); //Inicializa como maestro el bus I2C para la comunicación con los ADG715.

   pinMode(39, OUTPUT); //Reset de los ADG715.
   pinMode(14, OUTPUT); //Led Rojo
   pinMode(15, OUTPUT); //Led Verde
   pinMode(16, OUTPUT); //Zumbador

   lcd.clear();  //Borra la pantalla
   lcd.setCursor(0,0); lcd.print(ref[1]);     //Escribe la referencia del primer chip de la lista
   lcd.setCursor(0,1); lcd.print(descrip[1]); //Escribe la descripción del primer chip de la lista
   lcd.setCursor(0,2); lcd.print(descrip[0]);
   lcd.setCursor(0,3); lcd.print(descrip[0]);
}


void loop()
{
  if(indice!=indiceOld) {                      //Comprueba si se ha girado el selector
  lcd.setCursor(0,0); lcd.print(ref[indice]);  //Si es así, se escribe en el display la nueva descripción
  lcd.setCursor(0,1); lcd.print(descrip[indice]);
  lcd.setCursor(0,2); lcd.print(descrip[0]);
  lcd.setCursor(0,3); lcd.print(descrip[0]);
  indiceOld=indice;
  }

  if (indiceTest>0) {                          //Comprueba si se ha activado el pulsador
    VerificaChip(indiceVerif[indiceTest]);   //Si es así, se activa la función de verificación del chip
    }
 delay(200);
}

//Declaración de funciones

void SelPosicion() {

  byte estado;

  int MSB = digitalRead(pinEncoderA); //Considera la fase A como el bit más significativo
```

```
  int LSB = digitalRead(pinEncoderB); //Considera la fase B como el bit menos significativo

  estado = 2*MSB+LSB;   //Convierte el número binario en su correspondiente en baseo 10

  if(estado==0 || estado==3) {if (indice<numChips) {indice++;}} //Si es 0 ó 3, suma uno a indice
  if(estado==1 || estado==2) {if(indice>1){indice--;}}          //Si es 1 ó 2, resta uno a indice
 }

void StartTest() {
  indiceTest=indice;
  }

//En la siguiente función se encuentran las rutinas de verificación de cada uno de los chips
void VerificaChip(byte chip) {
  tone(16,2200,150);
  byte flagError=0;
  lcd.setCursor(0,3); lcd.print(descrip[0]);
  lcd.setCursor(0,3); lcd.print("Verificando Chip");
  byte j, s; //j Contador del bucle de verificación. s variable para el switch.
  if (chip<7)
  {
     if(chip<4){s=1;}
     else{s=2;}
  }
  else{s=chip;}

  switch(s)
  {
    case 1:    //*74HC00, 74HC08, 74HC32 y 74HC86*//
    DDRC=B11011011;   // Pines como salidas/entradas
    DDRL=B10110111;
    // Medida de consumos
    if(consumo(B10011011, B10110111, B10000000, B10000001)==1){error(); flagError=1;break;}
    // Verificación lógica
    for (j=0;j<4;j++)
      {
      PORTC=vectorC[chip][j]; PORTL=vectorL[chip][j];
      if (PINC!=respuestaC[chip][j] || PINL!=respuestaL[chip][j]){error(); flagError=1; break;}
      }
```

```
if(flagError==0){ok();}
break;

case 2:     //*74HC10, 74HC11 y 74HC27*/
DDRC=B11011111;   // Pines como salidas/entradas
DDRL=B10111011;
// Medida de consumos
if(consumo(B10011111, B10111011, B10000000, B10000001)==1){error();flagError=1; break;}
// Verificación lógica
for (j=0;j<8;j++)
  {
  PORTC=vectorC[chip][j]; PORTL=vectorL[chip][j];
  if (PINC!=respuestaC[chip][j] || PINL!=respuestaL[chip][j]){error();flagError=1; break;}
  }
if(flagError==0){ok();}
break;

case 7:    //*74HC02*//
DDRC=B11110110;   // Pines como salidas/entradas
DDRL=B11101101;
// Medida de consumos
cerrar(); delay(1);0, B11101101, B10000000
// Verificación lógica
for (j=0;j<4;j++)
  {
  PORTC=vectorC[chip][j]; PORTL=vectorL[chip][j];
  if (PINC!=respuestaC[chip][j] || PINL!=respuestaL[chip][j]){error();flagError=1; break;}
  }
if(flagError==0){ok();}
break;

case 8:    //*74HC04-4069UB*//
DDRC=B11010101;   // Pines como salidas/entradas
DDRL=B10101011;
// Medida de consumos
if(consumo(B10010101, B10101011, B10000000, B10000001)==1){error();flagError=1; break;}
// Verificación lógica
for (j=0;j<2;j++)
  {
```

```
  PORTC=vectorC[chip][j]; PORTL=vectorL[chip][j];
  if (PINC!=respuestaC[chip][j] || PINL!=respuestaL[chip][j]){error();flagError=1; break;}
  }
if(flagError==0){ok();}
break;

case 9:    //*74HC4511*//
DDRC=B11111111;   // Pines como salidas/entradas
DDRL=B00000001;
// Medida de consumos
if(consumo(B01111111, B00000001, B00000000, B00000001)==1){error();flagError=1; break;}
// Verificación lógica
digitalWrite(49, 1); //Vcc.
digitalWrite(35, 0);
delay(5);
if(PINL!=respuestaL[chip][8]){error(); flagError=1;break;}         //Primera linea de la tabla.
digitalWrite(35, 1); digitalWrite(34, 0);
delay(5);
if(PINL!=B00000001){error();flagError=1; break;}                   //Segunda linea de la tabla.
for(j=0;j<10;j++) //Numeros del 0 al 9
  {
  PORTC=vectorC[chip][j];
  delay(5);
  if(PINL!=respuestaL[chip][j]){error();flagError=1; break;}
  }
if(flagError==0){ok();}
break;

case 10:  //*74HC151*//
DDRC=B11001111;   // Pines como salidas/entradas
DDRL=B11111111;
// Medida de consumos
if(consumo(B01001111, B11111111, B00000000, B00000001)==1){error(); flagError=1;break;}
// Verificación lógica
digitalWrite(31, 1);
digitalWrite(49, 1);
digitalWrite(30, 0);
delay(5);
if(digitalRead(32)==0 || digitalRead(33)==1){error();flagError=1; break;} //Fun
```

```
    for(j=0;j<16;j++)
    {
      PORTC=vectorC[chip][j];
      PORTL=vectorL[chip][j];
      delay(5);
      if(PINC!=respuestaC[chip][j]){error();flagError=1; break;}
    }
    if(flagError==0){ok();}
    break;

    case 11:  //*74HC85*//
    {
    DDRC=B10001111;   // Pines como salidas/entradas
    DDRL=B11111111;
    // Medida de consumos
    if(consumo(B00001111, B11111111, B00000000, B00000001)==1){error(); flagError=1;break;}
    // Verificación lógica
    byte var=9;
    for(j=0;j<255;j++) //Bucle de verificación
          {
            var=j|B00000001; //Si el bit menos sgnificativo es 0 lo pone en 1, ya que este bit va a ser la alimentación
(Vcc) y siempre debe ser 1
            digitalWrite(30, 0);
            digitalWrite(37, bitRead(j, 0));
            PORTL=var; //Saca el número por el puerto L
            delay(5);
            if(bitRead(var, 1)>bitRead(j, 0)){if(digitalRead(31)==0 && digitalRead(32)==0 && digitalRead(33)==1){} //Si
cumple la condición pasa a la siguiente comprobación
                                              else{error();flagError=1; break;}
                                             }
            if(bitRead(var, 1)<bitRead(j, 0)){if(digitalRead(31)==1 && digitalRead(32)==0 && digitalRead(33)==0){} //Si
cumple la condición pasa a la siguiente comprobación
                                              else{error();flagError=1; break;}
                                             }
            if(bitRead(var, 1)==bitRead(j, 0)){//Compara los bits más significativos, si son iguales pasa a comparar los
segundos más significativos
                                                      if(bitRead(var, 3)>bitRead(var, 2)){if(digitalRead(31)==0 &&
digitalRead(32)==0 && digitalRead(33)==1){}
                                                      else{error(); flagError=1;break;}
```

```
                                                                    }
                                                        if(bitRead(var, 3)<bitRead(var, 2)){if(digitalRead(31)==1 &&
digitalRead(32)==0 && digitalRead(33)==0){}
                                                        else{error();flagError=1; break;}
                                                                  }
                                                        if(bitRead(var, 3)==bitRead(var, 2)) //Si los segundos más
significativos son iguales, pasa a comparar los terceros más significativos
                                                        {
                                                                  if(bitRead(var, 4)>bitRead(var, 5)){if(digitalRead(31)==0
&& digitalRead(32)==0 && digitalRead(33)==1){}
                                                                  else{error();flagError=1; break;}
                                                                                                        }
                                                                  if(bitRead(var, 4)<bitRead(var, 5)){if(digitalRead(31)==1
&& digitalRead(32)==0 && digitalRead(33)==0){}
                                                                  else{error();flagError=1; break;}
                                                                                                        }
                                                                  if(bitRead(var, 4)==bitRead(var, 5)) //Si los terceros
más significativos son iguales, pasa a comprobar los menos significativos
                                                                  {
                                                                            if(bitRead(var, 6)>bitRead(var,
7)){if(digitalRead(31)==0 && digitalRead(32)==0 && digitalRead(33)==1){}
                                                                            else{error();flagError=1; break;}
                                                                                                                  }
                                                                            if(bitRead(var, 6)<bitRead(var,
7)){if(digitalRead(31)==1 && digitalRead(32)==0 && digitalRead(33)==0){}
                                                                            else{error();flagError=1; break;}
                                                                                                                  }
                                                                            if(bitRead(var, 6)==bitRead(var, 7)) //Si todos
los bits son iguales, paso a la comprobación dependiente de la tres entradas de <, > o =
                                                                                {
                                                                                     digitalWrite(34, 1);digitalWrite(35,
0);digitalWrite(36, 0);
                                                                                            if(digitalRead(31)==0 &&
digitalRead(32)==0 && digitalRead(33)==1){}
                                                                                            else{error();
flagError=1;break;}
                                                                                     digitalWrite(34, 0);digitalWrite(35,
0);digitalWrite(36, 1);
                                                                                            if(digitalRead(31)==1 &&
digitalRead(32)==0 && digitalRead(33)==0){}
```

```
flagError=1;break;}

digitalRead(32)==1 && digitalRead(33)==0){}

break;}

0);digitalWrite(36, 1);

digitalRead(32)==0 && digitalRead(33)==0){}

break;}

0);digitalWrite(36, 0);

digitalRead(32)==0 && digitalRead(33)==1){}

break;}
```

```
        else{error();

digitalWrite(35, 1);
        if(digitalRead(31)==0 &&

        else{error();flagError=1;

digitalWrite(34, 1);digitalWrite(35,

        if(digitalRead(31)==0 &&

        else{error();flagError=1;

digitalWrite(34, 0);digitalWrite(35,

        if(digitalRead(31)==1 &&

        else{error();flagError=1;
```

```
                }
            }
        }
    }
        }
    if(flagError==0){ok();}
    }
    break;

    case 12:     ///*74HC74*//
    DDRC=B11001111;   // Pines como salidas/entradas
    DDRL=B10011111;
    // Medida de consumos
    if(consumo(B10001111, B10011111, B10000000, B10000001)==1){error();flagError=1; break;}
    // Verificación lógica
    PORTC=B10000001;//Comprobación de la primera linea de la tabla. Los dos flip-flop D a la vez
    PORTL=B10000011;
    delay(5);
    if(PINC!=B10010001 || PINL!=B10100011){error();flagError=1; break;}//error
```

```
PORTC=B10001000;//Comprobación de la segunda linea de la tabla. Los dos flip-flop D a la vez
PORTL=B10010001;
delay(5);
if(PINC!=B10101000 || PINL!=B11010001){error();flagError=1; break;}//error

PORTC=B10000000;//Comprobación de la tercera linea de la tabla. Los dos flip-flop D a la vez
PORTL=B10000001;
delay(5);
if(PINC!=B10110000 || PINL!=B11100001){error();flagError=1; break;}//error

PORTC=B10001011;//Cuarta linea tambien los dos a la vez. Pone el reloj en 0 ya
PORTL=B10010111;
subida(35);// Dos subidas de reloj (Una para cada flip-flop)
subida(46);
delay(5);
if(digitalRead(33)==0 || digitalRead(32)==1 || digitalRead(44)==0 || digitalRead(43)==1){error();flagError=1; break;}

PORTC=B10001001;//Quinta linea también los dos a la vez. Pone el reloj en 0 ya
PORTL=B10010011;
subida(35);// Dos subidas(Una para cada flip-flop)
subida(46);
delay(5);
if(digitalRead(33)==1 || digitalRead(32)==0 || digitalRead(44)==1 || digitalRead(43)==0){error();flagError=1; break;}

// Septima linea tmb los dos a la vez. Deja el reloj en 1 y comprueba que no cambien las salidas
digitalWrite(35, 1);
digitalWrite(46, 1);
delay(5);
if(digitalRead(33)==1 || digitalRead(32)==0 || digitalRead(44)==1 || digitalRead(43)==0){error();flagError=1; break;}

// Sexta linea. Pone los relojes a 0 y comprueba que no cambia.
digitalWrite(35, 0);//Pro
digitalWrite(46, 0);
delay(5);
if(digitalRead(33)==1 || digitalRead(32)==0 || digitalRead(44)==1 || digitalRead(43)==0){error();flagError=1; break;}
if(flagError==0){ok();}
break;

case 13:     //*74HC107*//
```

```
DDRC=B11001001;// Pines como salidas/entradas
DDRL=B11111111;
// Medida de consumos
if(consumo(B10001001, B11111111, B10000000, B10000001)==1){error();flagError=1; break;}
// Verificación lógica
PORTC=B10001001;
PORTL=B11101101;//Primera linea de la tabla
delay(5);
if(PINC!=B10101011){error();flagError=1;flagError=1; break;}

PORTL=B11111111;//Sexta linea de la tabla
delay(5);
if(PINC!=B10101011){error(); flagError=1;break;}// No cambio

PORTC=B10000000;//Segunda linea de la tabla
PORTL=B10110111;//Pone el reloj ya en alto para producir las bajadas
bajada(47);//Provoca las dos bajadas del reloj simulaneamente
bajada(44);
delay(10);
if(PINC!=B10100010){error(); flagError=1;break;}//No cambio


PORTC=B10000001;//Tercera linea de la tabla.
PORTL=B11110111;//Pone el reloj ya en alto para producir las bajadas
delay(5);
bajada(44);//Provoca las dos bajadas del reloj simulaneamente
bajada(47);
delay(5);
if(PINC!=B10010101){error();flagError=1; break;}

PORTC=B10001000;//Cuarta linea de la tabla
PORTL=B10111111;//Pone el reloj ya en alto para producir las bajadas
bajada(47);//Provoca las dos bajadas del reloj simulaneamente
bajada(44);
delay(5);
if(PINC!=B10101010){error();flagError=1; break;}

PORTC=B10001001;//Quinta linea de la tabla
PORTL=B11111111;
```

```
delay(5);
bajada(47);//Provoca las dos bajadas del reloj simulaneamente
bajada(44);
delay(5);
if(PINC!=B10011101){error();flagError=1; break;}
digitalWrite(47, 1);//Sube los dos relojes a 1
digitalWrite(44, 1);
bajada(47);//Provoca las dos bajadas del reloj simulaneamente
bajada(44);
delay(10);
if(PINC!=B10101011){error(); flagError=1;break;}
if(flagError==0){ok();}
break;

case 14:   //*74HC163*//
DDRC=B11111111; // Pines como salidas/entradas
DDRL=B11000001;
// Medida de consumos
if(consumo(B01111111, B11000001, B00000000, B00000001)==1){error();flagError=1; break;}
// Verificación lógica
PORTC=B00000000;//Prueba del Reset
PORTL=B00000001;
subida(36);
if(PINL!=B00000001){error();flagError=1;break;}//error

PORTC=B01111101;//Prueba de la carga en paralelo. Carga todo 1's
PORTL=B01000001;
subida(36);
if(PINL!=B01111111){error();flagError=1;break;}// Comprueba a la vez el carry
subida(36);
if(PINL!=B01111111){error();flagError=1;break;}// Comprueba que no cuenta
PORTC=B01000001; //Carga ahora todo 0's
subida(36);
if(PINL!=B01000001){error();flagError=1;break;}
subida(36);
if(PINL!=B01000001){error();flagError=1;break;}// Comprueba que no cuenta

PORTC=B00000000;// Reset para comprobar si cuenta, pone las salidas Q todas en 0
PORTL=B00000001;
```

```
    subida(36);
    PORTC=B01000001;//Comienza a comprobar si cuenta.
    PORTL=B11000001;
    for(j=193;j<250;j=j+4)
      {
        if(PINL!=j){error();flagError=1;break;}
        subida(36);
      }
    if(PINL!=B11111111){error();flagError=1;break;}// Este fuera del bucle para verificar también el Carry.
    subida(36);
    if(PINL!=193){error();flagError=1;break;}// Prueba a ver si vuelve a 0000.`

    PORTC=B00000001;//Comprueba que no cuenta si Enable P en L y enable T en H
    PORTL=B11000001;
    subida(36);
    if(PINL!=193){error();flagError=1;break;}// Prueba si sigue en 0000
    PORTC=B01111101;//Carga el 1111 para comprobar el carry
    PORTL=B01000001;
    subida(36);
    PORTC=B00000001;//Vuelve a la configuracion Enable P en L y enable T en H a ver si el carry está en 1
    PORTL=B11000001;
    if(PINL!=B11111111){error();flagError=1;break;}// Comprueba el Carry

    PORTC=B00000001;//Comprueba que no cuenta si Enable P en X y enable T en L, y a la vez mira que el carry es 0 aunque
las Qs sean 1111
    PORTL=B10000001;
    subida(36);
    if(PINL!=B10111101){error();flagError=1;break;}// Comprueba que el Carry es 0

    PORTC=B00000000;//Vuelve a comprobar el Reset
    PORTL=B00000001;
    subida(36);
    if(PINL!=B00000001){error();flagError=1;break;}//error.
    if(flagError==0){ok();}
    break;

    case 15:    //*74HC194*//
    {
    byte acum;
```

```
DDRC=B11111111; // Pines como salidas/entradas
DDRL=B11100001;
// Medida de consumos.
if(consumo(B01111111, B11100001, B00000000, B00000001)==1){error();flagError=1; break;}
// Verificación lógica
PORTC=B00000000; // Prueba del reset
PORTL=B00000001;
if(PINL!=B00000001){error();flagError=1; break;}

PORTC=B00000001; // Prueba la carga en paralelo de 0000
PORTL=B11000001;
subida(44);
if (PINL!=B11000001){error();flagError=1; break;}
PORTC=B00111101; // Prueba la carga en paralelo de 1111
subida(44);
if (PINL!=B11011111){error();flagError=1; break;}

PORTC=B00000000; // Vuelve a comprobar el reset
PORTL=B00000001;
if(PINL!=B00000001){error(); flagError=1;break;}

PORTC=B00000011; // Prueba el desplazamiento a la derecha
PORTL=B10000001;
acum=1;
for(j=129;j<=159;j=j+acum)
  {
   if(PINL!=j){error();flagError=1; break;}
   acum=2*acum;
   subida(44);
  }

PORTC=B00000001; // Prueba el desplazamiento a la derecha
PORTL=B10000001;
acum=1;
for(j=159;j>=129;j=j-acum)
 {
   if(PINL!=j){error();flagError=1; break;}
   acum=2*acum;
   subida(44);
```

```
     }

    PORTC=B01000001; // Prueba el desplazamiento a la izquierda
    PORTL=B01000001;
    acum=32;
    for(j=65;j<=95;j=j+acum)
     {
       if(PINL!=j){error();flagError=1; break;}
       acum=acum/2;
       subida(44);
     }
    PORTC=B00000001; // Prueba el desplazamiento a la izquierda
    PORTL=B01000001;
    acum=32;
    for(j=95;j>=65;j=j-acum)
     {
       if(PINL!=j){error();flagError=1; break;}
       acum=acum/2;
       subida(44);
     }

    PORTC=B01111111; // Prueba el hold
    PORTL=B00000001;
    subida(44);
    if(PINL!=B00000001){error();flagError=1; break;} // Comprueba si sigue en el estado en el que ha quedado antes
    PORTL=B01100001; // Con el reloj en 1 tampoco debe cambiar. Independientemente de S0 y S1 (Por eso pongo S1 en 1)
    if(PINL!=B01100001){error();flagError=1; break;}
    PORTL=B10000001; // Con el reloj en 0 tampoco debe cambiar. Independientemente de S0 y S1 (Por eso pongo S0 en 1)
    if(PINL!=B10000001){error();flagError=1; break;}
    if(flagError==0){ok();}
    }
    break;

   }
  lcd.setCursor(0,3); lcd.print(descrip[0]);
  indiceTest=0;
  }//Fin de VerificaChip

byte consumo (byte puertoC1, byte puertoL1, byte puertoC0, byte puertoL0)
```

```
{
    byte Iperm=0;
    digitalWrite(39, 0); // Reset a los ADG715
    delay(1);
    intensidad[0]=MedirConsumo(); //Medida de consumo
    cerrar(); delay(1);
    PORTC=puertoC1;   PORTL=puertoL1; delay(10); //Configuración de salidas en 1
    intensidad[1]=MedirConsumo(); // Medida de consumo
    PORTC=puertoC0;   PORTL=puertoL0; delay(10); //Configuración de salidas en 0
    intensidad[2]=MedirConsumo(); //Medida de consumo
    if(intensidad[1]-intensidad[0]>IMax || intensidad[2]-intensidad[0]>IMax){Iperm=1;} //Comparaciones de consumos
    return Iperm;
}
int MedirConsumo()
{
  int val=0;
  bitSet(ADCSRA, ADSC); //Inicio al conversión, set el bit ADSC
  while(bitRead(ADCSRA, ADSC)); //Espero a que termine la conversión, durante la conversión, el bit ADSC está en 1,
cuando acaba lo pone en 0
  val=256*ADCH+ADCL; //Coge el resultado, que está repartido en dos registros (ADCL y ADCH) y lo convierte en un entero
de 10 bits (número entre 0 y 1023)
  if(bitRead(ADCH, 1)==1) //Paso CA2, en caso de ser negativo (el bit de signo es el bit 1 de ADCH, por eso miro si este
bit es 1), calcula su valor.Obtiene así el número negativo
  {
  val=-1024+val;
  }
  return (val);
}
void error()
{
  digitalWrite(39, 0); //Reset a los ADG715 para abrir los switches
  digitalWrite(14, 1); // Enciende el led rojo
  lcd.setCursor(0,3); lcd.print(descrip[0]); // Escribe en el display "Chip defectuoso"
  lcd.setCursor(0,3); lcd.print("Chip defectuoso");
  tone(16,2200,100);// Doble pitido de error
  delay(100);
  tone(16,2200,100);
  delay(1000);
  digitalWrite(14, 0); // Apaga el led
```

```
}
void ok()
{
  digitalWrite(39, 0); //Reset a los ADG715 para abrir los switches
  digitalWrite(15, 1); // Enciende el led verde
  lcd.setCursor(0,3); lcd.print(descrip[0]); // Escribe en el display "Chip correcto"
  lcd.setCursor(0,3); lcd.print("Chip correcto");
  tone(16,2200,150);
  delay(1000);
  digitalWrite(15, 0); // Apaga el led
}
void subida (byte Psub)
{
  digitalWrite(Psub, 0);//Inicialmente en 0
  delay(1);
  digitalWrite(Psub, 1);//Subida del reloj
  delay(1); //Espera de un milisegundo para respetar el tiempo de respuesta
  digitalWrite(Psub, 0); //Vuelta a 0
}
void bajada (byte Pbaj)
{
  digitalWrite(Pbaj, 1);//Inicialmente en 1
  delay(1);
  digitalWrite(Pbaj, 0);//Bajada del reloj
  delay(1); //Espera de un milisegundo para respetar el tiempo de respuesta
  digitalWrite(Pbaj, 1); //Vuelta a 1
}
void cerrar() //Cierro los switches de los ADG715
{
    digitalWrite(39, 1);
    Wire.beginTransmission(0b1001000);
    Wire.write(0b11111111);// Cierra todos los switches
    Wire.endTransmission();
    Wire.beginTransmission(0b1001001);
    Wire.write(0b11111111);// Cierra todos los switches
    Wire.endTransmission();
}
```

ANEXO II

Se presentan, en este anexo, los *Datasheet* de los componentes usados para la realización de este trabajo:

-ADG715

-LCD

-Rotary Encoder

-Sección "*ADC – Analog to Digital Converter"* del *datasheet* completo del ATMega2560

-Buzzer

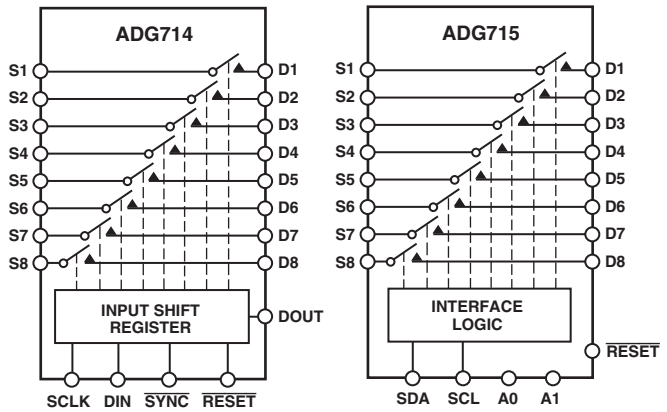
CMOS, Low Voltage Serially Controlled, Octal SPST Switches

ADG714/ADG715

FEATURES
- ADG714 SPI™/QSPI™/MICROWIRE™-Compatible Interface
- ADG715 I²C™-Compatible Interface
- 2.7 V to 5.5 V Single Supply
- ±2.5 V Dual Supply
- 2.5 Ω On Resistance
- 0.6 Ω On Resistance Flatness
- 100 pA Leakage Currents
- Octal SPST
- Power-On Reset
- Fast Switching Times
- TTL/CMOS-Compatible
- Small TSSOP Package

APPLICATIONS
- Data Acquisition Systems
- Communication Systems
- Relay Replacement
- Audio and Video Switching

FUNCTIONAL BLOCK DIAGRAMS



GENERAL DESCRIPTION

The ADG714/ADG715 are CMOS, octal SPST (single-pole, single-throw) switches controlled via either a 2- or 3-wire serial interface. On resistance is closely matched between switches and very flat over the full signal range. Each switch conducts equally well in both directions and the input signal range extends to the supplies. Data is written to these devices in the form of 8 bits, each bit corresponding to one channel.

The ADG714 uses a 3-wire serial interface that is compatible with SPI , QSPI, and MICROWIRE and most DSP interface standards. The output of the shift register DOUT enables a number of these parts to be daisy chained.

The ADG715 uses a 2-wire serial interface that is compatible with the I²C interface standard. The ADG715 has four hard wired addresses, selectable from two external address pins (A0 and A1). This allows the 2 LSBs of the 7-bit slave address to be set by the user. A maximum of four of these devices may be connected to the bus.

On power-up of these devices, all switches are in the OFF condition, and the internal registers contain all zeros.

Low power consumption and operating supply range of 2.7 V to 5.5 V make this part ideal for many applications. These parts may also be supplied from a dual ±2.5 V supply. The ADG714 and ADG715 are available in a small 24-lead TSSOP package.

PRODUCT HIGHLIGHTS

1. 2- or 3-wire serial interface
2. Single/dual supply operation. The ADG714 and ADG715 are fully specified and guaranteed with 3 V, 5 V, and ±2.5 V supply rails.
3. Low on resistance, typically 2.5 Ω
4. Low leakage
5. Power-on reset
6. Small 24-lead TSSOP package

REV. C

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 www.analog.com
Fax: 781/461-3113

ADG714/ADG715–SPECIFICATIONS[1] (V_{DD} = 5 V ± 10%, V_{SS} = 0 V, GND = 0 V unless otherwise noted.)

Parameter	B Version +25°C	B Version –40°C to +85°C	Unit	Test Conditions/Comments
ANALOG SWITCH				
Analog Signal Range		0 V to V_{DD}	V	
On Resistance (R_{ON})	2.5		Ω typ	V_S = 0 V to V_{DD}, I_S = 10 mA
	4.5	5	Ω max	
On Resistance Match Between Channels (ΔR_{ON})		0.4	Ω typ	
		0.8	Ω max	V_S = 0 V to V_{DD}, I_S = 10 mA
On Resistance Flatness ($R_{FLAT(ON)}$)	0.6		Ω typ	V_S = 0 V to V_{DD}, I_S = 10 mA
		1.2	Ω max	
LEAKAGE CURRENTS				V_{DD} = 5.5 V
Source OFF Leakage I_S (OFF)	±0.01		nA typ	V_D = 4.5 V/1 V, V_S = 1 V/4.5 V
	±0.1	±0.3	nA max	
Drain OFF Leakage I_D (OFF)	±0.01		nA typ	V_D = 4.5 V/1 V, V_S = 1 V/4.5 V
	±0.1	±0.3	nA max	
Channel ON Leakage I_D, I_S (ON)	±0.01		nA typ	V_D = V_S = 1 V, or 4.5 V
	±0.1	±0.3	nA max	
DIGITAL INPUTS (SCLK, DIN, $\overline{SYNC}$, A0, A1)				
Input High Voltage, V_{INH}		2.4	V min	
Input Low Voltage, V_{INL}		0.8	V max	
Input Current, I_{INL} or I_{INH}	0.005		μA typ	V_{IN} = V_{INL} or V_{INH}
		±0.1	μA max	
C_{IN}, Digital Input Capacitance[2]	3		pF typ	
DIGITAL OUTPUT ADG714 DOUT[2]				
Output Low Voltage		0.4	V max	I_{SINK} = 6 mA
C_{OUT} Digital Output Capacitance	4		pF typ	
DIGITAL INPUTS (SCL, SDA)[2]				
Input High Voltage, V_{INH}		0.7 V_{DD}	V min	
		V_{DD} + 0.3	V max	
Input Low Voltage, V_{INL}		–0.3	V min	
		0.3 V_{DD}	V max	
I_{IN}, Input Leakage Current	0.005		μA typ	V_{IN} = 0 V to V_{DD}
		±1	μA max	
V_{HYST}, Input Hysteresis	0.05 V_{DD}		V min	
C_{IN}, Input Capacitance	6		pF typ	
LOGIC OUTPUT (SDA)[2]				
V_{OL}, Output Low Voltage		0.4	V max	I_{SINK} = 3 mA
		0.6	V max	I_{SINK} = 6 mA
DYNAMIC CHARACTERISTICS[2]				
t_{ON} ADG714	20		ns typ	V_S = 3 V, R_L = 300 Ω, C_L = 35 pF
		32	ns max	
t_{ON} ADG715	95		ns typ	V_S = 3 V, R_L = 300 Ω, C_L = 35 pF
		140	ns max	
t_{OFF} ADG714	8		ns typ	V_S = 3 V, R_L = 300 Ω, C_L = 35 pF
		15	ns max	
t_{OFF} ADG715	85		ns typ	V_S = 3 V, R_L = 300 Ω, C_L = 35 pF
		130	ns max	
Break-Before-Make Time Delay, t_D	8		ns typ	V_S = 3 V, R_L = 300 Ω, C_L = 35 pF
		1	ns min	
Charge Injection	±3		pC typ	V_S = 2 V, R_S = 0 Ω, C_L = 1 nF
Off Isolation	–60		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 10 MHz
	–80		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 1 MHz
Channel-to-Channel Crosstalk	–70		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 10 MHz
	–90		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 1 MHz
–3 dB Bandwidth	155		MHz typ	R_L = 50 Ω, C_L = 5 pF
C_S (OFF)	11		pF typ	
C_D (OFF)	11		pF typ	
C_D, C_S (ON)	22		pF typ	
POWER REQUIREMENTS				V_{DD} = 5.5 V
I_{DD}	10		μA typ	Digital Inputs = 0 V or 5.5 V
		20	μA max	

NOTES
[1]Temperature range is as follows: B Version: –40°C to +85°C.
[2]Guaranteed by design, not subject to production test.

Specifications subject to change without notice.

SPECIFICATIONS[1] (V_{DD} = 3 V ±10%, V_{SS} = 0 V, GND = 0 V unless otherwise noted.)

Parameter	B Version +25°C	B Version –40°C to +85°C	Unit	Test Conditions/Comments
ANALOG SWITCH				
Analog Signal Range		0 V to V_{DD}	V	
On Resistance (R_{ON})	6		Ω typ	V_S = 0 V to V_{DD}, I_S = 10 mA
	11	12	Ω max	
On Resistance Match Between Channels (ΔR_{ON})		0.4	Ω typ	V_S = 0 V to V_{DD}, I_S = 10 mA
		1.2	Ω max	
On Resistance Flatness ($R_{FLAT(ON)}$)		3.5	Ω typ	V_S = 0 V to V_{DD}, I_S = 10 mA
LEAKAGE CURRENTS				V_{DD} = 3.3 V
Source OFF Leakage I_S (OFF)	±0.01		nA typ	V_S = 3 V/1 V, V_D = 1 V/3 V
	±0.1	±0.3	nA max	
Drain OFF Leakage I_D (OFF)	±0.01		nA typ	V_S = 1 V/3 V, V_D = 3 V/1 V
	±0.1	±0.3	nA max	
Channel ON Leakage I_D, I_S (ON)	±0.01		nA typ	V_S = V_D = 1 V, or 3 V
	±0.1	±0.3	nA max	
DIGITAL INPUTS (SCLK, DIN, $\overline{SYNC}$, A0, A1)				
Input High Voltage, V_{INH}		2.0	V min	
Input Low Voltage, V_{INL}		0.8	V max	
Input Current, I_{INL} or I_{INH}	0.005		μA typ	V_{IN} = V_{INL} or V_{INH}
		±0.1	μA max	
C_{IN}, Digital Input Capacitance[2]	3		pF typ	
DIGITAL OUTPUT ADG714 DOUT[2]				
Output Low Voltage		0.4	V max	I_{SINK} = 6 mA
C_{OUT} Digital Output Capacitance	4		pF typ	
DIGITAL INPUTS (SCL, SDA)[2]				
Input High Voltage, V_{INH}		0.7 V_{DD}	V min	
		V_{DD} + 0.3	V max	
Input Low Voltage, V_{INL}		–0.3	V min	
		0.3 V_{DD}	V max	
I_{IN}, Input Leakage Current	0.005		μA typ	V_{IN} = 0 V to V_{DD}
		±1	μA max	
V_{HYST}, Input Hysteresis	0.05 V_{DD}		V min	
C_{IN}, Input Capacitance	6		pF typ	
LOGIC OUTPUT (SDA)[2]				
V_{OL}, Output Low Voltage		0.4	V max	I_{SINK} = 3 mA
		0.6	V max	I_{SINK} = 6 mA
DYNAMIC CHARACTERISTICS[2]				
t_{ON} ADG714	35		ns typ	V_S = 2 V, R_L = 300 Ω, C_L = 35 pF
		65	ns max	
t_{ON} ADG715	130		ns typ	V_S = 2 V, R_L = 300 Ω, C_L = 35 pF
		200	ns max	
t_{OFF} ADG714	11		ns typ	V_S = 2 V, R_L = 300 Ω, C_L = 35 pF
		20	ns max	
t_{OFF} ADG715	115		ns typ	V_S = 2 V, R_L = 300 Ω, C_L = 35 pF
		180	ns max	
Break-Before-Make Time Delay, t_D	8		ns typ	V_S = 2 V, R_L = 300 Ω, C_L = 35 pF
		1	ns min	
Charge Injection	±2		pC typ	V_S = 1.5 V, R_S = 0 Ω, C_L = 1 nF
Off Isolation	–60		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 10 MHz
	–80		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 1 MHz
Channel-to-Channel Crosstalk	–70		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 10 MHz
	–90		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 1 MHz
–3 dB Bandwidth	155		MHz typ	R_L = 50 Ω, C_L = 5 pF
C_S (OFF)	11		pF typ	
C_D (OFF)	11		pF typ	
C_D, C_S (ON)	22		pF typ	
POWER REQUIREMENTS				V_{DD} = 3.3 V
I_{DD}	10		μA typ	Digital Inputs = 0 V or 3.3 V
		20	μA max	

NOTES
[1]Temperature range is as follows: B Version: –40°C to +85°C.
[2]Guaranteed by design, not subject to production test.

Specifications subject to change without notice.

ADG714/ADG715–SPECIFICATIONS[1]

DUAL SUPPLY (V_{DD} = +2.5 V ± 10%, V_{SS} = –2.5 V ± 10%, GND = 0 V unless otherwise noted.)

Parameter	B Version +25°C	B Version –40°C to +85°C	Unit	Test Conditions/Comments
ANALOG SWITCH				
Analog Signal Range		V_{SS} to V_{DD}	V	
On Resistance (R_{ON})	2.5		Ω typ	V_S = V_{SS} to V_{DD}, I_{DS} = 10 mA
	4.5	5	Ω max	
On Resistance Match Between Channels (ΔR_{ON})		0.4	Ω typ	V_S = V_{SS} to V_{DD}, I_{DS} = 10 mA
		0.8	Ω max	
On Resistance Flatness ($R_{FLAT(ON)}$)	0.6		Ω typ	V_S = V_{SS} to V_{DD}, I_{DS} = 10 mA
		1	Ω max	
LEAKAGE CURRENTS				V_{DD} = +2.75 V, V_{SS} = –2.75 V
Source OFF Leakage I_S (OFF)	±0.01		nA typ	V_S = +2.25 V/–1.25 V, V_D = –1.25 V/+2.25 V
	±0.1	±0.3	nA max	
Drain OFF Leakage I_D (OFF)	±0.01		nA typ	V_S = +2.25 V/–1.25 V, V_D = –1.25 V/+2.25 V
	±0.1	±0.3	nA max	
Channel ON Leakage I_D, I_S (ON)	±0.01		nA typ	V_S = V_D = +2.25 V/–1.25 V
	±0.1	±0.3	nA max	
DIGITAL INPUTS				
Input High Voltage, V_{INH}		1.7	V min	
Input Low Voltage, V_{INL}		0.7	V max	
Input Current, I_{INL} or I_{INH}	0.005		μA typ	V_{IN} = V_{INL} or V_{INH}
		±0.1	μA max	
C_{IN}, Digital Input Capacitance[2]	3		pF typ	
DIGITAL OUTPUT ADG714 DOUT[2]				
Output Low Voltage		0.4	V max	I_{SINK} = 6 mA
C_{OUT} Digital Output Capacitance	4		pF typ	
DIGITAL INPUTS (SCL, SDA)[2]				
Input High Voltage, V_{INH}		0.7 V_{DD}	V min	
		V_{DD} + 0.3	V max	
Input Low Voltage, V_{INL}		–0.3	V min	
		0.3 V_{DD}	V max	
I_{IN}, Input Leakage Current	0.005		μA typ	V_{IN} = 0 V to V_{DD}
		±1	μA max	
V_{HYST}, Input Hysteresis	0.05 V_{DD}		V min	
C_{IN}, Input Capacitance	6		pF typ	
LOGIC OUTPUT (SDA)[2]				
V_{OL}, Output Low Voltage		0.4	V max	I_{SINK} = 3 mA
		0.6	V max	I_{SINK} = 6 mA
DYNAMIC CHARACTERISTICS[2]				
t_{ON} ADG714	20		ns typ	V_S = 1.5 V, R_L = 300 Ω, C_L = 35 pF
		32	ns max	
t_{ON} ADG715	133		ns typ	V_S = 1.5 V, R_L = 300 Ω, C_L = 35 pF
		200	ns max	
t_{OFF} ADG714	8		ns typ	V_S = 1.5 V, R_L = 300 Ω, C_L = 35 pF
		18	ns max	
t_{OFF} ADG715	124		ns typ	V_S = 1.5 V, R_L = 300 Ω, C_L = 35 pF
		190	ns max	
Break-Before-Make Time Delay, t_D	8		ns typ	V_S = 1.5 V, R_L = 300 Ω, C_L = 35 pF
		1	ns min	
Charge Injection	±3		pC typ	V_S = 0 V, R_S = 0 Ω, C_L = 1 nF
Off Isolation	–60		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 10 MHz
	–80		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 1 MHz
Channel-to-Channel Crosstalk	–70		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 10 MHz
	–90		dB typ	R_L = 50 Ω, C_L = 5 pF, f = 1 MHz
–3 dB Bandwidth	155		MHz typ	R_L = 50 Ω, C_L = 5 pF
C_S (OFF)	11		pF typ	
C_D (OFF)	11		pF typ	
C_D, C_S (ON)	22		pF typ	
POWER REQUIREMENTS				V_{DD} = +2.75 V, V_{SS} = –2.75 V
I_{DD}	15		μA typ	Digital Inputs = 0 V or V_{DD}
		25	μA max	
I_{SS}	15		μA typ	
		25	μA max	

NOTES
[1]Temperature range is as follows: B Version: –40°C to +85°C.
[2]Guaranteed by design, not subject to production test.

Specifications subject to change without notice.

ADG714 TIMING CHARACTERISTICS[1, 2] (V_{DD} = 2.7 V to 5.5 V. All specifications –40°C to +85°C unless otherwise noted.)

Parameter	Limit at T_{MIN}, T_{MAX}	Unit	Conditions/Comments
f_{SCLK}	30	MHz max	SCLK Cycle Frequency
t_1	33	ns min	SCLK Cycle Time
t_2	13	ns min	SCLK High Time
t_3	13	ns min	SCLK Low Time
t_4	0	ns min	$\overline{SYNC}$ to SCLK Rising Edge Setup Time
t_5	5	ns min	Data Setup Time
t_6	4.5	ns min	Data Hold Time
t_7	0	ns min	SCLK Falling Edge to $\overline{SYNC}$ Rising Edge
t_8	33	ns min	Minimum $\overline{SYNC}$ High Time
t_9[3]	20	ns max	SCLK Rising Edge to DOUT Valid

NOTES
[1]See Figure 1.
[2]All input signals are specified with tr = tf = 5 ns (10% to 90% of V_{DD}) and timed from a voltage level of $(V_{IL} + V_{IH})/2$.
[3]C_L = 20 pF, R_L = 1 kΩ.

Specifications subject to change without notice.

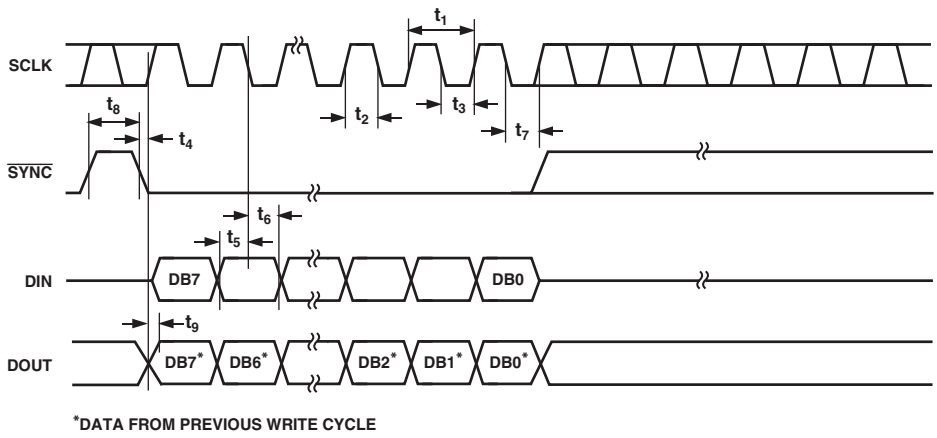


Figure 1. 3-Wire Serial Interface Timing Diagram

ADG715 TIMING CHARACTERISTICS[1] (V_{DD} = 2.7 V to 5.5 V. All specifications –40°C to +85°C unless otherwise noted.)

Parameter	Limit at T_{MIN}, T_{MAX}	Unit	Conditions/Comments
f_{SCL}	400	kHz max	SCL Clock Frequency
t_1	2.5	µs min	SCL Cycle Time
t_2	0.6	µs min	t_{HIGH}, SCL High Time
t_3	1.3	µs min	t_{LOW}, SCL Low Time
t_4	0.6	µs min	$t_{HD, STA}$, Start/Repeated Start Condition Hold Time
t_5	100	ns min	$t_{SU, DAT}$, Data Setup Time
t_6[2]	0.9	µs max	$t_{HD, DAT}$, Data Hold Time
	0	µs min	
t_7	0.6	µs min	$t_{SU, STA}$, Setup Time for Repeated Start
t_8	0.6	µs min	$t_{SU, STO}$, Stop Condition Setup Time
t_9	1.3	µs min	t_{BUF}, Bus Free Time Between a STOP Condition and a Start Condition
t_{10}	300	ns max	t_R, Rise Time of Both SCL and SDA When Receiving
	$20 + 0.1C_b$[3]	ns min	
t_{11}	250	ns max	t_F, Fall Time of SDA When Receiving
t_{11}	300	ns max	t_F, Fall Time of SDA When Transmitting
	$0.1C_b$[3]	ns min	
C_b	400	pF max	Capacitive Load for Each Bus Line
t_{SP}[4]	50	ns max	Pulsewidth of Spike Suppressed

NOTES
[1]See Figure 2.
[2]A master device must provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IH} min of the SCL signal) in order to bridge the undefined region of SCL's falling edge.
[3]C_b is the total capacitance of one bus line in pF. t_R and t_F measured between 0.3 V_{DD} and 0.7 V_{DD}.
[4]Input filtering on both the SCL and SDA inputs suppress noise spikes that are less than 50 ns.

Specifications subject to change without notice.

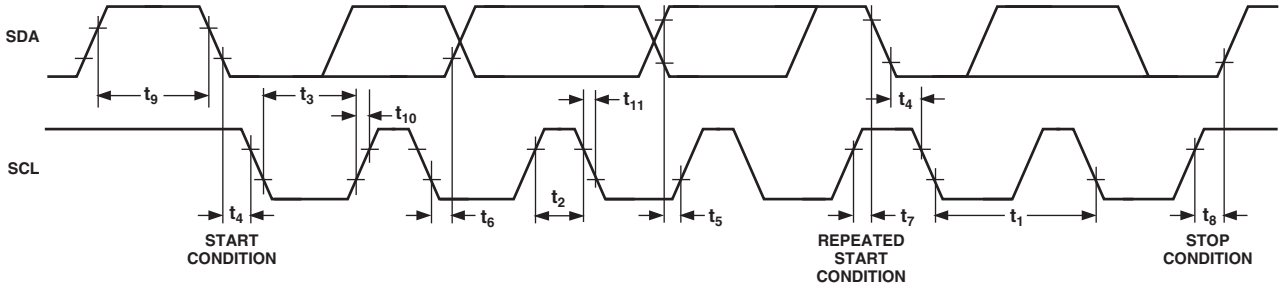


Figure 2. 2-Wire Serial Interface Timing Diagram

ABSOLUTE MAXIMUM RATINGS[1]
(T_A = 25°C unless otherwise noted.)

V_{DD} to V_{SS} . . . . . 7 V
V_{DD} to GND . . . . . –0.3 V to +7 V
V_{SS} to GND . . . . . +0.3 V to –3.5 V
Analog Inputs[2] . . . . . V_{SS} –0.3 V to V_{DD} +0.3 V
or 30 mA, Whichever Occurs First
Digital Inputs[2] . . . . . –0.3 V to V_{DD} +0.3 V
or 30 mA, Whichever Occurs First
Peak Current, S or D . . . . . 100 mA
(Pulsed at 1 ms, 10% Duty Cycle Max)
Continuous Current, S or D . . . . . 30 mA
Operating Temperature Range
Industrial (B Version) . . . . . –40°C to +85°C
Storage Temperature Range . . . . . –65°C to +150°C
Junction Temperature . . . . . 150°C

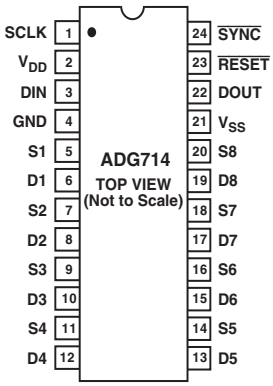
TSSOP Package
θ_{JA} Thermal Impedance . . . . . 128°C/W
θ_{JC} Thermal Impedance . . . . . 42°C/W
Lead Temperature, Soldering (10 sec) . . . . . 300°C
Infrared Reflow (20 sec) . . . . . 235°C

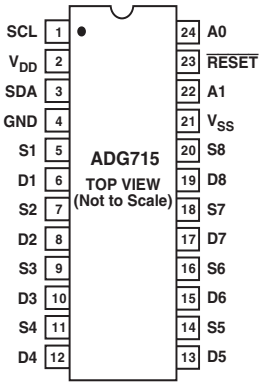
NOTES
[1]Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Only one absolute maximum rating may be applied at any one time.
[2]Overvoltages at IN, S, or D will be clamped by internal diodes. Current should be limited to the maximum ratings given.

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADG714/ADG715 feature proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

PIN CONFIGURATIONS
24-Lead TSSOP





ADG714 PIN FUNCTION DESCRIPTIONS

Pin No.	Mnemonic	Description
1	SCLK	Serial Clock Input. Data is clocked into the input shift register on the falling edge of the serial clock input. These devices can accommodate serial input rates of up to 30 MHz.
2	V_{DD}	Positive Analog Supply Voltage.
3	DIN	Serial Data Input. Data is clocked into the 8-bit input register on the falling edge of the serial clock input.
4	GND	Ground Reference
5, 7, 9, 11, 14, 16, 18, 20	Sx	Source. May be an input or output.
6, 8, 10, 12, 13, 15, 17, 19	Dx	Drain. May be an input or output.
21	V_{SS}	Negative Analog Supply Voltage. For single supply operation this should be tied to GND.
22	DOUT	Serial Data Output. This allows a number a parts to be daisy chained. Data is clocked out of the input shift register on the rising edge of SCLK. DOUT is an open-drain output that should be pulled to the supply with an external pull-up resistor.
23	$\overline{\text{RESET}}$	Active Low Control Input. Clears the input register and turns all switches to the OFF condition.
24	$\overline{\text{SYNC}}$	Active Low Control Input. This is the frame synchronization signal for the input data. When $\overline{\text{SYNC}}$ goes low, it powers on the SCLK and DIN buffers and the input shift register is enabled. Data is transferred on the falling edges of the following clocks. Taking $\overline{\text{SYNC}}$ high updates the switches.

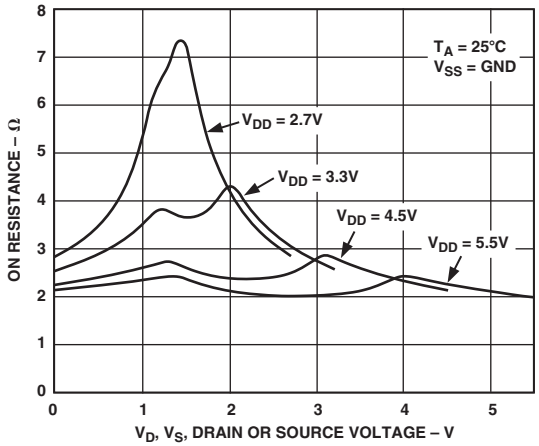
ADG715 PIN FUNCTION DESCRIPTIONS

Pin No.	Mnemonic	Description
1	SCL	Serial Clock Line. This is used in conjunction with the SDA line to clock data into the 8-bit input shift register. Clock rates of up to 400 kbit/s can be accommodated with this 2-wire serial interface.
2	V_{DD}	Positive Analog Supply Voltage
3	SDA	Serial Data Line. This is used in conjunction with the SCL line to clock data into the 8-bit input shift register during the write cycle and used to readback one byte of data during the read cycle. It is a bidirectional open-drain data line which should be pulled to the supply with an external pull-up resistor.
4	GND	Ground Reference
5, 7, 9, 11, 14, 16, 18, 20	Sx	Source. May be an input or output.
6, 8, 10, 12, 13, 15, 17, 19	Dx	Drain. May be an input or output.
21	V_{SS}	Negative Analog Supply Voltage. For single supply operation this should be tied to GND.
22	A1	Address Input. Sets the second least significant bit of the 7-bit slave address.
23	$\overline{\text{RESET}}$	Active Low Control Input. Clears the input register and turns all switches to the OFF condition.
24	A0	Address Input. Sets the least significant bit of the 7-bit slave address.

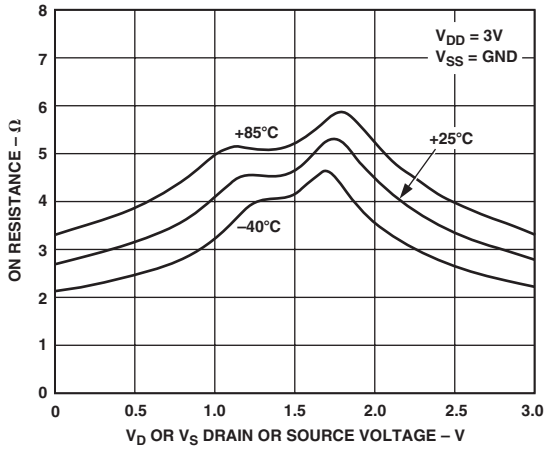
TERMINOLOGY

V_{DD}	Most positive power supply potential.
V_{SS}	Most negative power supply in a dual supply application. In single supply applications, this should be tied to ground.
I_{DD}	Positive Supply Current
I_{SS}	Negative Supply Current
GND	Ground (0 V) Reference
S	Source Terminal. May be an input or output.
D	Drain Terminal. May be an input or output.
R_{ON}	Ohmic resistance between D and S
ΔR_{ON}	On resistance match between any two channels, i.e., R_{ON} max–R_{ON} min.
$R_{FLAT(ON)}$	Flatness is defined as the difference between the maximum and minimum value of on resistance as measured over the specified analog signal range.
I_S (OFF)	Source leakage current with the switch "OFF."
I_D (OFF)	Drain leakage current with the switch "OFF."
I_D, I_S (ON)	Channel leakage current with the switch "ON."
V_D (V_S)	Analog voltage on terminals D and S
C_S (OFF)	"OFF" Switch Source Capacitance. Measured with reference to ground.
C_D (OFF)	"OFF" Switch Drain Capacitance. Measured with reference to ground.
C_D, C_S (ON)	"ON" Switch Capacitance. Measured with reference to ground.
C_{IN}	Digital Input Capacitance
t_{ON}	Delay time between loading new data to the shift register and selected switches switching on.
t_{OFF}	Delay time between loading new data to the shift register and selected switches switching off.
Off Isolation	A measure of unwanted signal coupling through an "OFF" switch.
Crosstalk	A measure of unwanted signal which is coupled through from one channel to another as a result of parasitic capacitance.
Charge Injection	A measure of the glitch impulse transferred from the digital input to the analog output during switching.
Bandwidth	The frequency at which the output is attenuated by –3 dBs.
On Response	The frequency response of the "ON" switch.
Insertion Loss	The loss due to the ON resistance of the switch. Insertion Loss = 20 $\log_{10}$ (V_{OUT} with switch/ V_{OUT} without switch.
V_{INL}	Maximum input voltage for Logic 0.
V_{INH}	Minimum input voltage for Logic 1.
I_{INL}(I_{INH})	Input current of the digital input.
I_{DD}	Positive Supply Current

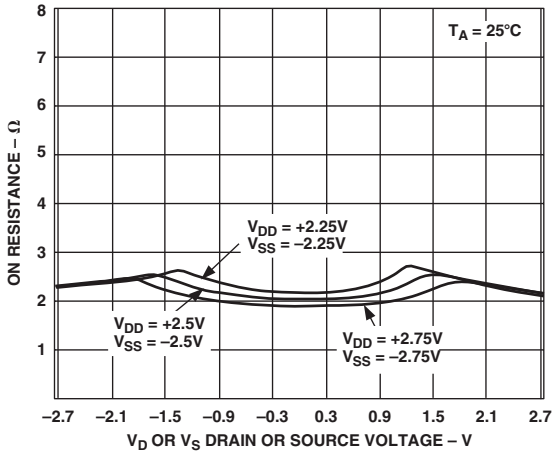
ADG714/ADG715–Typical Performance Characteristics



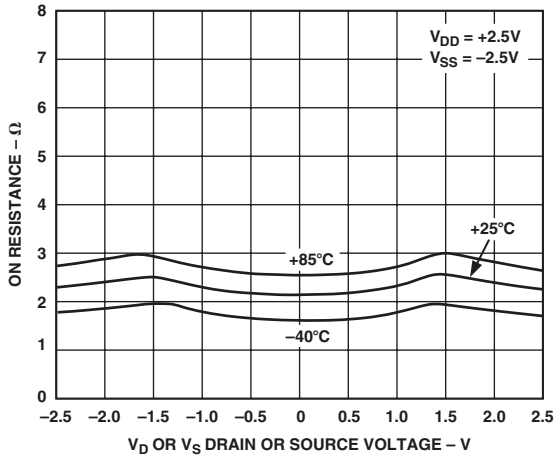
TPC 1. On Resistance as a Function of V_D (V_S) Single Supply



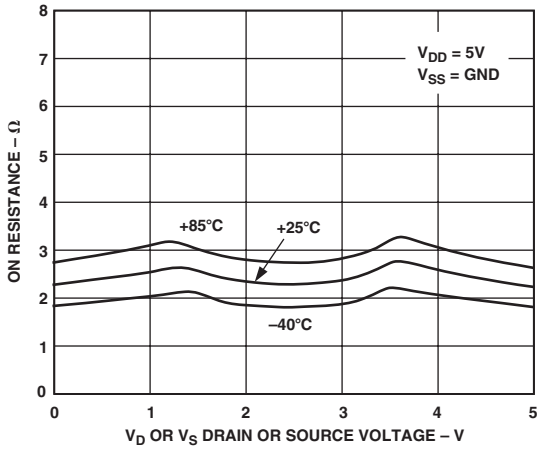
TPC 4. On Resistance as a Function of V_D (V_S) for Different Temperatures; V_{DD} = 3 V



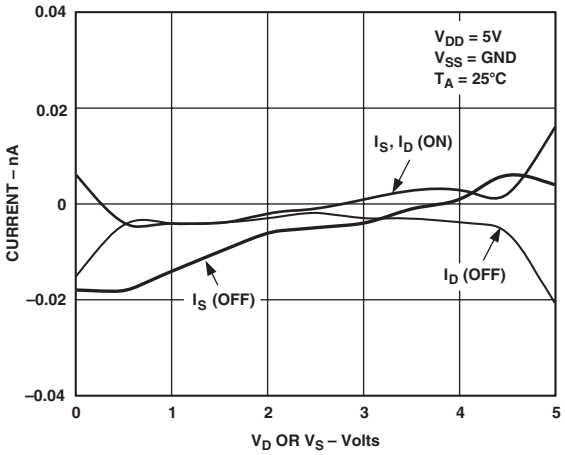
TPC 2. On Resistance as a Function of V_D (V_S); Dual Supply



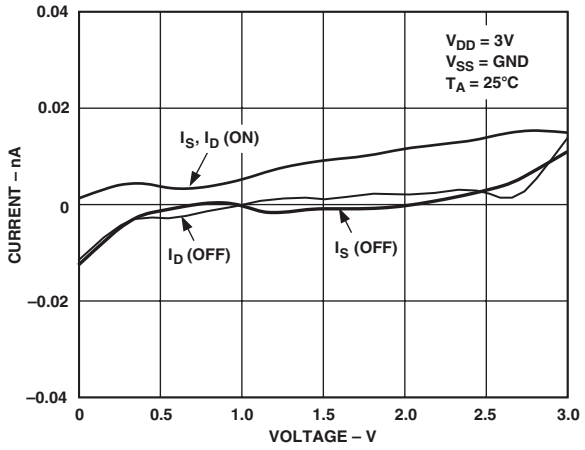
TPC 5. On Resistance as a Function of V_D (V_S) for Different Temperatures; Dual Supply



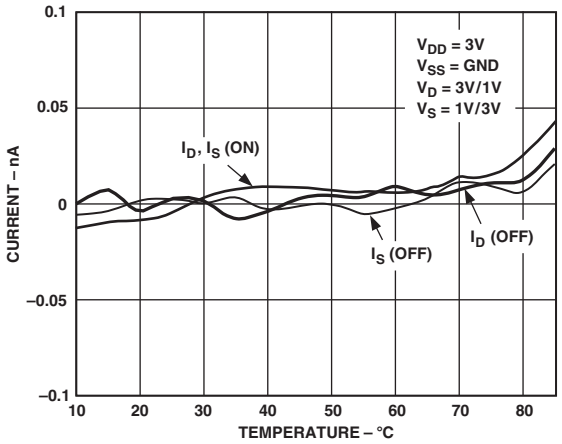
TPC 3. On Resistance as a Function of V_D (V_S) for Different Temperatures; V_{DD} = 5 V



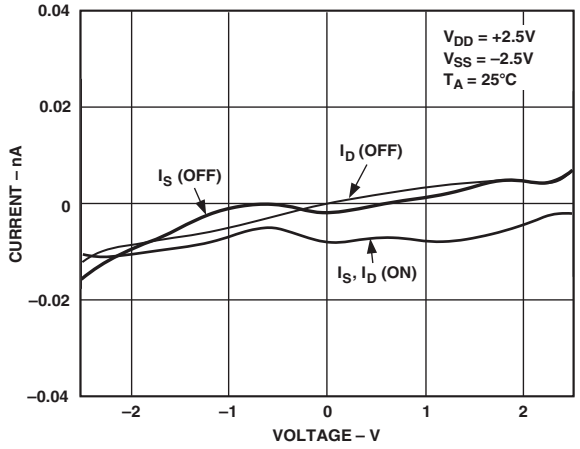
TPC 6. Leakage Currents as a Function of V_D (V_S)



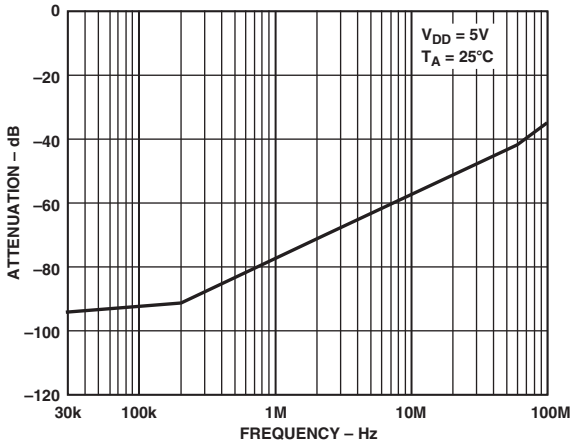
TPC 7. Leakage Currents as a Function of V_D (V_S)



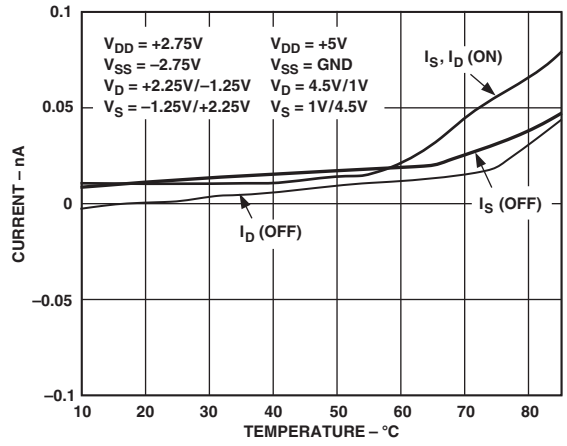
TPC 10. Leakage Currents as a Function of Temperature



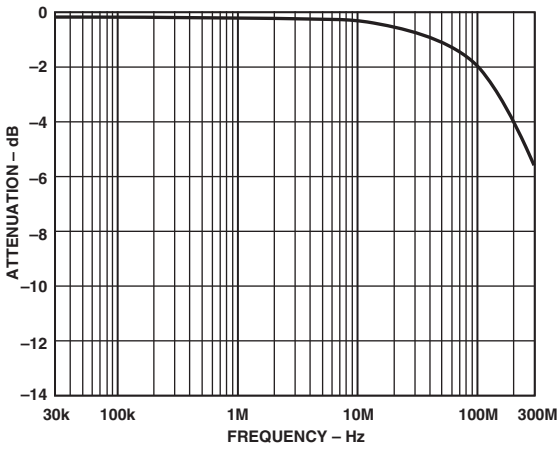
TPC 8. Leakage Currents as a Function of V_D (V_S) Dual Supply



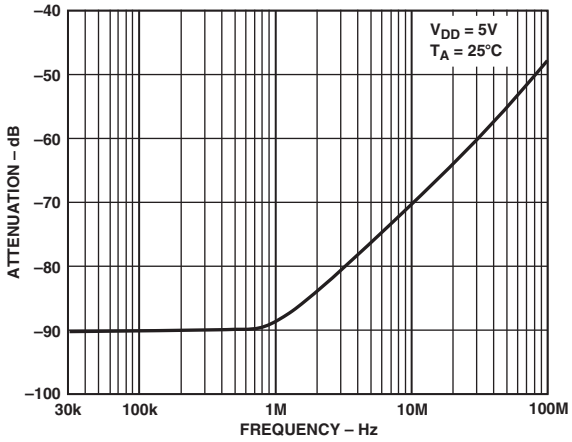
TPC 11. Off Isolation vs. Frequency



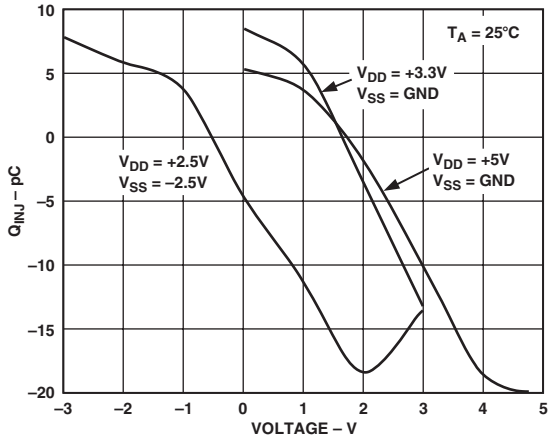
TPC 9. Leakage Currents as Function of Temperature



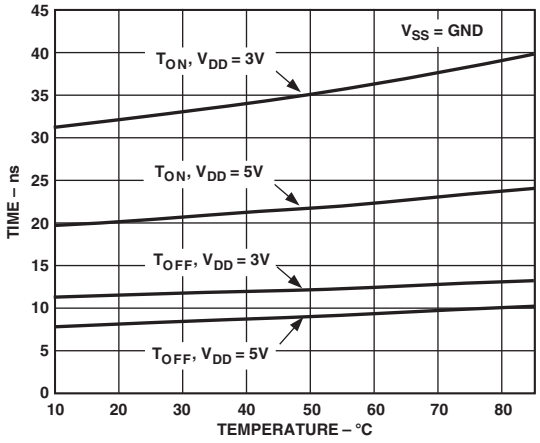
TPC 12. On Response vs. Frequency



TPC 13. Crosstalk vs. Frequency



TPC 14. Charge Injection vs. Source/Drain Voltage



TPC 15. T_{ON}/T_{OFF} Times vs. Temperature for ADG714

GENERAL DESCRIPTION

The ADG714 and ADG715 are serially controlled, octal SPST switches, controlled by either a 2- or 3-wire interface. Each bit of the 8-bit serial word corresponds to one switch of the part. A Logic 1 in the particular bit position turns on the switch, while a Logic 0 turns the switch off. Because each switch is independently controlled by an individual bit, this provides the option of having any, all, or none of the switches ON.

When changing the switch conditions, a new 8-bit word is written to the input shift register. Some of the bits may be the same as the previous write cycle, as the user may not wish to change the state of some switches. To minimize glitches on the output of these switches, the part cleverly compares the state of switches from the previous write cycle. If the switch is already in the ON condition, and is required to stay ON, there will be minimal glitches on the output of the switch.

POWER-ON RESET

On power-up of the device, all switches will be in the OFF condition and the internal shift register is filled with zeros and will remain so until a valid write takes place.

SERIAL INTERFACE

3-Wire Serial Interface

The ADG714 has a 3-wire serial interface ($\overline{SYNC}$, SCLK, and DIN), that is compatible with SPI, QSPI, MICROWIRE interface standards and most DSPs. Figure 1 shows the timing diagram of a typical write sequence.

Data is written to the 8-bit shift register via DIN under the control of the $\overline{SYNC}$ and SCLK signals. Data may be written to the shift register in more or less than eight bits. In each case the shift register retains the last eight bits that were written.

When $\overline{SYNC}$ goes low, the input shift register is enabled. Data from DIN is clocked into the shift register on the falling edge of SCLK. Each bit of the 8-bit word corresponds to one of the eight switches. Figure 3 shows the contents of the input shift register. Data appears on the DOUT pin on the rising edge of SCLK suitable for daisy chaining, delayed of course by eight bits. When all eight bits have been written into the shift register, the $\overline{SYNC}$ line is brought high again. The switches are updated with the new configuration and the input shift register is disabled. With $\overline{SYNC}$ held high, the input shift register is disabled, so further data or noise on the DIN line will have no effect on the shift register.

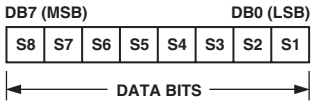


Figure 3. Input Shift Register Contents

SERIAL INTERFACE

2-Wire Serial Interface

The ADG715 is controlled via an I^2C-compatible serial bus. This device is connected to the bus as a slave device (no clock is generated by the switch).

The ADG715 has a 7-bit slave address. The five MSBs are 10010 and the two LSBs are determined by the state of the A0 and A1 pins.

The 2-wire serial bus protocol operates as follows:

1. The master initiates data transfer by establishing a START condition, which is when a high-to-low transition on the SDA line occurs while SCL is high. The following byte is the address byte that consists of the 7-bit slave address followed by a R/$\overline{W}$ bit (this bit determines whether data will be read from or written to the slave device).

   The slave whose address corresponds to the transmitted address responds by pulling the SDA line low during the ninth clock pulse (this is termed the acknowledge bit). At this stage, all other devices on the bus remain idle while the selected device waits for data to be written to or read from its serial register. If the R/$\overline{W}$ bit is high, the master will read from the slave device. However, if the R/$\overline{W}$ bit is low, the master will write to the slave device.

2. Data is transmitted over the serial bus in sequences of nine clock pulses (eight data bits followed by an acknowledge bit). The transitions on the SDA line must occur during the low period of SCL and remain stable during the high period of SCL.

3. When all data bits have been read or written, a STOP condition is established by the master. A STOP condition is defined as a low-to-high transition on the SDA line while SCL is high. In write mode, the master will pull the SDA line high during the tenth clock pulse to establish a STOP condition. In read mode, the master will issue a no acknowledge for the ninth clock pulse (i.e., the SDA line remains high). The master will then bring the SDA line low before the tenth clock pulse and then high during the tenth clock pulse to establish a STOP condition.

See Figure 4 for a graphical explanation of the serial interface.

A repeated write function gives the user flexibility to update the matrix switch a number of times after addressing the part only once. During the write cycle, each data byte will update the configuration of the switches. For example, after the matrix switch has acknowledged its address byte, and received one data byte, the switches will update after the data byte; if another data byte is written to the matrix switch while it is still the addressed slave device, this data byte will also cause a switch configuration update. Repeat read of the matrix switch is also allowed.

Input Shift Register

The input shift register is eight bits wide. Figure 3 illustrates the contents of the input shift register. Data is loaded into the device as an 8-bit word under the control of a serial clock input, SCL. The timing diagram for this operation is shown in Figure 2. The 8-bit word consists of eight data bits, each controlling one switch. MSB (Bit 7) is loaded first.

Write Operation

When writing to the ADG715, the user must begin with an address byte and R/$\overline{W}$ bit, after which the switch will acknowledge that it is prepared to receive data by pulling SDA low. This address byte is followed by the 8-bit word. The write operation for the switch is shown in the Figure 4.

READ Operation

When reading data back from the ADG715, the user must begin with an address byte and R/$\overline{W}$ bit, after which the switch will acknowledge that it is prepared to transmit data by pulling SDA low. The readback operation is a single byte that consists of the eight data bits in the input register. The read operation for the part is shown in Figure 5.

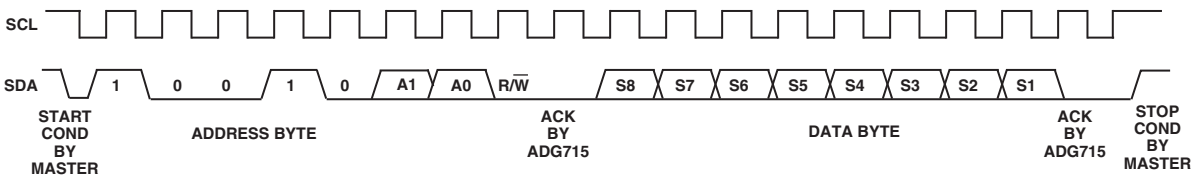


Figure 4. ADG715 Write Sequence

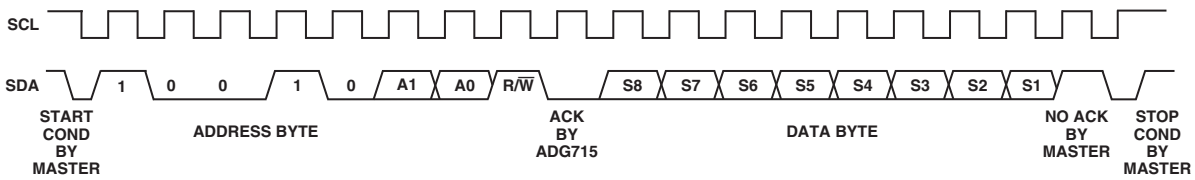


Figure 5. ADG715 Readback Sequence

APPLICATIONS

Multiple Devices On One Bus

Figure 6 shows four ADG715 devices on the same serial bus. Each has a different slave address since the state of their A0 and A1 pins is different. This allows each switch to be written to or read from independently.

Daisy-Chaining Multiple ADG714s

A number of ADG714 switches may be daisy-chained simply by using the DOUT pin. Figure 7 shows a typical implementation. The $\overline{\text{SYNC}}$ pin of all three parts in the example are tied together. When $\overline{\text{SYNC}}$ is brought low, the input shift registers of all parts are enabled, data is written to the parts via DIN, and clocked through the shift registers. When the transfer is complete, $\overline{\text{SYNC}}$ is brought high and all switches are updated simultaneously. Further shift registers may be added in series.

Power Supply Sequencing

When using CMOS devices, care must be taken to ensure correct power-supply sequencing. Incorrect power-supply sequencing can result in the device being subjected to stresses beyond those maximum ratings listed in the data sheet. Digital and analog inputs should always be applied after power supplies and ground. In dual supply applications, if digital or analog inputs may be applied to the device prior to the V_{DD} and V_{SS} supplies, the addition of a Schottky diode connected between V_{SS} and GND will ensure that the device powers on correctly. For single supply operation, V_{SS} should be tied to GND as close to the device as possible.

Decoding Multiple ADG714s Using an ADG739

The dual 4-channel ADG739 multiplexer can be used to multiplex a single chip select line to provide chip selects for up to four

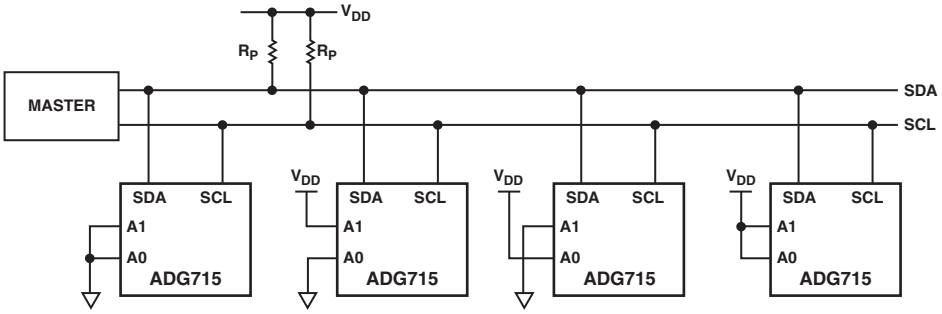


Figure 6. Multiple ADG715s On One Bus

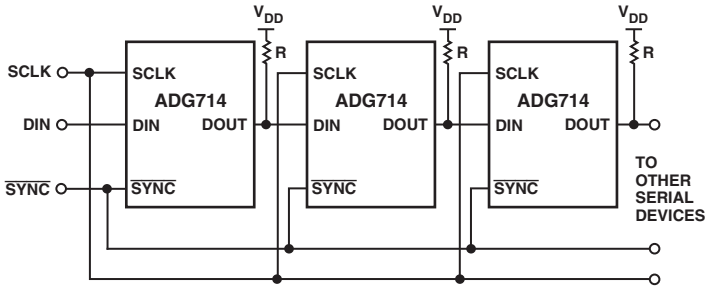


Figure 7. Multiple ADG714 Devices in a Daisy-Chained Configuration

devices on the SPI bus. Figure 8 illustrates the ADG739 and multiple ADG714s in such a typical configuration. All devices receive the same serial clock and serial data, but only one device will receive the $\overline{\text{SYNC}}$ signal at any one time. The ADG739 is a serially controlled device also. One bit programmable pin of the microcontroller is used to enable the ADG739 via $\overline{\text{SYNC2}}$, while another bit programmable pin is used as the chip select for the other serial devices, $\overline{\text{SYNC1}}$. Driving $\overline{\text{SYNC2}}$ low enables changes to be made to the addressed serial devices. By bringing $\overline{\text{SYNC1}}$ low, the selected serial device hanging from the SPI bus will be enabled and data will be clocked into its shift register on the falling edges of SCLK. The convenient design of the matrix switch allows for different combinations of the four serial devices to be addressed at any one time. If more devices need to be addressed via one chip select line, the ADG738 is an 8-channel device and would allow further expansion of the chip select scheme. There may be some digital feedthrough from the digital input lines because SCLK and DIN are permanently connected to each device. Using a burst clock will minimize the effects of digital feedthrough on the analog channels.

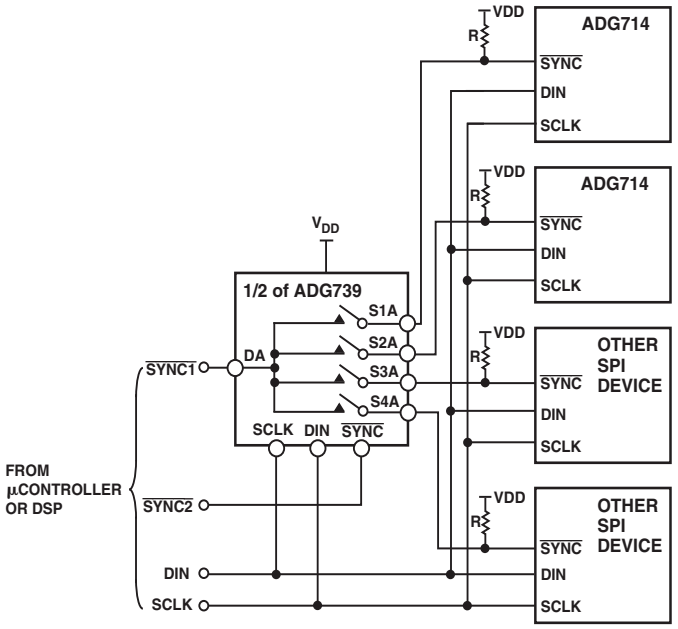
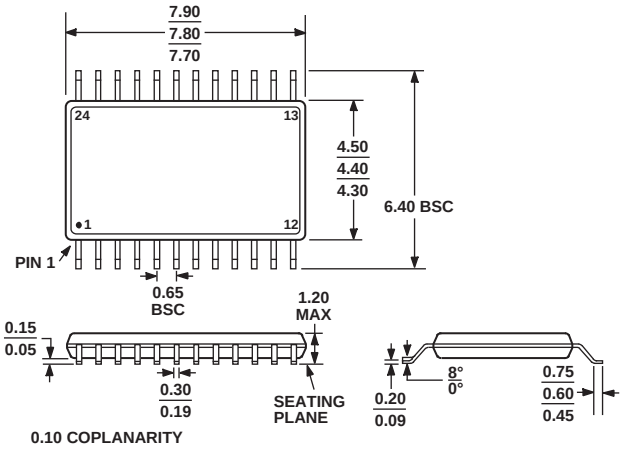


Figure 8. Addressing Multiple ADG714s Using an ADG739

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-153-AD

Figure 1. 24-Lead Thin Shrink Small Outline Package [TSSOP]
(RU-24)
Dimensions shown in millimeters

ORDERING GUIDE

Model[1]	Temperature Range	Package Description	Package Option
ADG714BRU-REEL	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG714BRU-REEL7	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG714BRUZ	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG714BRUZ-REEL	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG714BRUZ-REEL7	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG715BRU	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG715BRU-REEL	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG715BRU-REEL7	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG715BRUZ	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG715BRUZ-REEL	−40°C to +85°C	24-Lead TSSOP	RU-24
ADG715BRUZ-REEL7	−40°C to +85°C	24-Lead TSSOP	RU-24

[1] Z = RoHS Compliant Part.

REVISION HISTORY

1/13—Rev. B to Rev. C

Changes to Dual Supply Table Summary and I_{DD} Test Conditions/Comments 4
Changes to Ordering Guide 16

11/02—Rev. A to Rev. B

Edits to Features 1
Edits to General Description 1
Edits to Product Highlights 1
Edits to Specifications 3, 4
Edits to TPCs 2 and 5 10
Edits to TPCs 8 and 9 11
Edits to TPCs 14 12
Edits to Figure 8 15

LCD-20x4B

20x4 Liquid Crystal Display with HD44780 driver
(Blue Backlight and White Character)

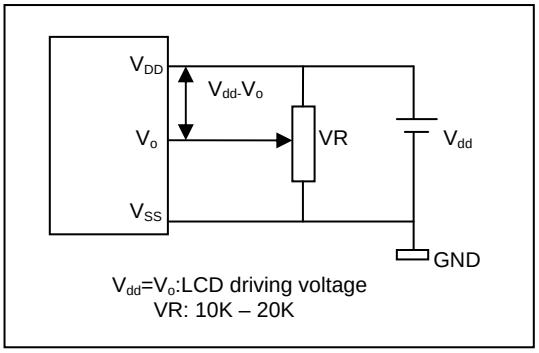
Specifications

Display Format	20 characters x 4 lines
Drive Method	1/16 duty, 1/5 bias
LCD Type	STN Blue
LED Backlight	White
Viewing direction	6: 00
Operating Temperature	-20°C to 60°C

Electrical Characteristics

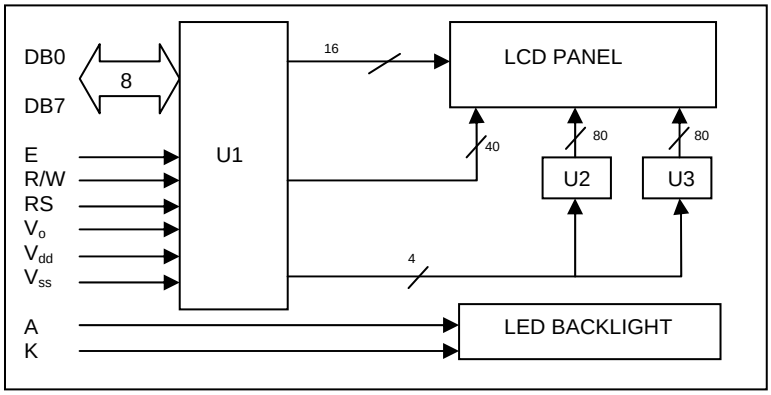
Parameter	Symbol	Condition	Stand value			UNIT
			Min.	Typ.	Max.	
Supply Voltage for Logic	V_{DD}-V_{SS}	-	4.5	5.0	5.5	V
Supply Voltage for LCD	V_{DD}-V_o	-	-	4.7	-	V
Input Logic HIGH	V_{IH}	-	2.2	-	V_{DD}	V
Input Logic LOW	V_{IL}	-	-0.3	-	0.6	V
Output Logic HIGH	V_{OH}	$-I_{OH}$=0.2mA	2.4	-	-	V
Output Logic LOW	V_{OL}	I_{OL}=1.2mA	-	-	0.4	V
Power Supply Current without backlight	V_{DD}	V_{DD}=5.0V	-	2.0	5	mA
Power Supply Current with backlight	I_{LED}	V_{LED}=5.0V	-	72	80	mA

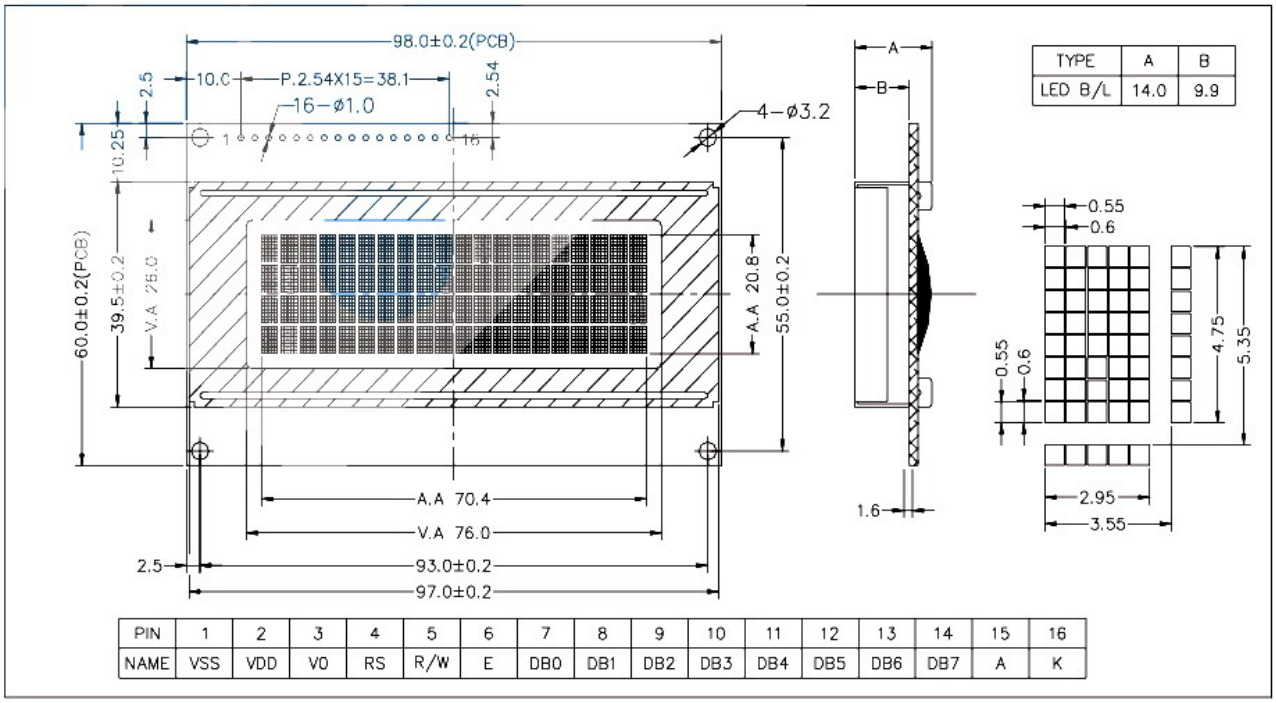
Power Supply



V_{dd}=V_o:LCD driving voltage
VR: 10K – 20K

Block Diagram





ALPS Manufacturer of Electronic Components/Parts Catalog

11mm Size Metal Shaft Type EC11 Series

Detail

Part number	EC11E09244AQ

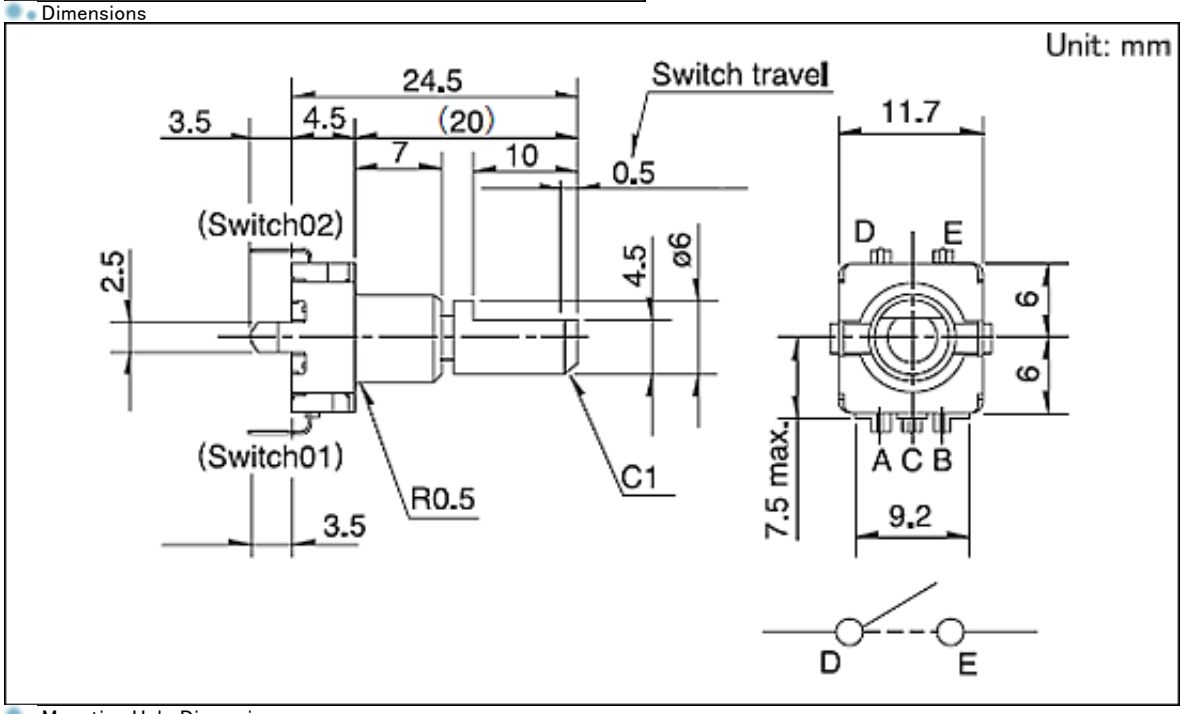
Structure	Vertical
Actuator configuration	Flat
Actuator length	20mm
Number of detent	18
Number of pulse	9
Push-on switch	With
Travel of push-on switch	1.5mm
Operating life	15,000 cycles
Minimum order unit (pcs.)	2,400

Operating temperature range		-40℃ to +85℃
Maximum operating current (Resistive load)		10mA
Electrical performance	Ratings	10mA 5V DC
	Output signal	Output of A and B signals, proportionate to phase difference
	Insulation resistance	100MΩ min. 250V DC
	Voltage proof	300V AC for 1 minute
Mechanical performance	Detent torque	10±7mN·m
	Push-pull strength	100N
Resistance to soldering heat	Manual soldering	350℃ max. 3s max.
	Dip soldering	260±5℃, 5±1s
Environmental performance	Cold	-40±3℃ for 240h
	Dry heat	85±3℃ for 240h
	Damp heat	60±2℃, 90 to 95%RH for 240h
Push-on switch specifications	Contact arrangement	Single pole single throw (Push-on)
	Travel	1.5±0.5mm
	Operating force	4±2N
	Rating	0.1A 5V DC (500μA 5V DC min. ratings)
	Contact resistance (Initial performance/After lifetime)	100mΩ max./200mΩ max.
	Operating life	20,000 times min.

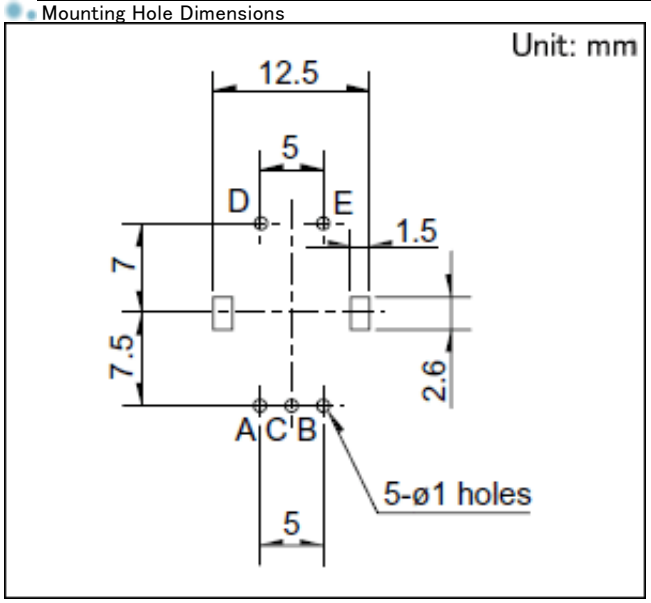
Certificate of Compliance to RoHS regulations	

Photo

Dimensions

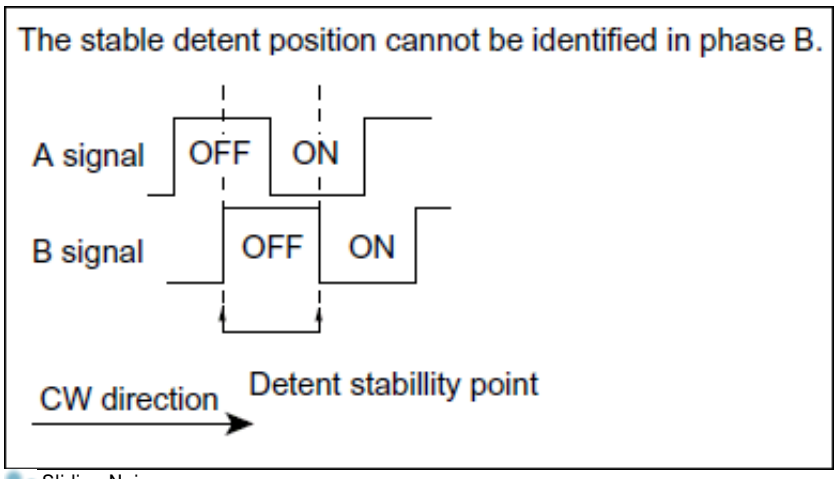


Mounting Hole Dimensions

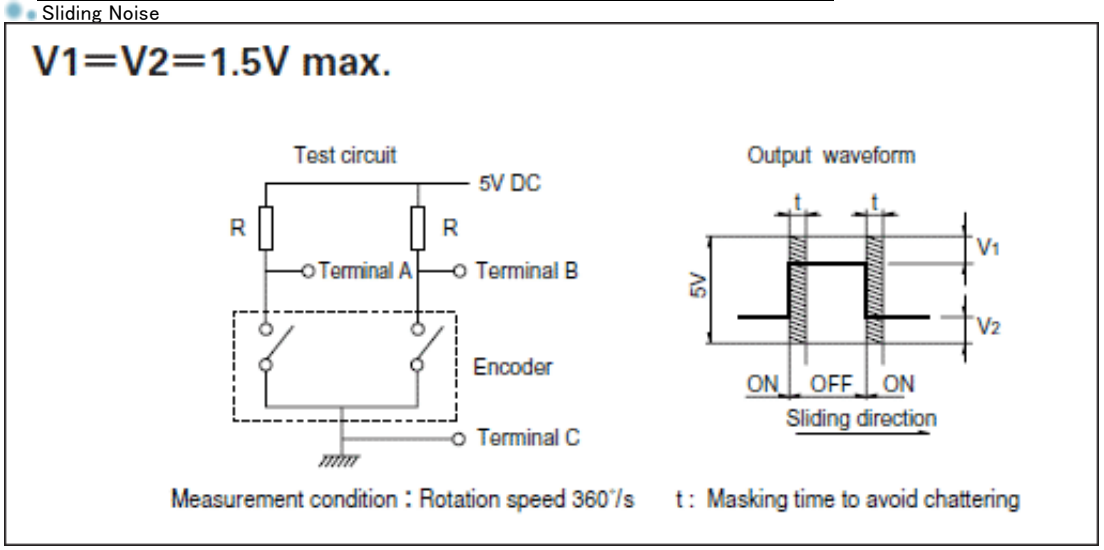


Viewed from mounting side.

Output Wave



Sliding Noise



At R=5kΩ Chattering: 3ms max. Bounce: 2ms max.

Packing Specifications

Tray

Number of packages (pcs.)	1 case / Japan	1,200
	1 case / export packing	2,400
Export package measurements (mm)		540×360×250

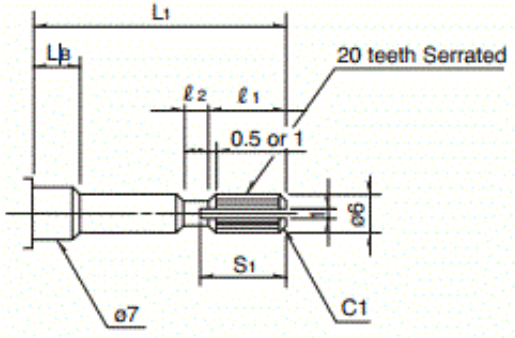
Product Varieties

Single-shaft Type

1) Knourled Type

Unit : mm

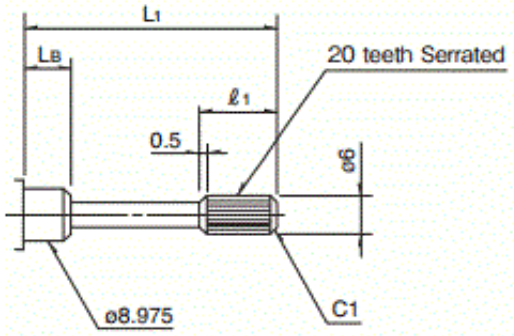
Style (Shaft diameter : φ6)



Detailed dimensions

L_1	L_B	ℓ_1	ℓ_2	S_1
20	7	6	1	7
25	10	10	2	11

Style (Shaft diameter : φ6)

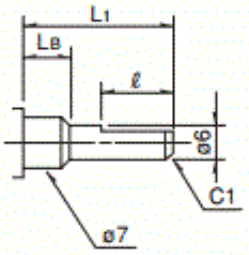


Detailed dimensions

L_1	L_B	ℓ_1
15	7	5
20	10	6

2) Flat Type Unit : mm

Style (Shaft diameter : ϕ6)

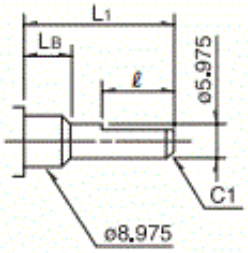


Detailed dimensions

	L_1	L_B	ℓ
※1	15	5	7
	15	7	5 (6)
	20	7	10 (12)
※2	25	10	12

※ 1 Does not comply with EC111
※ 2 L_B=7 for EC11B.

Style (Shaft diameter : ϕ6)

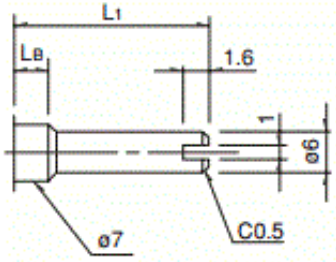


Detailed dimensions

L_1	L_B	ℓ
15	7	5
20	10	7

3) Slotted Type Unit : mm

Style (Shaft diameter : ϕ6)

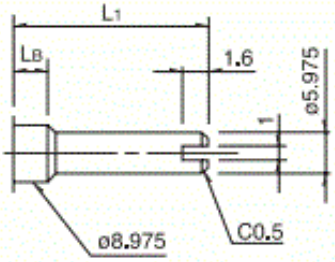


Detailed dimensions

	L_1	L_B
	15	7
	20	7
※	25	10

※ L_B=7 for EC11B.

Style (Shaft diameter : ϕ6)



Detailed dimensions

L_1	L_B
15	7
20	10

» NOTE Notes are common to this series/models.

1. This site catalog shows only outline specifications. When using the products, please obtain formal specifications for supply.
2. Please place purchase orders per minimum order unit (integer).
3. Products other than those listed in the above chart are also available. Please contact us for details.
4. This products can be used in vehicles.
   Although these products are designed to perform over a wide operating temperature range, please ensure that you receive and read the formal delivery specifications before use.

Inquiries about Products
For more information please contact: Products Information Center.
1-7, Yukigaya-otsukamachi, Ota-ku, Tokyo, 145-8501, Japan
Phone: +81 (3) 5499-8154

25. ADC – Analog to Digital Converter

25.1 Features

- **10-bit Resolution**
- **1 LSB Integral Non-linearity**
- **±2 LSB Absolute Accuracy**
- **13 µs - 260 µs Conversion Time**
- **Up to 76.9 kSPS (Up to 15 kSPS at Maximum Resolution)**
- **16 Multiplexed Single Ended Input Channels**
- **14 Differential input channels**
- **4 Differential Input Channels with Optional Gain of 10× and 200×**
- **Optional Left Adjustment for ADC Result Readout**
- **0V - V_{CC} ADC Input Voltage Range**
- **2.7V - V_{CC} Differential ADC Voltage Range**
- **Selectable 2.56V or 1.1V ADC Reference Voltage**
- **Free Running or Single Conversion Mode**
- **Interrupt on ADC Conversion Complete**
- **Sleep Mode Noise Canceler**

The ATmega640/1280/1281/2560/2561 features a 10-bit successive approximation ADC. The ADC is connected to an 8/16-channel Analog Multiplexer which allows eight/sixteen single-ended voltage inputs constructed from the pins of Port F and Port K. The single-ended voltage inputs refer to 0V (GND).

The device also supports 16/32 differential voltage input combinations. Four of the differential inputs (ADC1 & ADC0, ADC3 & ADC2, ADC9 & ADC8 and ADC11 & ADC10) are equipped with a programmable gain stage, providing amplification steps of 0 dB (1×), 20 dB (10×) or 46 dB (200×) on the differential input voltage before the ADC conversion. The 16 channels are split in two sections of 8 channels where in each section seven differential analog input channels share a common negative terminal (ADC1/ADC9), while any other ADC input in that section can be selected as the positive input terminal. If 1× or 10× gain is used, 8 bit resolution can be expected. If 200× gain is used, 7 bit resolution can be expected.

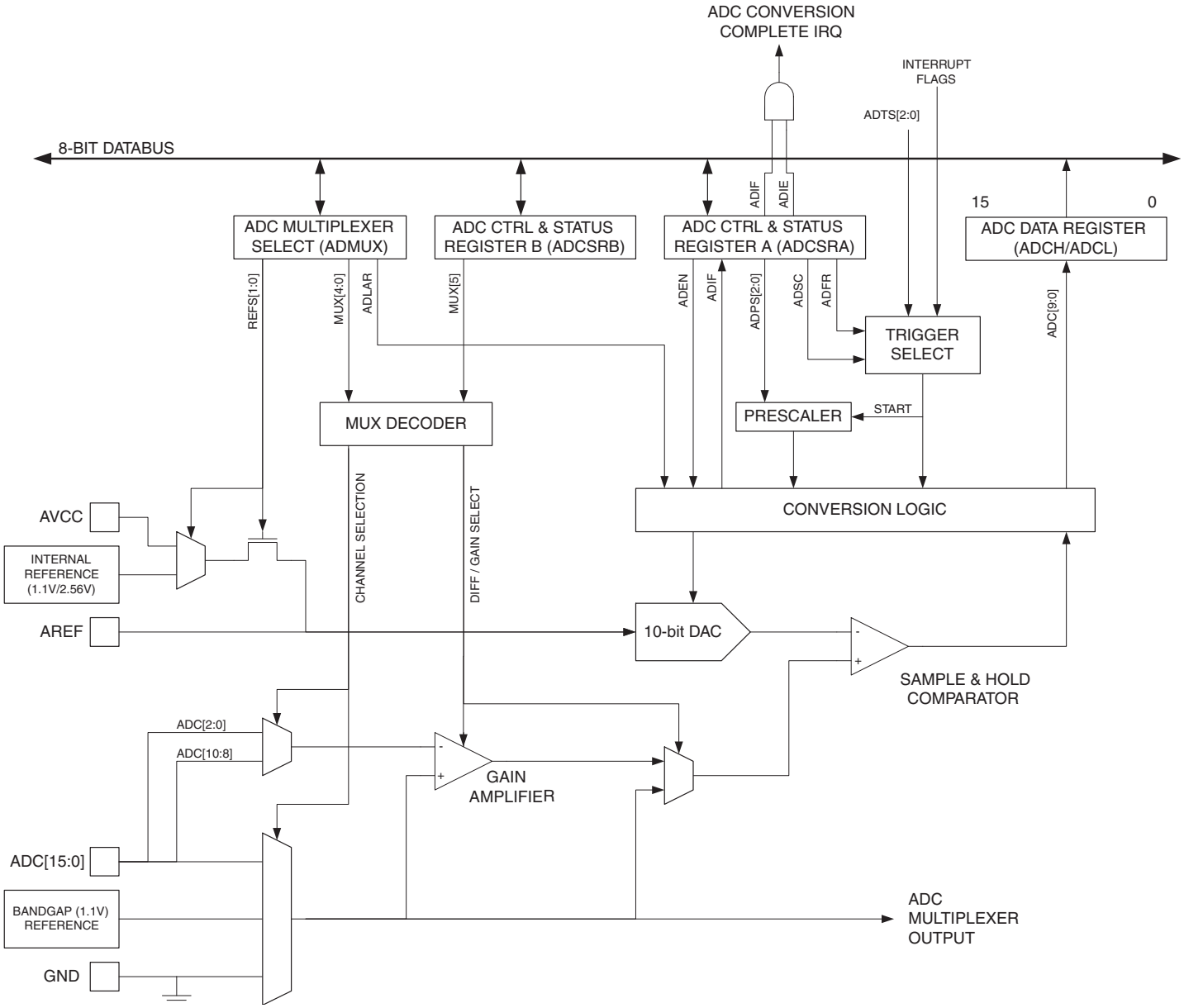
The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 25-1 on page 276.

The ADC has a separate analog supply voltage pin, AVCC. AVCC must not differ more than ±0.3V from V_{CC}. See the paragraph “ADC Noise Canceler” on page 283 on how to connect this pin.

Internal reference voltages of nominally 1.1V, 2.56V or AVCC are provided On-chip. The voltage reference may be externally decoupled at the AREF pin by a capacitor for better noise performance.

The Power Reduction ADC bit, PRADC, in “PRR0 – Power Reduction Register 0” on page 56 must be disabled by writing a logical zero to enable the ADC.

Figure 25-1. Analog to Digital Converter Block Schematic



25.2 Operation

The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB. Optionally, AVCC or an internal 1.1V or 2.56V reference voltage may be connected to the AREF pin by writing to the REFSn bits in the ADMUX Register. The internal voltage reference may thus be decoupled by an external capacitor at the AREF pin to improve noise immunity.

The analog input channel is selected by writing to the MUX bits in ADMUX and ADCSRB. Any of the ADC input pins, as well as GND and a fixed bandgap voltage reference, can be selected as single ended inputs to the ADC. A selection of ADC input pins can be selected as positive and negative inputs to the differential amplifier.

If differential channels are selected, the voltage difference between the selected input channel pair then becomes the analog input to the ADC. If single ended channels are used, the amplifier is bypassed altogether.

The ADC is enabled by setting the ADC Enable bit, ADEN in ADCSRA. Voltage reference and input channel selections will not go into effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power saving sleep modes.

The ADC generates a 10-bit result which is presented in the ADC Data Registers, ADCH and ADCL. By default, the result is presented right adjusted, but can optionally be presented left adjusted by setting the ADLAR bit in ADMUX.

If the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH, to ensure that the content of the Data Registers belongs to the same conversion. Once ADCL is read, ADC access to Data Registers is blocked. This means that if ADCL has been read, and a conversion completes before ADCH is read, neither register is updated and the result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL Registers is re-enabled.

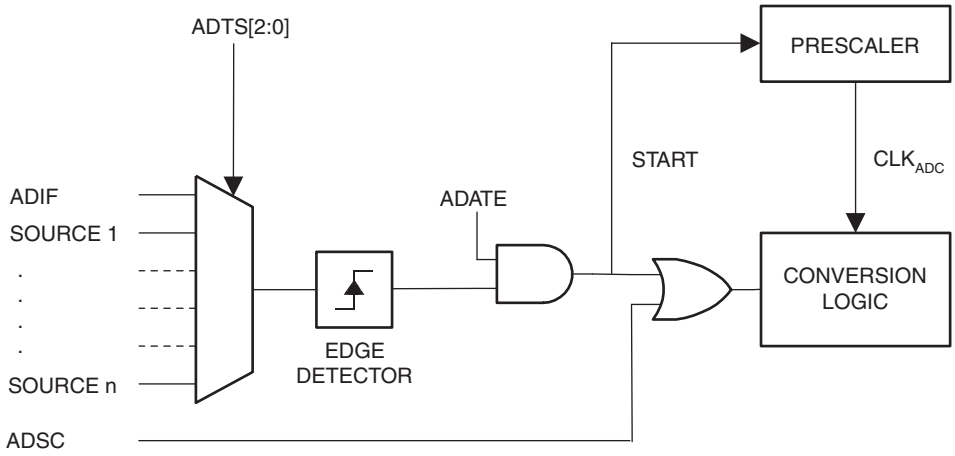
The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the Data Registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

25.3 Starting a Conversion

A single conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit stays high as long as the conversion is in progress and will be cleared by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

Alternatively, a conversion can be triggered automatically by various sources. Auto Triggering is enabled by setting the ADC Auto Trigger Enable bit, ADATE in ADCSRA. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB (See description of the ADTS bits for a list of the trigger sources). When a positive edge occurs on the selected trigger signal, the ADC prescaler is reset and a conversion is started. This provides a method of starting conversions at fixed intervals. If the trigger signal still is set when the conversion completes, a new conversion will not be started. If another positive edge occurs on the trigger signal during conversion, the edge will be ignored. Note that an Interrupt Flag will be set even if the specific interrupt is disabled or the Global Interrupt Enable bit in SREG is cleared. A conversion can thus be triggered without causing an interrupt. However, the Interrupt Flag must be cleared in order to trigger a new conversion at the next interrupt event.

Figure 25-2. ADC Auto Trigger Logic

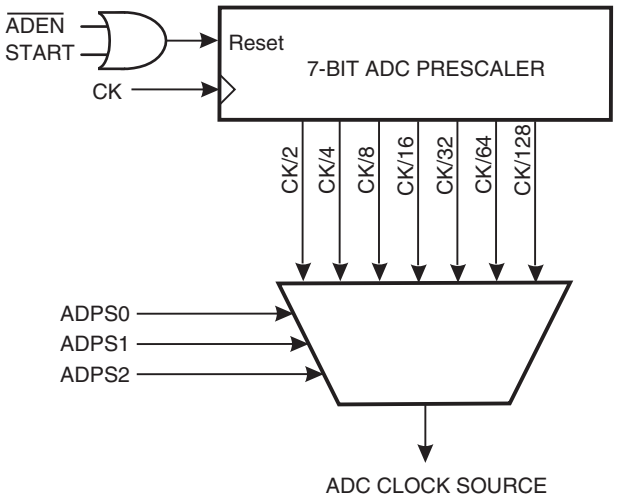


Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag, ADIF is cleared or not.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

25.4 Prescaling and Conversion Timing

Figure 25-3. ADC Prescaler



By default, the successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be as high as 1000 kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100 kHz. The prescaling is set by the ADPS bits in ADCSRA.

The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a single ended conversion by setting the ADSC bit in ADCSRA, the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 13 ADC clock cycles. The first conversion after the ADC is switched on (ADEN in ADCSRA is set) takes 25 ADC clock cycles in order to initialize the analog circuitry.

When the bandgap reference voltage is used as input to the ADC, it will take a certain time for the voltage to stabilize. If not stabilized, the first value read after the first conversion may be wrong.

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of an first conversion. When a conversion is complete, the result is written to the ADC Data Registers, and ADIF is set. In Single Conversion mode, ADSC is cleared simultaneously. The software may then set ADSC again, and a new conversion will be initiated on the first rising ADC clock edge.

When Auto Triggering is used, the prescaler is reset when the trigger event occurs. This assures a fixed delay from the trigger event to the start of conversion. In this mode, the sample-and-hold takes place two ADC clock cycles after the rising edge on the trigger source signal. Three additional CPU clock cycles are used for synchronization logic.

In Free Running mode, a new conversion will be started immediately after the conversion completes, while ADSC remains high. For a summary of conversion times, see Table 25-1 on page 281.

Figure 25-4. ADC Timing Diagram, First Conversion (Single Conversion Mode)

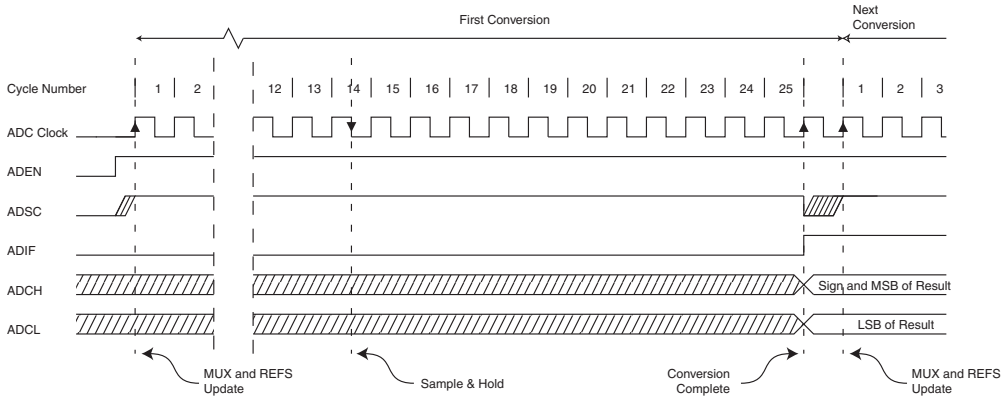


Figure 25-5. ADC Timing Diagram, Single Conversion

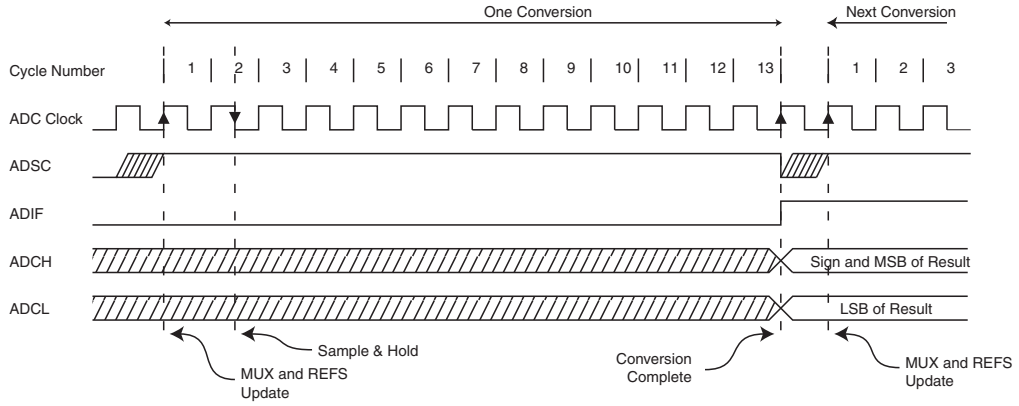


Figure 25-6. ADC Timing Diagram, Auto Triggered Conversion

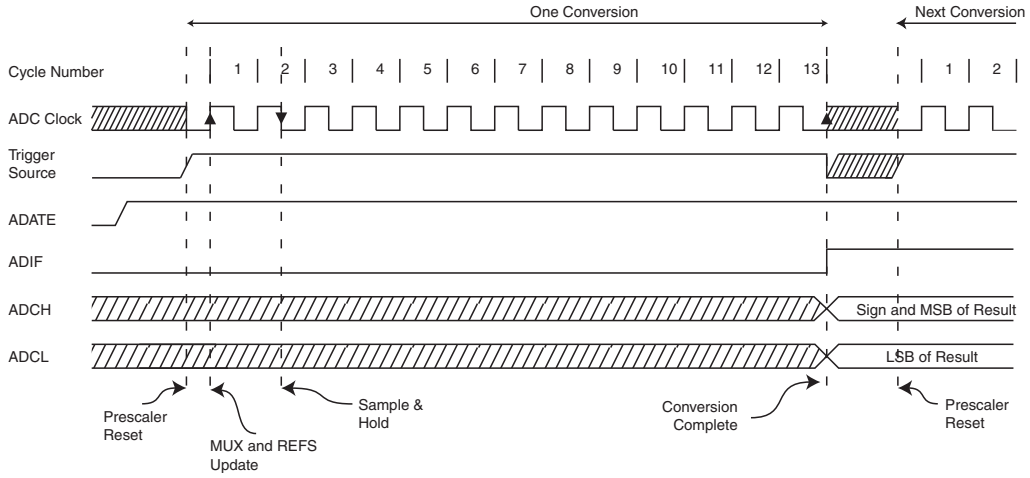


Figure 25-7. ADC Timing Diagram, Free Running Conversion

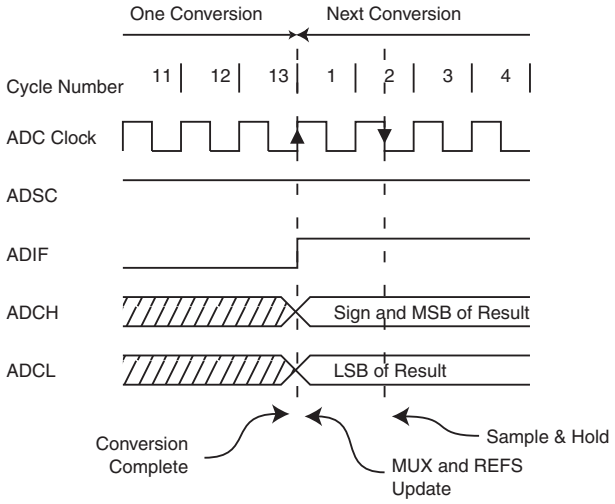


Table 25-1. ADC Conversion Time

Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5
Normal conversions, differential	1.5/2.5	13/14

25.4.1 Differential Channels

When using differential channels, certain aspects of the conversion need to be taken into consideration.

Differential conversions are synchronized to the internal clock CK_{ADC2} equal to half the ADC clock. This synchronization is done automatically by the ADC interface in such a way that the sample-and-hold occurs at a specific phase of CK_{ADC2}. A conversion initiated by the user (that is, all single conversions, and the first free running conversion) when CK_{ADC2} is low will take the same amount of time as a single ended conversion (13 ADC clock cycles from the next prescaled clock cycle). A conversion initiated by the user when CK_{ADC2} is high will take 14 ADC clock cycles due to the synchronization mechanism. In Free Running mode, a new conversion is initiated immediately after the previous conversion completes, and since CK_{ADC2} is high at this time, all automatically started (that is, all but the first) Free Running conversions will take 14 ADC clock cycles.

If differential channels are used and conversions are started by Auto Triggering, the ADC must be switched off between conversions. When Auto Triggering is used, the ADC prescaler is reset before the conversion is started. Since the stage is dependent of a stable ADC clock prior to the conversion, this conversion will not be valid. By disabling and then re-enabling the ADC between each conversion (writing ADEN in ADCSRA to “0” then to “1”), only extended conversions are performed. The result from the extended conversions will be valid. See “Prescaling and Conversion Timing” on page 278 for timing details.

25.5 Changing Channel or Reference Selection

The MUXn and REFS1:0 bits in the ADMUX Register are single buffered through a temporary register to which the CPU has random access. This ensures that the channels and reference selection only takes place at a safe point during the conversion. The channel and reference selection is continuously updated until a conversion is started. Once the conversion starts, the channel and reference selection is locked to ensure a sufficient sampling time for the ADC. Continuous updating resumes in the last ADC clock cycle before the conversion completes (ADIF in ADCSRA is set). Note that the conversion starts on the following rising ADC clock edge after ADSC is written. The user is thus advised not to write new channel or reference selection values to ADMUX until one ADC clock cycle after ADSC is written.

If Auto Triggering is used, the exact time of the triggering event can be indeterministic. Special care must be taken when updating the ADMUX Register, in order to control which conversion will be affected by the new settings.

If both ADATE and ADEN is written to one, an interrupt event can occur at any time. If the ADMUX Register is changed in this period, the user cannot tell if the next conversion is based on the old or the new settings. ADMUX can be safely updated in the following ways:

1. When ADATE or ADEN is cleared.
2. During conversion, minimum one ADC clock cycle after the trigger event.
3. After a conversion, before the Interrupt Flag used as trigger source is cleared.

When updating ADMUX in one of these conditions, the new settings will affect the next ADC conversion.

Special care should be taken when changing differential channels. Once a differential channel has been selected, the stage may take as much as 125 µs to stabilize to the new value. Thus conversions should not be started within the first 125 µs after selecting a new differential channel. Alternatively, conversion results obtained within this period should be discarded.

The same settling time should be observed for the first differential conversion after changing ADC reference (by changing the REFS1:0 bits in ADMUX).

25.5.1 ADC Input Channels

When changing channel selections, the user should observe the following guidelines to ensure that the correct channel is selected:

In Single Conversion mode, always select the channel before starting the conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the conversion to complete before changing the channel selection.

In Free Running mode, always select the channel before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the first conversion to complete, and then change the channel selection. Since the next conversion has already started automatically, the next result will reflect the previous channel selection. Subsequent conversions will reflect the new channel selection.

When switching to a differential gain channel, the first conversion result may have a poor accuracy due to the required settling time for the automatic offset cancellation circuitry. The user should preferably disregard the first conversion result.

25.5.2 ADC Voltage Reference

The reference voltage for the ADC (V_{REF}) indicates the conversion range for the ADC. Single ended channels that exceed V_{REF} will result in codes close to 0x3FF. V_{REF} can be selected as either AVCC, internal 1.1V reference, internal 2.56V reference or external AREF pin.

AVCC is connected to the ADC through a passive switch. The internal 1.1V reference is generated from the internal bandgap reference (VBG) through an internal amplifier. In either case, the external AREF pin is directly connected to the ADC, and the reference voltage can be made more immune to noise by connecting a capacitor between the AREF pin and ground. V_{REF} can also be measured at the AREF pin with a high impedant voltmeter. Note that V_{REF} is a high impedant source, and only a capacitive load should be connected in a system. The Internal 2.56V reference is generated from the 1.1V reference.

If the user has a fixed voltage source connected to the AREF pin, the user may not use the other reference voltage options in the application, as they will be shorted to the external voltage. If no external voltage is applied to the AREF pin, the user may switch between AVCC, 1.1V and 2.56V as reference selection. The first ADC conversion result after switching reference voltage source may be inaccurate, and the user is advised to discard this result.

If differential channels are used, the selected reference should not be closer to AVCC than indicated in “ADC Characteristics – Preliminary Data” on page 376.

25.6 ADC Noise Canceler

The ADC features a noise canceler that enables conversion during sleep mode to reduce noise induced from the CPU core and other I/O peripherals. The noise canceler can be used with ADC Noise Reduction and Idle mode. To make use of this feature, the following procedure should be used:

1. Make sure that the ADC is enabled and is not busy converting. Single Conversion mode must be selected and the ADC conversion complete interrupt must be enabled.
2. Enter ADC Noise Reduction mode (or Idle mode). The ADC will start a conversion once the CPU has been halted.
3. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the CPU and execute the ADC Conversion Complete interrupt routine. If another interrupt wakes up the CPU before the ADC conversion is complete, that interrupt will be executed, and an ADC Conversion Complete interrupt request will be generated when the ADC conversion completes. The CPU will remain in active mode until a new sleep command is executed.

Note that the ADC will not be automatically turned off when entering other sleep modes than Idle mode and ADC Noise Reduction mode. The user is advised to write zero to ADEN before entering such sleep modes to avoid excessive power consumption.

If the ADC is enabled in such sleep modes and the user wants to perform differential conversions, the user is advised to switch the ADC off and on after waking up from sleep to prompt an extended conversion to get a valid result.

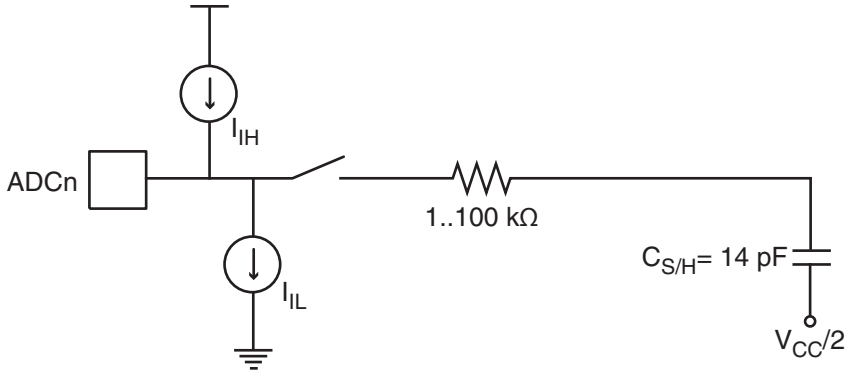
25.6.1 Analog Input Circuitry

The analog input circuitry for single ended channels is illustrated in Figure 25-8. An analog source applied to ADCn is subjected to the pin capacitance and input leakage of that pin, regardless of whether that channel is selected as input for the ADC. When the channel is selected, the source must drive the S/H capacitor through the series resistance (combined resistance in the input path).

The ADC is optimized for analog signals with an output impedance of approximately 10 kΩ or less. If such a source is used, the sampling time will be negligible. If a source with higher impedance is used, the sampling time will depend on how long time the source needs to charge the S/H capacitor, which can vary widely. The user is recommended to only use low impedant sources with slowly varying signals, since this minimizes the required charge transfer to the S/H capacitor.

Signal components higher than the Nyquist frequency ($f_{ADC}/2$) should not be present for either kind of channels, to avoid distortion from unpredictable signal convolution. The user is advised to remove high frequency components with a low-pass filter before applying the signals as inputs to the ADC.

Figure 25-8. Analog Input Circuitry



25.6.2 Analog Noise Canceling Techniques

Digital circuitry inside and outside the device generates EMI which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. Keep analog signal paths as short as possible. Make sure analog tracks run over the ground plane, and keep them well away from high-speed switching digital tracks.
2. The AVCC pin on the device should be connected to the digital V_{CC} supply voltage via an LC network as shown in Figure 25-9 on page 285.
3. Use the ADC noise canceler function to reduce induced noise from the CPU.
4. If any ADC port pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

Figure 25-9. ADC Power Connections, ATmega1281/2561.

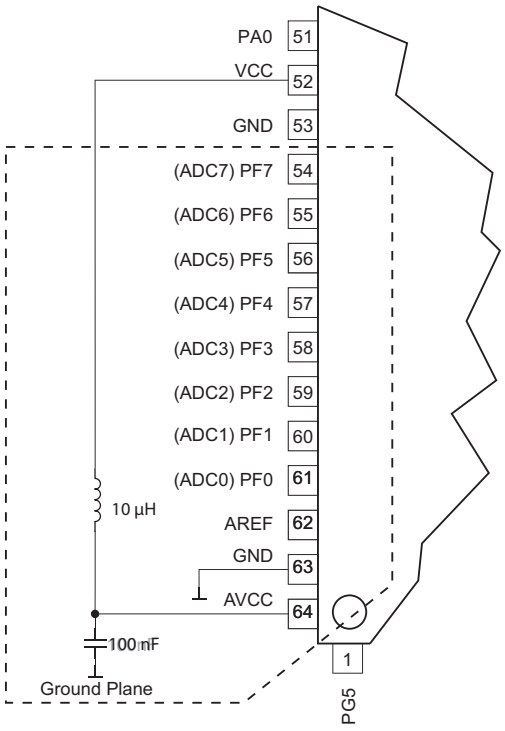
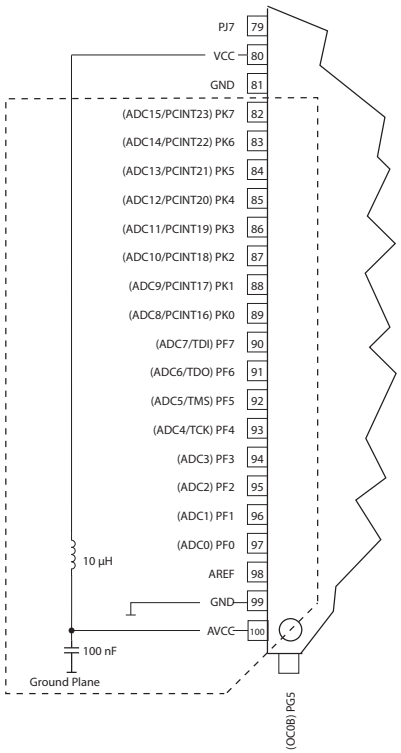


Figure 25-10. ADC Power Connections, ATmega640/1280/2560



25.6.3 Offset Compensation Schemes

The stage has a built-in offset cancellation circuitry that nulls the offset of differential measurements as much as possible. The remaining offset in the analog path can be measured directly by selecting the same channel for both differential inputs. This offset residue can be then subtracted in software from the measurement results. Using this kind of software based offset correction, offset on any channel can be reduced below one LSB.

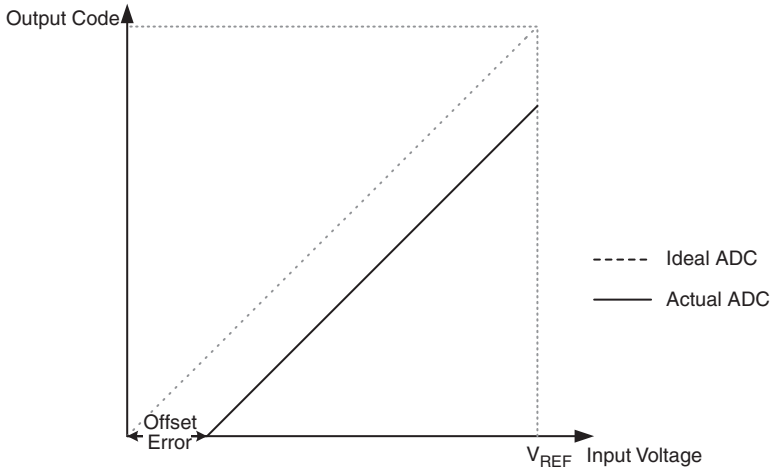
25.6.4 ADC Accuracy Definitions

An n-bit single-ended ADC converts a voltage linearly between GND and V_{REF} in 2^n steps (LSBs). The lowest code is read as 0, and the highest code is read as 2^n-1.

Several parameters describe the deviation from the ideal behavior:

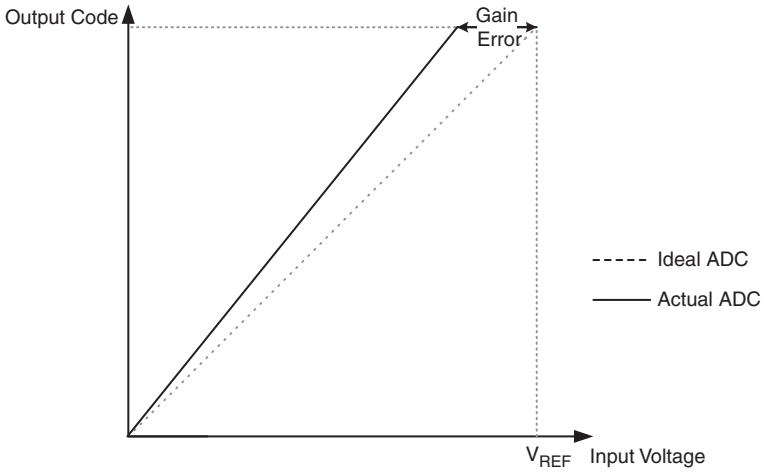
- Offset: The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

Figure 25-11. Offset Error



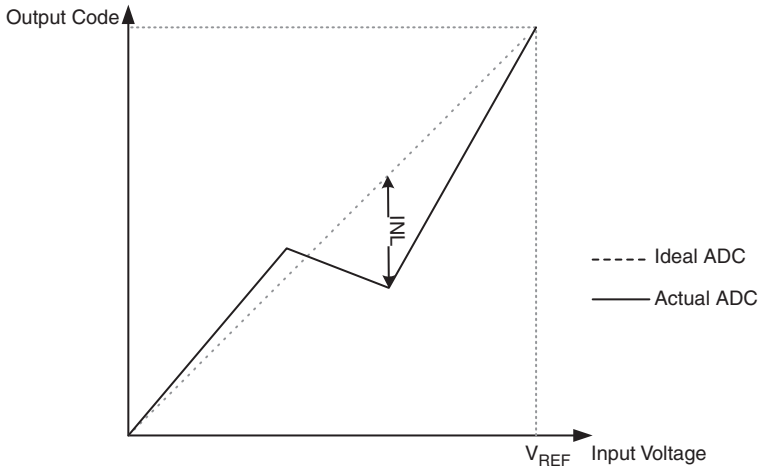
- Gain Error: After adjusting for offset, the Gain Error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB.

Figure 25-12. Gain Error



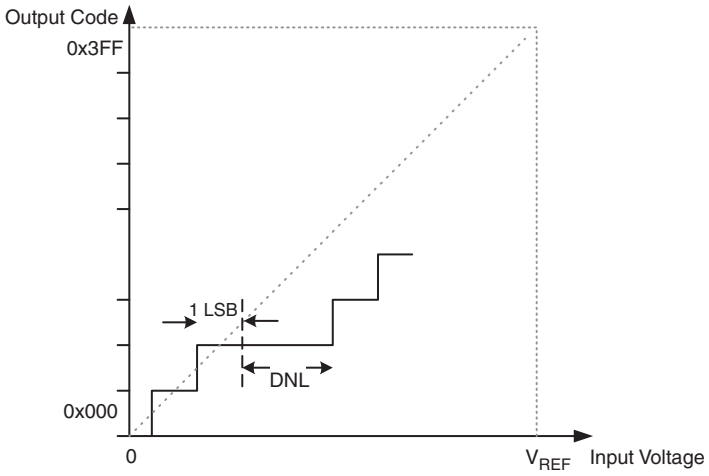
- Integral Non-linearity (INL): After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code. Ideal value: 0 LSB.

Figure 25-13. Integral Non-linearity (INL)



- Differential Non-linearity (DNL): The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1 LSB). Ideal value: 0 LSB.

Figure 25-14. Differential Non-linearity (DNL)



- Quantization Error: Due to the quantization of the input voltage into a finite number of codes, a range of input voltages (1 LSB wide) will code to the same value. Always ±0.5 LSB.
- Absolute Accuracy: The maximum deviation of an actual (unadjusted) transition compared to an ideal transition for any code. This is the compound effect of offset, gain error, differential error, non-linearity, and quantization error. Ideal value: ±0.5 LSB.

25.7 ADC Conversion Result

After the conversion is complete (ADIF is high), the conversion result can be found in the ADC Result Registers (ADCL, ADCH).

For single ended conversion, the result is

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

where V_{IN} is the voltage on the selected input pin and V_{REF} the selected voltage reference (see Table 25-3 on page 289 and Table 25-4 on page 290). 0x000 represents analog ground, and 0x3FF represents the selected reference voltage minus one LSB.

If differential channels are used, the result is

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot 512}{V_{REF}}$$

where V_{POS} is the voltage on the positive input pin, V_{NEG} the voltage on the negative input pin, and V_{REF} the selected voltage reference. The result is presented in two's complement form, from 0x200 (-512d) through 0x1FF (+511d). Note that if the user wants to perform a quick polarity check of the result, it is sufficient to read the MSB of the result (ADC9 in ADCH). If the bit is one, the result is negative, and if this bit is zero, the result is positive. Figure 25-15 shows the decoding of the differential input range.

Table 25-2 on page 289 shows the resulting output codes if the differential input channel pair (ADCn - ADCm) is selected with a gain of GAIN and a reference voltage of V_{REF}.

Figure 25-15. Differential Measurement Range

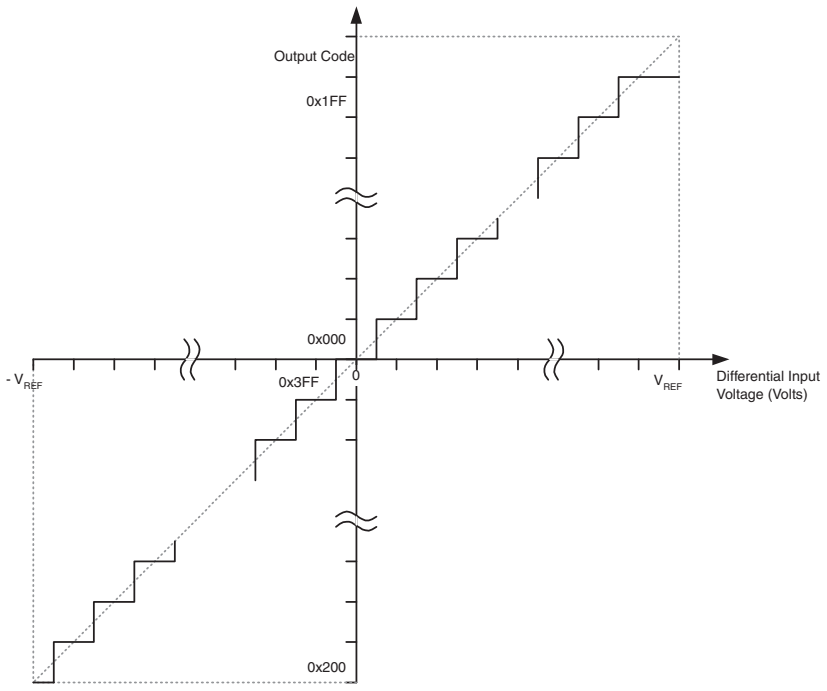


Table 25-2. Correlation Between Input Voltage and Output Codes

V_{ADCn}	Read Code	Corresponding Decimal Value
V_{ADCm} + V_{REF} / GAIN	0x1FF	511
V_{ADCm} + 0.999 V_{REF} / GAIN	0x1FF	511
V_{ADCm} + 0.998 V_{REF} / GAIN	0x1FE	510
...	...	...
V_{ADCm} + 0.001 V_{REF} / GAIN	0x001	1
V_{ADCm}	0x000	0
V_{ADCm} - 0.001 V_{REF} / GAIN	0x3FF	-1
...	...	...
V_{ADCm} - 0.999 V_{REF} / GAIN	0x201	-511
V_{ADCm} - V_{REF} / GAIN	0x200	-512

Example:

ADMUX = 0xFB (ADC3 - ADC2, 10x gain, 2.56V reference, left adjusted result).

Voltage on ADC3 is 300 mV, voltage on ADC2 is 500 mV.

ADCR = 512 × 10 × (300 - 500) / 2560 = -400 = 0x270.

ADCL will thus read 0x00, and ADCH will read 0x9C. Writing zero to ADLAR right adjusts the result: ADCL = 0x70, ADCH = 0x02.

25.8 Register Description

25.8.1 ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – REFS1:0: Reference Selection Bits**

These bits select the voltage reference for the ADC, as shown in Table 25-3. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

Table 25-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection[1]
0	0	AREF, Internal V_{REF} turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Internal 1.1V Voltage Reference with external capacitor at AREF pin
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Note: 1. If 10x or 200x gain is selected, only 2.56 V should be used as Internal Voltage Reference. For differential conversion, only 1.1V cannot be used as internal voltage reference.

- **Bit 5 – ADLAR: ADC Left Adjust Result**

The ADLAR bit affects the presentation of the ADC conversion result in the ADC Data Register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC Data Register immediately, regardless of any ongoing conversions. For a complete description of this bit, see “ADCL and ADCH – The ADC Data Register” on page 294.

- **Bits 4:0 – MUX4:0: Analog Channel and Gain Selection Bits**

The value of these bits selects which combination of analog inputs are connected to the ADC. See Table 25-4 for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

25.8.2 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	MUX5	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 3 – MUX5: Analog Channel and Gain Selection Bit**

This bit is used together with MUX4:0 in ADMUX to select which combination in of analog inputs are connected to the ADC. See Table 25-4 for details. If this bit is changed during a conversion, the change will not go in effect until this conversion is complete.

This bit is not valid for ATmega1281/2561.

Table 25-4. Input Channel Selections

MUX5:0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
000000	ADC0	N/A		
000001	ADC1			
000010	ADC2			
000011	ADC3			
000100	ADC4			
000101	ADC5			
000110	ADC6			
000111	ADC7			

Table 25-4. Input Channel Selections (Continued)

MUX5:0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
001000[1]	N/A	ADC0	ADC0	10x
001001[1]		ADC1	ADC0	10x
001010[1]		ADC0	ADC0	200x
001011[1]		ADC1	ADC0	200x
001100[1]		ADC2	ADC2	10x
001101[1]		ADC3	ADC2	10x
001110[1]		ADC2	ADC2	200x
001111[1]		ADC3	ADC2	200x
010000		ADC0	ADC1	1x
010001		ADC1	ADC1	1x
010010		ADC2	ADC1	1x
010011		ADC3	ADC1	1x
010100		ADC4	ADC1	1x
010101		ADC5	ADC1	1x
010110		ADC6	ADC1	1x
010111		ADC7	ADC1	1x
011000		ADC0	ADC2	1x
011001		ADC1	ADC2	1x
011010	N/A	ADC2	ADC2	1x
011011		ADC3	ADC2	1x
011100		ADC4	ADC2	1x
011101		ADC5	ADC2	1x
011110	1.1V (V_{BG})	N/A		
011111	0V (GND)			
100000	ADC8	N/A		
100001	ADC9			
100010	ADC10			
100011	ADC11			
100100	ADC12			
100101	ADC13			
100110	ADC14			
100111	ADC15			

Table 25-4. Input Channel Selections (Continued)

MUX5:0	Single Ended Input	Positive Differential Input	Negative Differential Input	Gain
101000[1]	N/A	ADC8	ADC8	10x
101001[1]		ADC9	ADC8	10x
101010[1]		ADC8	ADC8	200x
101011[1]		ADC9	ADC8	200x
101100[1]		ADC10	ADC10	10x
101101[1]		ADC11	ADC10	10x
101110[1]		ADC10	ADC10	200x
101111[1]		ADC11	ADC10	200x
110000		ADC8	ADC9	1x
110001		ADC9	ADC9	1x
110010		ADC10	ADC9	1x
110011		ADC11	ADC9	1x
110100		ADC12	ADC9	1x
110101		ADC13	ADC9	1x
110110		ADC14	ADC9	1x
110111		ADC15	ADC9	1x
111000		ADC8	ADC10	1x
111001		ADC9	ADC10	1x
111010		ADC10	ADC10	1x
111011		ADC11	ADC10	1x
111100		ADC12	ADC10	1x
111101	N/A	ADC13	ADC10	1x
111110	Reserved	N/A		
111111	Reserved	N/A		

Note: 1. To reach the given accuracy, 10x or 200x Gain should not be used for operating voltage below 2.7V.

25.8.3 ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

In Single Conversion mode, write this bit to one to start each conversion. In Free Running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

When this bit is written to one, Auto Triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the Data Registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a Read-Modify-Write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

- **Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

Table 25-5. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

25.8.4 ADCL and ADCH – The ADC Data Register

25.8.4.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

25.8.4.2 ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers. If differential channels are used, the result is presented in two's complement form.

When ADCL is read, the ADC Data Register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision (7 bit + sign bit for differential input channels) is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX, and the MUXn bits in ADMUX affect the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

- **ADC9:0: ADC Conversion Result**

These bits represent the result from the conversion, as detailed in "ADC Conversion Result" on page 288.

25.8.5 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	MUX5	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is reserved for future use. To ensure compatibility with future devices, this bit must be written to zero when ADCSRB is written.

- **Bit 2:0 – ADTS2:0: ADC Auto Trigger Source**

If ADATE in ADCSRA is written to one, the value of these bits selects which source will trigger an ADC conversion. If ADATE is cleared, the ADTS2:0 settings will have no effect. A conversion will be triggered by the rising edge of the selected Interrupt Flag. Note that switching from a trig-

ger source that is cleared to a trigger source that is set, will generate a positive edge on the trigger signal. If ADEN in ADCSRA is set, this will start a conversion. Switching to Free Running mode (ADTS[2:0]=0) will not cause a trigger event, even if the ADC Interrupt Flag is set.

Table 25-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

Note: Free running mode cannot be used for differential channels (see chapter "Differential Channels" on page 281).

25.8.6 DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0x7E)	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	DIDR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:0 – ADC7D:ADC0D: ADC7:0 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC7:0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

25.8.7 DIDR2 – Digital Input Disable Register 2

Bit	7	6	5	4	3	2	1	0	
(0x7D)	ADC15D	ADC14D	ADC13D	ADC12D	ADC11D	ADC10D	ADC9D	ADC8D	DIDR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:0 – ADC15D:ADC8D: ADC15:8 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding ADC pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the ADC15:8 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

Magnet Application Products

SD Series

Electromagnetic Buzzers
Pin Terminal

SD1209T3-A1, SD1209T5-A1(Applicable to automobile) / ø12mm TYPE

FEATURES

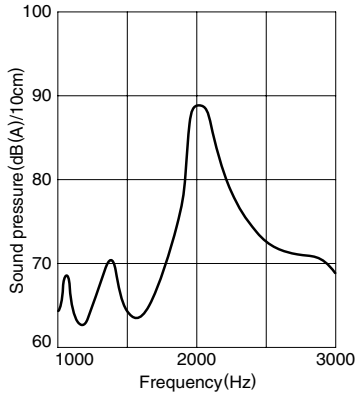
- These high reliability electromagnetic buzzers are applicable to automobile equipment.
- Compact, pin terminal type electromagnetic buzzer with 2048Hz output.
- Pin type terminal construction enables direct mounting onto printed circuit boards.

APPLICATIONS

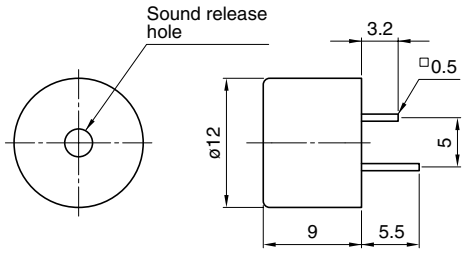
Clocks, travel watches, keyboards, toys, various alarms of automobile equipment.

FREQUENCY CHARACTERISTICS (SD1209T3-A1)

Around 2000Hz



SHAPES AND DIMENSIONS



Weight: 2g

•The longer pin is the + side. Dimensions in mm

SPECIFICATIONS

Part No.	SD1209T3-A1	SD1209T5-A1
Rated voltage Eo-p(V)	3	5
Operating voltage Eo-p(V)	1 to 4	3 to 6
Drive conditions:Rated voltage, square wave form [duty 50%]		
Frequency(Hz)	2048	2048
Sound pressure(dB(A)/10cm)min.	80	80
Current Io-p(mA)max.	60	60
DC resistance(Ω)typ.	50	90
Terminal construction	Pin terminal	Pin terminal
Operating temperature range	–40 to +85°C	
Storage temperature range	–40 to +85°C	

- Please ask us for details which is designed an another operating rated frequency.

SD160701, SD160709 / ø16mm TYPE

FEATURES

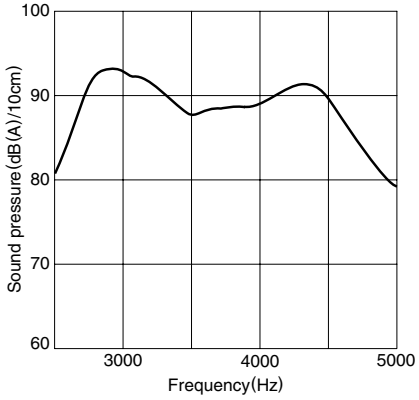
- These are thin-type electromagnetic buzzers with a height of only 7mm. They provide output over a relatively wide band, making them suitable for music.
- Pin type terminal construction enables direct mounting onto printed circuit boards.

APPLICATIONS

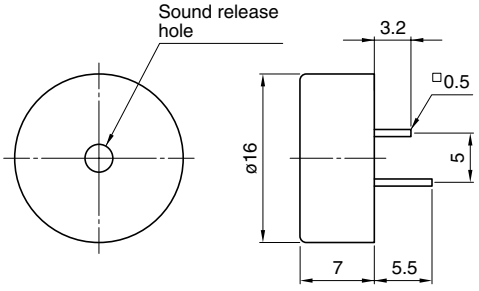
Travel watches, keyboards, handy-type equipment, toys.

FREQUENCY CHARACTERISTICS (SD160701)

Around 4000Hz



SHAPES AND DIMENSIONS



Weight: 3.5g

•The longer pin is the + side. Dimensions in mm

SPECIFICATIONS

Part No.	SD160701	SD160709
Rated voltage Eo-p(V)	3	5
Operating voltage Eo-p(V)	1 to 5	3 to 9
Drive conditions:Rated voltage, square wave form [duty 50%]		
Drive frequency(Hz)	4096	4096
Sound pressure(dB(A)/10cm)min.	80	80
Current Io-p(mA)max.	60	70
DC resistance(Ω)typ.	50	70
Terminal construction	Pin terminal	Pin terminal
Operating temperature range	–10 to +70°C	
Storage temperature range	–20 to +80°C	

- Please ask us for details which is designed an another operating rated frequency.

Magnet Application Products

SD Series

Electromagnetic Buzzers
Pin Terminal

SD1614T5-B1, SD1614TT-B1 / ø16mm TYPE

FEATURES

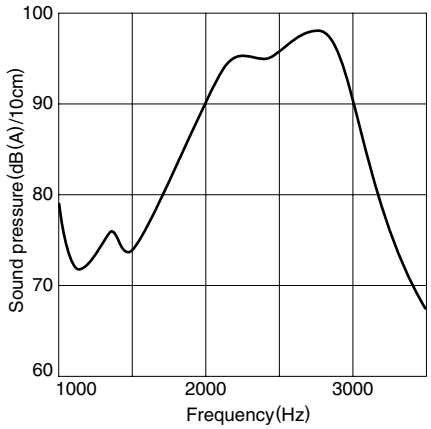
- These low-cost electromagnetic buzzers are suited to a wide range of applications.
- Pin type terminal construction enables direct mounting onto printed circuit boards.

APPLICATIONS

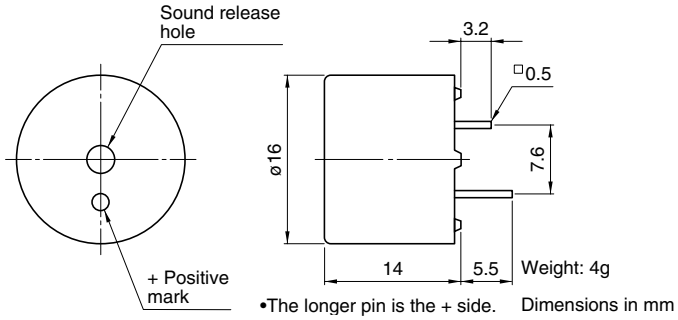
Office equipment, electronic games, toys, home appliances, key-boards.

FREQUENCY CHARACTERISTICS(SD1614T5-B1)

Around 2000Hz



SHAPES AND DIMENSIONS



SPECIFICATIONS

Part No.	SD1614T5-B1	SD1614TT-B1
Rated voltage Eo-p(V)	5	12
Operating voltage Eo-p(V)	3 to 6	6 to 15
Drive conditions: Rated voltage, square wave form [duty 50%]		
Drive frequency(Hz)	2048	2048
Sound pressure(dB(A)/10cm)min.	85	85
Current Io-p(mA)max.	80	100
DC resistance(Ω)typ.	70	120
Terminal construction	Pin terminal	Pin terminal
Operating temperature range	–10 to +70°C	
Storage temperature range	–20 to +80°C	

- Please ask us for details which is designed an another operating rated frequency.
- Products with 5mm terminal pitch are also available. Please substitute -B1 at the end of part number to -A1 in this case.

SD1614TT-B3M(Applicable to automobile) / ø16mm TYPE

FEATURES

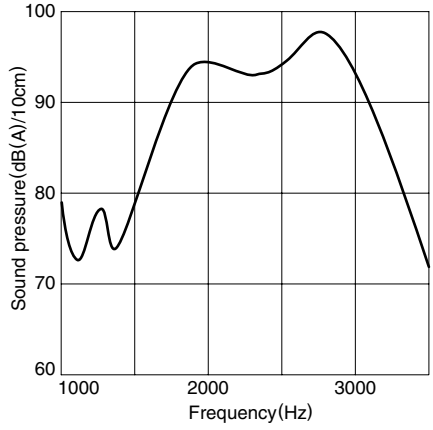
- These high reliability electromagnetic buzzers are applicable to automobile equipment.
- Pin type terminal construction enables direct mounting onto printed circuit boards.

APPLICATIONS

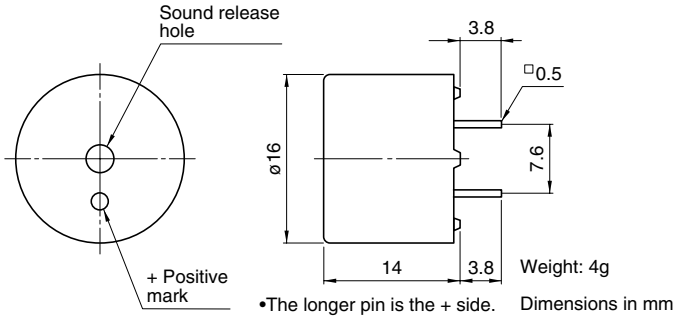
Various alarms of automobile equipment.

FREQUENCY CHARACTERISTICS(SD1614TT-B3M)

Around 2000Hz



SHAPES AND DIMENSIONS



SPECIFICATIONS

Part No.	SD1614TT-B3M
Rated voltage Eo-p(V)	12
Operating voltage Eo-p(V)	6 to 15
Drive conditions: Rated voltage, square wave form [duty 50%]	
Drive frequency(Hz)	2048
Sound pressure(dB(A)/10cm)min.	85
Current Io-p(mA)max.	100
DC resistance(Ω)typ.	120
Terminal construction	Pin terminal
Operating temperature range	–40 to +85°C
Storage temperature range	–40 to +85°C

- Please ask us for details which is designed an another operating rated frequency.
- Products with 5mm terminal pitch are also available. Part No.: SD1614TT-A5M

Magnet Application Products

SD Series

Electromagnetic Buzzers
Pin Terminal

SD1614T5-B5M(Applicable to automobile) / ø16mm TYPE

FEATURES

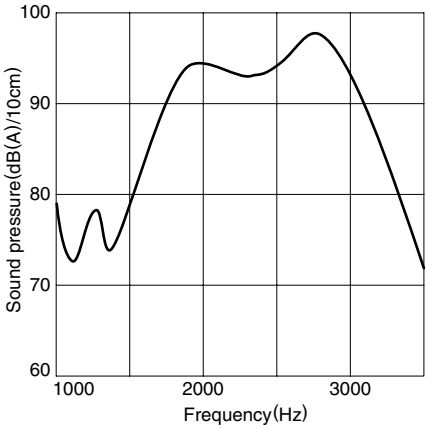
- These high reliability electromagnetic buzzers are applicable to automobile equipment.
- Pin type terminal construction enables direct mounting onto printed circuit boards.

APPLICATIONS

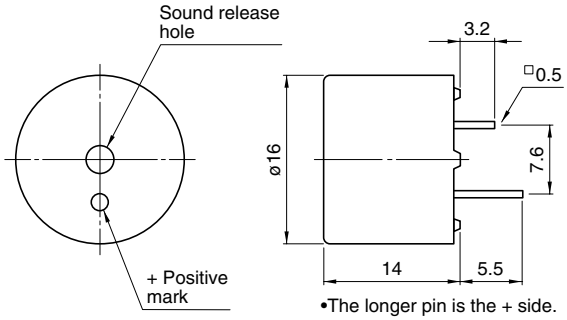
Various alarms of automobile equipment.

FREQUENCY CHARACTERISTICS(SD1614T5-B5M)

Around 2000Hz



SHAPES AND DIMENSIONS



•The longer pin is the + side.

Weight: 4g Dimensions in mm

SPECIFICATIONS

Part No.	SD1614T5-B5M
Rated voltage Eo-p(V)	5
Operating voltage Eo-p(V)	3 to 6
Drive conditions: Rated voltage, square wave form [duty 50%]	
Drive frequency(Hz)	2048
Sound pressure(dB(A)/10cm)min.	85
Current Io-p(mA)max.	80
DC resistance(Ω)typ.	70
Terminal construction	Pin terminal
Operating temperature range	–40 to +85°C
Storage temperature range	–40 to +85°C

- Please ask us for details which is designed an another operating rated frequency.

SD2412L5-T1M(Applicable to automobile) / ø24mm TYPE

FEATURES

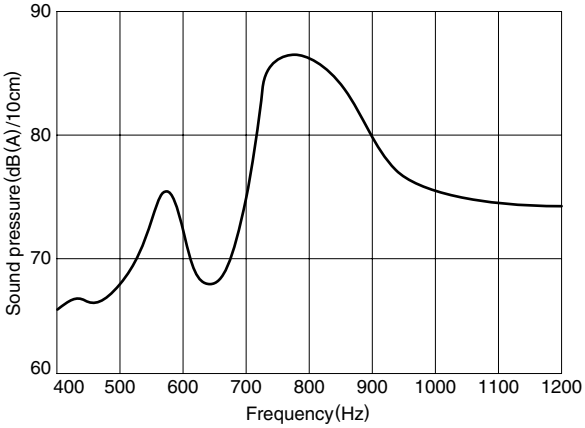
- Low-frequency type (800Hz output).
- Soft alarm tone.
- These high reliability electromagnetic buzzers are applicable to automobile equipments.
- Pin type terminal construction enables direct mounting onto printed circuit boards.

APPLICATIONS

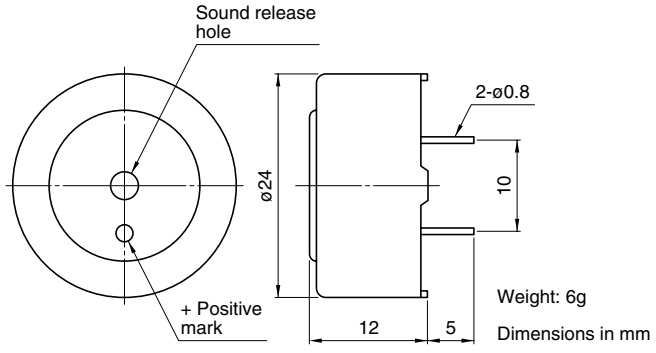
Various alarms of automobile equipment.

FREQUENCY CHARACTERISTIS(SD2412L5-T1M)

Around 800Hz



SHAPES AND DIMENSIONS



Weight: 6g

Dimensions in mm

SPECIFICATIONS

Part No.	SD2412L5-T1M
Rated voltage Eo-p(V)	5
Operating voltage Eo-p(V)	3 to 6
Drive conditions:Rated voltage, square wave form [duty 50%]	
Drive frequency(Hz)	800
Sound pressure(dB(A)/10cm)min.	80
Current Io-p(mA)max.	50
DC resistance(Ω)typ.	90
Terminal construction	Pin terminal
Operating temperature range	–40 to +85°C
Storage temperature range	–40 to +85°C

Magnet Application Products

SD Series

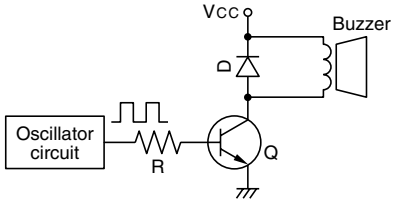
Electromagnetic Buzzers
Pin Terminal

SAFETY PRECAUTIONS FOR USING TDK ELECTROMAGNETIC BUZZER

Please be sure to read the specifications of TDK Electromagnetic Buzzer(hereinafter referred to as "the buzzer") before use.

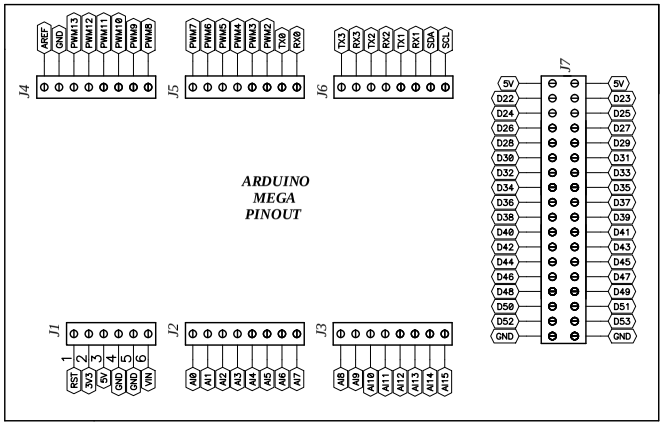
- Please pay sufficient attentions to the warnings for safe designing when using the buzzer.
  Incorrect usage may cause smoke or fire.
- Do not make sound on the buzzer when it is close to ears.
  Listening to the buzzer when it is close to ears, or continuant listening to it for an extended period, may result in hearing disorders.
  Thoroughly warn consumers not to turn on the sound when the buzzer is close to ears.
- Do not exceed the rated operation voltage range when using the buzzer.
  If it is exceeded beyond the rated operation voltage, a built-in coil may generate heat, resulting in deterioration of the coil, which may cause a drop in sound pressure or a change in the terminal resistance value.
  If a rare short circuit occurs due to heat generated in the built-in coil, a large electrical current flows to the equipment used, possibly causing it to emit smoke or fire.
- The buzzer is a magnetic type sounder.
  Do not use it for any other purpose than a sound generation.
- Do not use the buzzer in other than atmospheric air.
- Do not use the buzzer in the following environments (excluding cases in which countermeasures are taken).
  - Corrosive gases (Cl_2, NH_3, SOx, NOx, etc.)
  - Places to be exposed to dust.
  - Places where rain, fog, salt water, and the like will get on the buzzer.
- After installing the buzzer when designing an apparatus, confirm that there are no abnormalities by performing a reliability evaluation test.
- Do not exceed the rated operating temperature range when using the buzzer.
- Do not use for an extended period at relative humidity in excess of 90% (maximum wet-bulb temperature of 38°C).
- When using the buzzer for a safety device or warning device, take all surest measures such as providing a safety circuit or the joint use of another sounder with equivalent characteristics in order to prevent accidents.
- The buzzer may be influenced by electromagnetic waves, voltage changes, and ripple voltage.
  Check the buzzer in the mounted condition, and take measures such as safety circuits, shielding, etc.
- If the buzzer is molded or secured by adhesives, thermal expansion stresses from the mold or adhesive agent may cause deterioration in a soldered terminal part.
  Before molding or securing an adhesive, consider the type, amount, hardening conditions, adhesive properties, etc., of the sealing material, and confirm the reliability.
- Do not apply vibrations or shocks (such as dropping) larger than the specified.
- Do not bend the terminal pin when soldering to install on a circuit board.
- Do not apply larger than the specified force to the terminal pin.
- Do not apply force to the terminal pin at high temperatures (when soldering).
- When soldering a terminal pin, perform the soldering in clean conditions without contamination or rust in order to avoid poor connections.
- When performing dip soldering directly for installing on a printed circuit board, be careful so that smoke from the flux, or other foreign materials, will not get into the sound emission hole.
- Do not clean(excluding products that can be cleaned).
- With regards to devices in which the consumers can touch the buzzer, thoroughly warn consumers not to put foreign materials (such as needles, hairpins, garbage, etc.) into the sound emission hole.
- As the buzzer uses a coil, it has an inductive load.
  Protect the drive circuit by putting the diodes in parallel into the buzzer.
- Do not reconfigure the buzzer.
- Sound pressure changes depending on the shape of the part to be installed in a device, or the installation location.
  Check the buzzer in the mounted condition.
- The buzzer has polarities.
  When installing, be careful not to make a mistake with the polarities. Incorrect polarities will cause the following irregularities.
  - Without circuit type: Differences in sound pressure will be created.
  - With circuit type: No sound will be generated.
- Frequencies other than the rated frequency will change the characteristics (sound pressure), and may cause wide unevenness.
- As the buzzer uses a magnet, it generates a surrounding magnetic field.
  When designing a device, etc., confirm that there are no irregularities due to the magnetic field.
- The characteristics of the buzzer may change because of the influence of an outside magnetic field.
  Check the buzzer in the mounted condition, and take measures such as shielding, etc.
- Because of deterioration or damage, maintain in rated storage temperature range, avoid environments where there are sudden temperature changes, direct sunlight, corrosive gases and dust, and store wrapped in order to avoid applying stresses.

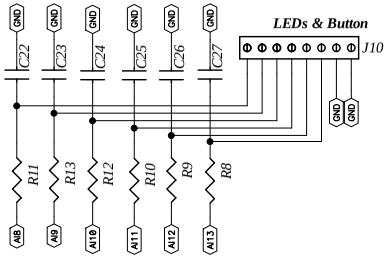
RECOMMENDED OPERATING CIRCUIT EXAMPLE

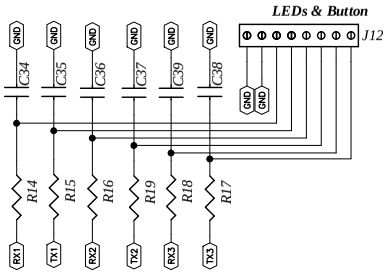


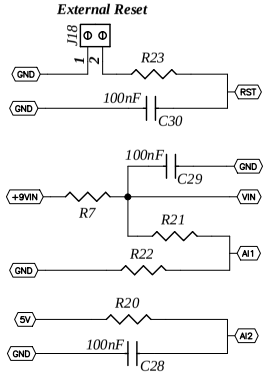
ANEXO III

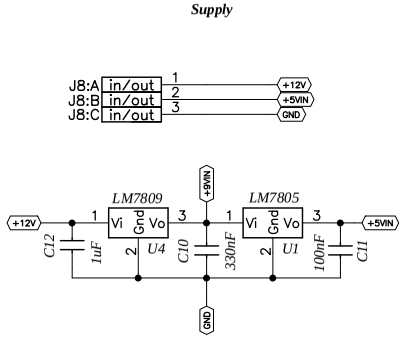
El anexo III contiene el esquemático y el layout de la placa de circuito impreso diseñada para este proyecto.

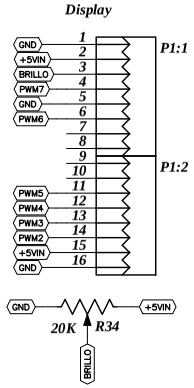


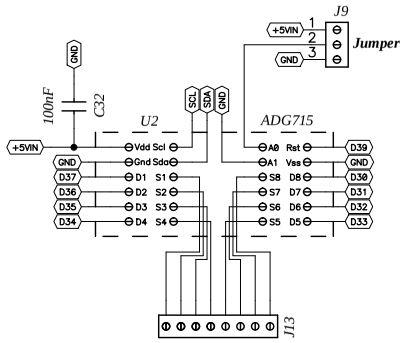


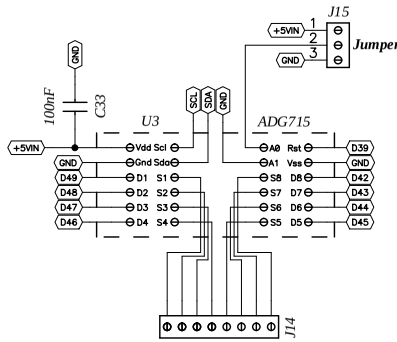












Title		
Size A3	Number	Rev
Date		Drawn by
Filename		Sheet of

RAUL ARENAZ
PEPE BARQUILLAS
CHEMA GARCIA
TFG – 2014/2015
GND 5V 12V
J4
J5
J6
J3
J7
J1

ANEXO IV

En el anexo IV se adjuntan los archivos con las especificaciones de dos de los sistemas de verificación de circuitos integrados comercializados. Éstos se han obtenido de las páginas web de las empresas que los comercializan.

El primero es fabricado por la compañía Ecuatoriana TESCA S.A. (Grupo MAESSA), especializada en montajes electromecánicos, instrumentación y prefabricación y mantenimiento de instalaciones industriales, petroleras, de generación y distribución eléctrica. El segundo pertenece a Kitek, empresa India dedicada principalmente al diseño de este tipo de dispositivos.

Order Code - 68004
Linear IC Functional Tester (Microprocessor Based)

It is the latest Microprocessor controlled linear IC Functional tester. It functionally test a variety of linear ICs including commonly used Operational amplifiers, Comparators, Transistor arrays, Optocouplers, Analog switches, Multiplexers, Voltage followers, Timers, A to D converters, D to A converters, cross point switches and other selected special function analog Ics.

TEST LIBRARY

OPERATIONAL AMPLIFIERS

LM101, LM107, LM108, LM112, LM124, LM143, LM144, LM148, LM149, LM158, LM192, Lm201, LM207, LM208,LM212, LM216, LM224, LM248, LM249, LM258, LM292, LM301, LM307, LM308, LM 312, LM316, LM324, LM343, LM348, LM349, LF351, LF356, LF357, LM358, LM392, LM709, LM714, LM725, LM741(8/14pin) LF747(10/14pin), LM748, LM1458, LM1558, LM2902, LM2924, LM 3301, LM3401, HA17080, HA18082, HA17083, HA17084, HA17301, HA17324, HA17358,, HA17458, HA17474, HA17558, HA17741(8/14pin), HA17747, HA17747, HA17902, HA17904, UCOPO1, UCOPO2, CA3130, CA3140, TL074, Tl084.

COMPARATORS

LM106, LM111(8/14pin) LF11, LM119, LM139, LM160(8/14pin), LM161, LM193, LM206, LM211(8/14pin), LF211, LM219, LM239, LM260(8/14pin), Lm261, LM293, LM311(8/14pin), LF311, LM319, LM339, LM360(8/14pin), LM361, LM393, LM710, LH2111, LH2211, LH2311, LM2901, LM3302, HA173302, HA17339, JA17393, HA17901, Ha17903,

TRANSISTORARRAYS

ULN2001, ULN2002, ULN2003, ULN2004, ULN2005, ULN2064, ULN2065, ULN2066, ULN2067, ULN2068, ULN2069, ULN2074,ULN2075, TD62501, TD62502, TD62503, TD62504, TD62505, TD62506, TD62507, 75064, 75065, 75066, 75067, 75068, 75069, 75074, 75075, 75430, 75446, 75447, 75448, 75449, 75450, 75460, 75465, 75466, 75467, 75468, 75469, 75470, 75491, 75492.

OPTOCOUPLERS

MCT2, MCT2E, 4N25, 4N26, 4N27, 4N28, 4N32, 4N33, 4N35, 4N36, 4N37, 6N135, 6N136, 6N137, 6N138, 6N139, TIL111, TIL112, TIL113, TIL114, TIL115, TIL116, TIL117, TIL118, TIL119, TIL127, TIL128, TIL153, TIL154, TIL155, TIL156, TIL157, TIL181, TIL189, TIL190, MOC3009, MOC3010, MOC3011, MOC3012, MOC20, MOC3021, MOC3022, MOC3023, HCPL2502, HCPL2601, HCPL2630, OP18012, OP18013, OP18014, Op18015.

ANALOG SWITCHES.MULTIPLEXERS

CD4016, CD4066, CD4051, CD4052, CD4053, HI200, HI201, HI300, HI301, HI302, HI303, HI304, HI305, Hi306, Hi307.

VOLTAGE FOLLOWERS

LM102, LM202, LM302 LM110,(14PIN), LM210(14PIN), LM310,(8/14PIN), LM2110, LM2210, LM2310.

TIMERS

LM555, NE555, HA17555

A TO D CONVERTERS

ADC0801, ADC0802, ADC0803, ADC0804.

D TO A CONVERTERS

DAC0800, DAC0806, DAC0807, DAC0808, HA 17008R, Ha17408,

MISCELLANEOUS LINEARICS

75182, 75183, 75189, DS7820A, Ds7830, DS8820A, Ds8830, Mc1489

Features:

- No personality cards/reference IC/IC data required.
- Automatic testing of IC after entering the IC number. No wiring is required.
- Continuous mode of testing ICs until Aborted. Number of times the test was successful id displayed.
- Buzzer to indicate bad IC
- Separate field for display of mode of operation (One/Con/Stp).
- Backspace, clear, select option and number keys for editing IC number.
- Remembers the IC number for repeated testing of same IC
- Self Test facility during Power ON and through keyboard
- Help key for short description of the various key functions
- Potential free 20 pin (ZIF Insertion Force) socket for easy insertion and removal of IC under test
- Computer interface through RS232C for enhancement of IC test library

Technical Specifications:

Package	Linear ICs and Analog & digital combined ICs of 6 , 8, 10, 14, 16, 18, 20, pins dual in line. (Also round pack if inserted after bending leads to suit DIP)
Range	Operational amplifiers, Comparators, Transistor arrays, Optocouplers, Analog switches, Multiplexers, Voltage Followers, Timers, A to D converters, D to A converters and other selected special function analog ICs,
Method	Output test with simulated inputs and at nominal loads. It performs only functional test and not parameter test.
Socket	20 pin DIP ZIF
Keyboard	24 feather touch keys
Display	16 character LCD dot matrix display
Voltage	230V ± 10% 50Hz AC
Enclosure	Sheet metal box with powder coating and screen printing

Note: Specifications are subject to change.

Tesca Technologies Pvt. Ltd.
305, Taru Chhaya Nagar, Tonk Road, Jaipur-302029, India
Tel: +91-141-2724326, Mob: +91-9413330765
Email: info@tesca.in, tesca.technologies@gmail.com
Website: www.tesca.in

DICT-03
UNIVERSAL IC TESTER

Features

- Tests a wide range of Digital IC's such as 74 Series, 40/45 Series of CMOS IC's.
- It can test Microprocessor 8085, 8086, Z80.
- It tests Peripherals like 8255, 8279, 8253, 8259, 8251, 8155, 6264,62256,8288,8284.
- It tests a wide range of Analog Ic's such as ADC, DAC, Opamp, 555, Transistor Arrays, Analog Switches, Waveform Generator, Line Drivers, Voltages Regulators, PLL's, VCO, PWM Generator, Sample & Hold, Voltages References, Opto couplers, Comparators, Voltages Followers and Others.
- It tests Seven segment display of common cathode & common anode type
- It has Auto search facility for Digital IC's.
- Test by: Truth table/sequence table comparison.
- ZIF: Two Nos. of 40 pin DIP ZIF sockets for Digital & Analog IC's.
- Keys: 50 cherry keys Key pad with numerical & functional keys.
- Display: 16x2 Backlit LCD Display
- Supply Input Voltage: 230V AC.

Device Supports

DIGITAL IC's

T.T.L. 74XXX series

7400 7401 7402 7403 7404 7405 7406 7407 7408 7409 7410 7411 7412 7413 7414 7415 7416 7417 7418 7419 7420 7421 7422 7423 7424 7425 7426 7427 7428 7430 7432 7433 7437 7438 7439 7440 7442 7443 7444 7445 7446 7447 7448 7449 7450 7451 7453 7454 7462 7464 7465 7470 7471 7472 7473 7474 7475 7476 7478 7483 7485 7486 7489 7490 7491 7492 7493 7495 7496 7497 74107 74109 74112 74113 74114 74116 74121 74122 74123 74125 74126 74128 74132 74133 74134 74135 74136 74137 74138 74139 74140 74141 74145 74147 74148 74150 74151 74152 74153 74154 74155 74156 74157 74158 74159 74160 74161 74162 74163 74164 74165 74166 74168 74169 74170 74172 74173 74174 74175 74176 74177 74180 74181 74182 74183 74184 74185 74189 74190 74191 74192 74193 74194 74195 74196 74197 74198 74221 74238 74240 74241 74242 74243 74244 74245 74246 74247 74248 74249 74251 74253 74256 74257 74258 74259 74260 74266 74273 74279 74280 74283 74290 74293 74298 74299 74322 74340 74341 74344 74347 74348 74350 74351 74352 74353 74354 74365 74366 74367 74368 74373 74374 74375 74377 74378 74379 74381 74382 74386 74390 74393 74395 74398 74399 74412 74423 74425 74426 74445 74447 74490 74521 74534 74540 74541 74543 74544 74563 74564 74573 74574 74575 74577 74580 74589 74590 74591 74595 74596 74620 74621 74622 74623 74638 74639 74640 74641 74642 74643 74645 74646 74647 74648 74649 74652 74668 74669 74670 74688 74689 74786 74800 74802 74804 74805 74808 74832 74841 74874 74901 74902 74903 74904 74906 74C923 74C925 74C926 74C927 74C928 74C929 74989 741244 741245 741623 741621 741639 741640 741641 741642 741643 741644 741645 743037 743038 74H01 74LS51 74H54 74L54 74L71 74H71 74LS73 74LS76 74LS78 74L85 74L86 74C89 74L93 74L95 74LS107 8280 8281 8290 8291

CMOS (40/45 XX SERIES)

4000 4001 4002 4006 4007 4008 4009 4010 4011 4012 4013 4014 4015 4017 4018 4019 4020 4021 4022 4023 4024 4025 4026 4027 4028 4029 4030 4031 4032 4033 4034 4035 4038 4040 4041 4042 4043 4044 4047 4048 4049 4050 4054 4055 4056 4060 4063 4067 4068 4069 4070 4071 4072 4073 4075 4076 4077 4078 4081 4082 4085 4086 4093 4094 4095 4096 4098 4099 40105 40106 40107 40109 40147 40160 40161 40162 40163 40174 40175 40181 40182 40192 40193 40194 40195 40244 40245 40257 40373 40374 40097 40098 4490 4502 4503 4504 4506 4507 4508 4510 4511 4512 4514 4515 4516 4518 4519 4520 4522 4526 4527 4528 4531 4532 4534 4538 4539 4541 4543 4544 4555 4556 4558 4562 4566 4572 4584 4585 4599 4723 4724 4727 4801 5801

CPU

8085 8086 V20 8088 8400(Z80) 6502 65C02 65SC02 8035 8039 8748 8749

PERIPHERAL

8155 8156 8205 8212 8216 8226 8237 8251 8253 8254 8255 8257 8259 8279 8282 8283 8284 8286 8287 8288 8250 82450 6350 6820 6821 6822 6840 6844 6845 6850 6851 6852 6854 6520 6521 6522 6524 6551 65C51 8420(Z80-PIO) 8430(Z80-CTC) 8440 (SIO-0) 8441 (SIO-1) 8442 (SIO-2) 8449 (SIO-90) 1852 1871 1879 2681

MEMORY

2102 2114 2115 2125 2149 6116 6264 62256 621024 9101 91L22 93412 93422 93425 4256 41256 5025

REAL TIME CLOCK

1879 5832 58167 82C8167 DS1287

PHASE FREQUENCY DETECTOR

MC4044 Mc4344

DECODER/ENCODER

1441 1442

SUPERVISORY CIRCUITRY

Dallas 1231 1232

SEVEN SEGMENT DISPLAY

Lt542 LT543

OSCILLATOR / DIVIDER

5369

LINEAR IC'S

ANALOG TO DIGITAL CONVERTER

ADC0800 ADC0801 ADC0802 ADC0803 ADC0804 ADC0805 ADC0808 ADC0816 ADC0817 AD574 7109 3162 ADC0809

Note : Specifications can be altered without notice in our constant efforts for improvement.

Kitek Technologies Pvt. Ltd., B-4, Lotus C.H.S., Plot No. 8, Sector - 7, Airoli, Navi Mumbai - 400708.
Tel.: 65116548 ◆ Telefax : 27694323 ◆ **Email:** knowl786@rediffmail.com ◆ **Web** : www.kitektechnologies.com

DICT-03
UNIVERSAL IC TESTER

DIGITAL TO ANALOG CONVERTER
DAC08 DAC0800 DAC0801 DAC0802 DAC0808 DAC1408 DAC1508 DAC558 DAC1020 DAC1021 DAC1022 DAC1220 DAC1221 DAC1222 DAC7520 DAC7521 DAC7523 DAC7524 DAC7530 DAC7531 DAC7533 DAC7541

COMPARATOR
106 111 160 193 211 260 293 311360 393 2903 17393 CA3290 111 119 139 161 211 219 239 261 311 319 339 361 2901 3302 17339 521 522 527 529 2111 2211 2311

OP-AMP
AD548 AD648 AD707 AD708 AD711 AD712 AD744 AD746 AD821 AD845 AD846 AD847 AD848 AD849 CA081 CA082 CA307 CA3100 CA3130 CA3140 CA3193 CA3240 CA3160 CA3260 LF 351 353 386 412 442 LM 10 11 101 107 108 112 118 143 155 156 157 158 201 207 208 212 218 255 256 257 258 301 307 308 312 318 343 355 356 357 358 411 441 530 531 532 538 714 725 741 748 1458 1558 2904 3080 33181 4250 4558 5204 5205 5230 5512 5532 5534 5535 13741 17301 17741 17458 17558 OP05 OP07 OP21 OP27 OP37 OPA121 OPA602 OPA606 OPA11 TL061 TL062 TL071 TL072 TL081 TL082 LT1013 709 1709 7611 7612 7621 AD713 AD840 AD841 AD842 CA084 LM 11 101 107 108 112 118 124 147 148 149 201 207 208 212 218 224 248 249 301 307 308 312 318 324 347 348 349 442 444 534 725 741 747 748 2900 2902 3303 3403 3503 5514 17301 17741 TL064 TL074 TL084 OPA404 7641 7642 LT1014 524 5533

VOLTAGE FOLLOWER
110 210 310 102 110 202 210 302 310

LINE DRIVERS & RECEIVER
1488 1489 3486 3487 75107 75108 75110 75114 75121 75174 75175 75176 75182 75183 75188 75189 75450 75451 75452 75453 75454 75477 75491 75492 75494 7820 7830 7831 7832 8820 8830 8831 8832 26LS31 26LS32 96174 96175 145406 10125

TRANSISTOR ARRAY
394 CA3028 CA3046 CA3053 CA3054 CA3146 CA3083 CA3086 ULN2001 ULN2003 ULN2004 ULN2005 ULN2015 ULN2011 ULN2013 ULN2014 ULN2021 ULN2023 ULN2024 ULN2069 ULN2025 ULN2064 ULN2065 ULN2068 ULN2801 ULN2803 ULN2815 ULN2804 ULN2805 ULN2811 ULN2813 ULN2814 ULN2821 54566 ULN2823 ULN2824 ULN2825 L601 L603 L604

ANALOG SWITCH
4016 4051 4052 4053 4066 6108 6208 11201 11202 11331 11508 11509 13201 13202 13331 11332 13332 13508 13509 14016 14051 14052 14053 14066 DG200 DG201 DG202 DG308 DG309 DG211 DG212 DG506 DG507 DG508 DG509

WAVEFORM GENERATOR
8038

TIMER
555 556 7555

PLL
565 567N 4046

VCO
131 231 331 131A 231A 331A 4151 566N

SAMPLE AND HOLD
198 298 398 5537

PWM GENERATOR
1524 2524 3524 494 594 3525

DPM IC
7106 7107 7116 7117 7126 7136 7137

OPTO-COUPLER
4N25 4N26 4N27 4N28 4N32 4N33 4N35 4N36 4N37 4N38 6N135 6N136 6N137 6N138 6N139 6N140 CNY171 CNY172 CNY173 H11A1 H11C1 H11C4 H11D1 H11D2 H11G1 H11G2 MCT2 MCT2E MCT6 MCT26 MCT210 MCA230 MCA231 MCA255 MOC3006 MOC3010 MOC3020 MOC3021 MOC3022 MOC3023 MOC3030 MOC3040 MOC3041 MOC3061 MOC3063 TIL111 TIL112 TIL113 TIL114 TIL115 TIL116 TIL117 TIL118 TIL119 TIL126 TIL155 HCPL 2531 2631 2731 2502 2602 2601 2630 3700 SFH600_0 SFH600_3 SFH610_1 SFH610_2 SFH611_1 SL5500 SL5501 PC817 PS2041 2501 EECF

CROSS POINT SWITCH
22100 22101 42100 45100

LATCH DRIVER
UCN4801 UCN5801

VOLTAGE REGULATOR
7805 78L05 78M05 78T05 7806 7808 7812 7815 7824 7905 79L05 79M05 7906 7908 7915 7918 117 117L 117M 217 217L 217M 317 317L 317M 337 18512 28512 38512 18525 28525 38525 2930 2931 723

VOLTAGE REFERENCE
185_1.2 185_2.5 285_1.2 285_2.5 385_1.2 385_2.5

DOT/BAR Display Driver
3914 3915

OPAMP AND COMPARATOR
192 292 392 2924

OVER VOLTAGE CROWBAR SENSING CIRCUIT
3423 3523

LED FLASHER
3909

FREQUENCY.TO VOLTAGE CONVERTER
2907 2917.

Note : Specifications can be altered without notice in our constant efforts for improvement.

Kitek Technologies Pvt. Ltd., B-4, Lotus C.H.S., Plot No. 8, Sector - 7, Airoli, Navi Mumbai - 400708.
Tel.: 65116548 ◆ Telefax : 27694323 ◆ Email: knowl786@rediffmail.com ◆ Web : www.kitektechnologies.com