LED Y CONSOLA SERIE

Un circuito electrónico

Descripción generada automáticamente con confianza bajavoid setup()

{

   pinMode( 3, OUTPUT) ;

   Serial.begin(9600) ;

}

void loop()

{

  if (Serial.available() > 0)

  {

   int x = Serial.parseInt();

   Serial.println ( x ) ;

   if (x >= 0 && x <= 255)

   {

    analogWrite (3, x) ;

    delay( 10);

   }

  }

}

BRILLO DE LOS LEDS

Imagen que contiene circuito, tabla

Descripción generada automáticamentevoid setup()

{

  pinMode( 3, OUTPUT) ;

  pinMode( 5, OUTPUT) ;

}

void loop()

{

  for ( int i= -255 ; i<=255 ; i++)

    {

    analogWrite (3, abs(i)) ;

    analogWrite (5, abs( abs(i) - 255)) ;

    delay( 10);

    }

}

BRILLO Y PULSADOR

int parar = 0 ;

int valor = -255 ;

Un circuito electrónico

Descripción generada automáticamente con confianza baja

void setup()

{

  pinMode( 3, OUTPUT) ;

  pinMode( 6, INPUT) ;

  Serial.begin(9600) ;

}

void loop()

{

  while ( parar == 0 )

  {

    analogWrite (3, abs(valor)) ;

    delay( 10);

    if ( valor == 255)

    {

      valor = -valor ;

    }

    else

    {

      valor++ ;

    }

    if ( digitalRead(6) == false )

    {

      parar = 1 ;

      Serial.println ( "para" ) ;

      delay( 500);

    }

  }

   while ( parar == 1 )

  {

    if ( digitalRead(6) == false )

    {

      parar = 0 ;

      Serial.println ( "sigue" ) ;

      delay( 500);

    }

  }

}

LED Y ZUMBADOR

// TONOS //

Un circuito electrónico

Descripción generada automáticamente con confianza baja// Definiendo la relación entre nota, período y frecuencia.

// el período está en microsegundos por lo que P = 1/f \* (1E6)

#define c3 7634

#define d3 6803

#define e3 6061

#define f3 5714

#define g3 5102

#define a3 4545

#define b3 4049

#define c4 3816 // 261 Hz

#define d4 3401 // 294 Hz

#define e4 3030 // 329 Hz

#define f4 2865 // 349 Hz

#define g4 2551 // 392 Hz

#define a4 2272 // 440 Hz

#define a4s 2146

#define b4 2028 // 493 Hz

#define c5 1912 // 523 Hz

#define d5 1706

#define d5s 1608

#define e5 1517 // 659 Hz

#define f5 1433 // 698 Hz

#define g5 1276

#define a5 1136

#define a5s 1073

#define b5 1012

#define c6 955

#define R 0 // Define una nota especial, 'R', para representar un silencio

// MELODIAS y TIEMPO //

// melody[] es una matriz de notas, acompañada de beats[],

// que establece la duración relativa de cada nota (# más alto, nota más larga)

// Melodía 1: Marcha Imperial de Star Wars

int melodia1[] = { a4, R, a4, R, a4, R, f4, R, c5, R, a4, R, f4, R, c5, R, a4, R, e5, R, e5, R, e5, R, f5, R, c5, R, g5, R, f5, R, c5, R, a4, R};

int tiempos1[] = { 50, 20, 50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 80, 50, 20, 50, 20, 50, 20, 40, 5, 20, 5, 60, 10, 40, 5, 20, 5, 60, 40};

// Melodía 2: Tema de Star Wars

int melodia2[] = { f4, f4, f4, a4s, f5, d5s, d5, c5, a5s, f5, d5s, d5, c5, a5s, f5, d5s, d5, d5s, c5};

int beats2[] = { 21, 21, 21, 128, 128, 21, 21, 21, 128, 64, 21, 21, 21, 128, 64, 21, 21, 21, 128 };

int MAX\_COUNT1 = sizeof (melodia1) / 2; // Duración de la melodía, para looping.

int MAX\_COUNT2 = sizeof (melodia2) / 2; // Duración de la melodía, para looping.

long tempo = 10000; // Establecer el tempo general

int pausa = 1000; // Establecer la duración de la pausa entre notas

int rest\_count = 50; // Variable de bucle para aumentar la duración del descanso (BLETCHEROUS HACK; Ver NOTAS)

int altavozSalida = 7; // Configure el altavoz en el pin digital 7

 /// TONO DE REPRODUCCIÓN //

// Pulse el altavoz para reproducir un tono durante una duración determinada

void playTone(int tono, int tempo, long duracion) {

  long tiempo\_transcurrido = 0;

  //Serial.println ( "tono tempo duracion " + String(tono)+ " " + String(tempo)+ " " + String(duracion) ) ;

  if (tono > 0) { // si este no es un tiempo de descanso, mientras que el tono tiene

    analogWrite (3, HIGH) ;

    // jugado menos tiempo que 'duración', altavoz de pulso ALTO y BAJO

    while (tiempo\_transcurrido < duracion) {

      digitalWrite (altavozSalida, HIGH) ;

      delayMicroseconds(tono / 2);

      // ABAJO

      digitalWrite (altavozSalida, LOW);

      delayMicroseconds(tono / 2);

      // Realizar un seguimiento de cuánto tiempo pulsamos

      tiempo\_transcurrido = tiempo\_transcurrido + tono;

      //Serial.println ( "tiempotranscurrido " + String(tiempo\_transcurrido)) ;

    }

    analogWrite (3, LOW) ;

  }

  else { // Tiempo de reposo; retardo de tiempos de bucle

    for (int j = 0; j < rest\_count; j++) { // Ver NOTA en rest\_count

      delayMicroseconds(duracion);

    }

  }

}

void setup() {

  pinMode(altavozSalida, OUTPUT);

  analogWrite (3, LOW);

  Serial.begin(9600) ;

}

void loop() {

  // Inicializar variables principales

   int tonoM = 0;

   int tiempo = 0;

   long duracion = 0;

   int potVal = 0;

  // melodia 1

  for (int i=0; i<MAX\_COUNT1; i++) {

    Serial.println ( "i: " + String(i) ) ;

    tonoM = melodia1[i];

    tiempo = tiempos1[i];

    duracion = tiempo \* tempo; // Configurar el tiempo

    playTone(tonoM,tiempo,duracion); // Una pausa entre notas

    delayMicroseconds(pausa);

    }

  for (int i=0; i<MAX\_COUNT2; i++) {

    Serial.println ( "i: " + String(i) ) ;

    tonoM = melodia2[i];

    tiempo = beats2[i];

    duracion = tiempo \* tempo; // Configurar el tiempo

    playTone(tonoM,tiempo,duracion); // Una pausa entre notas

    delayMicroseconds(pausa);

    }

}

Un circuito electrónico

Descripción generada automáticamente con confianza bajaTAREA BLOQUE 2

int encendido = 0 ;

int luz = 250 ;

void setup()

{

  pinMode( 10, OUTPUT) ;

  pinMode( 5, INPUT) ;

  pinMode( 6, INPUT) ;

  pinMode( 7, INPUT) ;

  Serial.begin(9600) ;

  analogWrite (10, LOW ) ;

}

void loop()

{

    if ( digitalRead(5) == false )

    {

      if ( encendido == 0)

      {

        encendido = 1 ;

        analogWrite (10, luz) ;

        Serial.println ( "luz " + String(luz)) ;

        delay(200);

      }

      else

      {

        encendido = 0 ;

        analogWrite (10, LOW) ;

        delay(200);

      }

    }

    if ((encendido == 1) && (digitalRead(7) == false))

    {

      if ( luz < 250){luz = luz + 50 ;}

      analogWrite (10, luz) ;

      Serial.println ( "luz " + String(luz)) ;

      delay(200);

    }

    if ((encendido == 1) && (digitalRead(6) == false))

    {

      if ( luz > 0){ luz = luz - 50 ;}

      analogWrite (10, luz) ;

      Serial.println ( "luz " + String(luz)) ;

      delay(200);

    }

 }