

Nombre: David Diaz-Rullo Torres	Proyecto : arduino(DDT-1)	Fecha:07/04/2015
	Índice (DDT -1.1)	Página 1 de 51
		Firma:



Diseño de un sistema de domotización con Arduino



David Diaz-Rullo torres

18/06/2015

IES Alonso Quijano (Quintanar de la orden)

Nombre: David Díaz-Rullo Torres	Proyecto : arduino(DDT-1)	Fecha:07/04/2015
	Índice	Página 2 de 51
	(DDT -1.1)	Firma:

INDICE

Objeto del proyecto pagina 3 hasta la 5

Temporalización del proyecto pagina 6

Memoria del proyecto pagina 7 hasta la 24

- 1.- **Montaje del control de las tres luces de las habitaciones** pagina 7-10
- 2.- **montaje del control de la iluminación de la luz del salón** pagina 11-13
- 3.- **control mediante un servo de la puerta de entrada a la casa.** Página 14-16
- 4.- **Control de la apertura de la puerta corredera de la puerta** Página 17-19
- 5.- **control de la apertura del salón mediante servos.** Página 20- 24

Esquemas eléctricos

- 1.- **Montaje del control de las tres luces de las habitaciones** página 25
- 2.- **montaje del control de la iluminación de la luz del salón** página 26
- 3.- **control mediante un servo de la puerta de entrada a la casa.** Página 27
- 4.- **Control de la apertura de la puerta corredera de la puerta** Página 28
- 5.- **control de la apertura del salón mediante servos.** Página 29

Pliego de condiciones pagina 30-41

Presupuesto página 4

Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 3 de 51
		Firma:

OBJETO

El motivo de la automatización de una vivienda puede ir desde el simple confort que puede ofrecer a una persona a la hora de no tener que desplazarse para apagar una luz, a su verdadero sentido que sería el de ayudar a esas personas con una movilidad reducida o con otros problemas que hacen que las cosas más simples sean un gran trabajo para ellos. Mi proyecto consigue que se aumente el nivel de vida de estas persona así como aumentar su autonomía las personas con problemas de movilidad no necesitaría que nadie le abriese la puerta para salir de su casa, no tendría que bajar del coche para pasar a su casa, no tendría que montarse en su silla para apagar o encender una simple luz, gracias a la placa controladora de Arduino capaz de realizar todas estas funciones con un precio reducido.

Con la domotización de nuestra casa intentamos satisfacer las necesidades del hombre, controlar todas las tareas básicas de una casa con el objeto de hacer más fácil su día a día y poder realizar las tareas más repetitivas del día a día desde un simple botón. Gracias a la placa controladora Arduino capaz de captar información del exterior y actuar de diferente forma dependiendo del valor obtenido. En este proyecto hemos conseguido automatizar las tareas de una vivienda como son el control de todas las luces de la casa, control o regulación de la luz, apertura automática de la puerta de la entrada a la casa y de la puerta corredera de la entrada todo con la placa Arduino mega 2560, mediante comandos.

Arduino es una plataforma de computación física se basa en una placa controladora con un software libre, que nos permite aparte de controlar todo lo que desees también permite tener en la red una gran cantidad de información sobre su utilización, consejos etc..

Las mayores ventajas de Arduino son su precio bajo, la posibilidad de implementarlo en cualquier sistema operativo, el entorno de programación simple capaz de ser utilizados por expertos que podrán sacar todo el provecho a su placa y por novatos que irán aprendiendo poco a poco como se programa con la programación utilizada por Arduino Processing que es fácil e intuitiva.

Arduino mega 2560 cuenta con 54 pines digitales que funcionan como entrada y salida; 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión mediante USB, un botón de reset y una entrada para la alimentación de la placa.

Las salidas PWM (modulación por ancho de pulsos) nos ofrecen el control de circuitos analógicos.

Las entradas y salidas digitales: solo tienen dos estados y pueden estar encendidas (HIGH) o apagada (LOW)

Las entradas analógicas las cuales nos permiten poder conectar a nuestra placa todo tipo de sensores y otros dispositivos con una salida variable sobre la cual deberes actuar de distinta forma en nuestro Arduino.

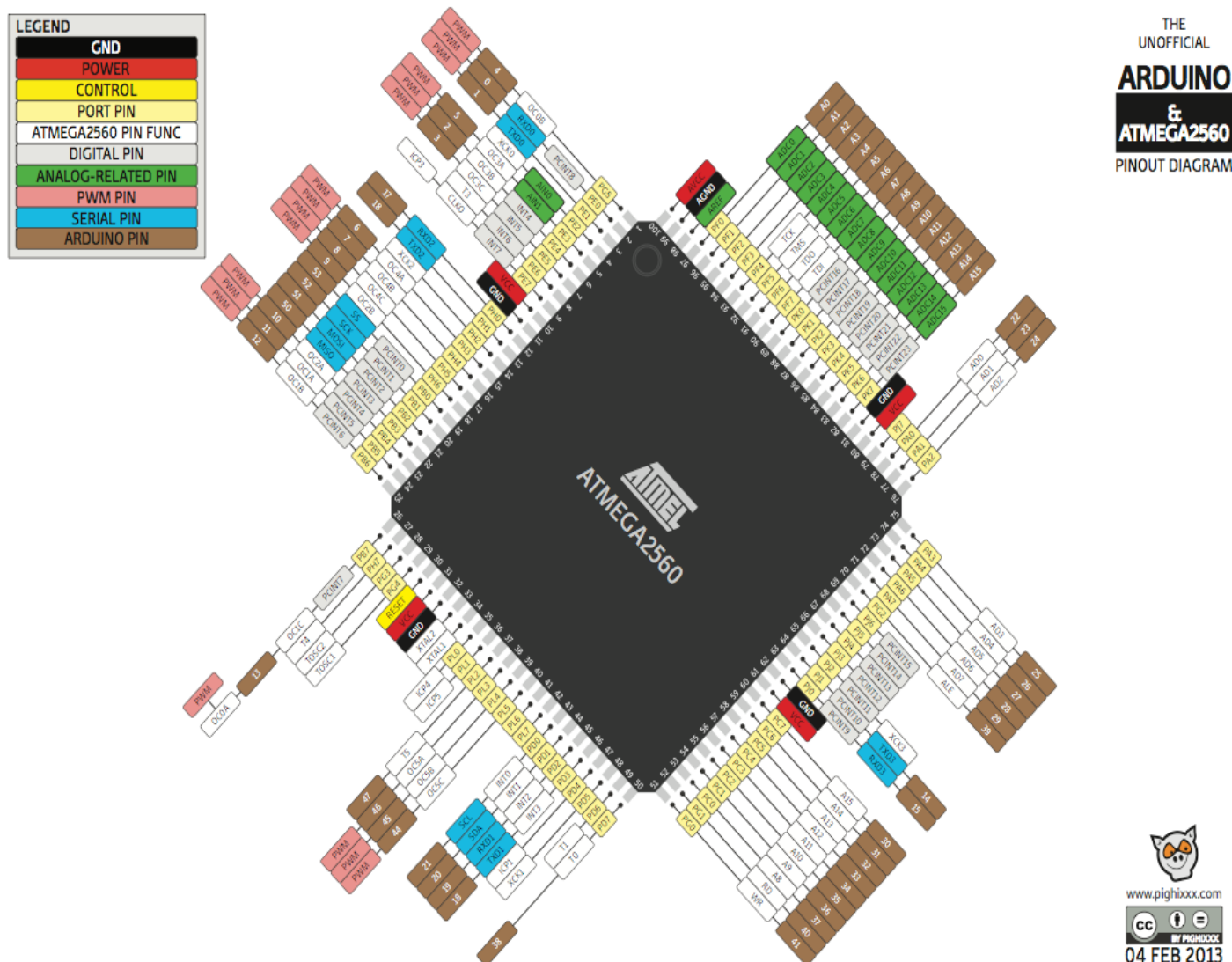
Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 4 de 51
		Firma:

Los puertos de comunicación que son utilizados a la hora de tener shield's bluetooth o wifi de tal modo que la placa puede ser controlada desde el exterior sin necesidad de cables o te permite estar conectado con el exterior para obtener información.

Arduino es capaz de ser alimentado por dos puntos y de diferentes formas dispone del puerto USB Jack del tipo B el utilizado comúnmente en las impresoras, también puede ser alimentado desde una entrada de alimentación que tiene Arduino con una entrada Jack, tiene otra entrada que se debería de utilizar como último recurso son unos pines colocados en la derecha desde donde se puede alimentar la placa.

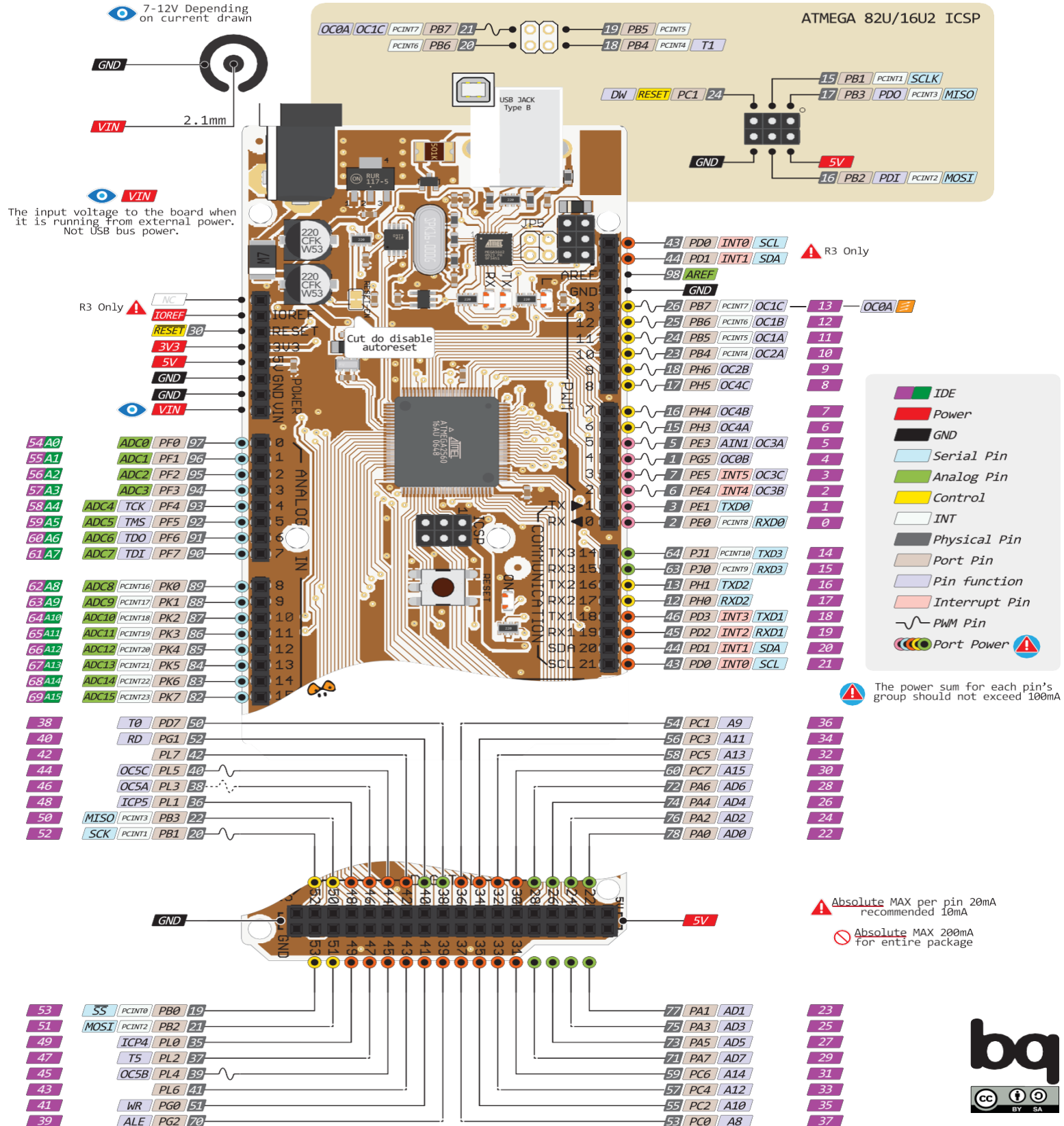
Para alimentar se necesita una fuente de alimentación capaz de ofrecer como mínimo 5V y 1A pero lo más aconsejado es una tensión entre los 7 y los 12V con 1 A.

Adjunto una foto de las conexiones del procesador de la placa para mayor aclaración



Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 5 de 51
		Firma:

También un esquema con todos los pines de los que dispone la placa at mega 2560



Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 6 de 51
		Firma:

TEMPORIZACION DEL PROYECTO

La realización del proyecto ha tenido un tiempo máximo de 3 meses.

El material necesario estaba a mi disposición para su montaje desde el primer día.

La primera semana y la segunda fue dedicada a la toma de contacto con Arduino así como el montaje del código para llegar a controlar las luces de mi maqueta con la pulsación de un botón se encendería una de ellas y al volver a pulsarlo esta se apagaría. Realizamos el control de tres luces una para cada habitación de la casa. A la vez que comenzabas el montaje de la maqueta.

La tercera y la cuarta semana estuvo dedicada a realizar el control de la intensidad de una luz esta luz seria colocada en el salón y tendría un máximo de 4 pulsaciones con la primera la luz se enciende con una luminosidad baja, con la segunda pulsación la luz aumentaría la intensidad, con la tercera pulsación la luz llegaría a su máximo nivel de luz, y la cuarta será utilizada para apagar la luz. También se realizo durante esta semana el control del servo que permitirá la apertura y cierre de la puerta de la entrada de la casa. Mediante un servo conectada a la puerta.

La quinta y la sexta semana fueron dedicadas a realizar el montaje del control de apertura de la puerta corredera de la entrada, tanto código, circuito y montaje sobre la maqueta. También realice durante esta semana el control de la ventana del salón y el comienzo del montaje del cableado para cada elemento de los circuitos de nuestra maqueta.

Séptima y octava semana fue dedicada a la documentación del proyecto y terminación del montaje de toda la instalación en la maqueta, comprobación del correcto funcionamiento del código.

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 7 de 51
		Firma:

DESCRIPCION DEL PROYECTO

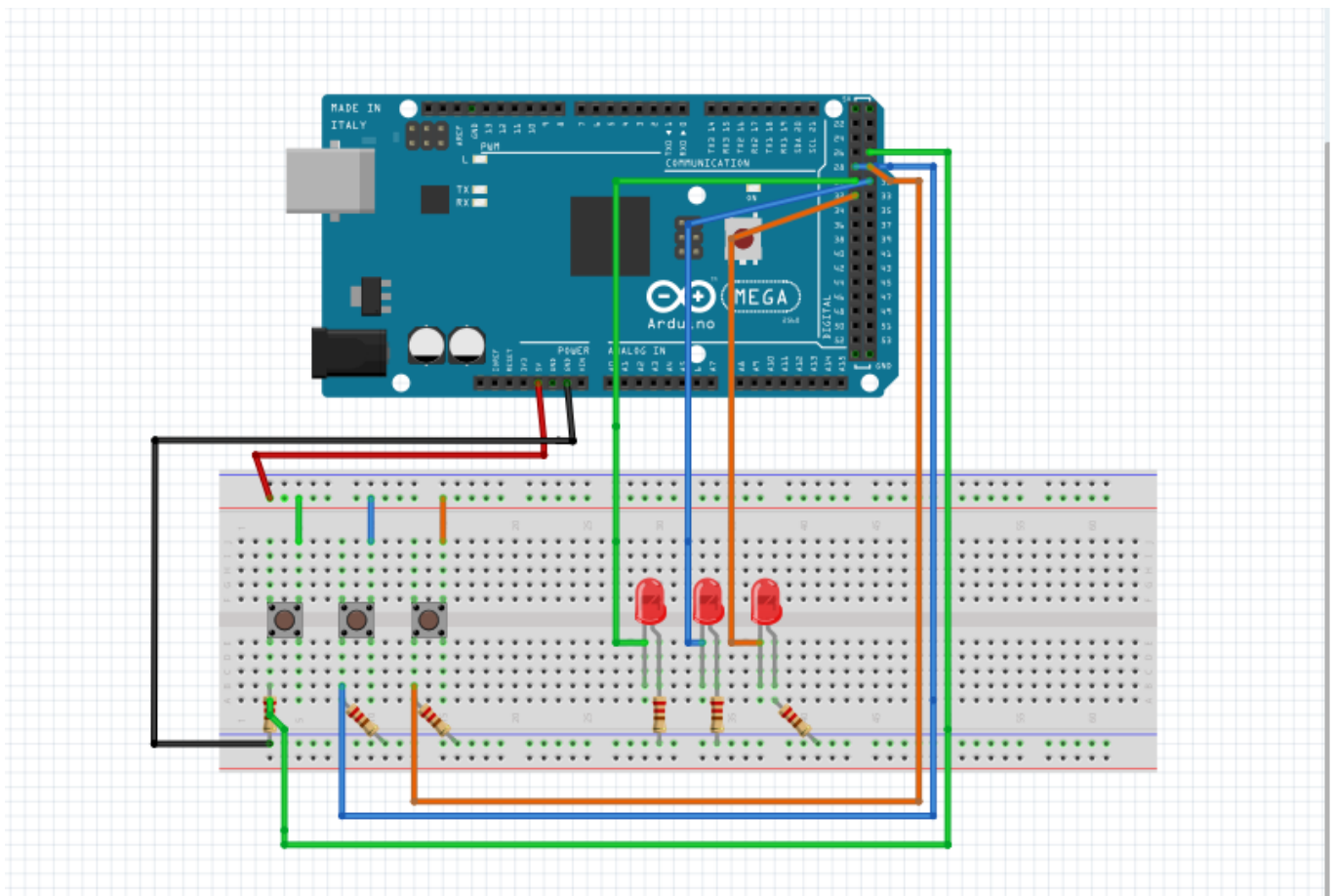
Para la realización de este proyecto es básico disponer de la placa Arduino Mega 2560, un ordenador para introducir el código. Y una placa placa board para todas las conexiones

1.- Montaje del control de las tres luces de las habitaciones

Para el montaje de las tres luces de las habitaciones vamos a necesitar.

- 1.-Tres led's
- 2.-tres pulsadores
- 3.-cableado
- 4.- 3 resistencias de 10k Ω y 3 resistencias de 330 Ω

Lo primero que hay que realizar es el montaje del circuito



Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 8 de 51
		Firma:

Para el montaje del circuito conectaremos el positivo y el GND de la placa Arduino a cada extremo de la placa de conexiones ,conectaremos un cable desde el positivo hacia una de las patillas de el pulsador cogeremos la salida 27 de la placa y la conectaremos a la patilla del otro extremo del pulsador en esta patilla también ira una resistencia de 10kΩ que su otro extremo ira conectado a el GND de la placa, haremos el mismo proceso con las salidas 28 y 29 por ultimo cogeremos las salidas 30, 31 y 32 a cada una de ellas le conectaremos el ánodo de las luces led's (la patilla larga) y al extremo del cátodo(la patilla más corta) conectaremos una resistencia de 330Ω con su otro extremo conectado a el GND de la placa.

Una vez realizado el montaje del circuito procederemos a realizar el código para su funcionamiento.

Utilizamos int para renombrar las salidas de la placa Arduino y pasen a llamarse como yo quiero, también se puede utilizar el int cuando queremos guardar las lecturas que hace Arduino de los botones en su memoria, para ello ponemos el int el nombre que queramos y 0.

```
//control de las luces
int LED1 = 30; //LED habitacion 1
int LED2 = 31; //LED habitacion 2
int LED3 = 32; //LED habitacion 3
int pulsador1 = 27; // primer pulsador
int pulsador2 = 28; // segundo pulsador
int pulsador3 = 29; // tercer pulsador

int estado1 = 0;      // Variable que indica el estado del pulsador
int anterior1 = 0;    // Variable que indica el estado anterior del pulsador
int encender1 = 0;    // Variable para controlar el encendido del led
int estado2 = 0;      // Variable que indica el estado del pulsador
int anterior2 = 0;    // Variable que indica el estado anterior del pulsador
int encender2 = 0;    // Variable para controlar el encendido del led
int estado3 = 0;      // variable que me indica el estado en que esta el pulsador
int anterior3 = 0;    //variable que me indica el estado en que estaba el led.
int encender3 = 0;    // variable para encender el led
```


Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 9 de 51
		Firma:

Ahora configuraremos las E/S de la placa para colocarlas como entrada o como salida para ellos utilizaremos el void setup que es una función fundamental de Arduino aquí se pondrá todo lo que necesitaremos que se ejecute solo una vez cuando se encienda la placa.

Por último procederemos a realizar el código que nos permitirá el correcto funcionamiento de nuestro circuito. Para eso utilizaremos la otra función básica que es el void loop donde pondremos las cosas que queremos que se repitan varias veces según que determinada acción pase. Para crear una salida deberemos poner la función pinMode (el nombre de la E/S, OUTPUT) ; y para crear una entrada pinMode (el nombre de la E/S, INPUT);

```
void setup () {
//control de la iluminacion
  pinMode (LED1, OUTPUT);      //configurado como salida
  pinMode (LED2, OUTPUT);      //configurado como salida
  pinMode (LED3, OUTPUT);      //El led configurado como salida
  pinMode (pulsador1, INPUT);   //configurado de entrada
  pinMode (pulsador2, INPUT);   //configurado de entrada
  pinMode (pulsador3, INPUT);   //pulsador configurado como entrada
}
```

La función digital Read se utiliza para leer el estado de alguna de las salida o las entras de la placa Arduino.

El if es una variable que nos indica que si se cumple la función que tiene se va a realizar determinada acción que viene escrita abajo.

El delay que hay puesto es una espera llamada anti rebote utilizado ya que Arduino pregunta por el estado de la entrada millones de veces por segundos y nuestro botón está formado por dos placas que al pulsarlas rebotan y se conectan y desconectan varias veces para que Arduino no lea las siguientes se le da una espera de no leer el estado durante 35 microsegundos.

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 10 de 51
		Firma:

```

void loop ()
//control de las luces
{
    estado1 = digitalRead(pulsador1); // guardamos el estado actual del pulsador
    if (estado1 && anterior1 == LOW){ //si se deja de accionar el pulsador
        encender1 = 1 - encender1;    // cambiamos el valor de encender
        delay (35);
    }
    anterior1 = estado1;                // lo almacenamos como estado anterior

    if (encender1) {                    // si encender1 es alto, encendemos
        digitalWrite(LED1, HIGH);
    }
    else {
        digitalWrite(LED1, LOW);       // en caso contrario apagamos
    }

    estado2 = digitalRead(pulsador2); // comprobamos el estado pulsador
    if (estado2 && anterior2 == LOW){ //si no se acciona el pulsador
        encender2 = 1 - encender2;    //cambiamos el valor de encender
        delay (35);
    }

    anterior2 = estado2;                // guardamos el estado del pulsador

    if (encender2) { //
        digitalWrite(LED2, HIGH); //encendemos
    }
    else {
        digitalWrite(LED2, LOW); //apagamos
    }
    estado3 = digitalRead(pulsador3); // para comprobar el estado pulsador
    if (estado3 && anterior3 == LOW){ // para cuando no pulsamos el pulsador
        encender3 = 1 - encender3;    //cambiamos el valor de encender
        delay(35);
    }

    anterior3 = estado3;                //para guardar el estado del pulsador

    if (encender3) { //
        digitalWrite(LED3, HIGH); //para encender
    }
    else{ // lo contrario
        digitalWrite(LED3, LOW);    //para apagar
    }
}

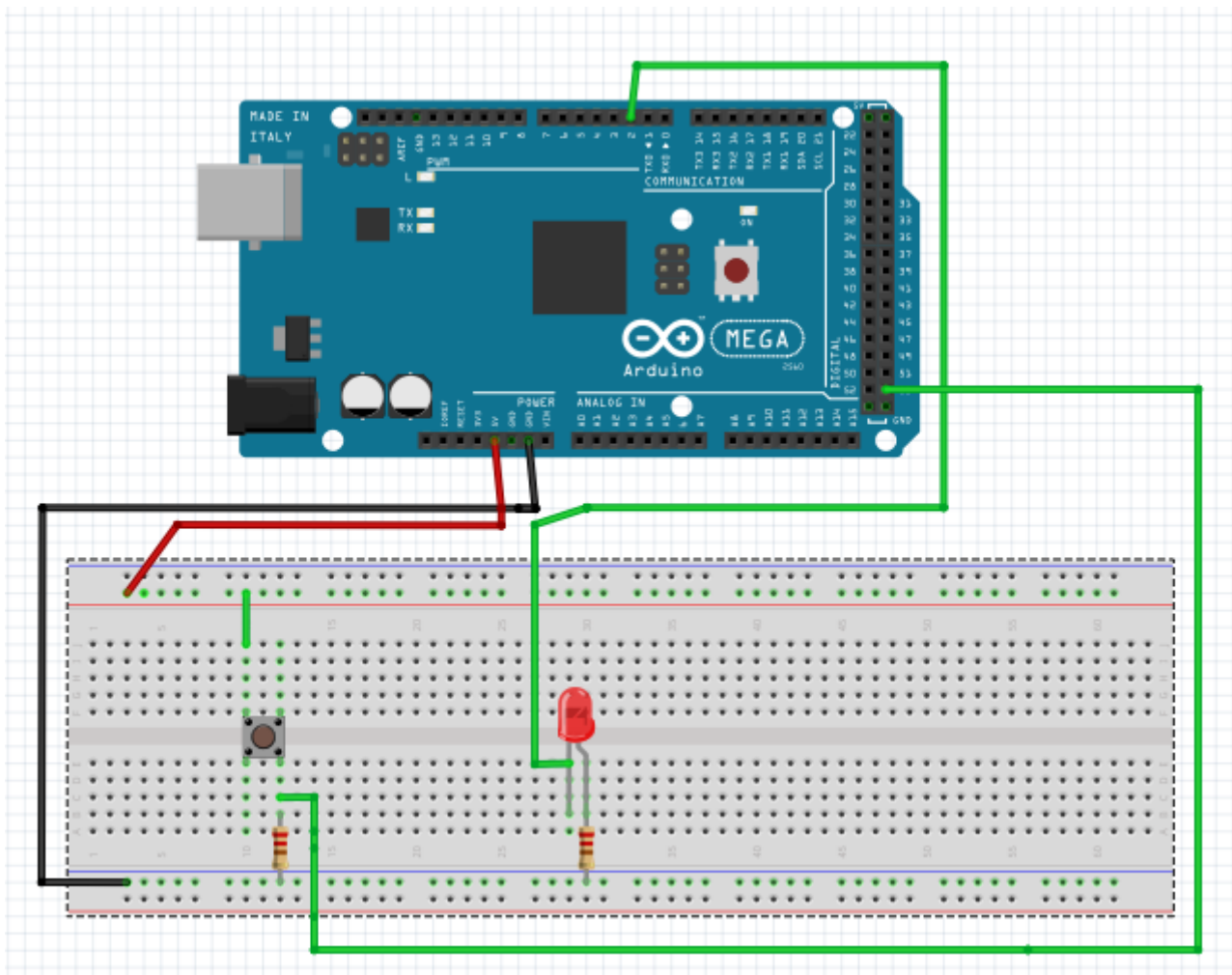
```

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 11 de 51
		Firma:

2.- montaje del control de la iluminación de la luz del salón

Para el montaje del control de la luminosidad de un led necesitaremos:

- 1.- Led
- 2.- Pulsador
- 3.- Una resistencia de 330Ω y otra de 10KΩ



Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 12 de 51
		Firma:

Para la conexión del control de la luminosidad de un punto de luz deberemos conectar el positivo y el GND de la placa Arduino a cada extremo de la placa de conexiones.

Conectaremos un extremo del pulsador a el positivo de la placa, después la salida 53 de la placa la conectaremos a el extremo opuesto del botón donde también ira conectada una resistencia de 10kΩ con su otro extremo conectado a la placa.

Por último conectaremos la salida 3 a el ánodo del led (la patilla mas larga) a el cátodo del diodo (la patilla mas corta) conectaremos una resistencia de 330Ω.

Una vez realizado el montaje del circuito procederemos a montar el código.

Comenzaremos como en el anterior dándole un nombre a nuestras variables y a los elementos utilizados mediante el const int una constante que dará para nuestra salida o entrada otro nombre durante todo el código.

```
//control de intensidad
const int LED4 = 3;           // pin del led
const int regulador = 53;     // pin del boton
int anterior4 = 0;            // variable que me dice el estado anterior de el boton
int estado4 = 0;              // variable que me indica el estado del pulasdor.
int brillo = 0;               // para almacenar la variable de encender
int intensidad = 0 ;          //para almacenar la variable

// -----
```

Ahora como siempre empezaremos a determinar las entradas y las salidas que vamos a tener mediante el pinMode. El serial begin nos indicara la velocidad de transferencia de datos entre el botón y la placa se recomienda el valor de 9600 baudios. Para crear una salida deberemos poner la función pinMode (el nombre de la E/S, OUTPUT) ; y para crear una entrada pinMode (el nombre de la E/S, INPUT);

```
void setup () {
//control de la intensidad de la luz
pinMode (LED4, OUTPUT);      //led configurado como salida
pinMode (regulador, INPUT);  //boton configurado como entrada
Serial.begin(9600);          // nos indica la velocidad con la que transfiere los datos la velocidad en baudios.
```

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 13 de 51
		Firma:

Por ultimo procederemos a crear el codigo que nos realizara la funcion en el aparatado de void loop lo primero que haremos sera leer el estado del boton mediante el digitalWrite para saber si esta en estado alto o bajo, una vez esto mediante la variable if y si se cumple la funcion de su interior se producira una accion o no, ahora deberemos utilizar la funcion map para poder controlar la intensidad y el estado de la intensidad del led lo guardaremos mediante el Serial.print en la memoria de arduino. Por ultimo saber que la funcion analogWrite nos permite que el led tenga varias intensidades.

```

void loop ()
//control de la intensidad
{
    estado4 = digitalRead(regulador); // para comprobar el estado del pulsador
    if(estado4 && !anterior4)         // para cuando no tocamos el pulsador
    {
        brillo++;                     // esto es para sumar el brillo
        if(brillo == 4)               // y esta es la cuenta
            brillo = 0;               //con esto reseteamos a la 4 pulsacion.
        Serial.print(brillo);         // aqui guardamos el brillo
    }
    intensidad = map(brillo, 0, 3, 0, 255); // funcion que controla la intensidad, y la reparticion de estas
    if(estado4 && !anterior4)           //
    {
        Serial.print(" - ");
        Serial.println(intensidad);
        delay(35);                     //un retardo para evitar el efecto rebote.
    }
    anterior4 = estado4;
    analogWrite(LED4, intensidad);     //Para poder controlar la intensidad
}

```

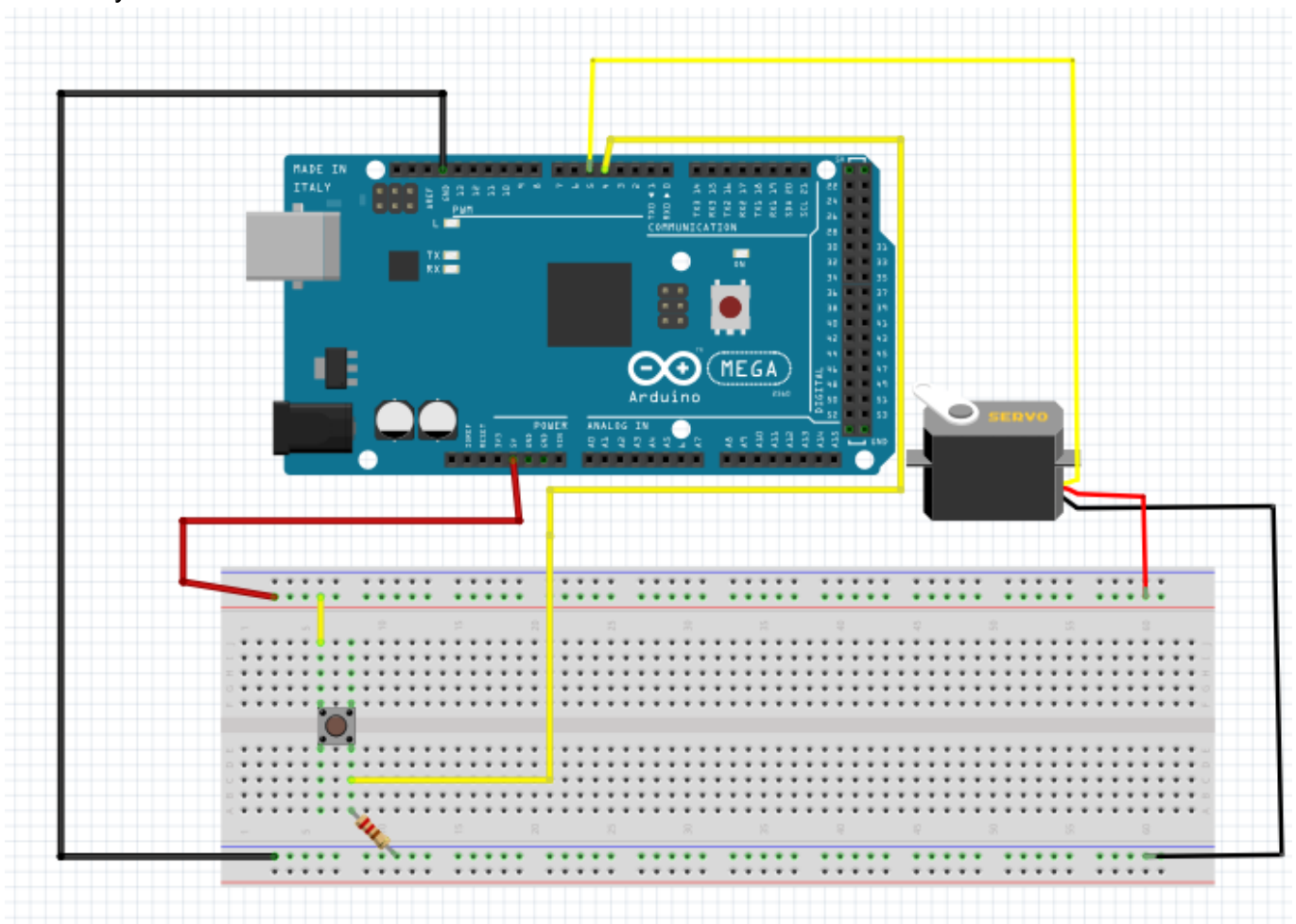
Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 14 de 51
		Firma:

3.- control mediante un servo de la puerta de entrada a la casa.

Lo primero que vamos a realizar será el circuito, para ello necesitamos

- 1.- Un servo
- 2.- cable para realizar las conexiones
- 3.- Un pulsador
- 4.- una resistencia de 10kΩ

Para realizar este montaje conectaremos el positivo y el GND de nuestra placa Arduino a la placa de conexiones, después cogeremos un positivo de la placa y lo conectaremos a una patilla del pulsador, en la patilla del otro extremo conectaremos la salida numero 4 de Arduino a la vez que tendrá una resistencia de 10kΩ con su otro extremo conectado a tierra una vez colocado el pulsador pondremos el servo para ello conectaremos el positivo y el negativo del servo a la placa de conexiones y el cable de datos del servo lo conectaremos a la salida 5 de Arduino y con esto tendríamos realizado nuestro circuito.



Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 15 de 51
		Firma:

Ahora procederemos a realizar el código lo primero para poder controlar un servo con Arduino será abrir una carpeta de la memoria de Arduino llamada servo.h para ello deberemos poner “#include <Servo.h>” con eso será suficiente después daremos nombres a nuestras salidas con la función const int para dar nombre al servo que hemos conectado utilizaremos la variable Servo y el nombre que queramos y también guardaremos el estado del botón y la posición del servo en la memoria de Arduino.

```
// control servo puerta casa
#include <Servo.h>
const int servoPin = 5;      // constante que indica el pin al que esta conectado el servo
const int buttonPin = 4;     // pin al que esta conectado el boton
int buttonState = 0;         // variable que me indica el estado del boton
int directionState = 0;      // variable que le indica la direccion a la que tiene que ir
Servo myservo;               // nombre que se le a dado a el servo
int pos = 0;                 // variable que guarda la posicion del servo
```

El apartado void setup de nuestro código es muy sencillo solo hara falta decir el pin que sera para mover el servo mediante la funcion .attach para utilizar correctamente deberas poner “el nombre de tu servo”.attach (pin que has elegido para el servo); y por ultimo el pin al que ira conectado el boton para hacer funcionar el servo.

```
void setup () {
// servo puerta casa
myservo.attach(5);           // pin al que esta conectado el servo y por donde vana salir las ordenes
pinMode(buttonPin, INPUT);   //pin del boton colocado como salida
```


Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 16 de 51
		Firma:

Por último crearemos el void loop para que nuestro servo realice los movimientos que queremos, Lo principal y lo que hacemos siempre es preguntar el estado del botón mediante la función DigitalRead una vez hecho esto con la función if haremos que si pulsamos el botón el servo se moverá desde la posición 0 hacia los 90° y después se vuelve a pulsar el botón el servo ira esta vez de 90 a 0° una vez realizado esto tendremos el movimiento de la puerta completado

```

void loop ()
//servo puerta casa
{ buttonState = digitalRead(buttonPin); //primero leemos el estado del boton.
  if (directionState == 0){ // le decimos la direccion a la que tiene que ir
    if (buttonState == HIGH) { // y si esta todo correcto enviamos un HIGH para mover el servo
      directionState = 1;

      for(pos = 0; pos < 90; pos=pos+1) // las direcciones a las que se mueve el servo
      {
        myservo.write(pos); // una vez que pulsamos el boton busca la direccion
        delay(35); // tiempo de espera anti rebote
      }
    }

  } else if (directionState == 1) { // y este es el movimiento contrario

    if (buttonState == HIGH) { // si el boton esta pulsado
      directionState = 0; // y la posicion del servo es 0

      for(pos = 90; pos>=0; pos=pos-1) // aqui tiene la cuenta para cambiar de direccion
      {
        myservo.write(pos); // escribe la direccion y se mueve hacia ella
        delay(35); // tiempo de espera anti rebote
      }
    }
  }
}

```

Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 17 de 51
		Firma:

4.- Control de la apertura de la puerta corredera de la puerta

Para realizar el circuito de el control de la puerta necesitaremos los siguientes materiales.

- 1.- cable telefónico
- 2.- 2 relés de 6V
- 3.- un motor de 5V
- 4.- 2 pulsadores
- 5.- 2 resistencias de 10kΩ

Para empezar cogeremos el positivo y el GND de Arduino y lo conectaremos a la placa de conexiones, una vez realizado esto llevaremos un positivo a una de las patillas de cada pulsador en las patillas del otro extremo conectaremos la salida 36 y 37 una para cada botón

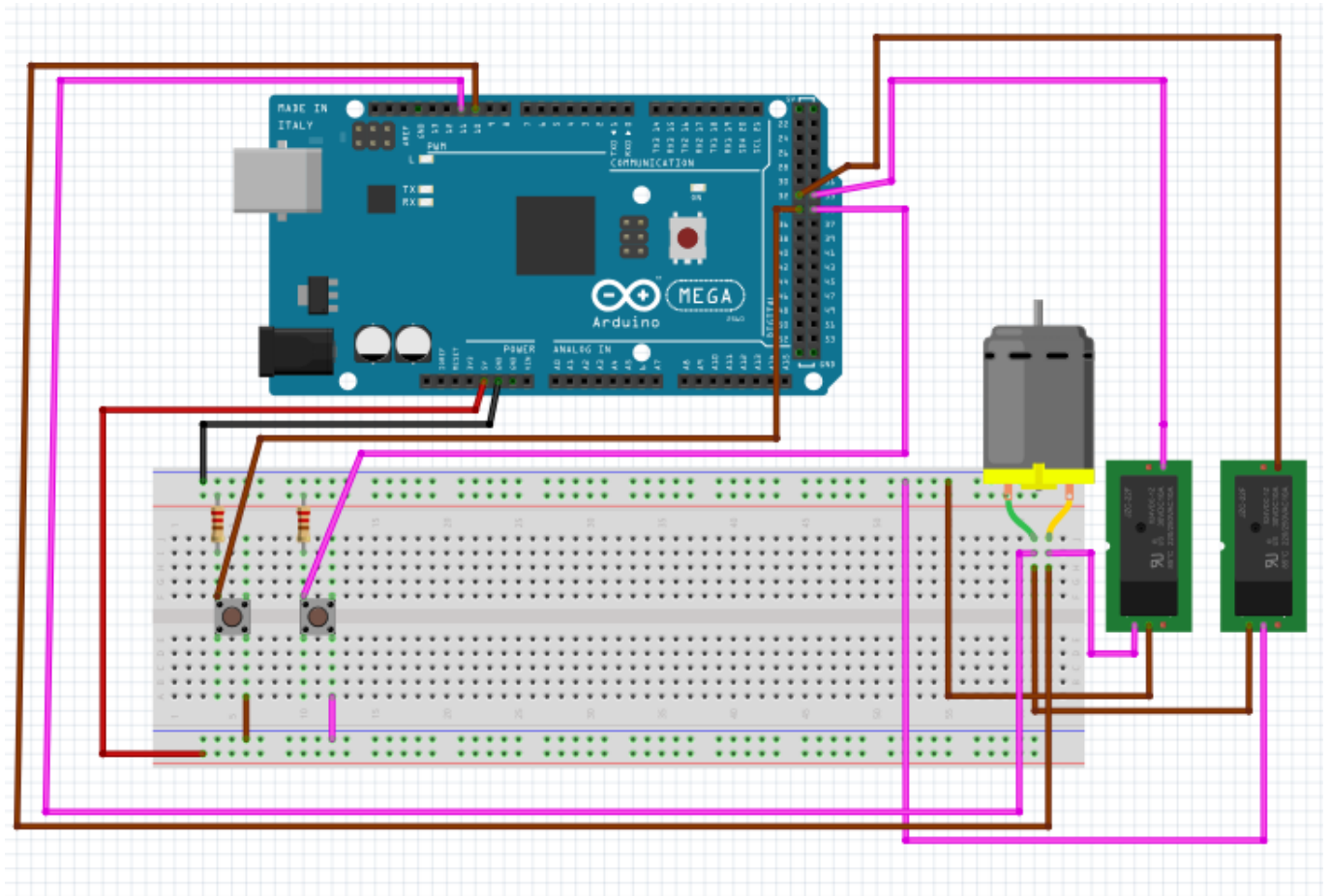
También a esta ira conectado una resistencia de 10kΩ que su extremo unido a el GND de la placa cada botón con una resistencia.

Las dos patillas del motor las conectaremos a dos líneas libres la placa de conexiones.

Una vez realizado esto procederemos a conectar los relés una patilla de cada bobina de los relés irán conectadas a el negativo de la placa la otra patilla ira conectada a la salida 35 para un relé y las salida 34 para el otro relé de tal forma que la bobina de los rele será conectado lo siguiente será conectar la patilla común de los relés a el negativo de la placa y la patilla normalmente abierta una a cada patilla del motor.

La salida 10 de Arduino ira conectada a una de las patillas del motor al igual que la salida 11 de Arduino ira conectada a la patilla contraria de la salida 10 de el motor.

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 18 de 51
		Firma:



Una vez realizado el circuito solo nos quedara hacer el código que nos permitirá su funcionamiento para ello empezaremos como siempre comenzaremos dando un nombre a nuestras salidas así como guardar los datos necesarios en la memoria de arduino

```
//puerta corredera de la entrada
int rele1 = 35 ;
int rele2 = 13;
int adelante = 10;
int atras = 11;
int inputPin1 = 36; // pulsador 1
int inputPin2 = 37; // pulsador 2
int p1;
int p2;
```

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 19 de 51
		Firma:

Una vez terminado de dar los nombres y de guardar datos en la memoria procederemos a decir las entradas y salidas que vamos a tener para crear una salida deberemos poner la función pinMode (el nombre de la E/S, OUTPUT) ; y para crear una entrada pinMode (el nombre de la E/S, INPUT);

```
void setup () {
//puerta corredera de la entrada
pinMode(rele1, OUTPUT);      //colocado para que cierre el rele y active el circuito del motor
pinMode(rele2, OUTPUT);      //colocado para que cierre el rele y active el circuito del motor
pinMode(inputPin1, INPUT);    //es el pulsador que activa la puerta
pinMode(inputPin2, INPUT);    // es el final de carrera que detiene la puerta y la cierra
pinMode(adelante, OUTPUT);    // es la salida que me da el positivo para accionar el motor
pinMode(atras, OUTPUT);       // es la salida que me activa activa el motor en la otra direccion
}
```

Ahora procederemos a realizar el montaje del código que nos realizara el movimiento de nuestra puerta realizaremos el código de tal manera que al pulsar el botón Arduino leerá un HIGH en el primer pulsador y me activara la salida del relé 1 que llevara el negativo a el motor a la misma vez me dará el positivo a el motor por medio de la salida adelante.la puerta comenzara a abrirse hasta llegar a un final de carrera que es un pulsador que nos dará un nivel bajo en las salidas del relé 1 y adelante se esperara 5 segundos hasta que accionara la salida del relé 2 que nos dará el negativo a el motor y la salida atrás que nos dará el positivo a el motor para conseguir que se cierre la puerta a los 5 segundos conseguirá cerrarse y las salidas relé 2 y atrás quedaran en un nivel bajo.

```
void loop ()
//puerta corredera de la entrada
{
p1=digitalRead(inputPin1);      //lo que nos dice que p1 es igual a una lectura de la salida inputPin1
p2=digitalRead(inputPin2);      //lo que nos dice que p2es igual a una lectura de la salida inputPin2
if (p1 == HIGH){                // si p1 es igual a alto
digitalWrite(rele1, HIGH);      // la salida rele 1 sera alto
digitalWrite (adelante, HIGH);  // la salida adelante sera alto
}
else if (p2 == HIGH)            // y si p2 es alto
{
digitalWrite(rele1, LOW );      // la salida rele1 sera baja
digitalWrite (adelante, LOW );  // la salida adelante sera baja

delay (5000);                   //se esperara un tiempo de 5 segundos
digitalWrite(rele2,HIGH);       // y la salida rele2 sera alto
digitalWrite(atras, HIGH);      // y la salida ataras sera alta
delay (5000) ;                  // una espera de 5 segundos funcionando
digitalWrite(rele2, LOW);       // y la salida rele 2 sera baja
digitalWrite(atras, LOW);       // y la salida atras sera baja
}
}
```

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 20 de 51
		Firma:

5.- control de la apertura del salón mediante servos.

Lo primero que debemos hacer es el circuito que tomara el control de nuestro código para ello necesitaremos

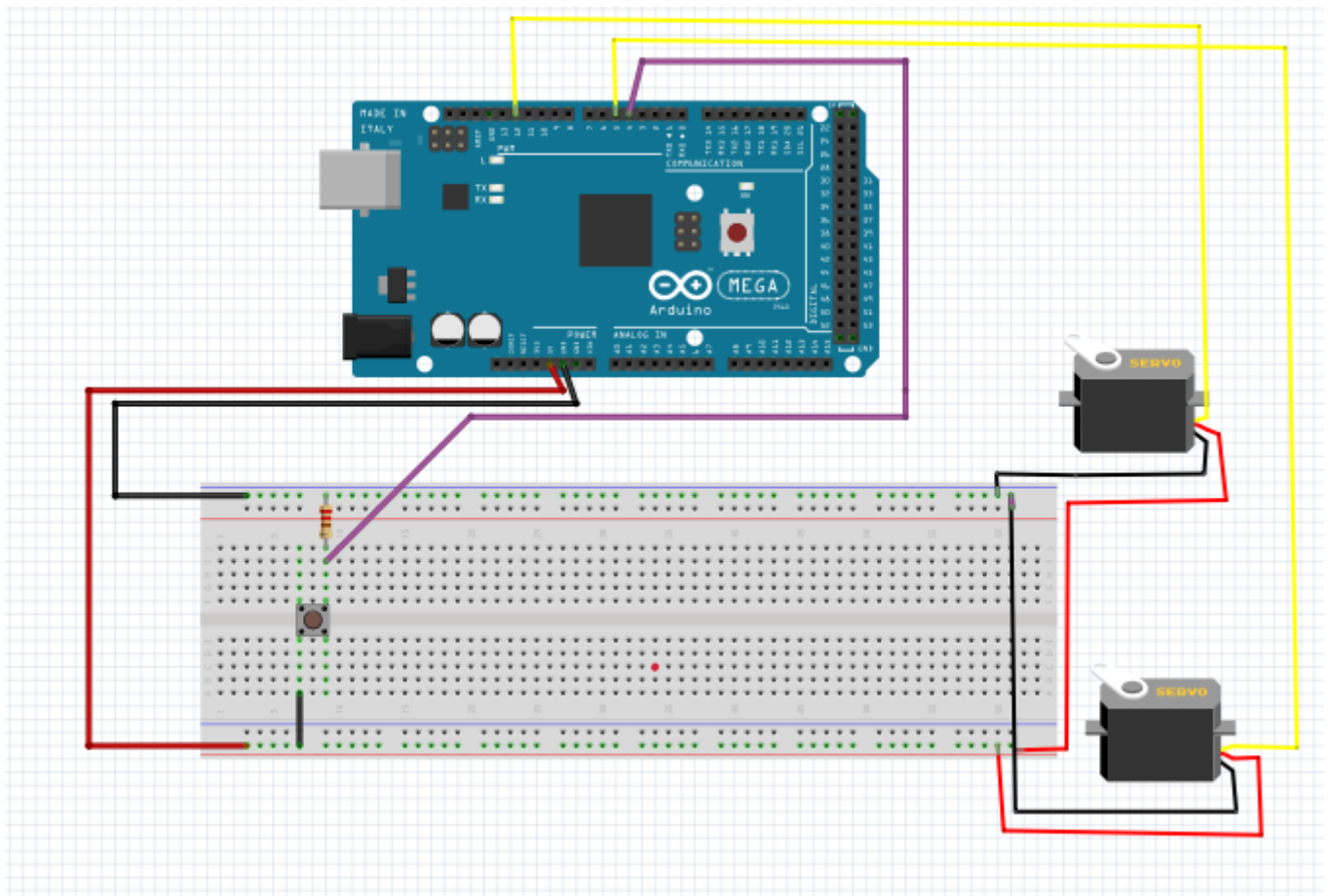
- 1.-2 servos
- 2.- un pulsador
- 3.- una resistencia de 10k Ω
- 4.- cable telefónico

Para realizar este circuito conectaremos el positivo y el GND de la placa Arduino a la placa de conexiones llevaremos un positivo de la placa hacia una patilla del pulsador en la patilla del otro extremo del pulsador conectaremos la entrada de Arduino numero 4 y una resistencia con un extremo colocado al mismo de la patilla de entrada y el otro extremo de la resistencia directo a el negativo de la placa de conexiones de esta manera el botón quedara instalado completamente.

Para los servos conectaremos su negativo a la placa de conexiones y haremos lo mismo para el positivo de los servos los conectaremos a la placa de conexiones con lo que quedaran alimentados.

Nos quedaría por conectar su entrada de datos que la de un servo ira colocada a la salida numero 5 de Arduino y la entrada de datos del segundo servo ira conectada a la entrada de Arduino numero 12. Con estas conexiones quedaría nuestro circuito completamente conectado.

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 21 de 51
		Firma:



Una vez realizado nuestro circuito procederemos al montaje del código. En este código e optado por repetir el mismo código con unos pocos cambios que más adelante que vemos por el motivo de un mejor funcionamiento de los servos ya que con un código puede inducir a errores al tener dos servos funcionando con un solo código guardando la posición en la que están sobre el mismo apartado.

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 22 de 51
		Firma:

Los primeros pasos para empezar a realizar el código son abrir dos librerías para servos una por cada servo y dar nombres a las salidas los servos no pueden tener el mismo nombre, y para que los dos funcionen al mismo tiempo utilizaremos el mismo botón o la misma entrada de información para crear el movimiento en nuestros servos.

```
//control de servo ventanas
#include <Servo.h>

const int servoPin1 = 5;
const int buttonPin1 = 4;

int buttonState1 = 0;
int directionState1= 0;

Servo myservol;

int pos1 = 0;
//control de servo ventanas2
#include <Servo.h>
const int servoPin2 = 12;    // constante que indica el pin al que esta conectado el servo
const int buttonPin2 = 4;    // pin al que esta conectado el boton
int buttonState2 = 0;        // variable que me indica el estado del boton
int directionState2 = 0;     // variable que le indica la direccion a la que tiene que ir
Servo myservo2;              // nombre que se le a dado a el servo
int pos2 = 0;                // variable que guarda la posicion del servo
```

Una vez terminada esta parte del código empezaremos a determinar las entradas y salidas que tendrá nuestro montaje, tendremos que poner la Salida de los dos servos mediante la función. attach, deben tener un orden de “el nombre de tu servo”.attach (pin que has elegido para el servo); para que funcione correctamente despues pondremos en los dos codigos el pin de entrada para que empiecen su movimiento que sera mediante el pinMode elegiremos el pin al que tenemos conectado el boton.en nuestro caso el 4.

```
void setup () {
// servos ventana
myservol.attach(5);
pinMode(buttonPin1, INPUT);
//servo ventana
myservo2.attach(12);           // pin al que esta conectado el servo y por donde vana salir las ordenes
pinMode(buttonPin2, INPUT);    //pin del boton colocado como salida
}
```


Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 23 de 51
		Firma:

Para realizar el código del primer servo habrá que seguir los siguientes pasos para que empiece a moverse nuestro servo lo primero que hace falta es preguntar el estado de nuestro botón y la posición en la que esta mientras no esté pulsado este no realizara ningún movimiento pero si esta pulsado y la posición del servo es 0 esta comprobación la hace mediante el cable conectado a él servo , el servo realizara una función que será moverse desde su posición 0 hasta la posición 1 a esta posiciones se les da valor mediante la función for. Una vez sacado estos datos la placa se dispone a atacar el pin del servo que antes hemos determinado.

```
void loop ()
//servo ventana salon
{ buttonStatel = digitalRead(buttonPin1);    //primero leemos el estado del boton.
  if (directionStatel == 0){                  // pregunta si la posicion del servo es 0
    if (buttonStatel == HIGH) {              // y si esta el boton en en estado alto
      directionStatel = 1;

      for(pos1 = 30; pos1 < 130; pos1=pos1+1)    // la direccion desde la que parte y hasta la que se mueve el servo
      {
        myservol.write(pos1); // una vez que pulsamos el boton busca la direccion del servo
        delay(35);            // tiempo de espera anti rebote
      }
    }

  } else if (directionStatel == 1) {           // y si el servo esta en la posicion contraria

    if (buttonStatel == HIGH) {               //y si el boton esta pulsado
      directionStatel = 0;                    // y la posicion del servo es 0
      for(pos1 = 130; pos1>=30; pos1=pos1-1)  // aqui tiene la cuenta para cambiar de direccion
      {
        myservol.write(pos1);                // una vez pulsado el boton por segunda vez con la funcion pasada hace que vuelva a la posicion 0 o inicial
        delay(35);                            // tiempo de espera anti rebote
      }
    }
  }
}
```

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 24 de 51
		Firma:

Para crear el código del segundo servo se deben realizar los mismos pasos que en el primero. Para mover nuestro servo lo primero que hace falta es preguntar el estado de nuestro botón y la posición en la que esta mientras no esté pulsado este no realizara ningún movimiento pero si esta pulsado y la posición del servo es 0 (esta comprobación la hace mediante el cable conectado a él servo), el servo realizara una función que será la de moverse desde su posición 0 hasta la posición 1 a estas posiciones se les da valor mediante la función for. Una vez sacado estos datos la placa se dispone a atacar el pin del servo que antes hemos determinado. La diferencia que tiene con el del principio está en el nombre que se le da a los valores que queremos que sean guardados, el pin a el que está conectado el servo así como su nombre, ya que la función que tiene que realizar es la misma.

```
void loop ()
//servo ventana
{ buttonState2 = digitalRead(buttonPin2);    //primero leemos el estado del boton.

  if (directionState2 == 0){                  // pregunta si la posicion del servo es 0

    if (buttonState2 == HIGH) {               // y si esta el boton en en estado alto
      directionState2 = 1;
      for(pos2 = 0; pos2 < 90; pos2 = pos2 +1)    // la direccion desde la que parte y hasta la que se mueve el servo
      {
        myservo2.write(pos2);                  // una vez que pulsamos el boton busca la direccion del servo
        delay(35);                             // tiempo de espera anti rebote
      }
    }

  } else if (directionState2 == 1) {           // y si el servo esta en la posicion contraria

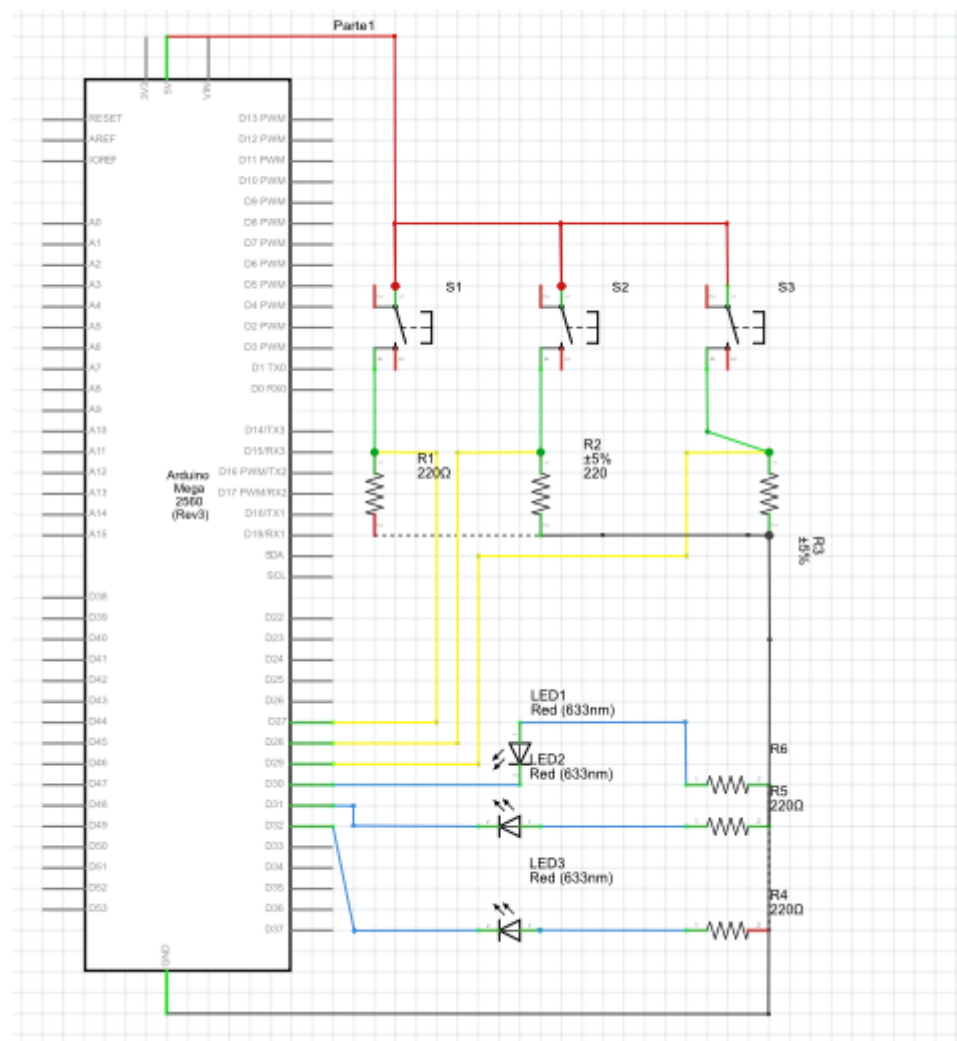
    if (buttonState2 == HIGH) {               //y si el boton esta pulsado
      directionState2 = 0;                    // y la posicion del servo es 0
      for(pos2 = 90; pos2 >=0; pos2 = pos2 -1)    // aqui tiene la cuenta para cambiar de direccion
      {
        myservo2.write(pos);                  // una vez pulsado el boton por segunda vez con la funcion pasada hace que vuelva a la posicion 0 o inicial
        delay(35);                             // tiempo de espera anti rebote
      }
    }
  }
}
```

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 25 de 51
		Firma:

ESQUEMAS ELECTRICOS DE LOS MONTAJES.

1.- Montaje del control de las tres luces de las habitaciones

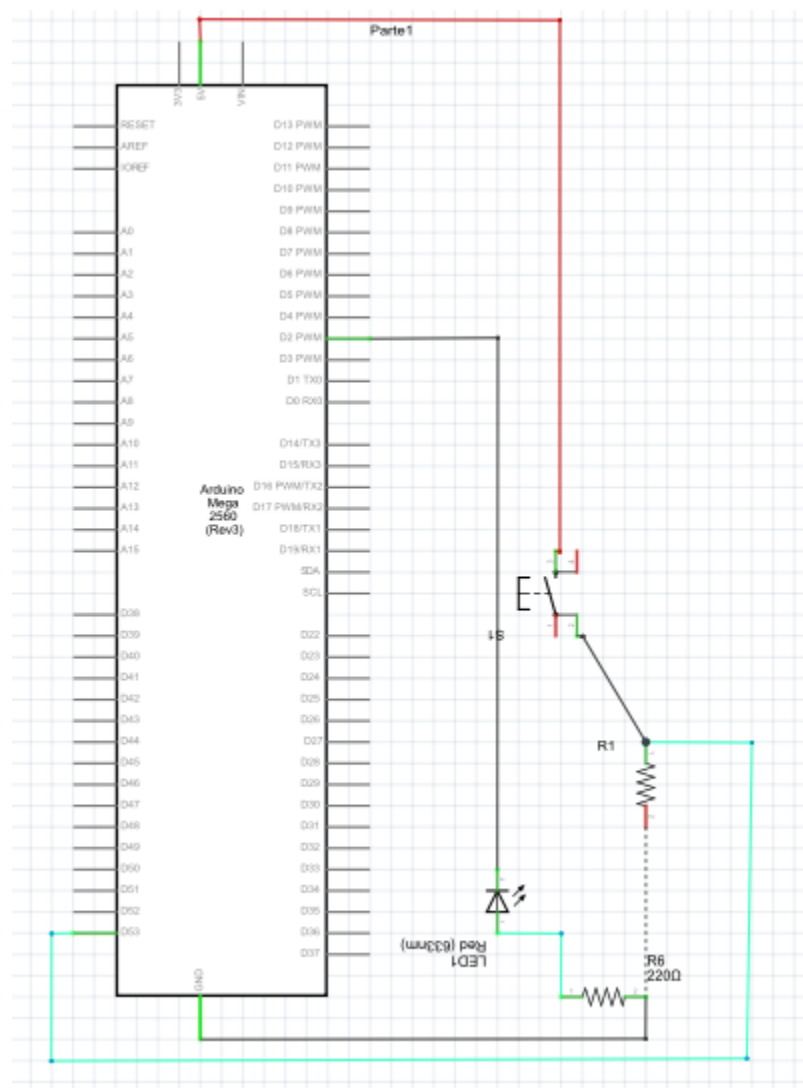
El esquema eléctrico del circuito correspondiente al montaje del control de las tres luces correspondientes a las habitaciones de mi maqueta es el siguiente.



Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 26 de 51
		Firma:

2.- montaje del control de la iluminación de la luz del salón

El esquema eléctrico correspondiente al montaje del circuito de la regulación de la luminosidad de la luz del salón es el siguiente



Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 27 de 51
		Firma:

3.- control mediante un servo de la puerta de entrada a la casa.

El esquema eléctrico correspondiente al montaje del servo que controlara la puerta de entrada a nuestra casa será el siguiente.

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 29 de 51
		Firma:

Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 30 de 51
		Firma:

4.- Control de la apertura de la puerta corredera de la puerta

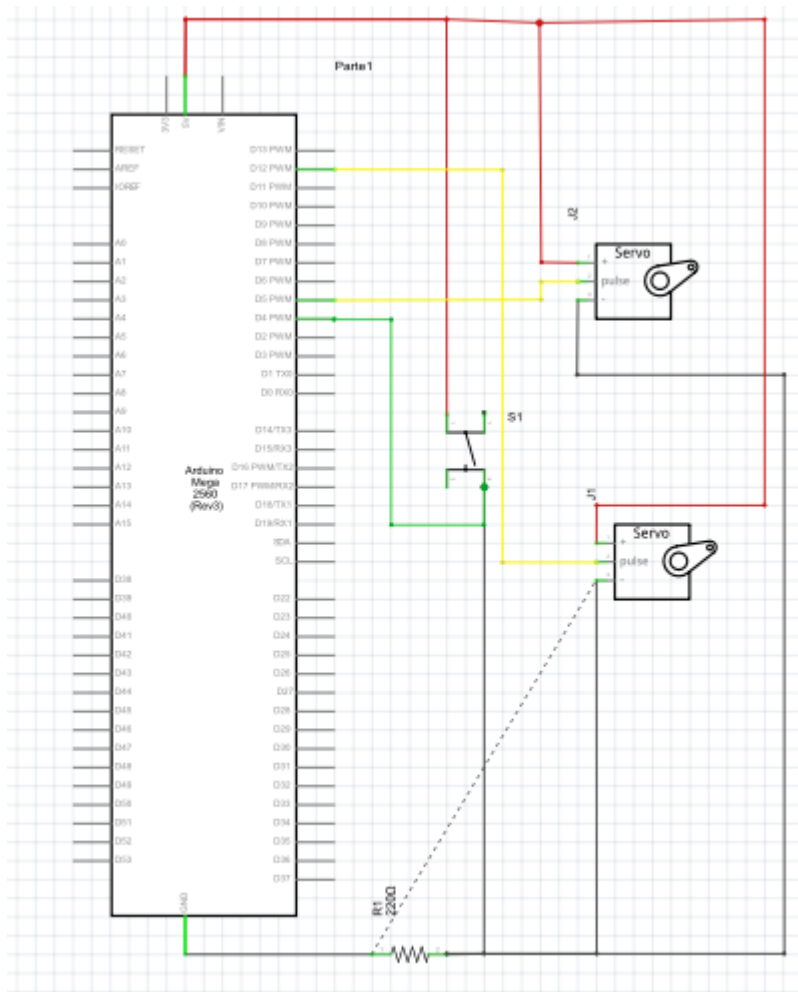
El esquema eléctrico correspondiente a el control de la apertura de la puerta corredera de la entrada será el siguiente.

Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 32 de 51
		Firma:

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 33 de 51
		Firma:

5.- control de la apertura del salón mediante servos.

El esquema eléctrico correspondiente a el control de la ventana del salón por medio de dos servos es el siguiente



Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 34 de 51
		Firma:

PLIEGO DE CONDICIONES

Para la realización de este proyecto serán necesario los siguientes materiales con las siguientes características aunque puedan tener otro formato siempre que cumplan las características dadas. El formato no es tan importante. Todo los materiales y herramientas que vienen a continuacion se podrán encontrar en la página Deal extreme

Pistola de calor para pegamento 20W (110~240V)



- Ideal para reparaciones del hogar manualidades y pasatiempos
 - Componente del calentador está hecho de resistencia de coeficiente positivo de temperatura (PTC)
 - Función de fusibles para un uso seguro
 - Toma barras de pegamento con un diámetro de 7 mm
 - Longitud del cable: 140 cm
 - El paquete incluye: 1 * Estuche de cuero, 4 palos * pegamento blanco, 2 barras de pegamento blanco *, 1 * pinzas,
 - La herramienta de nicking con 2 hojas *, 1 * 15cm regla de acero, 1 * clips de rosca
- Dimensiones: 5,2 x 4,29 en x 0,87 en (13,2 cm x 10,9 cm x 2,2 cm)
Peso: 12.7 oz (360 g)

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 35 de 51
		Firma:

Multifunción práctico Carpintería Sierra / Metal Hacksaw Set - Negro + Plata



Marca	N / A
Modelo	N / A
Cantidad	1
Color	Plata + Negro
Material	Acero inoxidable
Características	Sierra para metal, madera, hoja de sierra es ajustable 3-ángulo
Aplicación	Uso de corte
Contenido del paquete	1 x mango Hacksaw
	5 x Hoja de sierra
	1 x Madera de sierra
	1 x Ubicación de punzón
Dimensiones: 9.80 in x 4.13 in x 0.51 in (24,9 cm x 10,5 cm x 1,3 cm)	
Peso: 5,61 oz (159 g)	

Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 36 de 51
		Firma:

Cable con 40 Pin multicolor - multicolor (10 metros)



- Cable Color: Rainbow
- Material: Silicona
- Número de pines: 40
- Anchura: 6 cm
- Longitud del cable: 10 metros

Dimensiones: 39,37 x 2,36 en x 0,04 en (100,0 cm x 6,0 cm x 0,1 cm)

Peso: 28.85 oz (818 g)

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 37 de 51
		Firma:

RS-380SH eje corto Micro DC Motor de ventilador - hierro gris



Rango de voltaje: 6-9V

Voltaje clasificado: 6V

Corriente sin carga: Conecte el 6V voltaje trata 430mA

Corriente del rotor bloqueado: Conecte el 6V voltaje es aproximadamente 2

Conectar la velocidad sin carga de voltaje de 6V, aproximadamente 12000 vueltas por minuto

Longitud: 3,8 cm

Diámetro: 2,7 cm

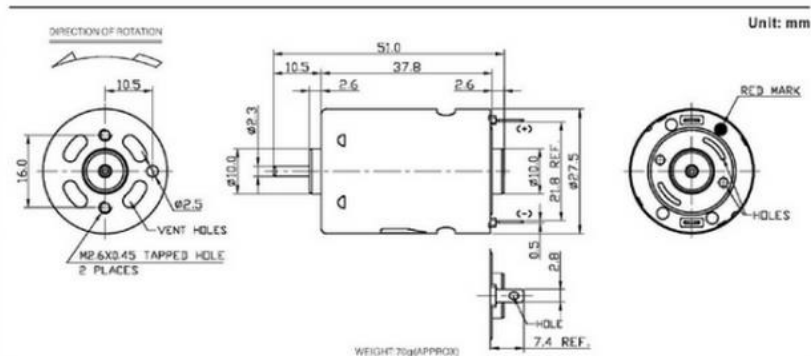
Largo del eje: 7,8 cm

Diámetro del eje: 2,3 cm

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 38 de 51
		Firma:

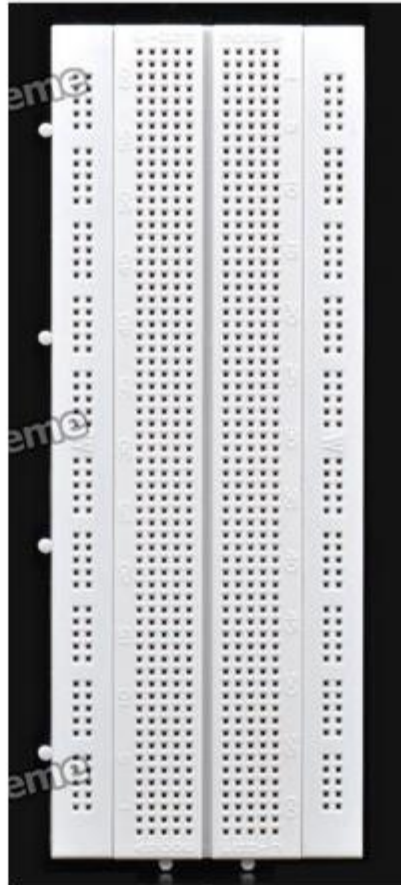
Ampliamente utilizado en equipos de automatización, accesorios autos, barriendo las máquinas, herramientas eléctricas, pequeños electrodomésticos, DC carga fan, máquinas de afeitar eléctricas, secador de pelo, masaje, secos de mano, bomba de agua, los aviones, equipos de enseñanza, los niños ' s juguetes, instrumentación y otros campos.

MODEL	VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY				STALL	
	OPERATING RANGE	NOMINAL	SPEED (r/min)	CURRENT (A)	SPEED (r/min)	CURRENT (A)	TORQUE (g.cm)	OUTPUT (W)	CURRENT (A)	TORQUE (g.cm)
RS-380SH-4045	4.5~9.0	7.2V CONSTANT	16200	0.5	14060	3.29	110	16	21.5	840
RS-380SH-12360	12~30	24V CONSTANT	6800	0.05	5640	0.24	71.2	4.1	1.2	420
RS-385SH-2270	9.0~24	12V CONSTANT	9800	0.15	8190	0.76	72.9	6.1	3.9	445
RS-385SH-10250	12~24	12V CONSTANT	2500	0.04	1890	0.13	32.5	0.63	0.4	135



Nombre: David Díaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 39 de 51
		Firma:

840 Punto de soldadura Breadboard



- Modelo: GL NO.12
- Color: Blanco
- 840 sin soldadura punto
- Permite que los componentes electrónicos para ser interconectados en un número casi infinito de formas de producir circuitos de trabajo
- Compatible con resistencias, transistores, diodos, LEDs, condensadores y otros tipos de componentes electrónicos
- Ideal para experimentar con el diseño de circuitos en los laboratorios

Dimensiones: 6,73 x 2,52 en x 0,31 en (17,1 cm x 6,4 cm x 0,8 cm)

Peso: 3,53 oz (100 g)

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 40 de 51
		Firma:

Arduino 9V 1A Adaptador de corriente - Negro (2-Flat-Pin Plug / cable 100cm)



alimentador de pared "conmutación de alta calidad AC a DC 9V 1000mA de corriente de pared fabricado específicamente para la electrónica. Estas son las fuentes de alimentación conmutadas que significa la salida se regula a 9V y la corriente de salida capaz es mucho mayor (1000mA!). Estos proyectos de energía más voluntad que no requieren más de 1000 mA de corriente.

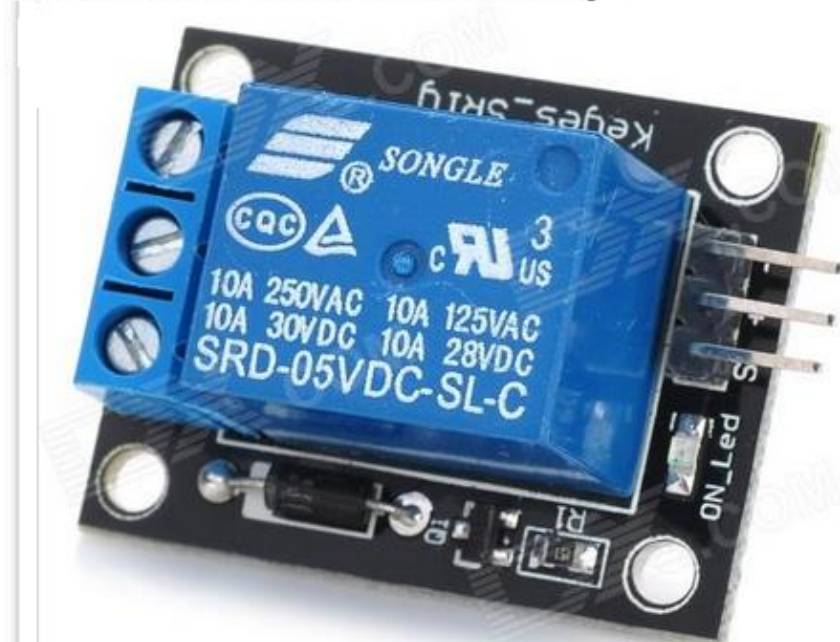
- Color: Negro
- Entrada: 100 ~ 240V, 50/60Hz
- Salida: 9V, 1A
- Funciona con la mayoría de dispositivos que utilicen un adaptador de 9 V y menos de 1A de la energía
- 2-plana patillas
- Conector de tamaño: 5,5 x 2,1 mm
- Longitud del cable: 100 cm

Dimensiones: 2,91 x 1,57 en en x 1,06 en (7,40 cm x 4,00 cm x 2,70 cm)

Peso: 2.54 oz (72 g)

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 41 de 51
		Firma:

Keyes Módulo de relé de 5V para Arduino (funciona con las Juntas de Oficiales Arduino)



El módulo de relé de 5V Arduino es adecuado para el desarrollo de SCM , control de electrodomésticos . Es con señal de control de 5V ~ 12V TTL que puede controlar la señal de DC / AC . El módulo es perfecto para la aplicación Arduino común.

- Marca: Keyes
- Color: azul + negro
- Apto para el desarrollo SCM , control de electrodomésticos
- 5V -12V señal de control TTL , puede controlar la señal DC / AC
- Puede controlar la carga 220V AC

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 42 de 51
		Firma:

Mini servo TowerPro SG90 9G con accesorios



El TowerPro SG90 servo mini es ligero, de alta calidad y la velocidad del rayo.El servo está diseñado para funcionar con casi todos los sistemas de radio de control.Es con un excelente rendimiento le lleva a otro horizonte de vuelo.El servo SG90 mini con accesorios es perfecto para R / C helicóptero, avión, coche, barco y el uso de camiones.

Dimensiones: 1,26 x 1,18 en en x 0,47 en
Peso: 0.49 oz

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 43 de 51
		Firma:

KEYES KT0053 Protoboard + Capacitores cerámicos Resistencias + + Más para Arduino - multicolor



General	
Marca	KEYES
Modelo	KT0053
Cantidad	1Piece
Color	blanco
Materiales	FR4
Especificaciones	
Manual de Inglés / Spec	No
Otras características	
Otras características	Un producto para Arduino que funciona con placas de Arduino oficiales.
Dimensiones y Peso	
Dimensiones	6.3 en x 3.54 en x en 1,57 (16 x 9 cm x 4 cm)

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 44 de 51
		Firma:

Peso	6,42 oz (182 g)
Contenido del KIT	
1 x 400-placa de conexiones 30 x cables macho macho 10 x F5 LEDs rojos 10 x F5 LEDs verdes 1 x RGB LEDs 10 x 10NF Condensadores cerámicos 10 x 100NF Condensadores cerámicos 5 x 100uF Condensadores cerámicos 10 x 330ohm resistencias 10 x 10Kohm resistencia 10 x 1Kohm resistencia 1 x interruptor de inclinacion (light green) 1 x Termistor (deep green) 1 x altavoz activo 1 x 9g servo (24cm cable) 5 x pulsadores 5 x interruptores triangulares basculantes 1 x pegatina resistencias 1 x potenciómetro 1 x reóstato	

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 45 de 51
		Firma:

Mejora Funduino Mega 2560 Módulo R3 (Compatible c / Oficial Arduino Mega 2560 R3) - Azul + Negro



Marca de fábrica

Funduino

Modelo

Funduino mega 2560 R3

Cantidad

1

Color

Azul + negro

Material

FR4

Características

ATMEGA16U2 más poderosa se expande el FLASH de la ISP. Además de tradicional de USB a puerto serie, permite a los usuarios definir otras funciones como el USB como ratón, teclado, rocking palanca, cámara, etc. En comparación con la versión 2011, este módulo agrega a SDA, SCL. Puerto IOREF da referencia de voltaje protector; un puerto adicional espacio se conserva en la izquierda de IOREF para uso futuro. Su circuito del reajuste también es mejorada y más estable.

Especificación

Micro controlador ATmega2560 tensión de funcionamiento 5V Voltaje de entrada (recomendado 7-12V voltaje de entrada (límites) 6-20V Digital I/O Pins 54 (de los cuales 14 proporcionar PWM salida analógica Pins 16 DC corriente por entrada-salida Pin 40 mA corriente para 3.3V Pin 50 mA Flash Memoria 256 KB de las cuales 4 KB utilizados por bootloader SRAM 8 KB EEPROM 4 KB reloj velocidad 16 MHz

Aplicación

Junta de desarrollo de aprendizaje

Manual de Inglés / Spec

<http://arduino.cc/en/Main/Software>

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 46 de 51
		Firma:

PRESUPUESTO

Para la realización de este proyecto necesitaremos una serie de materiales que los podemos encontrar todos a partir de la pagina deal extreme

Madera de balsa



pistola de silicona



Sierra

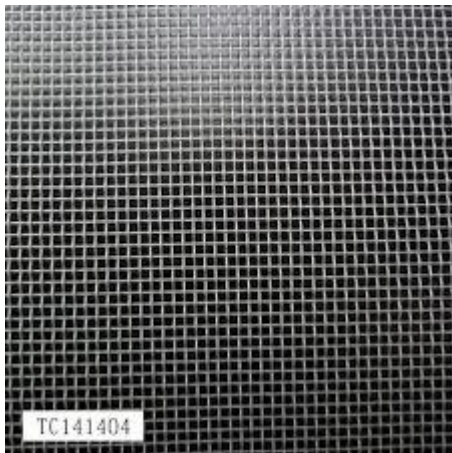


cableado telefónico



Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 47 de 51
		Firma:

Malla de hierro



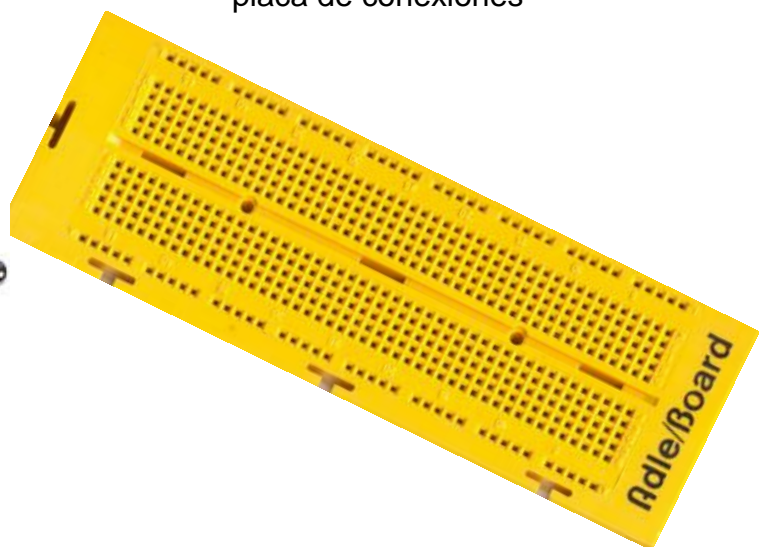
12 bisagras



Motor de 5V



placa de conexiones



Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 48 de 51
		Firma:

Alimentador de 6V 1000mA



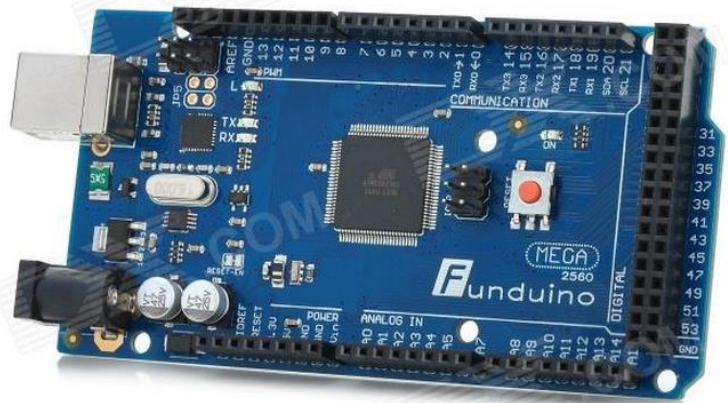
8 pulsadores



2 servos TowerPro SG9096



Placa Fundino mega modulo R3 2560



Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 49 de 51 Firma:

2 relés Finder



KIT ARDUINO KEYES KT0053 13.6550



Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 50 de 51
		Firma:

PRESUPUESTO PROYECTO ARDUINO		
Fecha:	19/06/2015	
Nombre:	David Diaz-Rullo Torres	
PROYECTO		
Producto	Descripción	Precio Total
Materias Primas	Componentes necesarios para la instalación del sistema	89,25 €
Equipos Utilizados	Equipos utilizados para el montaje de la instalación	117,90 €
Software y Herramientas	Software y herramientas de apoyo para el desarrollo	- €
COSTE TOTAL:		207,15 €
* Precios con I.V.A Incluido		

MATERIAS PRIMAS					
Materia prima	Descripción	Cantidad	Metros	Precio/unidad	Precio Total
Bobina cable	1x 400-placa de conexiones 30 x cables macho macho 10 x F5 LEDs rojos 10 x F5 LEDs verdes 1x RGB LEDs 10 x 10NF Condensadores cerámicos 10 x 100NF Condensadores cerámicos 5 x 100uF Condensadores cerámicos 10 x 330ohm resistencias 10 x 10Kohm resistencia 10 x 1Kohm resistencia 1x interruptor de inclinacion (light green) 1x Termistor (deep green) 1x altavoz activo 1x 9g servo (24cm cable) 5 x pulsadores 5 x interruptores triangulares basculantes 1x pegatina resistencias 1x potenciómetro 1x reóstato	1	10	1,00	10,00
Kit arduino		1		13,65	13,65
Madera	madera de balsa para carpinteria	1	2	7,00	14,00
Malla		1	0,5	3,00	1,50
Visagra		8		0,80	6,40
Alimentador	Arduino 9V 1A	1		3,80	3,80
Pulsador		8		0,50	4,00
Rele	rele de 6V	2		3,00	6,00
Motor	Motor de 5V	1		10,00	10,00
Servo	TowerPro SG90	2		2,70	5,40
Placa	Placa Fundino Mega Modulo R3 2560	1		14,50	14,50
COSTE TOTAL:					89,25
* Precios con I.V.A Incluido					

Nombre: David Diaz-Rullo Torres	Proyecto : Arduino(DDT-1)	Fecha:07/04/2015
	Objeto del proyecto (DDT -1.2)	Página 51 de 51
		Firma:

EQUIPOS UTILIZADOS				
Equipos Utilizados	Descripción	Coste		
Sierra		13,70€		
Pistola de silicona		17,20€		
Multímetro digital		20,00€		
PC sobremesa	Intel® Core™ i3 CPU 4.00GB RAM (Ya adquirido antes del proyecto)	-		
Taladro		50,00€		
Conjunto de brocas		10,00€		
Soldador de estaño		7,00€		
COSTE TOTAL:		117,90€		

* Precios con I.V.A Incluido

SOFTWARE Y HERRAMIENTAS				
Software y Herramientas	Descripción	Coste		
Arduino 1.6.5	Plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar	Software libre		
Microsoft Project	Es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.	Licencia académica		