

Martuino Piopio

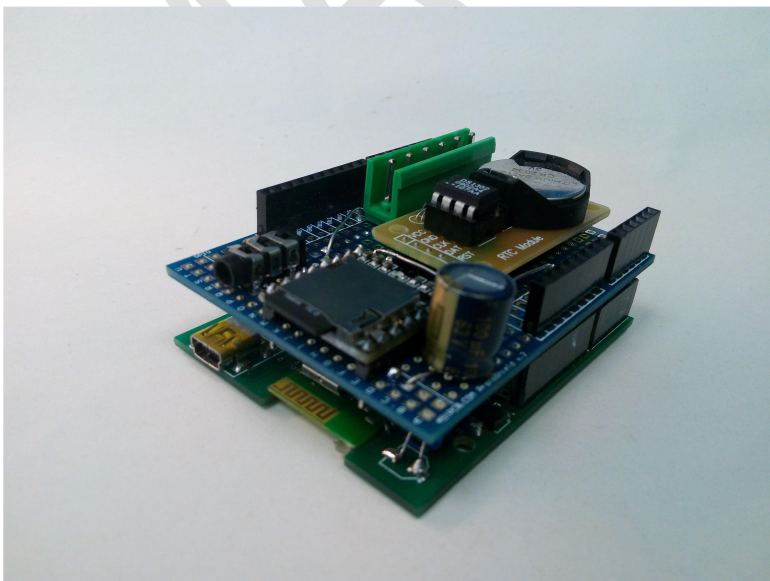
¿Qué es?

Hemos desarrollado con un Martuino un sistema con lo mínimo para la cría de pájaros de diferentes especies. Aunque no seamos expertos en este tema, hemos preparado un hardware con 3 salidas de 12 voltios dimmeables (variables) y 1 amperio cada una (si deseas más, habrá que ponerles radiador) y un reproductor MP3 para reproducir cantos de pájaros y poder enseñar a las crías a cantar. A las salidas se le pueden conectar tiras de Leds de 12 voltios y simular el amanecer o anochecer. Todo el control de las 3 salidas variables como del MP3, se hacen desde un programa en android, programando un horario para cada fase del día. Los programas son diarios, solo controlan 24 horas. Si no tienes teléfono móvil, también puedes programarlo con una conexión telnet.

Además se implementa una progresión entre los diferentes programas memorizados, así se pueden simular amaneceres o atardeceres.

Cada uno de vosotros hará el uso que deba para cada especie de pájaro. Martuino no se hace responsable del mal funcionamiento del control.

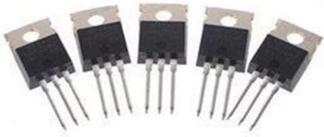
Dale rienda suelta a tú imaginación y utiliza este Martuino para lo que quieras, por ejemplo podrías dormir a un bebe con música y disminuyendo la luminosidad de unos leds poco a poco.



Material necesario.

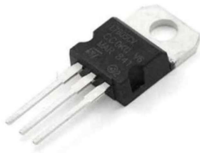
3 transistores TIP120, cómpralo en España (5,46€):

<http://www.e-ika.com/10pcs-tip120-to-220>



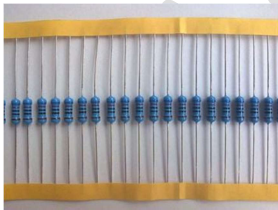
1 LM7805, regulador de 5 voltios, cómpralo en España (1,85€):

<http://www.e-ika.com/regulador-de-voltaje-lm7805-5v-1a-5uds>



3 resistencias de 10k, cómpralo en España (1,57€):

<http://www.e-ika.com/resistencia-de-10-kohm-14w>



Condensador electrolítico 10uF/50v, cómpralo en España (1,67€):

<http://www.e-ika.com/condensador-electrol%C3%ADtico-10uf-50v>



Reproductor Mp3 mini, puedes comprarlo en España (7,59€):

<http://www.e-ika.com/m%C3%B3dulo-mini-mp3-para-arduino-uno>



Una tarjeta microSD (máximo 32Gb) para el reproductor MP3.



Módulo reloj en tiempo real, lo puedes comprar en España (2,69€):

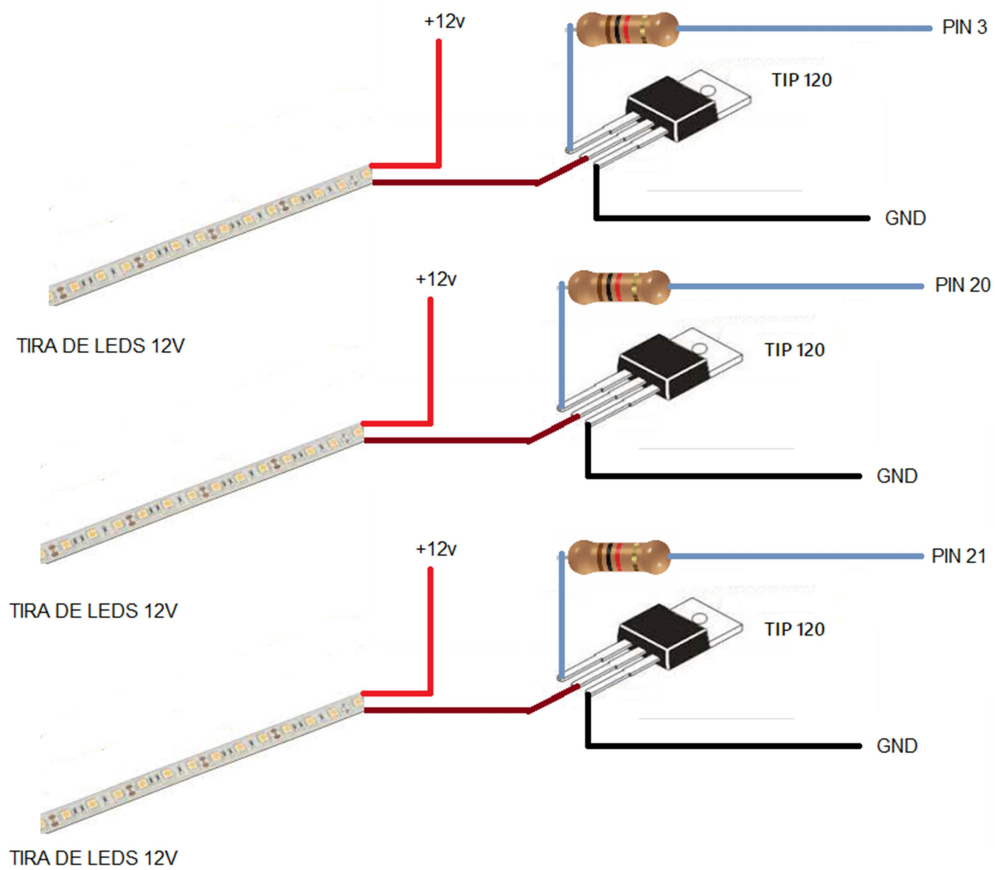
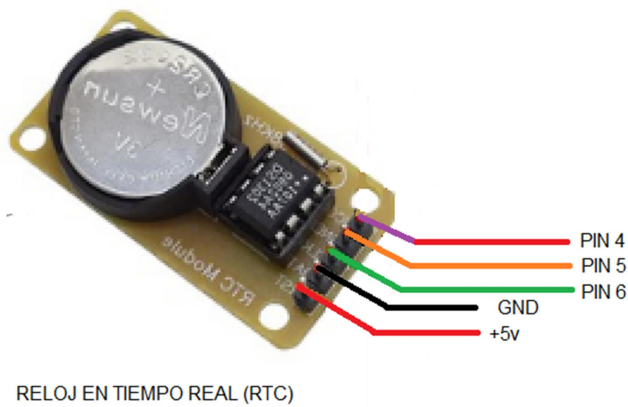
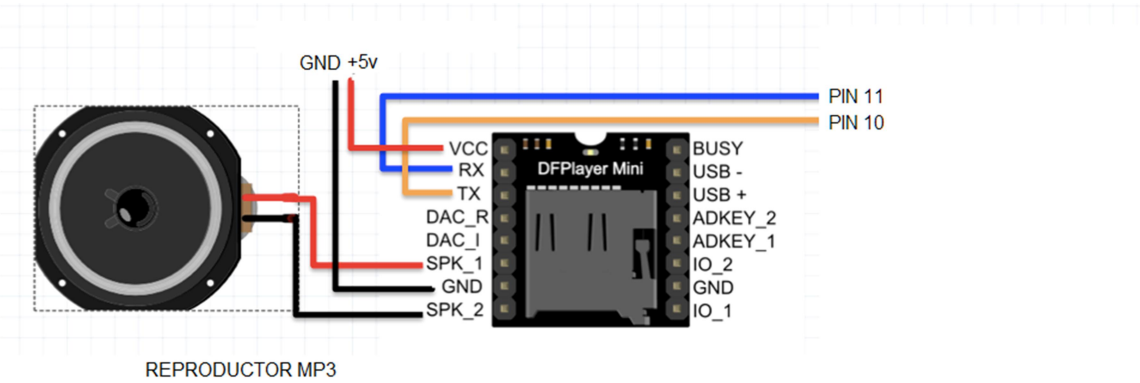
<http://www.e-ika.com/modulo-de-reloj-rtc-ds1302-para-arduino>



Algunos cables, un altavoz y por supuesto un Martuino.

Total sin contar la microSD, altavoz, los cables y Martuino, son 20,83€

Veamos el esquema de conexiones:

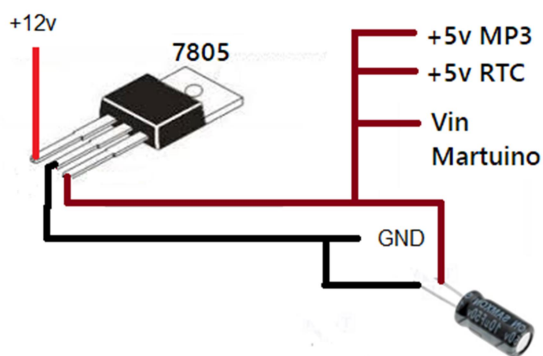


El comienzo

Lo primero que haremos es conectar todas las piezas anteriores conforme dice es esquema de conexiones. Donde aparece la palabra PIN corresponde al patillaje de la placa Martuino.

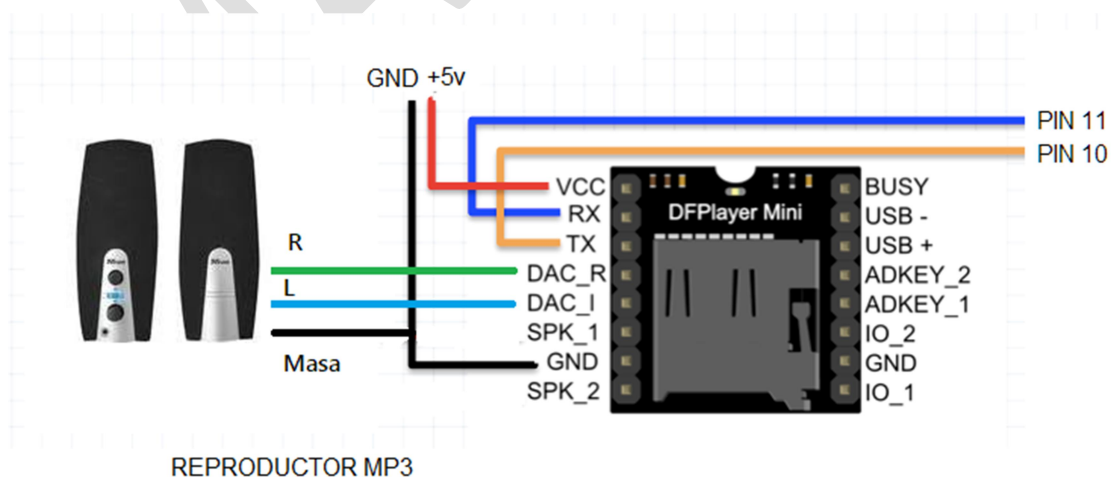
El circuito está hecho para funcionar a 12 v, por lo tanto después de grabar Martuino por el USB, desconecta este y recuerda poner el jumper Vusb a Vin.

Como funciona a 12 voltios tendremos que poner un regulador a 5 voltios, ya que todos los circuitos funcionan a 5 voltios, con este esquema:



El circuito completo también puede funcionar a 5 voltios sin necesidad de este alimentador, pero las tiras de leds que pongas deberán ser de 5 voltios.

El reproductor MP3 puede que no se oiga lo suficiente con un altavoz. Puede conectarse como sigue con unos altavoces con amplificador como los que usamos en cualquier P.C.:



El programa

El programa que debes de cargar en el Martuino, está basado en el ejemplo /wifi/WifiCharSevidor, pero le hemos añadido unos comandos que relatamos a continuación (los comandos están formados por letras en mayúsculas):

- T Te devuelve la hora del Martuino.
- F Pone en hora al Martuino. (por ejemplo: F027002017012057000001)
- Después de la F va el día del mes en 3 dígitos: 027
- A continuación el número de mes en 3 dígitos: 002
- Seguido el año en 3 dígitos: 017 (solo lleva unidad y decena del año)
- Seguido la hora en 3 dígitos: 012
- Seguido los minutos en 3 dígitos: 057
- Seguido los segundos en 3 dígitos: 000
- Seguido el día de la semana: 001 -> corresponde al Lunes(000=Domingo)
- Se utilizan 3 dígitos por abreviar el código del programa.
- S Pon un valor en una salida. S1 seguido de 3 dígitos con el valor 000-255
- Para S2 (salida 2) + 3 dígitos: S2255 = Salida 2 al máximo de potencia.
- Para S3 (salida 3) + 3 dígitos: S3000 = salida 3 apagada o mínima pot.
- El valor 000 corresponde a 0 voltios y el 255 corresponde a la alimentación, en este caso 12 voltios.
- P Reproduce una canción. P00 para parar la reproducción, P01 para reproducir la primera canción almacenada en la microSD y así hasta P99, si existe en la microSD.
- C programa una posición de memoria. Hay hasta 20 memorias, que irá desde la C000 hasta C019, seguido de los campos siguientes:
- hhhmmmv1v1v1v2v2v2v3v3v3mmm
- Donde hhh = hora del programa, mmm el minuto del programa. Será la hora en la que el programa empezará a ejecutarse. Formato de 3 dígitos.
- Luego irán los valores de las salidas v1v1v1 = 000-255 para la salida 1

V2v2v2 = 000-255 para la salida 2 y v3v3v3= 000-255 para la salida 3.

mmm = 001-099 para la canción a reproducir o 000 para parar la reproducción.

- L Lectura de una posición de memoria o programa. L00 a L19
- X Lectura de todas las memorias a la vez.
- ? Muestra una pequeña ayuda de estos comandos.

Después de cada letra ahí que pulsar Intro, o retorno de carro.

Para probar te puedes conectar con el wifi de tu ordenador al C3LYMP3 y poner la contraseña que corresponde a la MAC address de tú Martuino. Va impresa en la placa.

Una vez estés conectado, puedes abrir cualquier programa de telnet, y conectar con la IP 192.168.1.1 puerto 23. Una vez hayas conectado podrás poner los comandos descritos anteriormente, en mayúsculas, y recibirás contestación del Martuino.

El software lo puedes descargar de aquí: <http://www.martuino.com/code/c3lym.ino>

Veamos algunos fragmentos del software:

-El RTC (Real Time Clock) chip DS1302:

//Librería necesaria

```
#include <DS1302.h>
```

//Variables necesarias

```
char buftime[50];
```

```
String diasem;
```

```
int ano = 2016;
```

```
int mes = 1;
```

```
int dia = 1;
```

```
int diasemana = 0;
```

```
int hora = 0;
```

```
int minuto = 0;
```

```
int segundo = 0;
```

//Pines del RTC

```
const int kCePin = 5; // Chip Enable
```

```
const int kIoPin = 6; // Input/Output
```

```
const int kSclkPin = 7; // Serial Clock
```

// Crea un objeto DS1302.

```
DS1302 rtc(kCePin, kIoPin, kSclkPin);
```

```

//Función para devolver el día de la semana según su número
String dayAsString(const Time::Day day) {
switch (day) {
    case Time::kSunday: return "Domingo";
    case Time::kMonday: return "Lunes";
    case Time::kTuesday: return "Martes";
    case Time::kWednesday: return "Miercoles";
    case Time::kThursday: return "Jueves";
    case Time::kFriday: return "Viernes";
    case Time::kSaturday: return "Sabado";
}

    return "(Sin dia)";
}

//Función que recoge la hora y fecha
void printTime() {
    // Coge la fecha y hora del chip DS1302.
    Time t = rtc.time();
    // Nombre del día de la semana.
    const String day = dayAsString(t.day);
    // Dale formato y lo introduces en Bufer.

    snprintf(buftime, sizeof(buftime), "%s %02d:%02d:%02d
%02d-%02d-%04d ",day.c_str(),t.hr, t.min, t.sec,
t.date, t.mon, t.yr);
    diasem = day.c_str();
    ano = t.yr;
    mes = t.mon;
    dia = t.date;
    hora = t.hr;
    minuto = t.min;
    segundo =t.sec;

}

void setup()
{
    //Inicializamos RTC DS1302
    rtc.writeProtect(false);
    rtc.halt(false);
}

```

Cuando queremos leer la hora y fecha solo tendremos que hacer:

```
printTime();
```

```
Serial.println(buftime); //Imprimira la fecha y hora por el puerto serie
```

Y cuando ponerlo en hora se pondrá así:

```

//Variable Time que da el formato adecuado.
Time t(2000 + ano, mes, dia, hora, minuto, segundo, Time::kSunday);
//Ahora lo enviamos al chip RTC
rtc.time(t);

```


-El MP3 conectado al puerto serie 1:

//Librería necesaria

```
#include <DFPlayer_Mini_Mp3.h>

Void setup()

{

    //Inicializamos MP3
    Serial1.begin(9600);
    mp3_set_serial (Serial1); //Pon Serial para el
    módulo mp3 DFPlayer-mini
    mp3_set_volume (25);
    mp3_stop();

//Reproduce la canción 1 de la tarjeta microSD

    mp3_play_physical(1);
//Repite esta canción
    mp3_play_repeat();

//Parar la canción en curso
    mp3_stop();
```

La hora y fecha se leerá cada segundo. Esta hora será comparada con la de todas las memorias y cuando coincida con alguna de ellas actuará lo memorizado en esa memoria o programa.

También hemos implementado otra cosa, para que los saltos de luz (en las salidas) no sea brusco, el software calcula desde la memoria que está ahora, cuando coincida con la hora del Martuino, hasta la siguiente memoria, la diferencia de valores en las salidas, en las 3, y cada minuto que pasa aumentará o disminuirá el valor de las salidas para ajustarse al valor de la siguiente cuando llegue a su hora. Es decir, si ponemos que a las 10 horas la salidas 1 tenga un valor de 000 y las 12 horas tenga un valor de 127, el hará el siguiente calculo:

Minutos que hay entre las 10 y la 12 = 120 minutos

Diferencia entre valores de la salida 1 de las 10 a las 12, $127 - 0 = 127$

Luego divide el valor por los minutos -> $127 / 120 = 1,05$

Por lo tanto desde las 10 horas hasta las 12 la salida 1 irá aumentando cada minuto en 1,05 su valor. Así cuando sean las 12 horas su valor será el de 127, pero progresivamente. Si no existiera este método, pasaríamos de las 10 horas que estaría apagada la salida 1, a las 12 horas se encendería la mitad de la potencia la salida 1 de golpe. Así se puede simular amaneceres o atardeceres. Esto es para todas las salidas independientemente.

Siempre debemos de poner el primer programa con los valores de salidas a 000 para que no empiece calculando y se enciendan poco a poco las salidas.

Y el último programa también deberá contener las salidas a 000, si no calculará el siguiente valor y nunca se apagarán.

En la aplicación para Android hemos cambiado los valores de las salidas que pueden ser desde 000 hasta 255, por 0% hasta 100%. Más entendibles para aquellos que no se aclaran con el sistema binario.

La aplicación para Android te la puedes descargar del Play Store de google:

<https://play.google.com/store/apps/details?id=com.martuino.piopio&hl=es>

o buscando Piopio Martuino.

La APP para Android

1.- Introducción.

La app Piopio para Android controla el hardware C3LYMP3 de Martuino.

Con ella se pueden programar hasta 20 horarios diferentes para 3 salidas y un reproductor MP3. Cada salida es regulable entre 0% y 100% .

Cuando se llega a un horario programado, la app calcula el tiempo que falta para llegar al segundo horario programado y va aumentando o disminuyendo la luminosidad de cada salida según los valores guardados en cada horario. Es decir, los valores de las salidas no cambian bruscamente cuando se alcanza la hora programada, si no, que cada programa realiza un cálculo hacia el siguiente para que el cambio sea progresivo.

Por lo tanto programe los horarios en orden, empezando siempre por uno con todas las salidas a 0%. El reproductor MP3 tiene la opción de poner el número de canción según se copiaron en la tarjeta microSD. Si el número es 0 la reproducción se para.

El hardware necesario lo puede encontrar en:

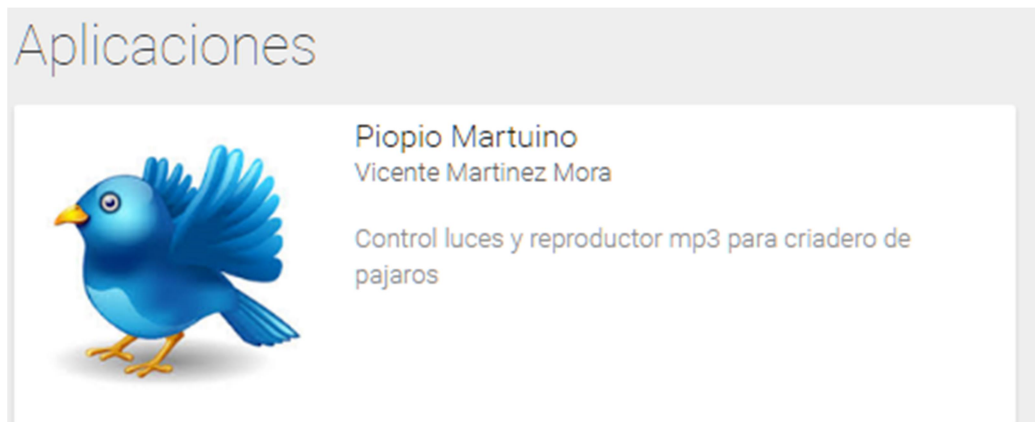
<http://www.martuino.com/martuino.html>



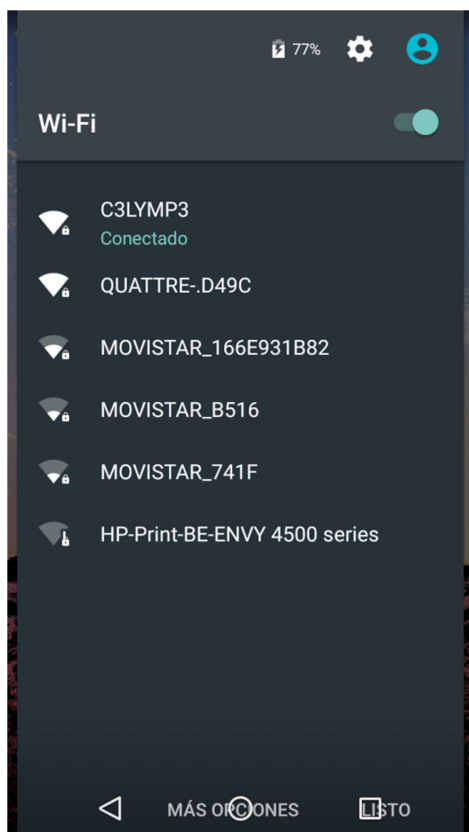
Imagen de la App.

2. Como funciona.

Lo primero será instalar la aplicación desde Google Play. Busque la aplicación como “Piopio martuino”.



Una vez instalada antes de abrirla deberá conectar con el hardware mediante su Wi-Fi. Busque con su teléfono el Wi-Fi “C3LYMP3” y conéctese:



La contraseña viene en una etiqueta en el hardware. Una vez conectado podrá abrir la App.

Veamos que es cada cosa en la App:

Pulse aquí para volver a conectar en caso de fallo.

71% 18:43

Piopio

Fecha y hora del hardware.

Miercoles 18:43:49

Programas:

Hora/Min: Salida 1: Salida 2: Salida 3: MP3:

18:35	0%	0%	0%	0
22:30	0%	0%	0%	0
00:00	0%	0%	0%	0
00:00	0%	0%	0%	0
00:00	0%	0%	0%	0
00:00	0%	0%	0%	0
00:00	0%	0%	0%	0
00:00	0%	0%	0%	0

Cada línea es un horario para programar. La que está en verde es el último horario que se ejecutó. Por lo tanto en el cálculo de los niveles de las salidas influirá el siguiente.

Pulse aquí para cambiar la hora

Pulse aquí para cambiar el valor

Nivel salida 1

Nivel actual de cada salida. Con este elemento se pueden modificar los niveles manualmente. Este corresponde a la salida 3.

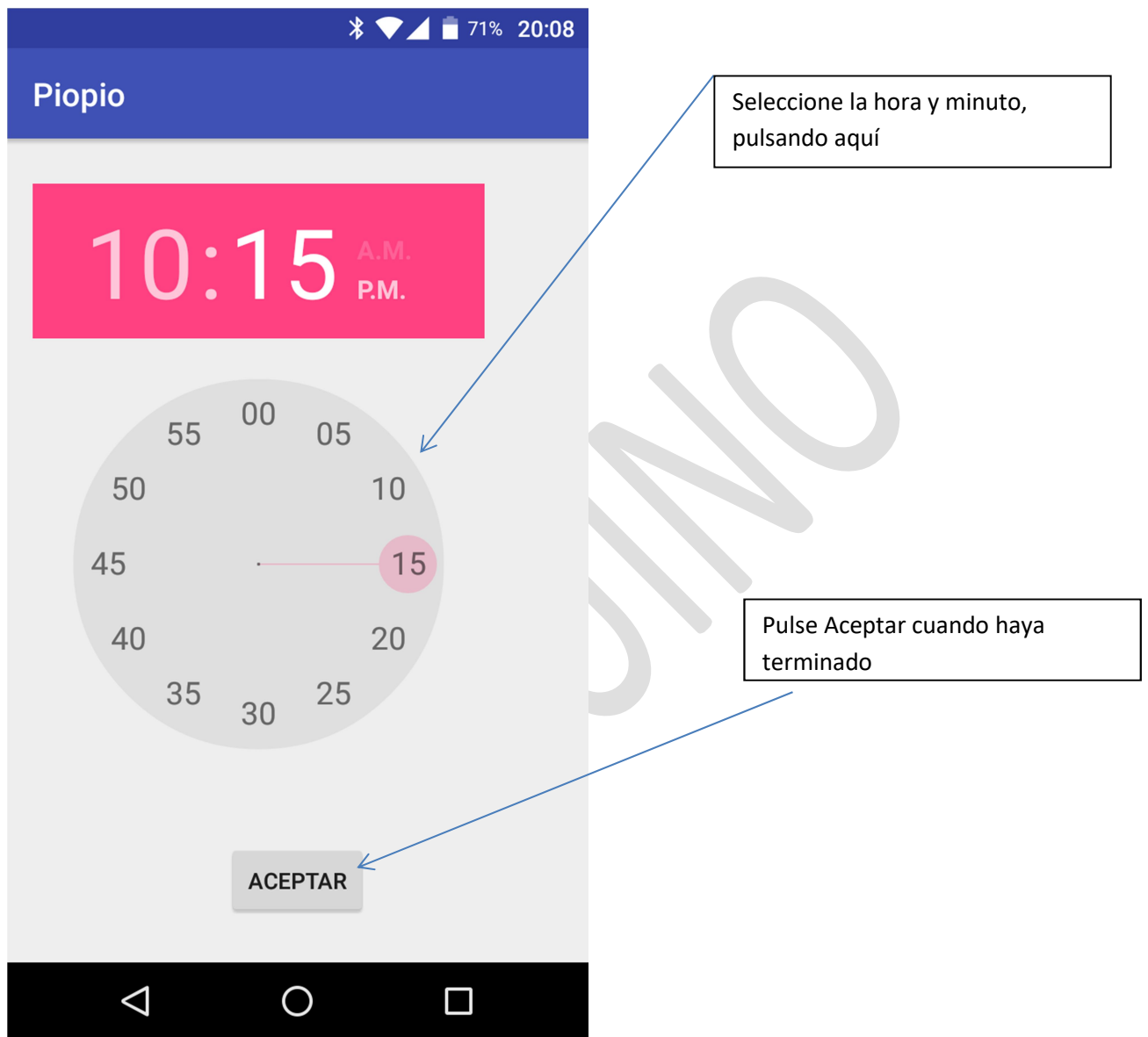
49,8 % 0,0 % 0,0 %

ENVIAR

Nivel salida 2

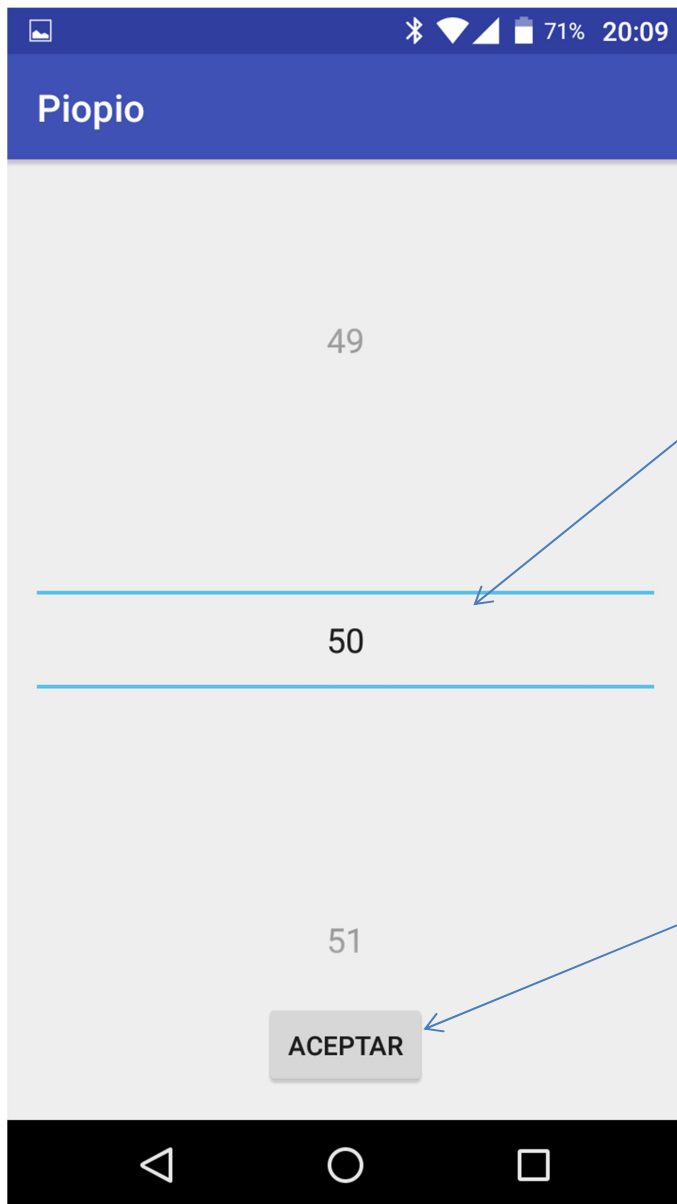
Cuando modifique un horario o valor debe de pulsar aquí para enviárselo al hardware. Hasta que no pulse aquí puede modificar varios horarios a la vez.

Al pulsar sobre la hora de cualquier horario aparecerá la siguiente pantalla:



La nueva hora aparecerá en el lugar donde pulsó.

Al pulsar sobre cualquier valor de las salidas o MP3 aparecerá esta pantalla:



Seleccione un valor entre 0 y 100 moviendo hacia arriba o abajo o poniendo el número con el teclado.

Pulse Aceptar cuando haya finalizado.

El nuevo valor aparecerá donde pulsó.

Para poner en hora el hardware solo tiene que pulsar sobre la etiqueta donde aparece la hora del hardware (vea página 3). Si pulsa se envía la hora que tenga su teléfono móvil .

Si abre la App sin conectar previamente el Wi-Fi le saldrá el **error “Conecte el wifi al C3LYMP3”**. Tan solo tiene que seleccionar el Wi-Fi y conectarse (página 2). Después para actualizar la conexión pulse sobre la etiqueta “Programas:” o reinicie la aplicación.

Cuando abra la aplicación **directamente se establecerá la conexión con el hardware**, mostrando los horarios memorizados, en verde el horario donde está situado en el momento actual, y **la fecha y hora** en la parte superior que **se irá actualizando cada 10 segundos**.

Los valores que aparecen en la parte baja de la aplicación también se actualizan cada 10 segundos, mostrando el valor real. Moviendo la bolita modificará el valor de cada salida desde el 0% al 100%.

Establezca siempre un horario de inicio donde los valores sean 0 y uno de fin donde también los valores sean 0. Por el medio podrá poner los horarios que quiera. En la App solo pueden ponerse 8 horarios. Estos horarios se repiten diariamente.

Como hemos comentado **el paso de un horario a otro se hace progresivamente**. Así cuando se llega a un horario, se colocan los valores de ese horario para todas las salidas y se calcula el incremento o decremento hasta llegar al siguiente horario. Esto se hace así, se mira la diferencia en minutos entre el horario actual y el siguiente y la diferencia del valor de cada salida. Esto provoca que cada minuto el valor de las salidas vaya modificándose. Por ejemplo, pongamos que a las 22:30 los valores están todos a 0% y el siguiente horario sea a las 23:00 y el valor de la salida 1 sea del 50%. Cuando sean las 22:30 se fijará el valor de las salidas a 0% y **se hará el cálculo siguiente**:

- Diferencia de minutos = 30
- Diferencia de la salida 1 pasa del 0% al 50%, es ascendente.
- Dividirá el $50/30 = 1,66\%$ cada minuto.

Cada minuto el valor de la salida 1 irá subiendo 1,66% hasta llegar a las 23:00 con el valor de 50%. Cuando sean las 23:00 comenzará el cálculo con el siguiente horario.

El número de canción seleccionada en un horario **se reproduce infinitamente** hasta que cambie en otro horario o se pare, al encontrar un 0 como canción en el campo MP3. Soporta hasta 32Gb de microSD, con FAT16 o FAT32.

Las salidas soportan hasta 1 Amperio a 12 voltios.

El hardware se alimenta con 12 voltios continua.