

Riego programable con Telegram, ESP32 placa dos relés y DS3231

Riego para macetas con ESP32 placa dos relés y modulo reloj DS3231

Podemos programarlo mediante bot de telegram los días de riego y hora,minutos,segundos de inicio y de fin de riego

Conexiones:

Bomba:16

DS3231: SDA:21, SCL:22

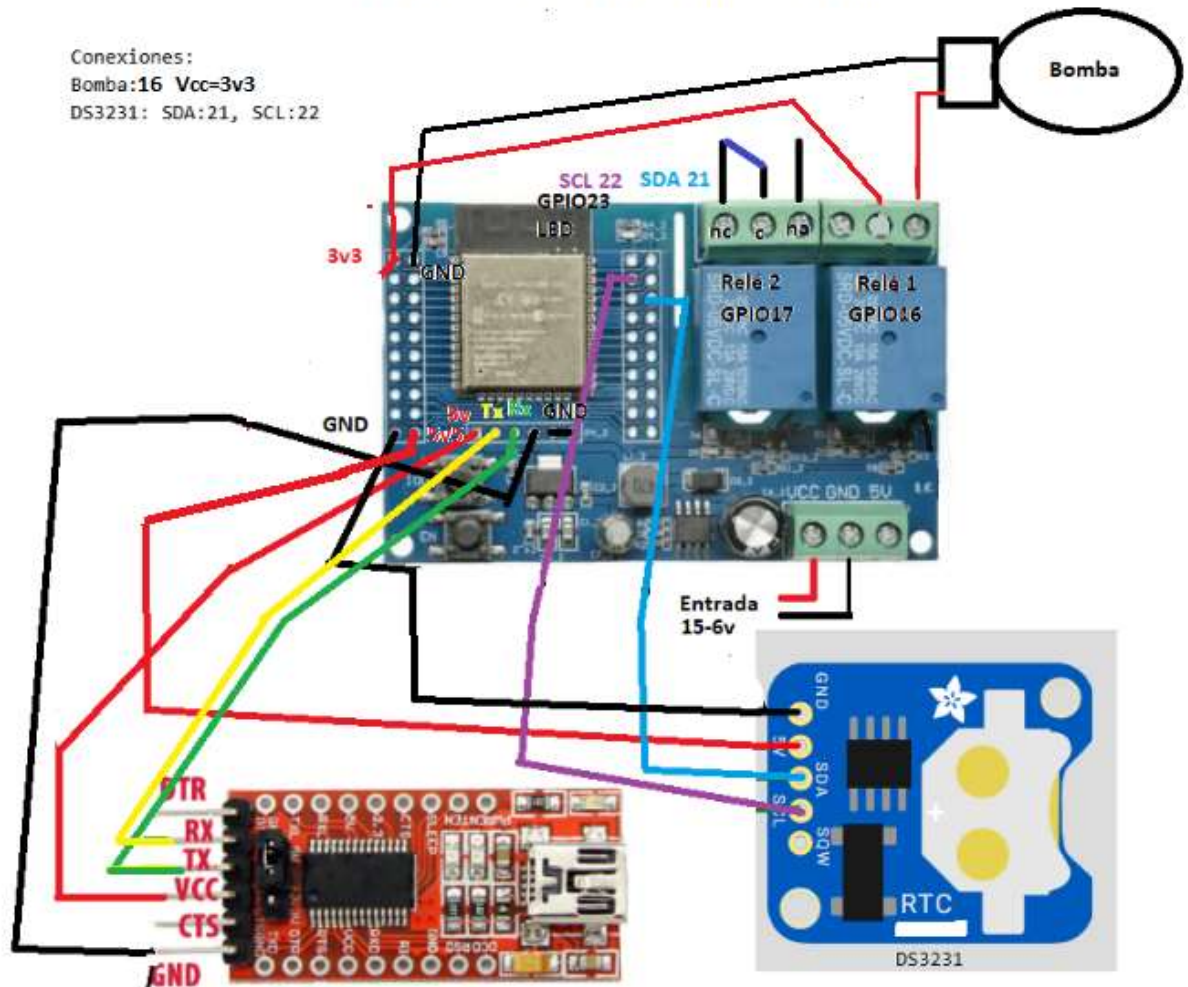
ÓRDENES:

```
inicio //información comandos";
on // activa bomba";
off // deactiva bomba";
info// muestra valor EEPROM";
eeprom // muestra valor EEPROM";
a //año reloj (a.25)";
ho //hora reloj (ho.22)";
me //mes reloj (me.9)";
di //dia reloj (di.25)";
mi //minutos reloj (mi.25)";
se //segundos reloj (se.25)";
d0 //Borra días de riego";
db //ver días de riego";
dd //días de riego en binario (dd.01111111)";
dr //días de riego (dr.d) d l m x j v s";
ir //hora,minuto,segundo de inicio de riego (ir.13,01,10)";
fr //hora,minuto,segundo de fin de riego (fr.13,01,50)" ;
f //ajustar fechaTiempo (f.d,m,y,h,m,s) (f.1,6,25,13,10,15)" ;
```

- Crear Bot de telegram
<https://randomnerdtutorials.com/telegram-request-esp32-esp8266-nodemcu-sensor-readings/> o utilizar uno propio.
- Cambiar en el código .ino los valores de vuestra WIFI y los valores del BOT de Telegram.

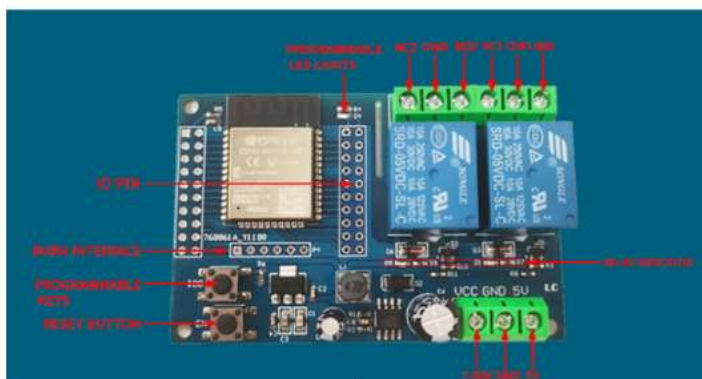
CONTROL RIEGO ESP32 PLACA DOS RELES CON TELEGRAM

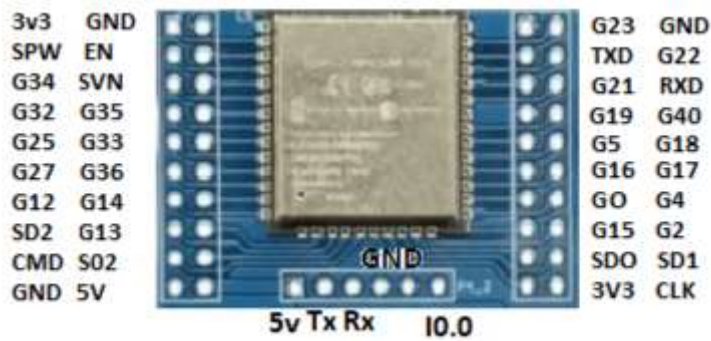
Conexiones:
 Bomba:16 Vcc=3v3
 DS3231: SDA:21, SCL:22



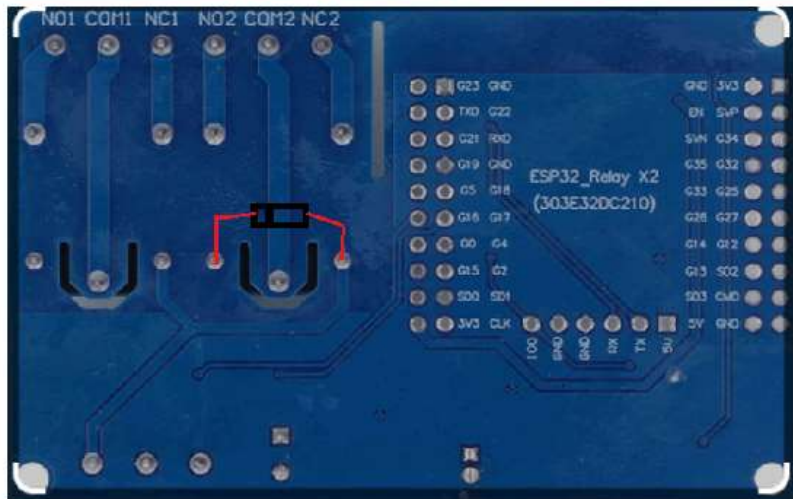
TTL serial port module	ESP32 development board
GND	GND
TX	RX
RX	TX
5V	5V

ESP32 Dev Module





Problema reinicio al conectar bomba por picos de corriente, solución poder diodo en antiparalelo en bobina rele



/*

[ESP32_TelegramTemporizadorRiegoDS3231.ino](#)

https://adafruit.github.io/RTCLib/html/_r_t_clib_8h_source.html

<https://github.com/adafruit/RTCLib/tree/master>

<https://randomnerdtutorials.com/esp32-ds3231-real-time-clock-arduino/#more-164824>

//.....

Placa ESP32 con dos relés

GPIO0 Botón de placa

GPIO16 Relé 1 (más externo)

GPIO17 Relé 2

GPIO23 LED

RTC_DS3231 SDA 21, SCL 22

Programar como: ESP32 DEV Module

PROBLEMAS AL ENCENDER LA BOMBA: se reinicia

RESUELTO, pongo un diodo antiparalelo en la bobina del rele y alimento la bomba a 3v3

on: activa bomba

off: desactiva bomba

Enciende la bomba la fecha programada y el tiempo programado

Programamos:

- Dias de riego

```
String diasDeLaSemanaCompleto[7] = {"Domingo", "Lunes", "Martes", "Miercoles", "Jueves",  
"Viernes", "Sabado"}; //para mostrar dias completos
```

```
// bit    0      1      2      3      4      5      6
```

```
// valor  1      2      4      8     16     32     64
```

```
byte diasSemanaRiego = B00000000; //pone a 1 los dias de riego bit 7.....0
```

```
String diasDeLaSemana[7] = {"d", "l", "m", "x", "j", "v", "s"}; // para entrar dias de riego
```

- HoraRiego de inicio riego

```
int horaRiego= 12;
```

```
int minutoRiego= 1;
```

```
int segundoRiego= 0;
```

- HoraFinRiego de fin de riego en s.

```
int horaFinRiego= 12;
```

```
int minutoFinRiego= 1;
```

```
int segundoFinRiego=30;
```

- byte diasSemanaRiego con 1 riega, o no riega

```
*/
```

```
#include <EEPROM.h>
```

```
#include <Wire.h>
```

```
#include <RTCLib.h>
```

```
#include <WiFi.h>
```

```
#include <WiFiClientSecure.h>
```

```
#include <UniversalTelegramBot.h>
```

```
#include <ArduinoJson.h>
```

```
//..... Variables .....
```

```
const char* ssid = "....."; //Cambiar
```

```
const char* password = "....."; //Cambiar
```

```
// Initialize Telegram BOT
```

```
#define BOTtoken "....." //Cambiar
```

```
#define CHAT_ID "....." //Cambiar
```

```
WiFiClientSecure client;
```

```
UniversalTelegramBot bot(BOTtoken, client);
```

```
bool estadoBomba = false; //estado bomba
```

```
bool automaticoBomba = true; //true: riego con reloj; false: riego con on
```

```
//Chequea mensajes cada segundo
```

```
int botRequestDelay = 1000;
```

```
unsigned long lastTimeBotRan;
```

```

int Bomba = 16; //Rele mas externo de la placa

String data[2]; //matiz de datos para entrar valores separados por ,
String data1[6]; //matiz de datos para entrar valores FECHA separados por :

String diasDeLaSemanaCompleto[7] = {"Domingo", "Lunes", "Martes", "Miercoles", "Jueves",
"Viernes", "Sabado"}; //para mostrar dias completos
        // bit    0        1        2        3        4        5        6
        // valor  1        2        4        8        16       32       64
byte diasSemanaRiego = 0b00000010; //pone a 1 los dias de riego bit 7.....0
String diasDeLaSemana[7] = {"d", "l", "m", "x", "j", "v", "s"}; // para entrar dias de riego

String message = ""; //mensaje recibido a,b
String valor=""; // valor numerico para entrar
String texto=""; //texto a mostrar
String value=""; //lee dato

int valor_inicial_EEprom; // para ver si ya esta configurada los valores
const int logitud_memoria = 1000; //para almacenar los valores en EEPROM
const int x=5;
const int memoria_diasSemanaRiego= 2*x;
//const int memoria_ndias_riego= 2*x;
const int memoria_anno= 3*x;
const int memoria_dia= 4*x;
const int memoria_mes= 6*x;
const int memoria_hora= 7*x;
const int memoria_minutos= 8*x;
const int memoria_segundos= 9*x;
const int memoria_horaRiego= 10*x;
const int memoria_minutoRiego= 11*x;
const int memoria_segundoRiego= 12*x;
const int memoria_horaFinRiego= 13*x;//65
const int memoria_minutoFinRiego= 14*x;//70
const int memoria_segundoFinRiego=15*x;

//reloj
int diames= 4;
int mes=8;
int anno=25;
int hora= 13;
int minutos=0;
int segundos=0;

//programa riego
int horaRiego= 12;
int minutoRiego= 40;
int segundoRiego= 30;
//fin riego

```

```

int horaFinRiego= 12;
int minutoFinRiego= 41;
int segundoFinRiego=10;

unsigned long HoraActual ;//en seg
unsigned long HoraRiego ;//en seg
unsigned long HoraFinRiego ;//en seg

// Variable para almacenar dirección MAC
uint8_t baseMac[6];

//.....
RTC_DS3231 rtc;

String fechaHoraFormatearda(){
    DateTime now = rtc.now();
    String yearStr = String(now.year(), DEC);
    String monthStr = (now.month() < 10 ? "0" : "") + String(now.month(), DEC);
    String dayStr = (now.day() < 10 ? "0" : "") + String(now.day(), DEC);
    String hourStr = (now.hour() < 10 ? "0" : "") + String(now.hour(), DEC);
    String minuteStr = (now.minute() < 10 ? "0" : "") + String(now.minute(), DEC);
    String secondStr = (now.second() < 10 ? "0" : "") + String(now.second(), DEC);
    String dayOfWeek = diasDeLaSemanaCompleto[now.dayOfTheWeek()];

    return dayOfWeek + ", " + yearStr + "/" + monthStr + "/" + dayStr + " " + hourStr + ":" + minuteStr +
    ":" + secondStr;
}

void enviar(String info){
    Serial.println(":" + info);
    bot.sendMessage(CHAT_ID, info, ""); //telegram
}

//*****FUNCIONES*****
String cadenaByteInBinary(byte b) {
    //devuelve como string Byte en binario
    String resultado ="";
    for (int i = 7; i >= 0; i--) {
        if (b & (1 << i)) {
            resultado+='1';

        } else {
            resultado+='0';
        }
    }
    return resultado;
}

```

```

}

//.....
String obtenerDiasCompleto(){
    //Devuelve los dias de la semana que estan a 1 en diaDeLaSemana
    String resultado = "";
    for (int i = 7; i >= 0; i--) {
        if (bitRead(diasSemanaRiego, i) == 1){
            resultado+= diasDeLaSemanaCompleto[i]+ " ";
        }
    }
    return resultado;
}

//.....
int existeEnArray(String array[], int longitud, String valorBuscado) {
    //Devuelve el n° de elemento de la matriz si existe, sino devuelve 10
    for (int i = 0; i < longitud; i++) {

        if (array[i] == valorBuscado) {
            return i; // Se encontró el valor
        }
    }
    return 10; // No se encontró el valor
}

//.....
String mostrarHora(int vhora,int vminuto,int vsegundo){
    //formatea fecha H:m:s
    String vvhora = (vhora < 10 ? "0" : "") + String(vhora, DEC);
    String vvminuto = (vminuto < 10 ? "0" : "") + String(vminuto, DEC);
    String vvsegundo = (vsegundo < 10 ? "0" : "") + String(vsegundo, DEC);
    return String(vvhora) + ":" + String(vvminuto) + ":" + String(vvsegundo) ;
}

String verHoras(){
    DateTime now = rtc.now();

    HoraActual = now.hour()*60*60 + now.minute()*60 + now.second() ; //pasamos a seg
    HoraRiego = horaRiego*60*60 + minutoRiego*60 + segundoRiego;
    HoraFinRiego = horaFinRiego*60*60 + minutoFinRiego*60 + segundoFinRiego;
    String textoFechas = "Hora Actual: "+String(HoraActual) + "= "+
    mostrarHora(now.hour(),now.minute(),now.second())+
    "\n Hora Riego: " +String(HoraRiego) + "= "+
    mostrarHora(horaRiego,minutoRiego,segundoRiego)+
    "\n Hora FinRiego: " + String(HoraFinRiego) + "= " +
    mostrarHora(horaFinRiego,minutoFinRiego,segundoFinRiego);
    return textoFechas;
}

//.....
void verEEPROM() {

```

```

//muestra contenido de la memoria
texto = "verEEPROM():\n";
texto += fechaHoraFormatearda();
texto += "\n diasSemanaRiego: " + String(EEPROM.read(memoria_diasSemanaRiego));
texto += " (" + cadenaByteInBinary(EEPROM.read(memoria_diasSemanaRiego)) + ") (" +
obtenerDiasCompleto()+"");
texto += "\n horaRiego: " + String(EEPROM.readInt(memoria_horaRiego)) ;
texto += ":" + String(EEPROM.readInt(memoria_minutoRiego)) ;
texto += ":" + String(EEPROM.readInt(memoria_segundoRiego)) ;
texto += "\n hora Fin Riego: " + String(EEPROM.readInt(memoria_horaFinRiego)) ;
texto += ":" + String(EEPROM.readInt(memoria_minutoFinRiego)) ;
texto += ":" + String(EEPROM.readInt(memoria_segundoFinRiego)) ;
enviar(texto);
} //verEEPROM

void informacion() {
  DateTime now = rtc.now();
  texto = "informacion():\n";
  texto += fechaHoraFormatearda();
  texto += "\n Hora: " + String(now.day())+"/"+String(now.month())+"/"+String(now.year());
  texto += "\n Hora de inicio de riego: " + mostrarHora(horaRiego,minutoRiego,segundoRiego) ;
  texto += "\n Hora de fin de riego: " + mostrarHora(horaFinRiego,minutoFinRiego,segundoFinRiego);
  texto += "\n Dias de riego : " + String(diasSemanaRiego) ;
  texto += " (" + cadenaByteInBinary(EEPROM.read(memoria_diasSemanaRiego)) + ") (" +
obtenerDiasCompleto()+"");
  enviar(texto);
} //

void inicio(){
  texto = "inicio():\n";
  texto += fechaHoraFormatearda();
  texto += "\n Ordenes:" ;
  texto += "\n inicio //información comandos";
  texto += "\n on // activa bomba";
  texto += "\n off // deactiva bomba";
  texto += "\n info// muestra valor EEPROM";
  texto += "\n eeprom // muestra valor EEPROM";
  texto += "\n a //año reloj (a.25)";
  texto += "\n ho //hora reloj (ho.22)";
  texto += "\n me //mes reloj (me.9)";
  texto += "\n di //dia reloj (di.25)";
  texto += "\n mi //minutos reloj (mi.25)";
  texto += "\n se //segundos reloj (se.25)";
  texto += "\n d0 //Borra días de riego";
  texto += "\n db //ver días de riego";
  texto += "\n dd //días de riego en binario (dd.01111111)";
  texto += "\n dr //días de riego (dr.d) d l m x j v s";
  texto += "\n ir //hora,minuto,segundo de inicio de riego (ir.13,01,10)";
  texto += "\n fr //hora,minuto,segundo de fin de riego (fr.13,01,50)" ;

```



```

    texto += "\n f //ajustar fechaTiempo (f.d,m,y,h,m,s) (f.1,6,25,13,10,15)" ;

    enviar( texto);

}

bool isScheduledON(DateTime now){
//bool isScheduledON(date)
    Serial.println(verHoras()); //PARA DEPURAR
    HoraActual = now.hour()*60*60 + now.minute()*60 + now.second() ; //pasamos a seg
    HoraRiego = horaRiego*60*60 + minutoRiego*60 + segundoRiego;
    HoraFinRiego = horaFinRiego*60*60 + minutoFinRiego*60 + segundoFinRiego;
    bool DiaCondicion =(bitRead(diasSemanaRiego, now.dayOfTheWeek() )== 1 ); //si esta a uno el
    bit de diasSemanaRiego
    bool HoraCondicion = (HoraActual > HoraRiego && HoraActual < HoraFinRiego);

    if (DiaCondicion && HoraCondicion)
    {
        return true;
    }
    return false;
}

//Gestiona mensajes recibidos
void handleNewMessages(int numNewMessages) {

    Serial.println("handleNewMessages");
    Serial.println(String(numNewMessages));

    for (int i=0; i<numNewMessages; i++) {

        String chat_id = String(bot.messages[i].chat_id);
        if (chat_id != CHAT_ID){
            enviar("Unauthorized user");
            continue;
        }

        value = bot.messages[i].text;
        Serial.println(message);

        String from_name = bot.messages[i].from_name;

        DateTime now = rtc.now(); //obtiene fecha hora actuales

        //-----fin telegram-----

        for (int i = 0; i < 2 ; i++) {
            int index = value.indexOf(".");

```

```

    data[i] = value.substring(0, index); // .toInt();
    value = value.substring(index + 1);
}
Serial.println(value); // Presenta valor.
message = data[0]; // obtiene mensaje separando orden y valor
if (data[0] == data[1]) {data[1] = "1";} // para evitar error al no meter 2 parametros
valor = data[1];

if (message == "on") { // activa bomba

    digitalWrite(Bomba, HIGH);
    estadoBomba = true;
    automaticoBomba = false; // riego manual
    enviar("Estado_Bomba= Activada");
}
// .....
else if (message == "off") { // desactiva bomba
    digitalWrite(Bomba, LOW);
    estadoBomba = false;
    automaticoBomba = true; // riego con temporizador
    enviar("Estado_Bomba= Desactivada");
}
// .....
else if (message == "info") { // muestra valor EEPROM
    informacion();
}
// .....
else if (message == "eeprom") { // muestra valor EEPROM
    verEEPROM();
}
// .....
else if (message == "se") { // minutos reloj se.22
    segundos = valor.toInt();
    EEPROM.writeInt(memoria_segundos, segundos);
    EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
    rtc.adjust(DateTime(200+anno, mes, diames, hora, minutos, segundos));
    enviar(String("minutos reloj: ") + valor + " " + fechaHoraFormateada());
}
// .....
else if (message == "mi") { // minutos reloj mi.22
    minutos = valor.toInt();
    EEPROM.writeInt(memoria_minutos, minutos);
    EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
    rtc.adjust(DateTime(200+anno, mes, diames, hora, minutos, segundos));
    enviar(String("minutos reloj: ") + valor + " " + fechaHoraFormateada());
}
// .....
else if (message == "di") { // dia reloj di.3

```

```

diames = valor.toInt();
EEPROM.writeInt(memoria_dia,diames);
EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
rtc.adjust(DateTime(2000+anno,mes, diames, hora, minutos, segundos));
enviar(String("minutos reloj: ") + valor + " "+ fechaHoraFormatearda());
}
//.....
else if (message=="me"){ //mes reloj me.6
mes = valor.toInt();
EEPROM.writeInt(memoria_mes, mes);
EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
rtc.adjust(DateTime(2000+anno,mes, diames, hora, minutos, segundos));
enviar(String("minutos reloj: ") + valor + " "+ fechaHoraFormatearda());
}
//.....
else if (message=="ho"){ //hora reloj ho.22
hora = valor.toInt();
EEPROM.writeInt(memoria_hora, hora);
EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
rtc.adjust(DateTime(2000+anno,mes, diames, hora, minutos, segundos));
enviar(String("minutos reloj: ") + valor + " "+ fechaHoraFormatearda());
}
//.....
else if (message=="a"){ //año reloj a.25
anno = valor.toInt();
EEPROM.writeInt(memoria_anno, anno);
EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
rtc.adjust(DateTime(2000+anno,mes, diames, hora, minutos, segundos));
enviar(String("minutos reloj: ") + valor + " "+ fechaHoraFormatearda());
}
//.....
else if (message=="b"){ //rele 16, 17
Bomba = valor.toInt();
pinMode(Bomba, OUTPUT);
digitalWrite(Bomba, LOW);

enviar(String("Salida bomba: ") + String(Bomba));
}
//.....
//.....
else if (message=="d0"){ //Borra días de riego
diasSemanaRiego = 0;
EEPROM.writeInt(memoria_diasSemanaRiego,diasSemanaRiego);
EEPROM.commit(); // Guarda los cambios en la memoria EEPROM

enviar(String("Dias semana: " + obtenerDiasCompleto() + "("+
EEPROM.readInt(memoria_diasSemanaRiego)+")");
}
//.....

```

```

else if (message=="db"){ //ver días de riego
rtc.adjust(DateTime(200+anno,mes, diames, hora, minutos, segundos));
enviar(String("minutos reloj: ") + valor + " " + fechaHoraFormateada());
enviar(String("Dias semana: " + obtenerDiasCompleto() + "("+
EEPROM.readInt(memoria_diasSemanaRiego)+")"));
}
//.....
else if (message=="dd"){ //días de riego en binario
char *p; //par pasar de binario a byte
long decimalValue = strtol(valor.c_str(), &p, 2);
diasSemanaRiego = (byte)decimalValue;
EEPROM.writeInt(memoria_diasSemanaRiego,diasSemanaRiego);
EEPROM.commit(); // Guarda los cambios en la memoria EEPROM

enviar(String("Dias semana: " + obtenerDiasCompleto() + "("+
EEPROM.readInt(memoria_diasSemanaRiego)+")"));
}
//.....
else if (message=="dr"){ //días de riego dr. d l m x j v s

int nElements = sizeof(diasDeLaSemana)/sizeof(diasDeLaSemana[0]); //tamaño
int n= existeEnArray(diasDeLaSemana, nElements, valor); //comprueba si existe dia
if (n != 10) {
bitSet(diasSemanaRiego, n); //pone a uno ese dia
}

EEPROM.writeInt(memoria_diasSemanaRiego,diasSemanaRiego);
EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
enviar( " Dias Semana: " + obtenerDiasCompleto()+ " (" +
cadenaByteInBinary(diasSemanaRiego)+ ")");
}

//.....
// PONER ENTRAR datos inicio riego r,h,m,s ir.15,10,26
else if (message=="ir"){ //hora,minuto,segundo de inicio riego

//SEPARA POR , h;m;s
for (int i = 0; i < 3 ; i++) {
int index = valor.indexOf(",");
data1[i] = valor.substring(0, index);.toInt();
valor = valor.substring(index + 1);
}
horaRiego = data1[0].toInt();
minutoRiego = data1[1].toInt();
segundoRiego= data1[2].toInt();

EEPROM.writeInt(memoria_horaRiego,horaRiego) ;
EEPROM.writeInt(memoria_minutoRiego,minutoRiego);

```

```

EEPROM.writeInt(memoria_segundoRiego,segundoRiego);

EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
texto = "Actual: " + fechaHoraFormatearda();
texto += "\n Fecha de Riego: " + String(horaRiego) + ":" +String(minutoRiego)+":"
+String(segundoRiego);
texto += "\n Fecha de fin de Riego: " + String(horaFinRiego) + ":"
+String(minutoFinRiego)+":" +String(segundoFinRiego);
texto += "\n Dias de riego" + obtenerDiasCompleto()+ " (" +
cadenaByteInBinary(diasSemanaRiego)+ ")" ;
enviar(texto);
}
//.....
// PONER ENTRAR datos inicio riego fr.h,m,s  ifr.15,10,26
else if (message=="fr"){ //hora,minuto,segundo de inicio riego

//SEPARA POR , h;m;s
for (int i = 0; i < 3 ; i++) {
    int index = valor.indexOf(",");
    data1[i] = valor.substring(0, index);//.toInt();
    valor = valor.substring(index + 1);
}
horaFinRiego = data1[0].toInt();
minutoFinRiego = data1[1].toInt();
segundoFinRiego= data1[2].toInt();

EEPROM.writeInt(memoria_horaFinRiego,horaFinRiego) ;
EEPROM.writeInt(memoria_minutoFinRiego,minutoFinRiego);
EEPROM.writeInt(segundoFinRiego,segundoFinRiego);

EEPROM.commit(); // Guarda los cambios en la memoria EEPROM
texto = "Actual: " + fechaHoraFormatearda();
texto += "\n Fecha de Riego: " + String(horaRiego) + ":" +String(minutoRiego)+":"
+String(segundoRiego);
texto += "\n Fecha de fin de Riego: " + String(horaFinRiego) + ":"
+String(minutoFinRiego)+":" +String(segundoFinRiego);
texto += "\n Dias de riego" + obtenerDiasCompleto()+ " (" +
cadenaByteInBinary(diasSemanaRiego)+ ")" ;
enviar(texto);
}
//.....
/* PONER ENTRAR RELOJ f.D,M,A,h,m,s */
else if (message=="f"){ //Pone fecha f.D,M,A,h,m,s

//SEPARA POR : D,M,A,h,m
Serial.println(valor);
for (int i = 0; i < 6 ; i++) {

    int index = valor.indexOf(",");

```

```

        data1[i] = valor.substring(0, index);//.toInt();
        valor = valor.substring(index + 1);

    }
    diames = data1[0].toInt();
    mes = data1[1].toInt();
    anno = data1[2].toInt();
    hora = data1[3].toInt();
    minutos= data1[4].toInt();
    segundos= data1[5].toInt();
    EEPROM.writeInt(memoria_segundos, segundos);
    EEPROM.writeInt(memoria_minutos, minutos);
    EEPROM.writeInt(memoria_hora, hora);
    EEPROM.writeInt(memoria_dia,diames);
    EEPROM.writeInt(memoria_mes, mes);
    EEPROM.writeInt(memoria_anno, anno);

    EEPROM.commit(); // Guarda los cambios en la memoria EEPROM

    rtc.adjust(DateTime(2000+anno,mes, diames, hora, minutos, segundos));
    enviar(fechaHoraFormatearda());
}

else if (message == "inicio") { //información comandos
    inicio();
}

else{ //ninguna orden valida
    Serial.println("error");
    enviar(String("Error: ") + message);
}
}

}

void setup() {
    Serial.begin(9600);
    EEPROM.begin(logitud_memoria);
    pinMode(Bomba, OUTPUT);
    digitalWrite(Bomba, LOW);
    // carga valores nuevos -1 SI NO ESTA INICIADA ESA POSICIÓN DE MEMORIA HASTA
    255Bit DE MEMORIA

    if ( EEPROM.readInt(memoria_diasSemanaRiego) < 0) {
    EEPROM.writeInt(memoria_diasSemanaRiego, diasSemanaRiego);} //tiempo encendido bomba 1m.
    // if ( EEPROM.readInt(memoria_ndias_riego) < 0) { EEPROM.writeInt(memoria_ndias_riego,
    dia);}

```

```

if ( EEPROM.readInt(memoria_anno) < 0) { EEPROM.writeInt(memoria_anno, anno);}
if ( EEPROM.readInt(memoria_mes) < 0) { EEPROM.writeInt(memoria_mes, mes);}
if ( EEPROM.readInt(memoria_dia) < 0) { EEPROM.writeInt(memoria_dia, diames);}
if ( EEPROM.readInt(memoria_hora) < 0) { EEPROM.writeInt(memoria_hora, hora);}
if ( EEPROM.readInt(memoria_minutos) < 0) { EEPROM.writeInt(memoria_minutos, minutos);}
if ( EEPROM.readInt(memoria_segundos) < 0) { EEPROM.writeInt(memoria_segundos,
segundos);}

//if ( EEPROM.readInt(memoria_dia_riego) < 0) { EEPROM.writeInt(memoria_dia_riego,
dia_riego);}
if ( EEPROM.readInt(memoria_horaRiego) < 0) { EEPROM.writeInt(memoria_horaRiego,
horaRiego);}
if ( EEPROM.readInt(memoria_minutoRiego) < 0) { EEPROM.writeInt(memoria_minutoRiego,
minutoRiego);}
if ( EEPROM.readInt(memoria_segundoRiego) < 0) { EEPROM.writeInt(memoria_segundoRiego,
segundoRiego);}

if ( EEPROM.readInt(memoria_horaFinRiego) < 0) { EEPROM.writeInt(memoria_horaFinRiego,
horaFinRiego);}
if ( EEPROM.readInt(memoria_minutoFinRiego) < 0) {
EEPROM.writeInt(memoria_minutoFinRiego, minutoFinRiego);}
if ( EEPROM.readInt(memoria_segundoFinRiego) < 0) {
EEPROM.writeInt(memoria_segundoFinRiego,segundoFinRiego);}

EEPROM.commit(); // Guarda los cambios en la memoria EEPROM

//LEE VALOR DE MEMORIA Y PONE EN HORA RELOJ INTERNO
segundos= EEPROM.readInt(memoria_segundos);
minutos= EEPROM.readInt(memoria_minutos);
hora=EEPROM.readInt(memoria_hora) ;
diames= EEPROM.readInt(memoria_dia);
mes= EEPROM.readInt(memoria_mes);
anno= EEPROM.readInt(memoria_anno);

diasSemanaRiego = EEPROM.readInt(memoria_diasSemanaRiego);

horaRiego= EEPROM.readInt(memoria_horaRiego);
minutoRiego= EEPROM.readInt(memoria_minutoRiego);
segundoRiego= EEPROM.readInt(memoria_segundoRiego);

horaFinRiego= EEPROM.readInt(memoria_horaFinRiego);
minutoFinRiego= EEPROM.readInt(memoria_minutoFinRiego);
segundoFinRiego= EEPROM.readInt(memoria_segundoFinRiego);
delay(1000);

if (!rtc.begin()) {
Serial.println(F("Couldn't find RTC"));

```

```

while (1);
}

Serial.print("Connecting Wifi: ");
Serial.println(ssid);

WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for api.telegram.org

while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}
//PARA DEPURAR
Serial.println("");
Serial.println("WiFi connected");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
//PARA DEPURAR

enviar( "ESP32TelegramTemporizadorRiegoDS3231: " + String(WiFi.localIP()));

if (rtc.lostPower()) {
  rtc.adjust(DateTime(2025, 13, 8, 12, 0, 0)); // Ejemplo de ajuste manual
} inicio();

}

//.....
void loop() {
  DateTime now = rtc.now();

  if (estadoBomba == false && automaticoBomba== true && isScheduledON(now)) // Apagado y
  debería estar encendido
  {
    digitalWrite(Bomba, HIGH);
    estadoBomba = true;
    texto= String("Riego Activado") + String("\n") +String( verHoras());
    enviar(texto);
  }
  else if (estadoBomba == true && automaticoBomba== true && !isScheduledON(now)) // Encendido
  y debería estar apagado
  {
    digitalWrite(Bomba, LOW);
    estadoBomba = false;
    texto= String("Riego Desactivado") + String("\n") +String( verHoras());
  }
}

```



```

    enviar(texto);

}

//-----telegram-----
if (millis() > lastTimeBotRan + botRequestDelay) {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

    while(numNewMessages) {
        Serial.println("Mensaje recibido"); //PARA DEPURAR
        handleNewMessages(numNewMessages);
        numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
}

delay(1000);
}

```