

```

// Aquest programa està adreçat als clients de FADISHOP que hagin adquirit
// la tarja FADICLOCK en alguna de les seves versions http://www.fadishop.eu
// Aquest programa serveix de referència per fer modificacions. ARDUINO UNO REV2
// Per a que resulti entenedor no està optimitzat. Es treballa a nivell de byte i bit.
// Serveix per comprovar alarma_1 i temperatura per la pantalla de degug.
// Properament serà revisat i millorat. ARDUINO ALPHA VERSIÓ 0023.
// Siusplau no suprimeixi aquestes línies i afegeixi el seu nom o entitat sota.
// AUTOR ORIGINAL: LLEONARD GARCÍA I LLOP 30 abril 2012 http://www.fadishop.eu
// AUTORITZATS: vosté
#include "Wire.h"
// Variables generals
int I2C_DS3232=0xD0 >> 1; // Adreça perifèric I2C sense LSB(read/write).
byte DS3232_00, DS3232_01, DS3232_02, DS3232_03, DS3232_04, DS3232_05, DS3232_06;
byte DS3232_07, DS3232_08, DS3232_09, DS3232_0A;
byte DS3232_0B, DS3232_0C, DS3232_0D;
byte DS3232control=0b00011100, DS3232status=0b11001000 ,DS3232aging ,DS3232Msb ,DS3232Lsb;
int command = 0; // This is the command char, in ascii form, sent from the serial port
int z=0;
bool restaurar_alarmes=0;
// CONFIGURACIÓ HORA EN CLAR //////////////////////////////////////
byte SegonsRtc=50; // RTC: segons en decimal.
byte MinutsRtc=30; // RTC: minuts en decimal.
byte HoraRtc=18; // RTC: hora en decimal.
byte DiaMesRtc=1; // RTC: Dia mensual en decimal.
byte MesRtc=5; // RTC: Mes en decimal.
byte AnyRtc=12; // RTC: Mes en decimal.
byte DiaSemRtc=1; // RTC: Dia setmanal en decimal.
bool ModeAmpmRtc=false; // RTC: false:24h, true:12h PM-AM.
////////////////////////////////////
// CONFIGURACIÓ DE L'ALARMA_1 EN CLAR //////////////////////////////////////
byte SegonsAlarma1=10; // Alarma1: segons en decimal.
byte MinutsAlarma1=31; // Alarma1: minuts en decimal.
byte HoraAlarma1=18; // Alarma1: hora en decimal.
byte DiaMesAlarma1=8; // Alarma1: Dia mensual en decimal.
byte MesAlarma1=5; // Alarma1: Mes en decimal.
byte DiaSemAlarma1=1; // Alarma1: Dia setmanal en decimal.
bool ModeDiaAlarma1=true; // Alarma1: false:dia mensual, true:dia setmanal.
bool ModeAmpmAlarma1=false; // Alarma1: false:24h, true:12h PM-AM.
int reactibleAlarma1=0; // Alarma1: reactibilitat: (0 un cop en data/hora/minut/segon).
// Alarma1: reactibilitat: (1 cada hora/minut/segon).

```

```

// Alarma1: reactibilitat: (2 cada en minut/segon).
// Alarma1: reactibilitat: (3 cada en segons).
//////////////////////////////////// Alarma1: reactibilitat: (4 generador de señal de 1 segon).
////////////////////////////////////
// CONFIGURACIÓ DE L'ALARMA_2 EN CLAR //////////////////////////////////////
byte MinutsAlarma2=31; // Alarma2: minuts en decimal.
byte HoraAlarma2=18; // Alarma2: hora en decimal.
byte DiaMesAlarma2=8; // Alarma2: Dia mensual en decimal.
byte MesAlarma2=05; // Alarma2: Mes en decimal.
byte DiaSemAlarma2=01; // Alarma2: Dia setmanal en decimal.
bool ModeDiaAlarma2=true; // Alarma2: false:dia mensual, true:dia setmanal.
bool ModeAmpmAlarma2=false; // Alarma2: false:24h, true:12h PM-AM.
int reactibleAlarma2=0; // Alarma1: reactibilitat: (0 un cop en data/hora/minut).
// Alarma1: reactibilitat: (1 cada hora/minut).
// Alarma1: reactibilitat: (2 cada en minuts).
//////////////////////////////////// Alarma1: reactibilitat: (3 i 4 generador de señal de 1 minut).
//*****
byte LutBcdDec[128]={0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x0A,0x0B,0x0C,0x0D,0x0E,0x0F,0x10,0x11,0x12,0x13,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x14,0x15,0x16,0x17,0x18,0x19,0x1A,0x1B,0x1C,0x1D,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x1E,0x1F,0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x28,0x29,0x2A,0x2B,0x2C,0x2D,0x2E,0x2F,0x30,0x31,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x3A,0x3B,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x3C,0x3D,0x3E,0x3F,0x40,0x41,0x42,0x43,0x44,0x45,0x46,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x47,0x48,0x49,0x4A,0x4B,0x4C,0x4D,0x4E,0x4F,0x50,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
//*****
byte LutDecBcd[64]={0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x10,0x11,0x12,0x13,0x14,0x15,
0x16,0x17,0x18,0x19,0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28,0x29,0x30,0x31,
0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x40,0x41,0x42,0x43,0x44,0x45,0x46,0x47,
0x48,0x49,0x50,0x51,0x52,0x53,0x54,0x55,0x56,0x57,0x58,0x59,0x60,0x61,0x62,0x63};
//*****
byte AxM1[5]={0b00000000,0b00000000,0b00000000,0b00000000,0b10000000};
byte AxM2[5]={0b00000000,0b00000000,0b00000000,0b10000000,0b10000000};
byte AxM3[5]={0b00000000,0b00000000,0b10000000,0b10000000,0b10000000};
byte AxM4[5]={0b00000000,0b10000000,0b10000000,0b10000000,0b10000000};
//*****
byte BoolByte (byte a, bool b)
{
if ((bool)b == true){return a;
}
}

```

```

        else {return 0b00000000;
    }
}

//*****
byte FastDecBcd(int num){
//    return LutDecBcd[num&&0x3f];
    return LutDecBcd[num];
}

//*****
byte FastBcdDec(int gg){
//    return LutBcdDec[gg&&0x7f];
    return LutBcdDec[gg];
}

//*****
void CodificarRtc()
{
    DS3232_00=FastDecBcd(SegonsRtc&0x7f);
    DS3232_01=FastDecBcd(MinutsRtc&0x7f);
    DS3232_02=(FastDecBcd(HoraRtc&0x3f) | BoolByte(0b01000000,ModeAmpmRtc));
    DS3232_03=FastDecBcd(DiaSemRtc&0x07);
    DS3232_04=FastDecBcd(DiaMesRtc&0x3f);
    DS3232_05=FastDecBcd(MesRtc&0x1f);
    DS3232_06=FastDecBcd(AnyRtc&0x7f);
}

//*****
void DecodificarRtc()
{
    SegonsRtc=FastBcdDec(DS3232_00&0x7f);
    MinutsRtc=FastBcdDec(DS3232_01&0x7f);
    HoraRtc=(FastBcdDec(DS3232_02&0x3f));
    DiaSemRtc=FastBcdDec(DS3232_03&0x07);
    DiaMesRtc=FastBcdDec(DS3232_04&0x3f);
    MesRtc=FastBcdDec(DS3232_05&0x1f);
    AnyRtc=FastBcdDec(DS3232_06&0x7f);
    z=z+1;
}

//*****
void DecodificarAlarma1()
{
    DS3232_0A=FastBcdDec(DS3232_0A&0x3f);

```

```

        DS3232_09=FastBcdDec(DS3232_09&0x3f);
        DS3232_08=(FastBcdDec(DS3232_08&0x7f));
        DS3232_07=FastBcdDec(DS3232_07&0x7f);
    }
    //*****
void codificarAlarma1()
{
    DS3232_07=(FastDecBcd(SegonsAlarma1 & 0x7f) | AxM1[reactibleAlarma1]);
    DS3232_08=(FastDecBcd(MinutsAlarma1 & 0x7f) | AxM2[reactibleAlarma1]);
    DS3232_09=(FastDecBcd(HoraAlarma1 & 0x3f) | AxM3[reactibleAlarma1] | BoolByte(0b01000000,ModeAmpmAlarma1));
    if (ModeDiaAlarma1==false){
        DS3232_0A=(FastDecBcd(DiaMesAlarma1 & 0x3f) | AxM4[reactibleAlarma1] | BoolByte(0b01000000,ModeDiaAlarma1));}
    else{DS3232_0A=((FastDecBcd(DiaSemAlarma1& 0x07)) | AxM4[reactibleAlarma1]) | BoolByte(0b01000000,ModeDiaAlarma1);}
}
//*****
void codificarAlarma2()
{
    DS3232_0B=((FastDecBcd(MinutsAlarma2& 0x7f)) | (AxM2[reactibleAlarma2]));
    DS3232_0C=((FastDecBcd(HoraAlarma2& 0x3f)) | AxM3[reactibleAlarma2] | BoolByte(0b01000000,ModeAmpmAlarma2));
    if (ModeDiaAlarma2==false){
        DS3232_0D=((FastDecBcd(DiaMesAlarma2& 0x3f)) | AxM4[reactibleAlarma2] | BoolByte(0b01000000,ModeDiaAlarma2));}
    else{DS3232_0D=((FastDecBcd(DiaSemAlarma2& 0x07)) | AxM4[reactibleAlarma2]) | BoolByte(0b01000000,ModeDiaAlarma2);}
}
//*****
void ficarAlm1DS3232(){
    codificarAlarma1();
    delay(100);
    Wire.beginTransaction(I2C_DS3232); // START + adreça periféric I2C.
    Wire.send(0x07); // Adreça interna.
    Wire.send(DS3232_07); // Segons.
    Wire.send(DS3232_08); // Minuts.
    Wire.send(DS3232_09); // Hora.
    Wire.send(DS3232_0A); // Dia semanal/mensual.
    Wire.endTransmission(); // STOP.

    DS3232control=DS3232control | 0b00000100; // Pin INT per alarmes (INTCN=1).
    DS3232control=DS3232control | 0b00000001; // Habilita alarma_1 (A1IE=1) INT0 pin2 B.2.
    DS3232status=DS3232status & 0b11111110; // Borra flag alarma1 (A1IF=0).

    delay(100);

```

```

Wire.beginTransmission(I2C_DS3232);           // START + adreça perifèric I2C.
Wire.send(0x0E);                             // Adreça interna.
Wire.send (DS3232control);                   // Byte de control.
Wire.send (DS3232status);                    // Byte de control/status.
Wire.endTransmission();                      // STOP.

attachInterrupt(0,AtenAlarma,FALLING);       // Activada interrupció INT0 pin2 B.2. per nivell baix.
}
//*****
void ficarAlm2DS3232(){
  codificarAlarma2();
  delay(100);
  Wire.beginTransmission(I2C_DS3232);        // START + adreça perifèric I2C.
  Wire.send(0x0B);                           // Adreça interna.
  Wire.send (DS3232_0B);                      // Minuts.
  Wire.send (DS3232_0C);                      // Hora.
  Wire.send (DS3232_0D);                      // Dia semanal/mensual.
  Wire.endTransmission();                    // STOP.

  DS3232control=DS3232control | 0b00000100; // Pin INT per alarmes (INTCN=1).
  DS3232control=DS3232control | 0b00000010; // Habilita alarma_2 (A2IE=1) INT1 pin3 B.0.
  DS3232status=DS3232status & 0b11111101;  // Borra flag alarmal (A2IF=0).

  delay(100);
  Wire.beginTransmission(I2C_DS3232);        // START + adreça perifèric I2C.
  Wire.send(0x0E);                             // Adreça interna.
  Wire.send (DS3232control);                   // Byte de control.
  Wire.send (DS3232status);                    // Byte de control/status.
  Wire.endTransmission();                      // STOP.

  attachInterrupt(0,AtenAlarma,FALLING);     // Activada interrupció INT1 pin3 B.0. per nivell baix
}
//*****
void FicarEnHoraDs3232(){
  CodificarRtc();
  Wire.beginTransmission(I2C_DS3232);        // START + adreça perifèric I2C.
  Wire.send(0x00);                             // Adreça interna.
  Wire.send (DS3232_00);                       // Segons.
  Wire.send (DS3232_01);                       // Minuts.
  Wire.send (DS3232_02);                       // Hora.

```

```

Wire.send (DS3232_03);           // Dia de la setmana.
Wire.send (DS3232_04);           // Dia del mes.
Wire.send (DS3232_05);           // Mes.
Wire.send (DS3232_06);           // Any.
Wire.endTransmission();         // STOP.
}
//*****
void AtenAlarma()
{restaurar_alarmes=!restaurar_alarmes;
}

//*****
void ReactivarAlm1()
{delay(100);                     // Obviable. Cortesia anterior op I2C.
Wire.beginTransmission(I2C_DS3232); // START + adreça perifèric I2C.
Wire.send(0x0F);                 // Adreça interna control/estat.
Wire.endTransmission();         // STOP
Wire.requestFrom(I2C_DS3232, 1); // RESTART + @esclau +1byte + STOP
DS3232status=Wire.receive();    // Byte de control/estat.
delay(100);                     // Obviable. Cortesia anterior op I2C.
if ((DS3232status & 0b00000001) == 0b00000001){ // Flag alarmal=1? A1IF=1?
Serial.println(" alarma 1 ");
DS3232status=DS3232status & 0b11111110; // Restaura flag alarmal (A1IF=0).
Wire.beginTransmission(I2C_DS3232); // START + adreça perifèric I2C.
Wire.send(0x0F);                 // Adreça interna.
Wire.send (DS3232status);       // Byte de control/status.
Wire.endTransmission();        // STOP.
}
else {Serial.println(" no alarma 1 ");
}
}

//*****
void ReactivarAlm2()
{delay(100);                     // Obviable. Cortesia anterior op I2C.
Wire.beginTransmission(I2C_DS3232); // START + adreça perifèric I2C.
Wire.send(0x0F);                 // Adreça interna control/estat.
Wire.endTransmission();         // STOP
Wire.requestFrom(I2C_DS3232, 1); // RESTART + @esclau +1byte + STOP
DS3232status=Wire.receive();    // Byte de control/estat.
delay(100);                     // Obviable. Cortesia anterior op I2C.
}

```

```

    if ((DS3232status & 0b00000010) == 0b00000010){; // Flag alarma=1? A1IF=1?
        Serial.println(" alarma 2 ");
        DS3232status=DS3232status & 0b11111101; // Restaura flag alarma (A1IF=0).
        Wire.beginTransmission(I2C_DS3232); // START + adreça perifèric I2C.
        Wire.send(0x0F); // Adreça interna.
        Wire.send(DS3232status); // Byte de control/status.
        Wire.endTransmission(); // STOP.
    }
    else {Serial.println(" no alarma 2 ");
    }
}
}
}
//*****
void AraAvuiDs3232(){
    delay(100);
    Wire.beginTransmission(I2C_DS3232); // START + adreça perifèric I2C.
    Wire.send(0x00); // Adreça interna
    Wire.endTransmission(); // STOP
    Wire.requestFrom(I2C_DS3232, 16); // RESTART + @esclau +7bytes + STOP
    DS3232_00=Wire.receive(); // Segons.
    DS3232_01=Wire.receive(); // Minuts
    DS3232_02=Wire.receive(); // Hora
    DS3232_03=Wire.receive(); // Dia de la setmana
    DS3232_04=Wire.receive(); // Dia del mes
    DS3232_05=Wire.receive(); // Mes
    DS3232_06=Wire.receive(); // Any
    DS3232_07=Wire.receive(); // ALARMA_1 TEST
    DS3232_08=Wire.receive(); // ALARMA_1 TEST
    DS3232_09=Wire.receive(); // ALARMA_1 TEST
    DS3232_0A=Wire.receive(); // ALARMA_1 TEST
    DS3232_0B=Wire.receive(); // ALARMA_2 TEST
    DS3232_0C=Wire.receive(); // ALARMA_2 TEST
    DS3232_0D=Wire.receive(); // ALARMA_2 TEST
    DS3232control=Wire.receive(); // CONTROL TEST
    DS3232status=Wire.receive(); // STATUS TEST
    DecodificarRtc();
// DecodificarAlarma1();
    Serial.print("num:");
    Serial.print(z, DEC);
    Serial.print(" ");
    Serial.print(HoraRtc, DEC);

```

```

Serial.print(":");
Serial.print(MinutsRtc, DEC);
Serial.print(":");
Serial.print(SegonsRtc, DEC);
Serial.print(" ");
Serial.print(DiaMesRtc, DEC);
Serial.print("/");
Serial.print(MesRtc, DEC);
Serial.print("/");
Serial.print(AnyRtc, DEC);
Serial.print(" ");
Serial.print(DS3232control, DEC);
Serial.print(" ");
Serial.print(DS3232status, DEC);
Serial.print(" a");
Serial.print(DS3232_07, DEC);
Serial.print(" a");
Serial.print(DS3232_08, DEC);
Serial.print(" a");
Serial.print(DS3232_09, DEC);
Serial.print(" a");
Serial.print(DS3232_0A, DEC);
Serial.print(" b");
Serial.print(DS3232_0B, DEC);
Serial.print(" b");
Serial.print(DS3232_0C, DEC);
Serial.print(" b");
Serial.print(DS3232_0D, DEC);
Serial.print(" ");
Serial.print(DS3232Msb, DEC);
Serial.print(" ");
Serial.print(DS3232Lsb, DEC);
Serial.print("\n");
}
//*****
void setup() {
  Wire.begin();
  Serial.begin(9600);
  FicarEnHoraDs3232();
  ficarAlm1DS3232();
}

```



```

ficarAlm2DS3232();
Serial.print("\nCICLE      HORA      DATA      control      status      ALARMES      TEMPERATURA\n");
}
//*****
void loop() {
  SollicitarTemp(); // Sol·licita temperatura.
  AraAvuiDs3232(); // LLegeix data-hora.
  TemperaturaDs3232(); // Llegix temperatura encomanada.
  if ((bool)restaurar_almes == true) { // Semafor. true-> cal restaurar.
    restaurar_almes=!restaurar_almes;
    ReactivarAlm1();
    ReactivarAlm2();
  }

  delay(413); // Espera.
}
//*****
void SollicitarTemp()
{delay(100); // Obviable. Cortesia anterior op I2C.
  Wire.beginTransmission(I2C_DS3232); // START + adreça perifèric I2C.
  Wire.send(0x0F); // Adreça interna control/estat.
  Wire.endTransmission(); // STOP
  Wire.requestFrom(I2C_DS3232, 1); // RESTART + @esclau +1bytes + STOP
  DS3232status=Wire.receive(); // Byte de control/estat
  if (( DS3232status & 0b00000100)== 0b00000000) // Bit BSY=0 de control/estat? Conversió esdevinguda?
    {DS3232control=DS3232control | 0b00100000; // (SI)Forçar una conversió (CONV=1).
    delay(100); // (SI)Obviable. Cortesia anterior op I2C.
    Wire.beginTransmission(I2C_DS3232); // (SI)START + adreça perifèric I2C.
    Wire.send(0x0E); // (SI)Adreça interna.
    Wire.send (DS3232control); // (SI)Byte de control.
    Wire.endTransmission(); // (SI)STOP.
    } // (NO)Deixar escapar. No sol·licitar conversió..
}
//*****
void TemperaturaDs3232()
{delay(100); // Obviable. Cortesia anterior op I2C.
  Wire.beginTransmission(I2C_DS3232); // START + adreça perifèric I2C.
  Wire.send(0x0F); // Adreça interna control/estat.
  Wire.endTransmission(); // STOP
  Wire.requestFrom(I2C_DS3232, 1); // RESTART + @esclau +1byte + STOP
  DS3232status=Wire.receive(); // Byte de control/estat.
}

```

```
if (( DS3232status & 0b00000100)== 0b00000000)// Bit BSY=0 de control/estat? Lliure?
{delay(100); // (SI)Obviaable. Cortesia anterior op I2C.
Wire.beginTransmission(I2C_DS3232); // (SI)START + adreça perifèric I2C.
Wire.send(0x11); // (SI)Adreça interna MSB TEMPERATURA
Wire.endTransmission(); // (SI)STOP
Wire.requestFrom(I2C_DS3232, 2); // (SI)RESTART + @esclau +2bytes + STOP
DS3232Msb=Wire.receive(); // (SI)Part entera.
DS3232Lsb=Wire.receive(); // (SI)Part decimal.
} // (NO)Deixar escapar. No llegir la temperatura.
}
//*****
//*****lleonardgarcía*****http://www.fadishop.eu*****
//*****
```