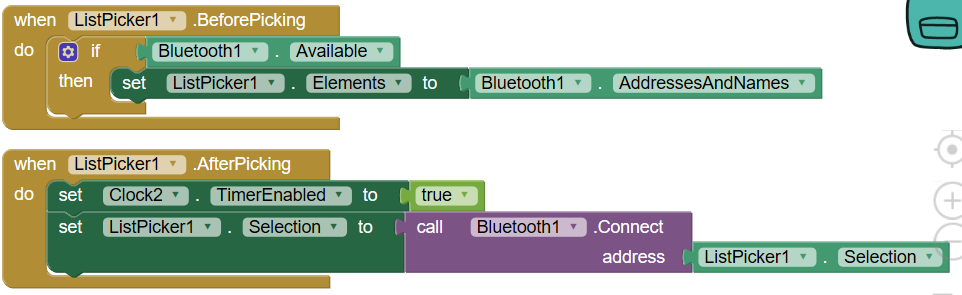
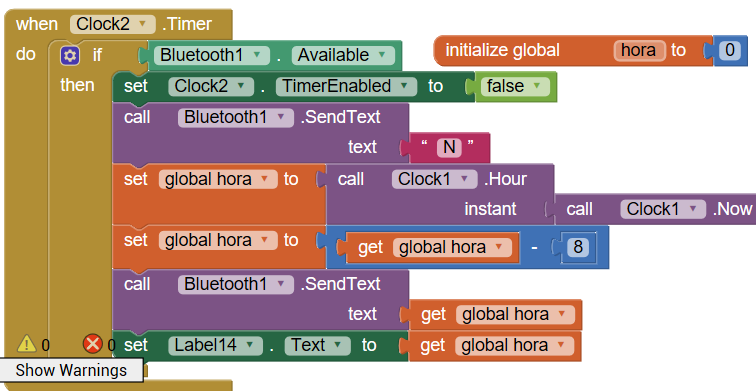
|  |  |
| --- | --- |
|  | Dos relojes  Clock 1 para verificar en cada intervalo si hay informacion que envia Arduino, intervalo 1 segundo (1000) siempre enable  Clock 2 inicialmente esta en disable   * Al conectar se da enable=true * Se espera 5 segundos (5000) * Si conecto a BT entonces 1)actualize hora 2) pide valores de sensors actuals |
|  |  |

CONNECT TO BLUETOOTH



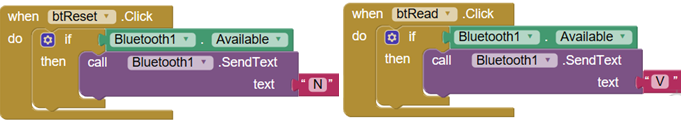
Los valores de la hora se decrementan en 8, asi solo se envia 1 CHAR, y luego Arduino le suma 8



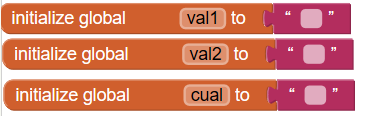
SEND TO COMMANDS TO ARDUINO, SOLO SE MANDA CHAR

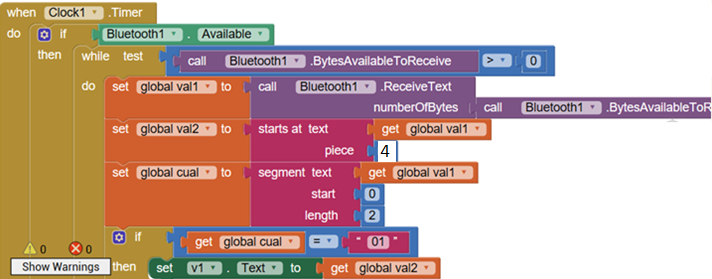
Send “V” para que envie lecturas historiacas de sensors

Send “N” para que envie lecturas actuales de sensores



Checking incoming messages from Arduino





Para los valores actuales el valor es “99”

#include <TimeAlarms.h>

#include <Time.h>

#include <SoftwareSerial.h>

SoftwareSerial BT1(4, 2); // RX | TX

int MaxAddr = 20;

String lectura[20];

bool test = false;

String FromSerial = "";

int SENSOR1 = 6;

int SENSOR2 = 7;

int SENSOR3 = 8;

int SENSOR4 = 9;

int Addr = 0;

int AA1 = A1;

int AA2 = A2;

int AA3 = A3;

int AA4 = A4;

void setup()

{

pinMode(AA1, INPUT);

pinMode(AA2, INPUT);

pinMode(AA3, INPUT);

pinMode(AA4, INPUT);

pinMode(SENSOR1, OUTPUT);

pinMode(SENSOR2, OUTPUT);

pinMode(SENSOR3, OUTPUT);

pinMode(SENSOR4, OUTPUT);

//setTime(8,29,0,1,1,11); // set time to Saturday 8:29:00am Jan 1 2011

//Alarm.alarmRepeat(8,30,0, MorningAlarm);

BT1.begin(9600); //Iniciar el BT1

Serial.begin(9600);

for (int i=0; i< MaxAddr; i++)

lectura[i]="-";

lectura[0]="0 0 0 0 ";

lectura[1]="0 0 0 0 ";

lectura[2]="0 0 0 0 ";

lectura[3]="0 0 0 0 ";

AlarmRepeat(7,59);

}

void loop(){

Alarm.delay(1000); // IMPORTANTE !!!!! espere 1 segundo para que funcione alarma

// INTERACTUA por el serial de ARDUINO

//================================

if (Serial.available() )

{

char c1= Serial.read();

FromSerial += c1;

if (FromSerial =="h" || FromSerial=="H")

{

int hh = hour();

int mm = minute();

int ss = second();

Serial.println(String(hh)+ " "+ String(mm)+ " "+ String(ss));

FromSerial = "";

}

if (FromSerial =="g" || FromSerial=="G")

{

SendSerialData();

FromSerial="";

return;

}

if (FromSerial> "23")

{

FromSerial="";

Serial.println("Vuelva a escribir comando");

return;

}

if (FromSerial.length()==2)

{

int hora2 = FromSerial.toInt();

Serial.println("Hora Fijada a las "+ String(hora2));

AlarmRepeat(hora2, 59);

FromSerial="";

}

}

// Interactua por el Bluetooth

//======================

if (BT1.available()>=1)

{

int entrada = BT1.read(); //Leer un caracter en ascii

if (test) Serial.println(entrada);

if (entrada == 86) SendValues(); // Sent 'V' 86 ascii, send all values stored

if (entrada == 78) GetNow(); // Sent 'N' 78 ascii, send current measures

if (entrada == 86 || entrada == 78) return;

int hora = entrada - 40; // ascii 49 =1 50 =2

//if (Addr==0) SetAddress(hora);

AlarmRepeat(hora,0);

BT1.print("Hora Set "+String(hora));

}

}

void AlarmRepeat(int hora, int minuto)

{

//setTime(6,0,0,1,1,11); // set time to Saturday 8:29:00am Jan 1 2011

// CADA VEZ QUE SETEA LA FECHA DEBE CREAR LAS ALARMAS

setTime(hora, minuto, 0, 1, 11, 11);

Alarm.alarmRepeat(8,0,0, SetValues); // 8 am

Alarm.alarmRepeat(11,0,0, SetValues); // 11 am

Alarm.alarmRepeat(15,0,0, SetValues); // 3 pm

Alarm.alarmRepeat(17,0,0, SetValues); // 5 pm

}

void SendSerialData()

{

for (int i=0; i< MaxAddr; i++)

Serial.println(String(i+1) + " - " +lectura[i]);

}

void SetAddress(int hora)

{

if (hora < 8)

{

Addr = 0;

return;

}

if (hora < 11)

{

Addr = 1;

return;

}

if (hora < 15)

{

Addr =2;

return;

}

Addr = 3;

}

// current values

void GetNow()

{

// encender sensores

SENSORES(1);

// read sensors

int value1 = abs(map(analogRead(AA1), 500,0,0,100));

int value2 = abs(map(analogRead(AA2), 500,0,0,100));

int value3 = abs(map(analogRead(AA3), 500,0,0,100));

int value4 = abs(map(analogRead(AA4), 500,0,0,100));

String valores = "099 " +String(value1)+ " "+ String(value2)+ " " + String(value3)+ " " + String(value4);

if (test) Serial.println(valores);

BT1.print(valores);

// apagar sensores

SENSORES(0);

}

void SENSORES(int valor)

{

// 0 apagar sensores

// 1 encender sensores

if (valor==0) delay(500);

digitalWrite(SENSOR1, valor);

digitalWrite(SENSOR2, valor);

digitalWrite(SENSOR3, valor);

digitalWrite(SENSOR4, valor);

if (valor==1) delay(2000);

}

void SetValues()

{

// encender sensores

SENSORES(1);

int value1 = abs(map(analogRead(AA1), 500,0,0,100));

int value2 = abs(map(analogRead(AA2), 500,0,0,100));

int value3 = abs(map(analogRead(AA3), 500,0,0,100));

int value4 = abs(map(analogRead(AA4), 500,0,0,100));

String valores = String(value1)+ " "+ String(value2)+ " " + String(value3)+ " " + String(value4);

Serial.println(valores);

if (Addr >= MaxAddr)

{

for (int i=0; i< MaxAddr-1; i++)

lectura[i] = lectura[i+1];

lectura[MaxAddr-1] = "-";

Addr = Addr -1;

}

//Serial.println((String)Addr+ " "+valores);

lectura[Addr] = valores;

Addr++;

// APAGAR SENSORES

SENSORES(0);

}

void SendValues()

{

int espere= 1100;

for (int i=0; i< MaxAddr; i++)

{

int uno = i/4;

double dos = i;

dos = (dos/4-(double)uno)\*4+1;

int dos2 = dos;

String posi = String(uno) + String(dos2);

String SendVal = posi + " " + lectura[i];

if (test) Serial.println("0"+ SendVal+" ");

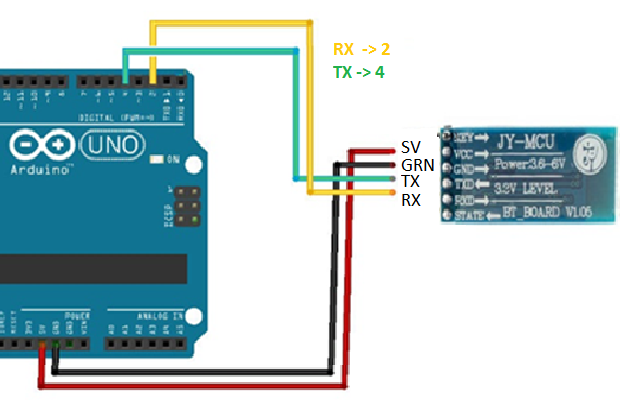
BT1.print("0"+ SendVal+" ");

delay(espere);

}

GetNow();

}

}

SoftwareSerial BT1(4, 2); // RX | TX

El VCC verifique que no es 3.5 V sino entre 4.5 -5.0 v