

EcoDuino

Kit automatico de plantación



Contenidos

- [1 Introducción](#)
- [2 Especificaciones](#)
 - [2.1 Documentos](#)
- [3 Diagrama](#)
 - [3.1 wiki de Sensores](#)
- [4 Código de ejemplo](#)
 - [4.1 Lea los valores del sensor](#)
 - [4.2 Resultado](#)
 - [4.3 Pruebe la bomba de agua](#)
 - [4.4 Ejemplo de riego automático de flores](#)
- [5 FAQ](#)
- [6 Mas documentación](#)

Introduction

Introducción

EcoDuino está evolucionando. Ahora el EcoDuino tiene una nueva carcasa. Está protegido de las salpicaduras de agua, por lo que es seguro utilizarlo junto a sus plantas. El EcoDuino ahora tiene un Atmega32U4 que elimina la necesidad de un adaptador. Los bocetos se pueden subir simplemente a través de Mirco USB, igual que Arduino Leonardo. Otra mejora es que ahora el sensor DS18B20 está directamente soportado.

EcoDuino está diseñado por DFRobot para ayudarle a cultivar plantas. Utilizando una serie de microcontroladores, sensores y actuadores, el sistema EcoDuino puede hacer que sus esfuerzos para cultivar plantas sean mucho más fáciles.

En este sistema, los sensores se utilizan para recoger datos que pueden mostrar las condiciones de la planta como la temperatura, humedad, intensidad de la luz, etc...

Si lo desea, EcoDuino puede enviarle un mensaje y decirle cómo van sus plantas a través de las comunicaciones inalámbricas. También regará sus plantas automáticamente cuando tengan sed, o en un intervalo predeterminado.

Lo bueno del EcoDuino es que está desarrollado en base a Arduino, lo que significa que no sólo puedes programar EcoDuino en el entorno IDE de Arduino, sino que también puedes usar cualquier hardware compatible con Arduino en tu sistema EcoDuino.

Advertencia: Los cables empaquetados con los sensores no son correctos, le sugerimos que utilice los cables anaranjados adjuntos.

Especificaciones

- Fuente de alimentación de la placa: 6~12V DC
- Bootloader: leonardo
- 4 puertos de E/S analógicas, 5 puertos de E/S digitales
- Terminal para la conexión de una varilla de carbono (sensor de humedad del suelo)
- Terminal para interconectar un sensor de temperatura DS18B20 (sensor de temperatura del suelo)
- Borne para la conexión de un motor o de una electroválvula (solenoides)
- Potenciómetro para ajustar el valor umbral de humedad del suelo del riego
- Ranura Xbee
- Mirco USB
- Terminal de tornillos de 3.5mm
- Dimensiones de la placa: 75 x 50 mm
- Alimentación de la bomba de agua: 4.5~12V DC
- Altura de bombeo: 200cm
- Capacidad de caudal: 100-350L/H

- Rango de potencia:: 0.5W-5W
- Dimensiones de la bomba: 38x38x29mm

Documentos

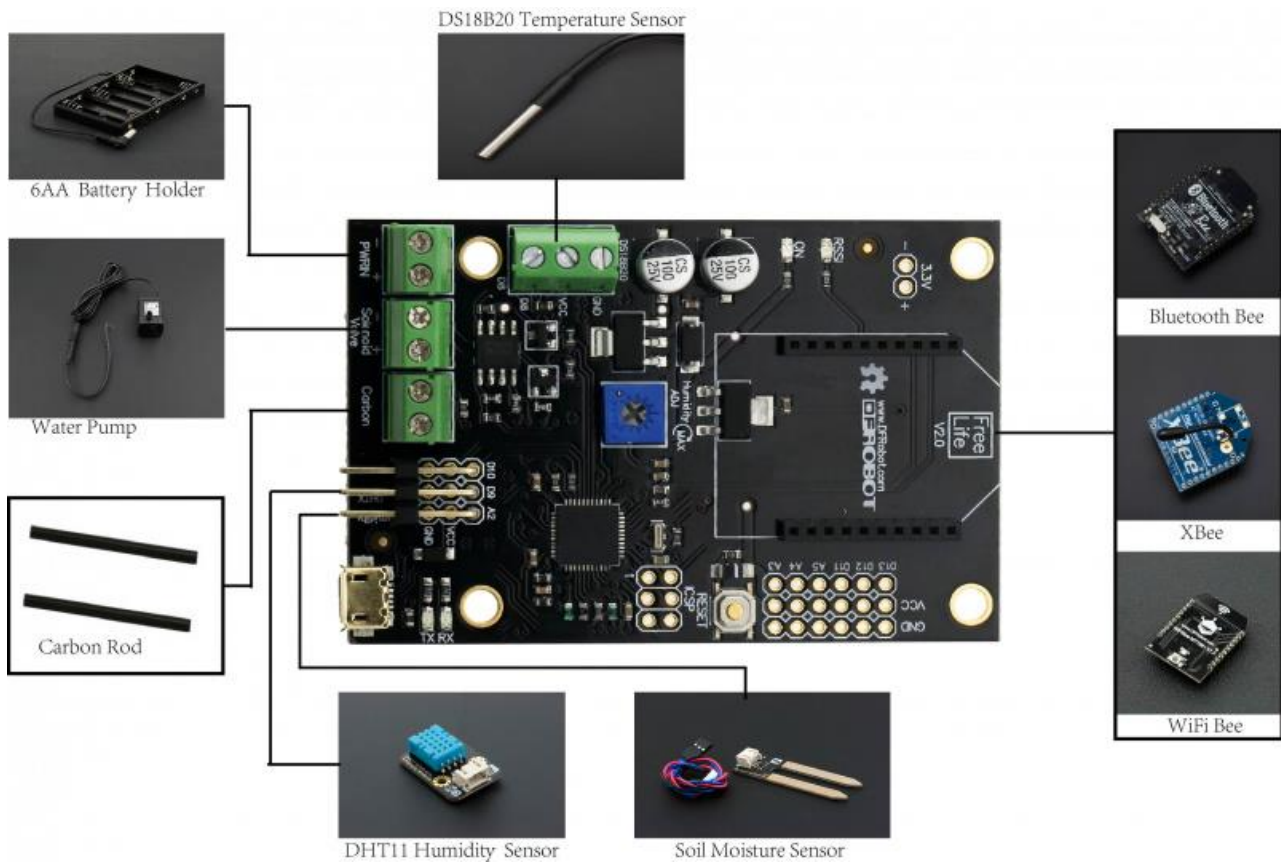
- [DHT11 Datasheet](#)
- [Esquematicos](#)
- [DHT11 libreria](#)

Por favor, instale las librerías antes de probar los códigos de ejemplo

Diagrama



- Este kit no contiene carbono, XBEE, DS18B20.



Esquema general

- Sensor de humedad del suelo: cable azul (A2), cable rojo (VCC), cable negro (GND)
- Sensor de humedad DHT11: cable verde (D9), cable rojo (VCC), cable negro (GND)
- Bomba: cable marrón (+), cable azul (-)
- Soporte de la batería: cable rojo (+), cable negro (-)
- El potenciómetro azul está conectado al pin A1 de la placa de control principal. El usuario puede leer el valor y ajustar el umbral para el riego automático.

Wiki de Sensores

- [Sensor de humedad del suelo](#)
- [Sensor de humedad DHT11](#)
- [Sensor de temperatura DS18B20](#)

Código de ejemplo

Leer los valores del sensor

```
#include <dht11.h>
dht11 DHT;
#define MOISTURE_PIN A2  /soil Moisture sensor/
#define DHT11_PIN      9  //DHT11

int airHumidity;  //environment humidity
int airTemperature;  // environment temperature
int soilHumidity;  //soil moisture

void setup(){
  Serial.begin(9600);
}

void loop(){
  int chk;
  chk = DHT.read(DHT11_PIN);  //Read Data
  switch (chk){
    case DHTLIB_OK:
      Serial.print("OK,\t");
      break;
    case DHTLIB_ERROR_CHECKSUM:
      Serial.print("Checksum error,\t");
      break;
    case DHTLIB_ERROR_TIMEOUT:
      Serial.print("Time out error,\t");
      break;
    default:
      Serial.print("Unknown error,\t");
      break;
  }
  airHumidity=DHT.humidity;
  airTemperature=DHT.temperature;
  soilHumidity=analogRead(MOISTURE_PIN);
}
```

```

Serial.print("airHumidity:");
Serial.print(airHumidity);
Serial.print(",\t");
Serial.print("airTemperature:");
Serial.print(airTemperature);
Serial.print(",\t");
Serial.print("soilHumidity:");
Serial.println(soilHumidity);

delay(1000);
}

```

Resultados

Abra el monitor del Puerto serial, Baud rate: 9600.

```

COM26 (Arduino Leonardo)
OK,   airHumidity:66, airTemperature:26,   soilHumidity:0
OK,   airHumidity:66, airTemperature:26,   soilHumidity:0
OK,   airHumidity:66, airTemperature:26,   soilHumidity:0
OK,   airHumidity:66, airTemperature:26,   soilHumidity:0
OK,   airHumidity:66, airTemperature:26,   soilHumidity:0
OK,   airHumidity:66, airTemperature:26,   soilHumidity:62
OK,   airHumidity:66, airTemperature:26,   soilHumidity:273
OK,   airHumidity:66, airTemperature:26,   soilHumidity:268
OK,   airHumidity:66, airTemperature:26,   soilHumidity:268
OK,   airHumidity:66, airTemperature:26,   soilHumidity:265
OK,   airHumidity:66, airTemperature:26,   soilHumidity:268
OK,   airHumidity:66, airTemperature:26,   soilHumidity:274
OK,   airHumidity:66, airTemperature:26,   soilHumidity:290
OK,   airHumidity:66, airTemperature:26,   soilHumidity:0
OK,   airHumidity:66, airTemperature:26,   soilHumidity:0

```

Prueba la bomba de agua

```

void setup() {
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

```

```

    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
}

void loop() {
    pumpOn();
    delay(1000);
    pumpOff();
    delay(1000);
}
//open pump
void pumpOn()
{
    digitalWrite(5, HIGH);
    digitalWrite(6, HIGH);
}
//close pump
void pumpOff()
{
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
}

```

Ejemplo de riego automático de flores

Nota: Este programa de ejemplo no utiliza DHT11. Puede ajustar el umbral para encender y apagar la bomba. Cuando la humedad del suelo es inferior al umbral, encenderá la bomba.

```

#define MOISTURE_PIN A2

int soilHumidity;
int setHumidity = 50;      //Set the pump trigger threshold
void setup() {
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);

    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
}

void loop() {
    soilHumidity = map(analogRead(MOISTURE_PIN), 0, 1023, 0, 100);    //Map
    analog value to 0~100% soil moisture value
    if (soilHumidity < setHumidity) {
        pumpOn();
    } else {
        pumpOff();
    }
}

```

```
//open pump
void pumpOn() {
    digitalWrite(5, HIGH);
    digitalWrite(6, HIGH);
}
//close pump
void pumpOff() {
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
}
```

FAQ

Para cualquier pregunta, consejo o ideas interesantes para compartir, por favor visite el sitio web de [DFRobot Forum](#).