Arduino KY-015 Temperature and humidity sensor module

From

http://www.dx.com/p/arduino-digital-temperature-humidity-sensor-module-121350

Contents

Digital Temperature and Humidity Sensor Module

The module related presentations

DHT11 digital temperature and humidity sensor is a calibrated digital signal output temperature and humidity combined sensor, which Application-specific modules capture technology and digital temperature and humidity sensor technology to ensure that products with high reliability and excellent Long-term stability. The product has excellent quality, fast response, anti-interference ability, high cost and other advantages. Single Wire serial interface that allows quick and easy system integration. Ultra-small size, low power consumption, signal transmission distance Up to 20 meters, making it to the class of applications and even the most demanding applications is the best choice. Products for the 4-pin single row Pin package, easy connection.

Specifications

- Supply voltage: 3.3 ~ 5.5V DC
- Output: single-bus digital signal
- Measuring range: Humidity 20-90% RH, Temperature 0 ~ 50 °C
- Accuracy: Humidity + -5% RH, temperature + -2 °C
- Resolution: Humidity 1% RH, temperature 1 °C
- Long-term stability: <± 1% RH / Year

Notes

1. to avoid the use of the condensation conditions
2. long-term storage temperature 10-40 °C, humidity below 60%
3. the use of power and ground connection to be correct, so as not to damage the sensor

Use

About DHT11 specific timing problems we can refer to its datasheet, see the following modules, such as our company He connected with the Arduino board

Schematic

- Arduino pin 8 --> Pin S module
- Arduino GND --> Pin - module
- Arduino +5 --> Pin Middle

Hardware Requirements

- Arduino controller × 1
- USB data cable × 1

- DHT 11 module × 1

Example Code

```
//KY015 DHT11 Temperature and humidity sensor
int DHpin = 8;
byte dat [5];
byte read_data () {
  byte data;
  for (int i = 0; i < 8; i ++) {
    if (digitalRead (DHpin) == LOW) {
      while (digitalRead (DHpin) == LOW); // wait for 50us
      delayMicroseconds (30); // determine the duration of the high level to determine the data is '0 'or
      if (digitalRead (DHpin) == HIGH)
        data |= (1 << (7-i)); // high front and low in the post
      while (digitalRead (DHpin) == HIGH); // data '1 ', wait for the next one receiver
     }
  }
return data;
}

void start_test () {
  digitalWrite (DHpin, LOW); // bus down, send start signal
  delay (30); // delay greater than 18ms, so DHT11 start signal can be detected

  digitalWrite (DHpin, HIGH);
  delayMicroseconds (40); // Wait for DHT11 response

  pinMode (DHpin, INPUT);
  while (digitalRead (DHpin) == HIGH);
  delayMicroseconds (80); // DHT11 response, pulled the bus 80us
  if (digitalRead (DHpin) == LOW);
  delayMicroseconds (80); // DHT11 80us after the bus pulled to start sending data

  for (int i = 0; i < 4; i ++) // receive temperature and humidity data, the parity bit is not considered
    dat[i] = read_data ();

  pinMode (DHpin, OUTPUT);
  digitalWrite (DHpin, HIGH); // send data once after releasing the bus, wait for the host to open the ne
}

void setup () {
  Serial.begin (9600);
  pinMode (DHpin, OUTPUT);
}

void loop () {
  start_test ();
  Serial.print ("Current humdity =");
  Serial.print (dat [0], DEC); // display the humidity-bit integer;
  Serial.print ('.');
  Serial.print (dat [1], DEC); // display the humidity decimal places;
  Serial.println ('%');
  Serial.print ("Current temperature =");
  Serial.print (dat [2], DEC); // display the temperature of integer bits;
  Serial.print ('.');
  Serial.print (dat [3], DEC); // display the temperature of decimal places;
  Serial.println ('C');
  delay (700);
}
```

- This page was last modified on 28 October 2014, at 18:51.