SUNROM TECHNOLOGIES

For more products visit our website http://www.sunrom.com

Document: Datasheet Date: 20-Jun-12 Model #: 3732 Product's Page: www.sunrom.com/p-1141.html

DHT11 - Humidity and Temperature Sensor

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds.

Features

- Full range temperature compensated
- Relative humidity and temperature measurement
- Calibrated digital signal
- Outstanding long-term stability
- Extra components not needed
- Long transmission distance
- Low power consumption
- 4 pins packaged and fully interchangeable

Details

This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness. Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process.
The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package.

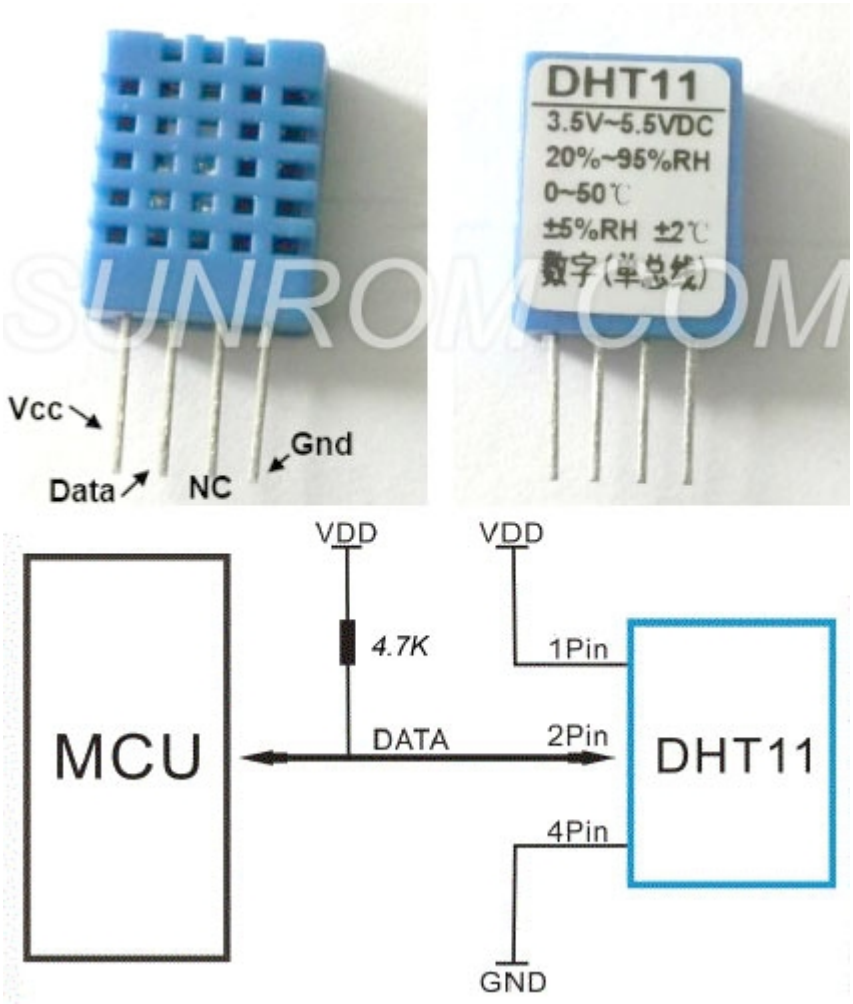
Specifications

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3	5	5.5	V
Current supply	Measuring	0.5		2.5	mA
	Stand-by	100	Null	150	uA
	Average	0.2	Null	1	mA

Typical Application



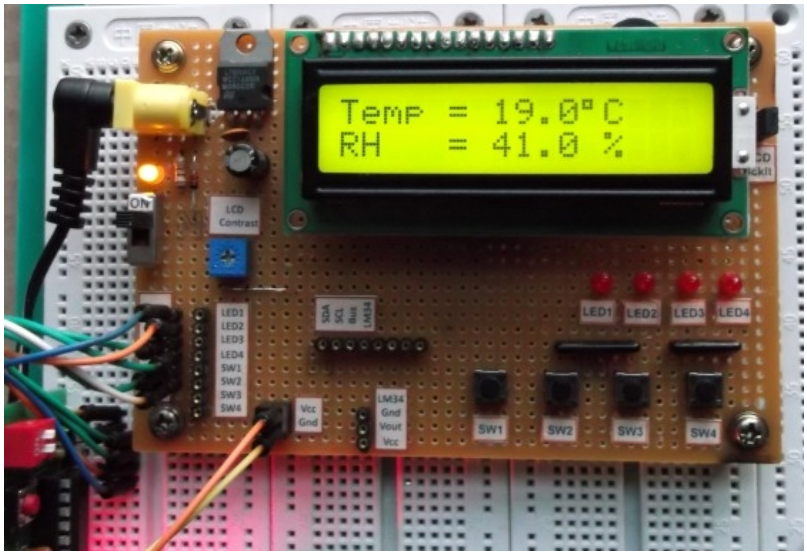
DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

SDK (Software Development Kit)

Download source code + project articles by clicking following link

http://www.sunrom.com/files/3732.zip

It contains details for AVR, PIC and Arduino projects.



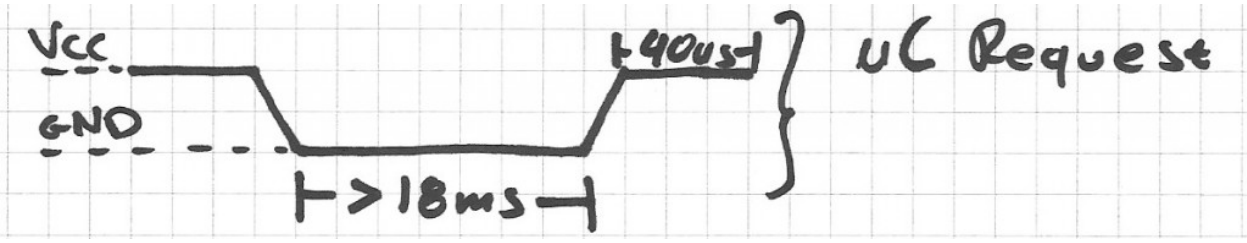
Communication Process: Serial Interface (Single-Wire Two-Way)

The interesting thing in this module is the protocol that uses to transfer data. All the sensor readings are sent using a single wire bus which reduces the cost and extends the distance. In order to send data over a bus you have to describe the way the data will be transferred, so that transmitter and receiver can understand what says each other. This is what a protocol does. It describes the way the data are transmitted. On DHT-11 the 1-wire data bus is pulled up with a resistor to VCC. So if nothing is occurred the voltage on the bus is equal to VCC.

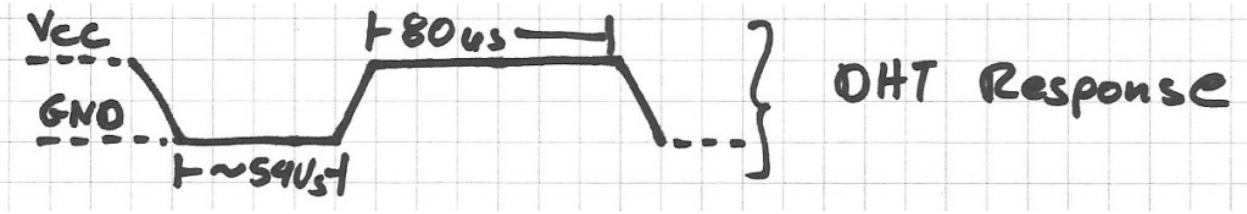
Communication Format can be seperated into three stages

1) Request
2) Response
3) Data Reading

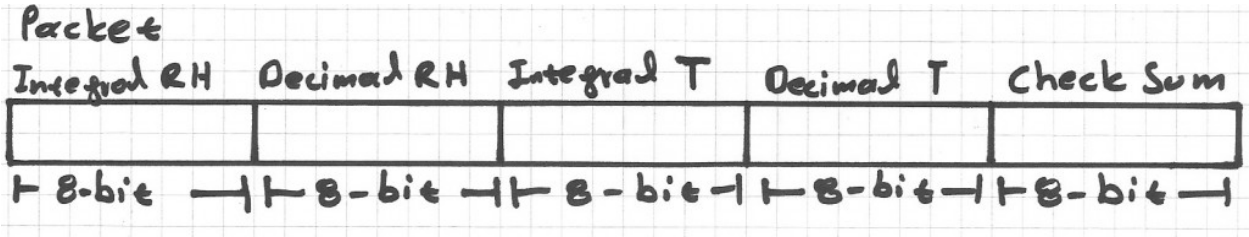
1) **Request:** To make the DHT-11 to send you the sensor readings you have to send it a request. The request is, to pull down the bus for more than **18ms** in order to give DHT time to understand it and then pull it up for **40uS**.



2) **Response:** What comes after the request is the DHT-11 response. This is an automatic reply from DHT which indicates that DHT received your request. The response is ~54uS low and 80uS high.



3) **Data Reading:** What will come after the response is the sensor data. The data will be packed in a packet of 5 segments of 8-bits each. Totally 5×8 =40bits.



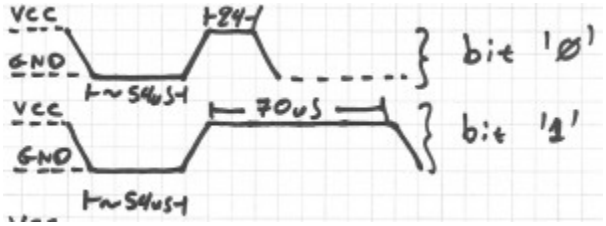
First two segments are Humidity read, integral & decimal. Following two are Temperature read in Celsius, integral & decimal and the last segment is the Check Sum which is the sum of the 4 first

segments. If Check Sum's value isn't the same as the sum of the first 4 segments that means that data received isn't correct.

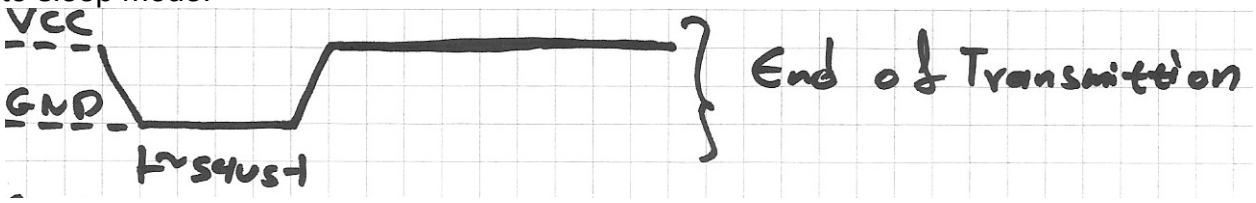
How to Identify Bits: Each bit sent is a follow of ~54uS Low in the bus and ~24uS to 70uS High depending on the value of the bit.

Bit '0' : ~54uS Low and ~24uS High

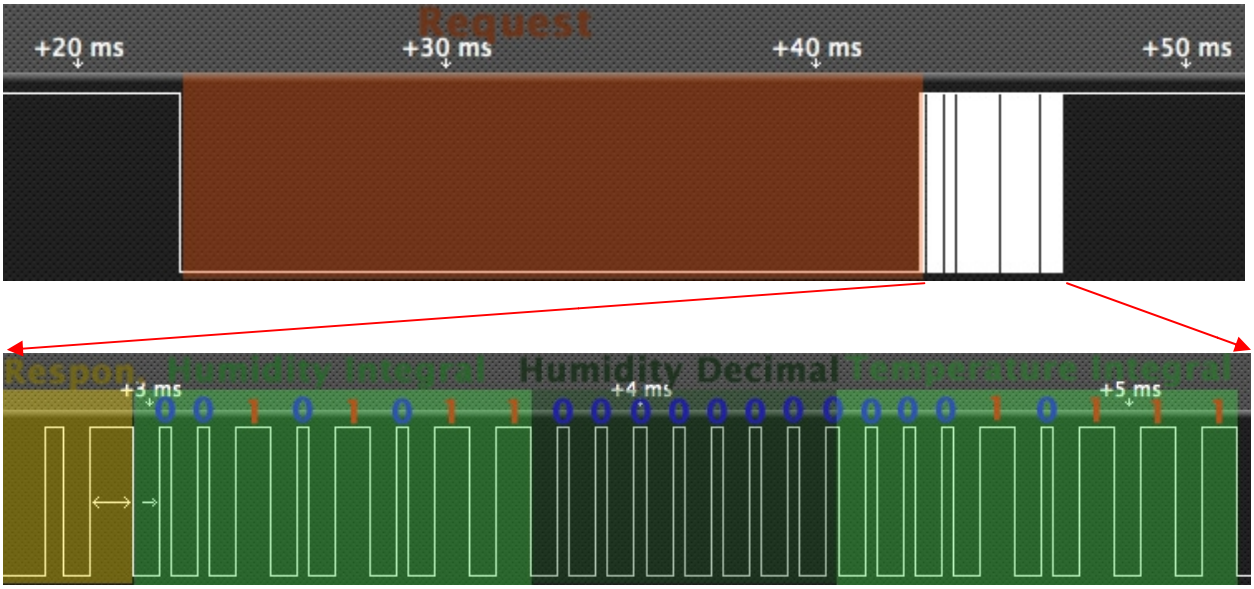
Bit '1' : ~54uS Low and ~70uS High



End Of Frame: At the end of packet DHT sends a ~54uS Low level, pulls the bus to High and goes to sleep mode.



Logic Analyzer Snapshots: In the following image you can see the request sent from the MCU to the DHT and following the packet. Because the request has very long duration as you can see is about 20mS and packet received is in uS we can't view the data bits. So it is exapanded in next view.



If we zoom at the data bits we can read the values. You can see after the Request follows the Response, and Data bits. I have drawn some color notes to be more understandable.

If we decode the above data we have.

Humidity 0b00101011.0b00000000 = 43.0% (43 is integral part and .0 is decimal part)

Temperature 0b00010111 = 23 C.

The last two segments can't be seen in this image because of zoom.

Implementation:

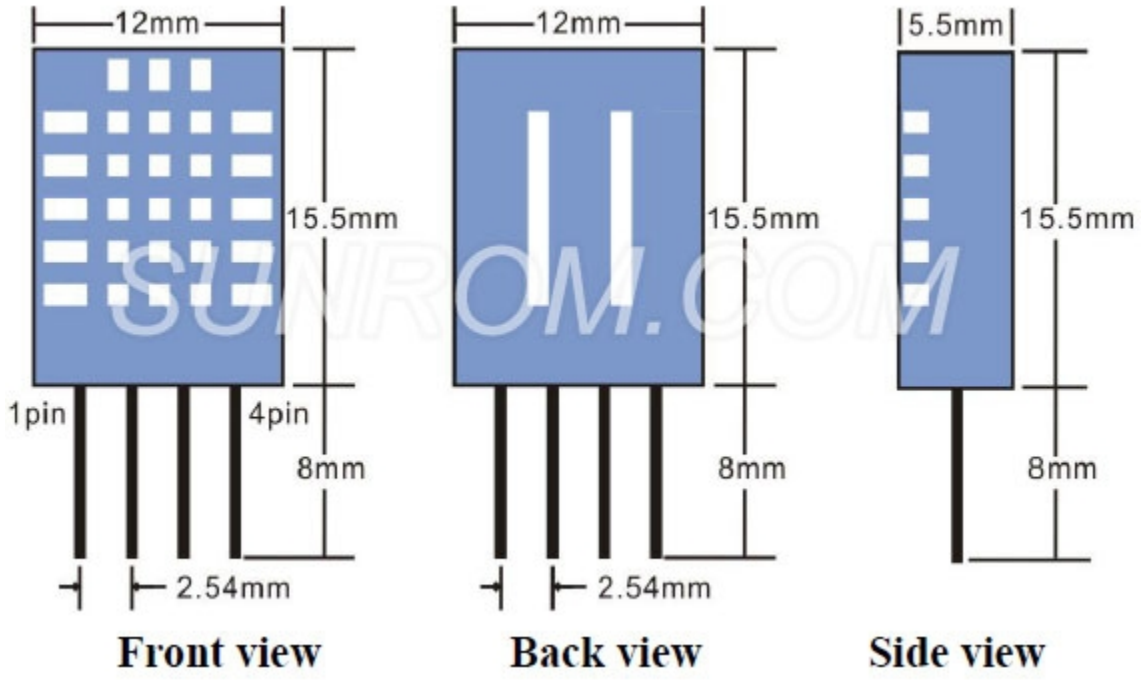
What we have to do to read a DHT-11 sensor is:

1) Send request
2) Read response
3) Read each data segment and save it to a buffer
4) Sum the segments and check if the result is the same as CheckSum

If the CheckSum is correct, the values are correct so we can use them. If CheckSum is wrong we discard the packet.

To read the data bits can use a counter and start count uSeconds of High level. For counts > 24uS we replace with bit '1'. For counts <=24 we replace with bit'0'

Dimensions (mm)



Front view **Back view** **Side view**