jDHT v1.01

B4J Additional Library

07.04.2019

Table of Contents

Overview

jDHT is an open source B4J library for the digital-output relative humidity & temperature sensors DHT11 & DHT22 connected to a Raspberry Pi.

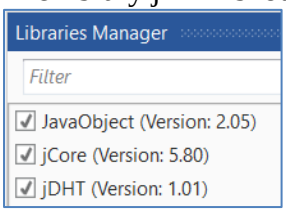
B4J is a development tool for desktop, server and IoT solutions by Anywhere Software.

The library is written in B4J (requires v5.80 or higher) and published on the B4J Forum.

Getting Started

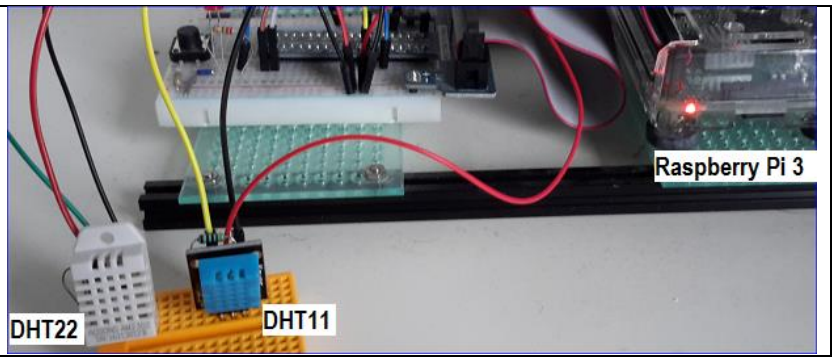
Install

- Download the B4J Library, Source Code and Examples.
- Unzip the jDHT.zip to a folder of choice.
- Copy the files jdht.jar, jdht.xml, dht.jar to the B4J additional libraries folder.
- Copy libdht.so to the app folder.
- The library jDHT should be listed in the B4J IDE Files Manager tab:



- Lookup folder Examples on how to use.

Prototype

The prototype setup has a Raspberry Pi 3 (running latest Raspian version) with a DHT11 and DHT22 connected.	

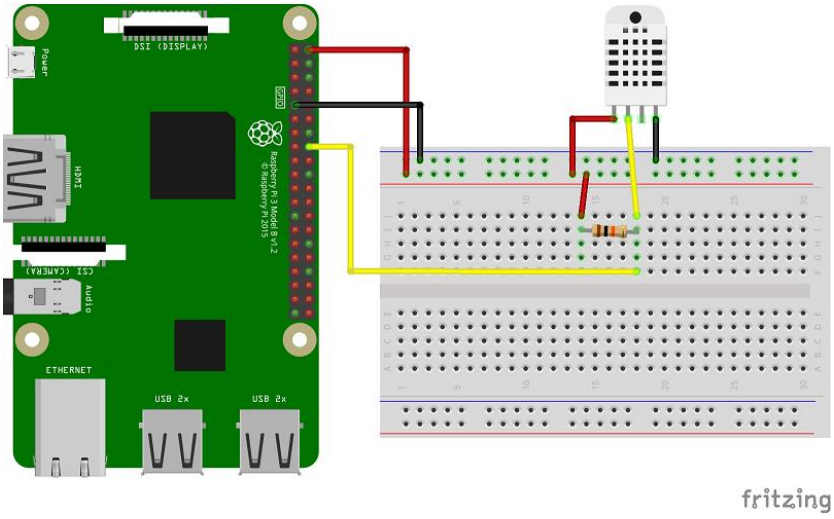
Wiring

The wiring used for testing the DHT11 & DHT22.

```
DHT11 = Raspberry Pi 3
5v = 5v
Data = Physical Pin #16 [BCM23], 4.7K Resistor to 5v
GND = GND

DHT22 = Raspberry Pi 3
5v = 5v
Data = Physical Pin #15 [BCM22], 4.7K Resistor to 5v
GND = GND
```

Circuit DHT22



Library

Object

The library object is DHT.

Declare:

```
Private dht22 as DHT
```

Properties

Temperature

Temperature As Float

Get the temperature in degrees Celsius.

If the value cannot be read, NaN is returned.

Humidity

Humidity As Float

Get the relative humidity in %RH.

If the value cannot be read, NaN is returned.

Methods

Initialize

Initialize(SensorType As Int, Pin As Int)

Initializes the object with the sensor type and GPIO Pin (BCM numbering is used).

Sensor Types:

- 2302 (AM2302)
- 11 (DHT-11)
- 22 (DHT-22 [DHT22 also named as AM2302])

Example Initialize

DHT22 connected to a Raspberry Pi physical pin 15 (=BCM22).

```
Private dht22 as DHT
dht22.Initialize(22, 22)
```

Events

- There are no events.

Code Example

B4J non-UI Example with a timer with DHT11 & DHT22 sensors tested on the prototype.

```
Sub Process_Globals
  Private dht11 As DHT
  Private dht22 As DHT
  Private timer1 As Timer
End Sub

Sub AppStart (Args() As String)
  dht11.Initialize(11, 23)
  dht22.Initialize(22, 22)
  timer1.Initialize("timer1", 2000)
  timer1.Enabled = True
  StartMessageLoop
End Sub

Sub timer1_tick
  Log($"DHT11 ${DateTime.Time(DateTime.Now)}: ${dht11.Temperature}*,
${dht11.Humidity}%"$)
  Log($"DHT22 ${DateTime.Time(DateTime.Now)}: ${dht22.Temperature}*,
${dht22.Humidity}%"$)
End Sub
```

Output

```
dht11 09:51:01: 19.9*, 41.7%
dht22 09:51:01: 19.9*, 41.7%
dht11 09:51:03: 18.0*, 44.6%
dht22 09:51:03: 18.0*, 44.6%
dht11 09:51:11: 18.0*, 56.4%
dht22 09:51:12: 18.0*, 56.4%
dht11 09:51:23: 18.7*, 71.7%
dht22 09:51:23: 18.7*, 71.7%
```

Testing B4J-Bridge

Example on how to test the library using the B4J-Bridge running on the Raspberry Pi.
Test setup: B4J v7.31, openjdk 11, DHT22, RPi 3B+ running stretch.

Wiring

```
DHT22=RPi
Pin#1:GND=GND
Pin#2:not connected
Pin#3:Signal=two connections: resistor to 5v, line to physical pin 16
(GPIO4,BCM23)
Pin#4:VCC=5v
```

Prepare RPi

- created a folder home/pi/b4j
- copy the B4J bridge
- start first time sudo java -jar b4j-bridge.jar
- stop the b4j-bridge (ctrl+c)
- cd new folder is created /home/pi/b4j/tempjars
- copy libdht.so to the folder /home/pi/b4j/tempjars
- start again sudo java -jar b4j-bridge.jar

B4J IDE

In the B4J IDE connect to the B4J bridge and run the program.

```
Sub Process_Globals
 Private dht22 As DHT
 Private timer1 As Timer
End Sub

Sub AppStart (Args() As String)
 Log("AppStart")
 ' Init the sensor with sensortype and pin (BCM)
 dht22.Initialize(22, 23)
 timer1.Initialize("timer1", 2000)
 timer1.Enabled = True
 StartMessageLoop
End Sub

Sub timer1_tick
 Log($"DHT22 ${DateTime.Time(DateTime.Now)}: ${dht22.Temperature}*,
${dht22.Humidity}%"$)
End Sub
```

Output

Example – ignore the warnings

```
AppStart
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by de.rwbl.dht.dht
(file:/home/pi/b4j/tempjars/AsyncInput1) to field
java.lang.ClassLoader.sys_paths
WARNING: Please consider reporting this to the maintainers of
de.rwbl.dht.dht
WARNING: Use --illegal-access=warn to enable warnings of further illegal
reflective access operations
WARNING: All illegal access operations will be denied in a future release
DHT22 14:49:33: NaN*, 38.5%
DHT22 14:49:35: 19.5*, 38.5%
DHT22 14:49:37: 19.5*, 38.5%
DHT22 14:49:39: 19.5*, 38.6%
DHT22 14:49:41: 19.5*, 38.6%
DHT22 14:49:43: 19.5*, 38.7%
DHT22 14:49:45: 19.5*, 38.7%
DHT22 14:49:47: 19.5*, 38.7%
DHT22 14:49:49: 19.5*, 38.6%
DHT22 14:49:51: 19.5*, 38.6%
```

Python

It is useful to test the DHT sensor connected to the RPi by using Python (part of the RPi OS). The Adafruit Python DHT Sensor Library is a good source to use. Lookup the internet for instructions & examples.

Test Script

Used this script for testing.
Note the data pin being 23 (Physical pin 16, GPIO. 4, BCM 23).

```
#Libraries
import Adafruit_DHT as dht
from time import sleep
#Set DATA pin
DHT = 23
while True:
    #Read Temp and Hum from DHT22
    h,t = dht.read_retry(dht.DHT22, DHT)
    #Print Temperature and Humidity on Shell window
    print('Temp={0:0.1f}*C  Humidity={1:0.1f}%'.format(t,h))
    sleep(5) #Wait 5 seconds and read again
```

Run

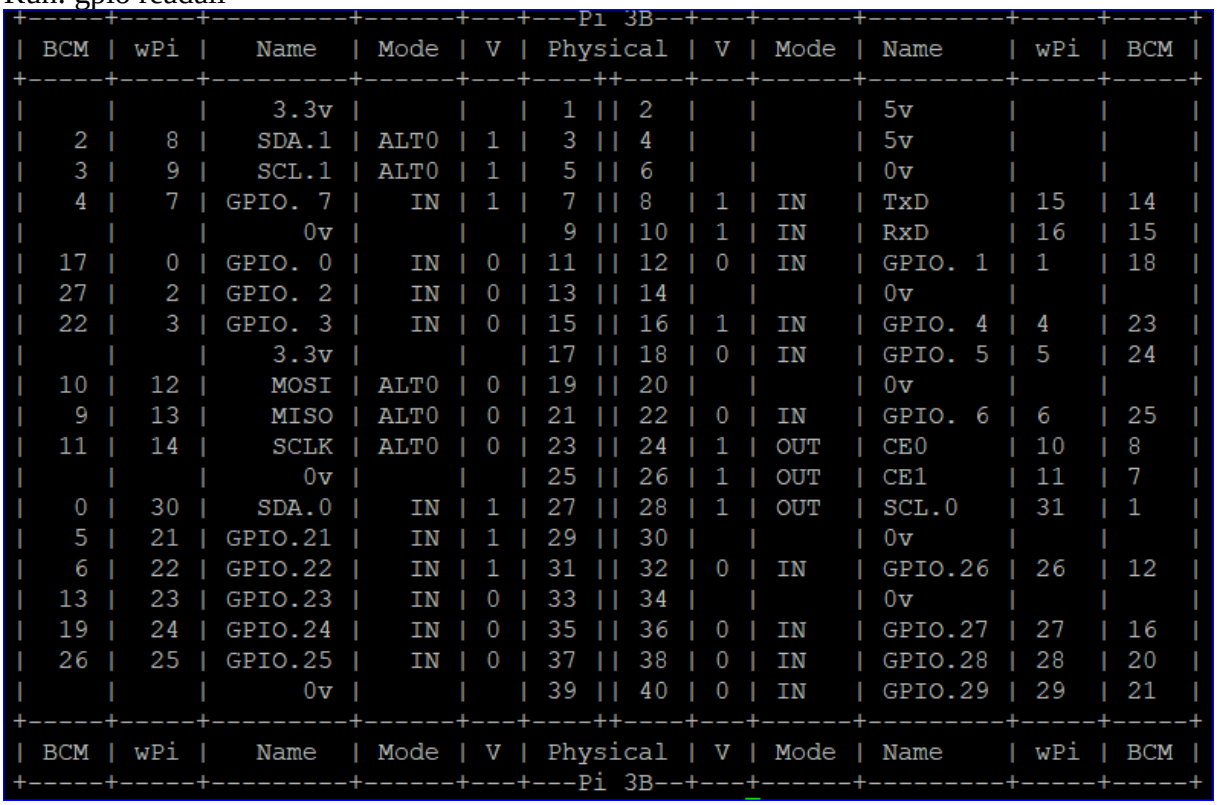
```
sudo python dht22.py
```

Output
(ignore the first reading)

```
Temp=19.4*C  Humidity=38.5%
Temp=18.4*C  Humidity=38.8%
Temp=18.3*C  Humidity=38.7%
Temp=18.4*C  Humidity=38.7%
```

GPIO ReadAll Table

Run: gpio readall



BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM
		3.3v			1 || 2			5v		
2	8	SDA.1	ALT0	1	3 || 4			5v		
3	9	SCL.1	ALT0	1	5 || 6			0v		
4	7	GPIO. 7	IN	1	7 || 8	1	IN	TxD	15	14
		0v			9 || 10	1	IN	RxD	16	15
17	0	GPIO. 0	IN	0	11 || 12	0	IN	GPIO. 1	1	18
27	2	GPIO. 2	IN	0	13 || 14			0v		
22	3	GPIO. 3	IN	0	15 || 16	1	IN	GPIO. 4	4	23
		3.3v			17 || 18	0	IN	GPIO. 5	5	24
10	12	MOSI	ALT0	0	19 || 20			0v		
9	13	MISO	ALT0	0	21 || 22	0	IN	GPIO. 6	6	25
11	14	SCLK	ALT0	0	23 || 24	1	OUT	CE0	10	8
		0v			25 || 26	1	OUT	CE1	11	7
0	30	SDA.0	IN	1	27 || 28	1	OUT	SCL.0	31	1
5	21	GPIO.21	IN	1	29 || 30			0v		
6	22	GPIO.22	IN	1	31 || 32	0	IN	GPIO.26	26	12
13	23	GPIO.23	IN	0	33 || 34			0v		
19	24	GPIO.24	IN	0	35 || 36	0	IN	GPIO.27	27	16
26	25	GPIO.25	IN	0	37 || 38	0	IN	GPIO.28	28	20
		0v			39 || 40	0	IN	GPIO.29	29	21
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM