

Electronic design professionals can leverage worldwide “maker” trend and use off-the-shelf hard- and software to kick off an IoT business project. Popular **Raspberry Pi** computer board is a good example. This article provides practical guidance and explains how Raspberry Pi can be used to build a compelling **IoT node mockup** for

- manufacturer-independent **evaluation** purposes,
- feasibility **studies** or
- **demonstration** for management and/or customers

Raspberry Pi (short: “**RasPi**”) is a small versatile play-and-play computer board which works out-of-the-box and offers common interfaces for users and peripherals. It comes with an onboard OS and many generic software ingredients for all kind of embedded computing applications. Customers can add application-specific circuitry via I/O pins or use I²C bus which is offered

by most IoT peripherals like sensors, controllers, secure elements, etc.

IoT (“Internet of Things”) stands for network applications with IoT devices (“nodes”) used at unattended locations. Typical IoT applications are **monitoring and reporting remote conditions**. By nature, IoT operators usually do not sit in front of IoT devices as these nodes are located at multiple places where the remote IoT work is done. Consequently, **IoT devices** must offer a couple of **fundamental functions**:

- (wireless) network connectivity
- “headless” design (i.e. no display, keyboard or mouse)
- remote control for operators
- forwarding and visualization of locally collected IoT data
- (mutual) authentication and secure messaging
- automatic startup and independent, autonomous operation
- application-specific function block

Amazingly, Raspberry Pi meets all these requirements out-of-box and free-of-charge. In addition, it offers user controllable GPIOs for customization allowing users to add an application-specific subsystem quickly. Dedicated onboard OS functions plus easy-to-learn Python programming language are offering flexible **hardware control**. This is why RasPi works fine as a **generic IoT node**, although in most cases it will not be considered for a large-scale commercial IoT roll-out (e.g. because of price, CPU, board dimensions, security). But an IoT node mockup based on RasPi is an efficient approach for evaluation and demonstration and might be worthwhile to consider esp. for startups or IoT newbies. It can be used in pre-selection project stage before you buy evaluation boards or a design kit from a specific manufacturer.

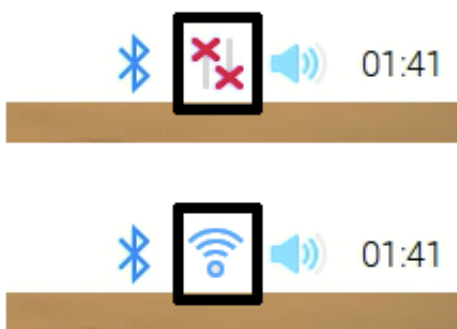
In this article we are introducing our **RasPi IoT node**, a simple project based on a simple IoT application, but many fundamental IoT functions have been covered: headless WLAN design, remote control, webserver, stand-alone operation. Our *RasPi IoT node* project is a “**toolbox**”, elements can be used as

a **starting point for a custom mockup design** according to your needs. As a sample IoT application for this project we used an environmental sensor (humidity, temperature).

SETUP OF RASBERRY PI AS A GENERIC IOT NODE

Fortunately, latest RasPi versions (Pi3 or Pi4) have onboard WLAN function, so you can easily prepare it for mentioned headless operation which is standard operation mode for IoT devices. As a start and in order to prepare for wireless operation we have to connect a USB keyboard, USB mouse and HDMI display to our RasPi board. A new RasPi should boot with a graphical user interface after power up. For headless operation three actions are required:

1. **Enable WLAN.** Click on the wireless network icon in the top right-hand corner of the screen, select your network from the drop-down menu and enter associated security key. This will enable WLAN connection of your RasPi IoT node mockup you might want to verify right away using built-in Internet browser.



2. **Identify RasPi's IP address.** Click on berry icon in top left-hand corner and use pull down menu to navigate via "Accessories" tap to "Terminal", click on it to open a terminal session. Command `hostname - I` returns IP address `192.168.178.31`

```
pi@iotnode:~$ hostname -I
192.168.178.31 2003:c1:2721:f000:f323:2081:dba5:b3dd
pi@iotnode:~$
```

3. **Change default UI to command line interface (“CLI”).** Click on berry icon in top left-hand corner and use pull down menu to navigate via “Preferences” to „Raspberry Pi Configuration“. Besides other stuff you should specify a friendly hostname, e.g. „**iotnode**“. As a start it makes sense to select “Auto Login” “as “Login as user shutdown pi”. Under “Interfaces” **enable I²C** and **SSH**. Finally, you change „Boot“ option to „To CLI“.

Then shutdown your RasPi, remove display, keyboard and mouse and reboot. Now you can use a PC to connect via WLAN to your RasPi IoT node mockup. Windows as well as MacOS come with onboard terminal programs offering command line interface for users. A built-in **SSH client** is being used to connect securely to a remote computer. Enter `ssh pi@192.168.178.31` or `ssh pi@iotnode` followed by assigned password for user “pi”, then you enjoy full **remote control** via command line interface and access to complete RasPi OS (Debian) command set which is based on GNU/Linux.

```
kerstens-Mini:~ kersten$ ssh pi@192.168.178.31
pi@192.168.178.31's password:
Linux iotnode 5.4.72-v7+ #1356 SMP Thu Oct 22 13:56:54 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Nov  4 20:10:53 2020
pi@iotnode:~$
```

Now we can start to customize our *RasPi IoT node* hard- and software according to our needs.

HARDWARE SETUP – HUMIDITY/TEMPERATURE SENSOR ON I²C BUS

For our *RasPi IoT node* project we have chosen a popular IoT application:

remote environmental sensing, so we have to add a few components. Related RasPi hardware extension is done via GPIO pins available on a 40-pin connector. It is a good idea to use a small standard breadboard and to connect it with a flat ribbon cable (see Figure 1).

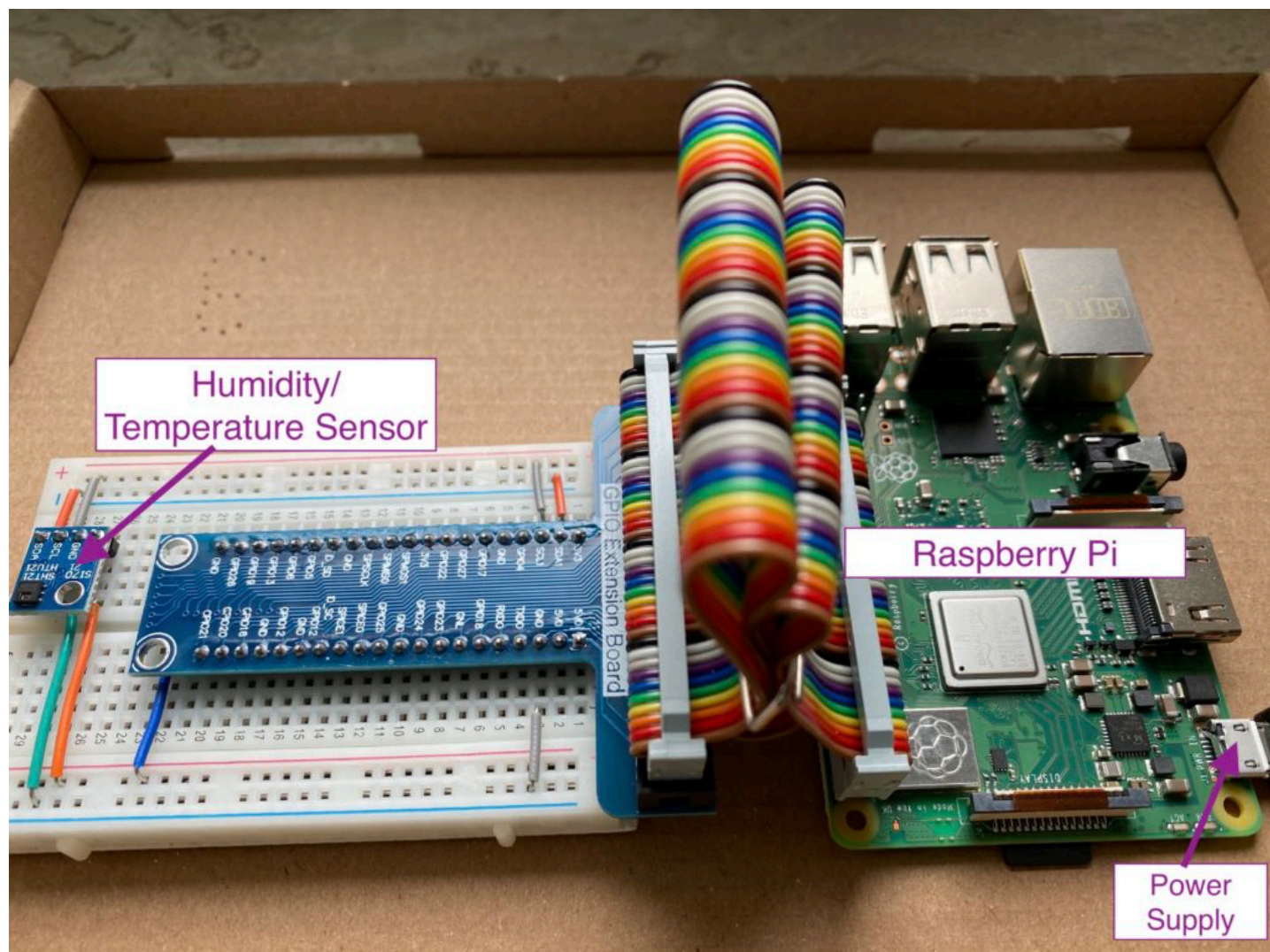


Figure 1: RasPi IoT node Project Hardware (© chip.info.com)

Used sample sensor is **HTU21D** (manufacturer is TE Connectivity) resp. **Si7021-A20** (manufacturer is Silicon Labs) which are functionally equivalent. Datasheet is telling us how to connect the sensor to an I²C bus.

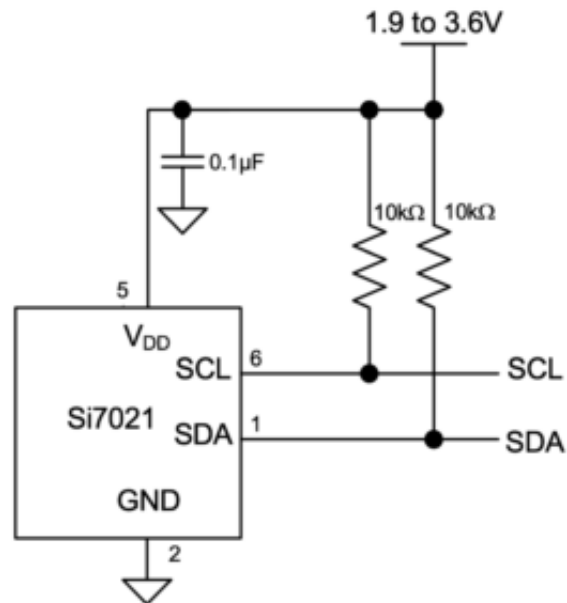


Figure 2: Si7021 I²C connection (© Silicon Labs)

In order to simplify breadboard wiring you can buy ready-to-use modules from many local dealers, e.g. like SparkFun <https://www.sparkfun.com> (in the US) or <https://cheaptronic24.de> (in Germany). Connection with I²C bus can be done with 4 wires only.



to continue reading and for download of complete article please proceed to www.chip-info.com. thank you ;-)

Not
on t
info

available
[www.chip-](http://www.chip-info.com)

HW INITIALIZATION AND FUNCTIONAL TEST

Let us do this step-by-step. As a first step we use onboard **i2c-tools** package