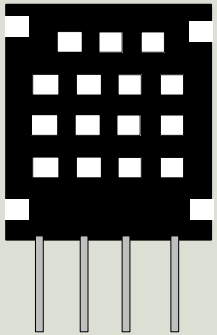
https://www.halvorsen.blog



Raspberry Pi and AM2320

Temperature and Humidity Sensor with I2C Interface

Exploring different premade Libraries for AM2320

Hans-Petter Halvorsen

Contents

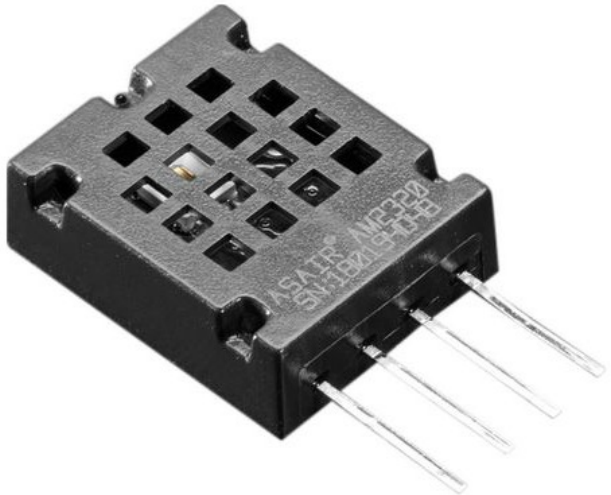
Introduction

Hans-Petter Halvorsen

Table of Contents

Introduction

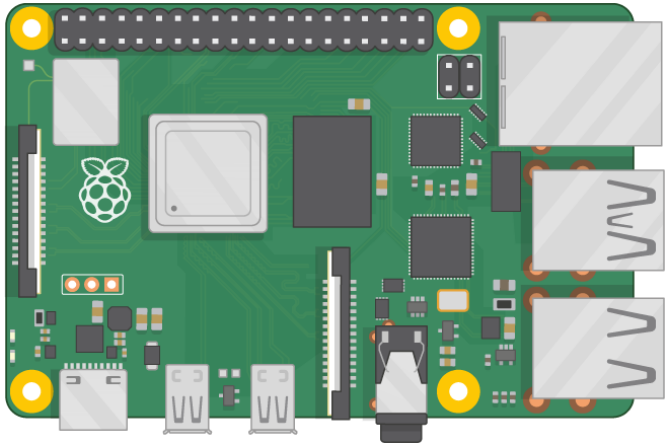
This Tutorial will demonstrate the use of a **AM2320** Temperature and Humidity Sensor in combination with Raspberry Pi and Python

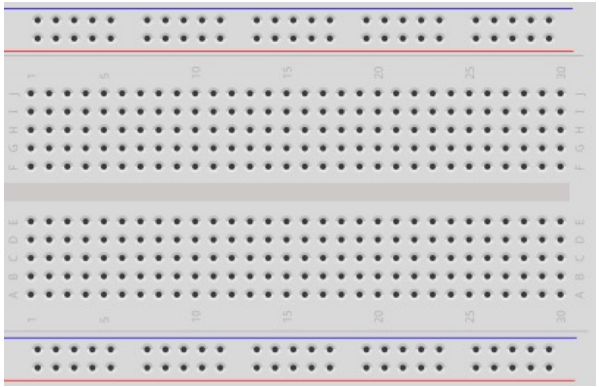


https://www.adafruit.com/product/3721

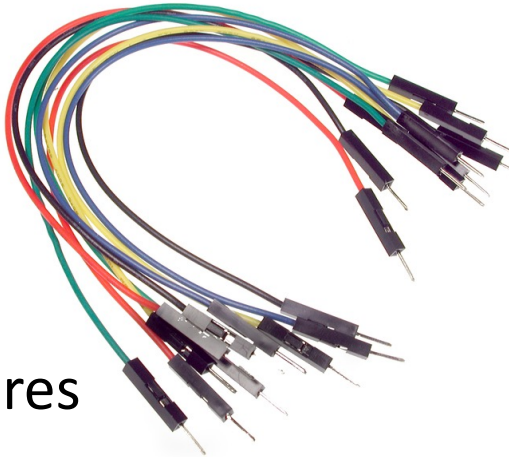
Hardware

Raspberry Pi



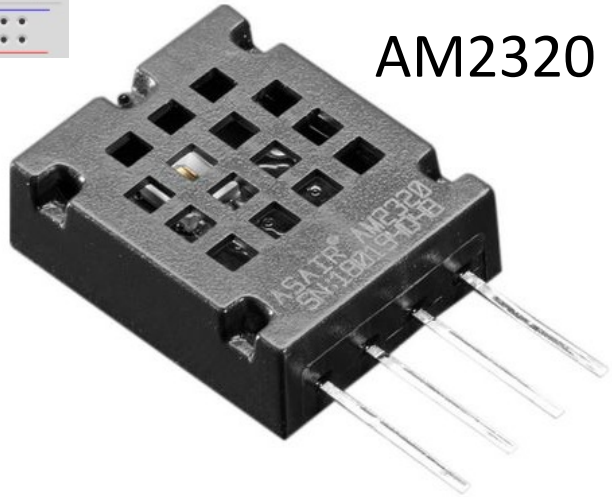


Breadboard



Wires

AM2320



AM2320

Temperature and Humidity Sensor

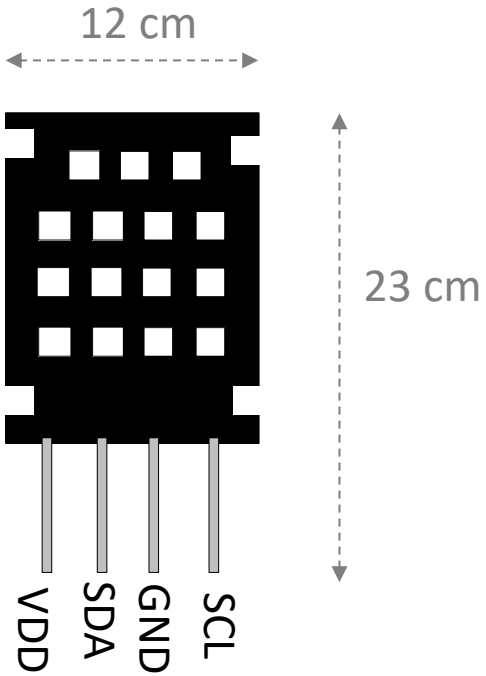
Hans-Petter Halvorsen

Table of Contents

AM2320 Sensor

- Temperature and Humidity Sensor
- **I2C** Interface
- Range: $-40°C$ to $+80°C$ and 0 to $100\%RH$
- Accuracy: Humidity $\pm 3\%RH$ and Temperature ± 0.5℃ according to the Datasheet
- Sampling Rate: 0.5Hz, this means the minimum interval between readings is 2 seconds
- I2C address: **0x5C** (cannot be changed)
- Price: about $4
- Sensor Overview: https://learn.adafruit.com/adafruit-am2320-temperature-humidity-i2c-sensor
- Datasheet: https://cdn-shop.adafruit.com/product-files/3721/AM2320.pdf

AM2320 Sensor



I2C Interface

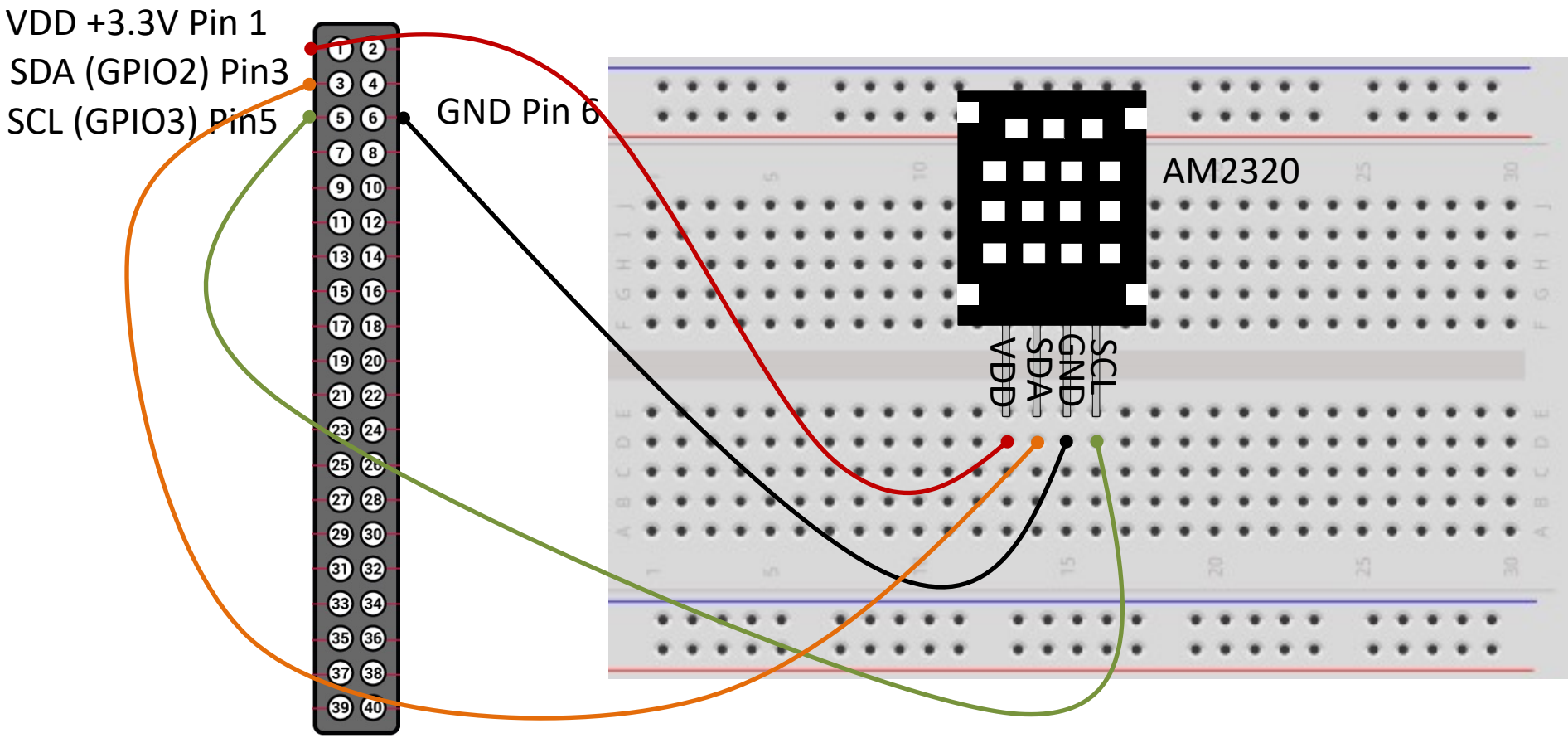
Pin Overview:

- **VDD** – Power, 3-5VDC
- **SDA** - I2C data in/out, requires a pullup resistor of 2-10KΩ to VDD
- **GND** - Ground
- **SCL** - I2C clock in, requires a pullup resistor of 2-10KΩ to VDD

Note! The Raspberry Pi has built-in pull up resistors on SDA/SCL, so there is no need to add external pullup resistors

https://learn.adafruit.com/adafruit-am2320-temperature-humidity-i2c-sensor/pinouts

AM2320 Wiring



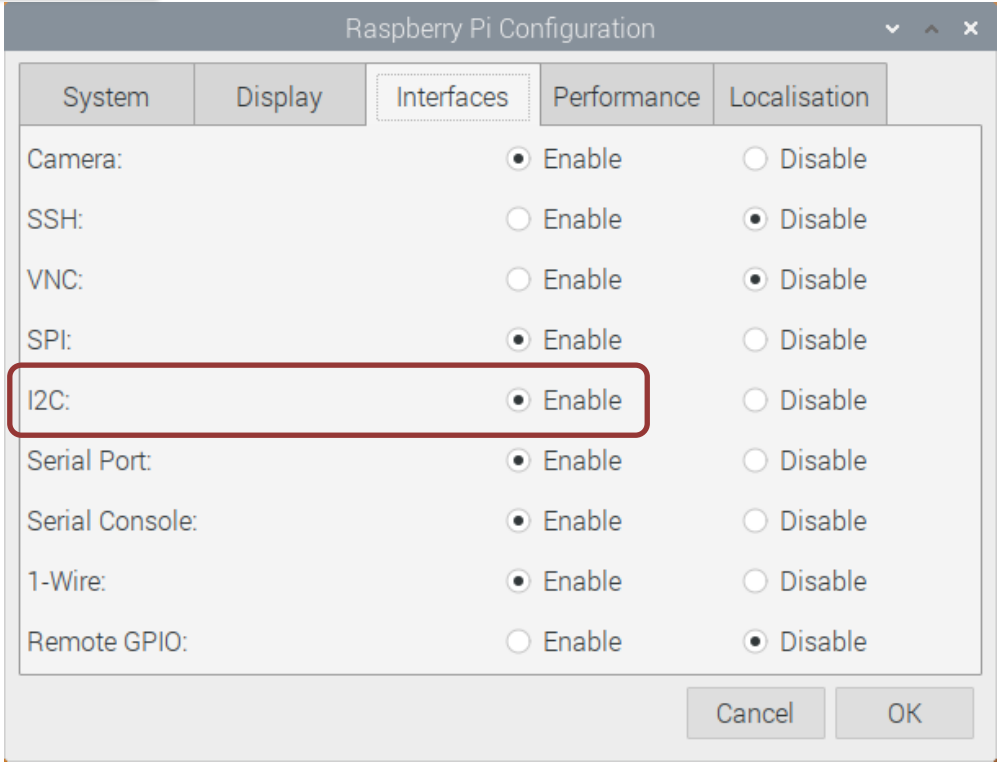
I2C Interface

Hans-Petter Halvorsen

Table of Contents

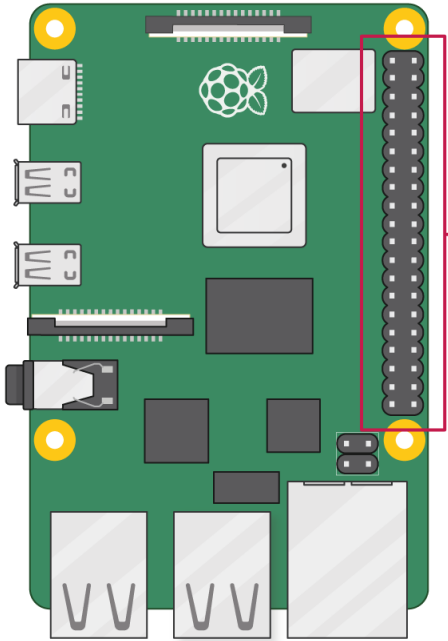
Access I2C on Raspberry Pi

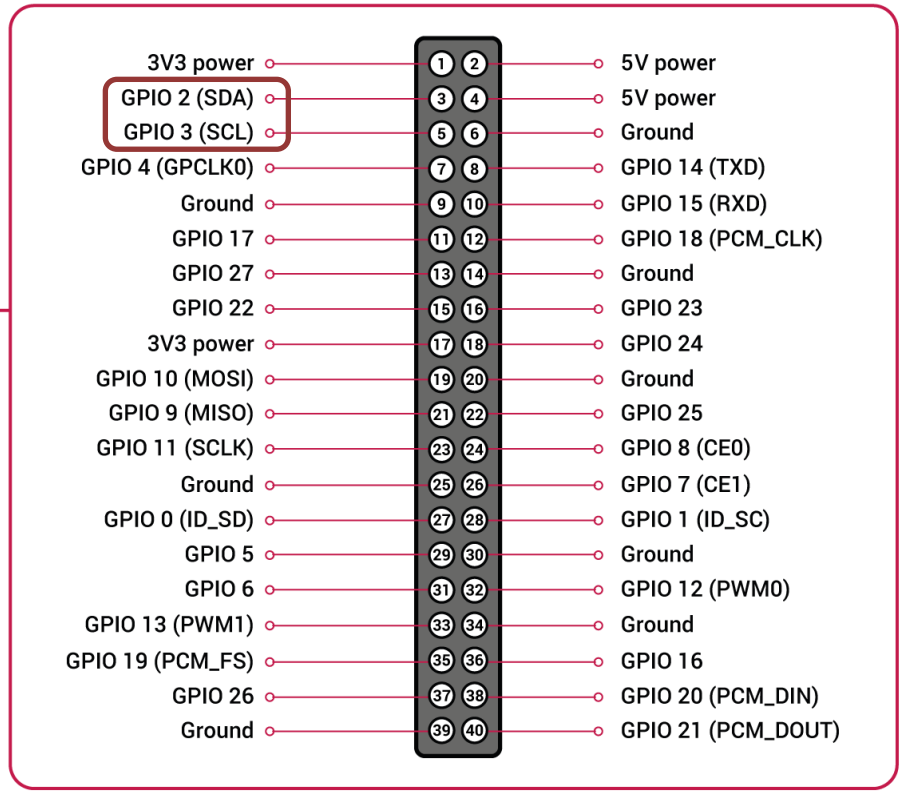
You need to Enable I2C on the Raspberry Pi



I2C Wiring on Raspberry Pi

GPIO 40 pins Connector





Note! The I2C pins include a fixed 1.8 kΩ pull-up resistor to 3.3v.

Detecting I2C Devices

Install I2C Tools on the Raspberry Pi:

```
sudo apt-get install -y i2c-tools
```

Detecting and Find the Address of the I2C Device using the `i2cdetect` command:

```
sudo i2cdetect -y 1
```

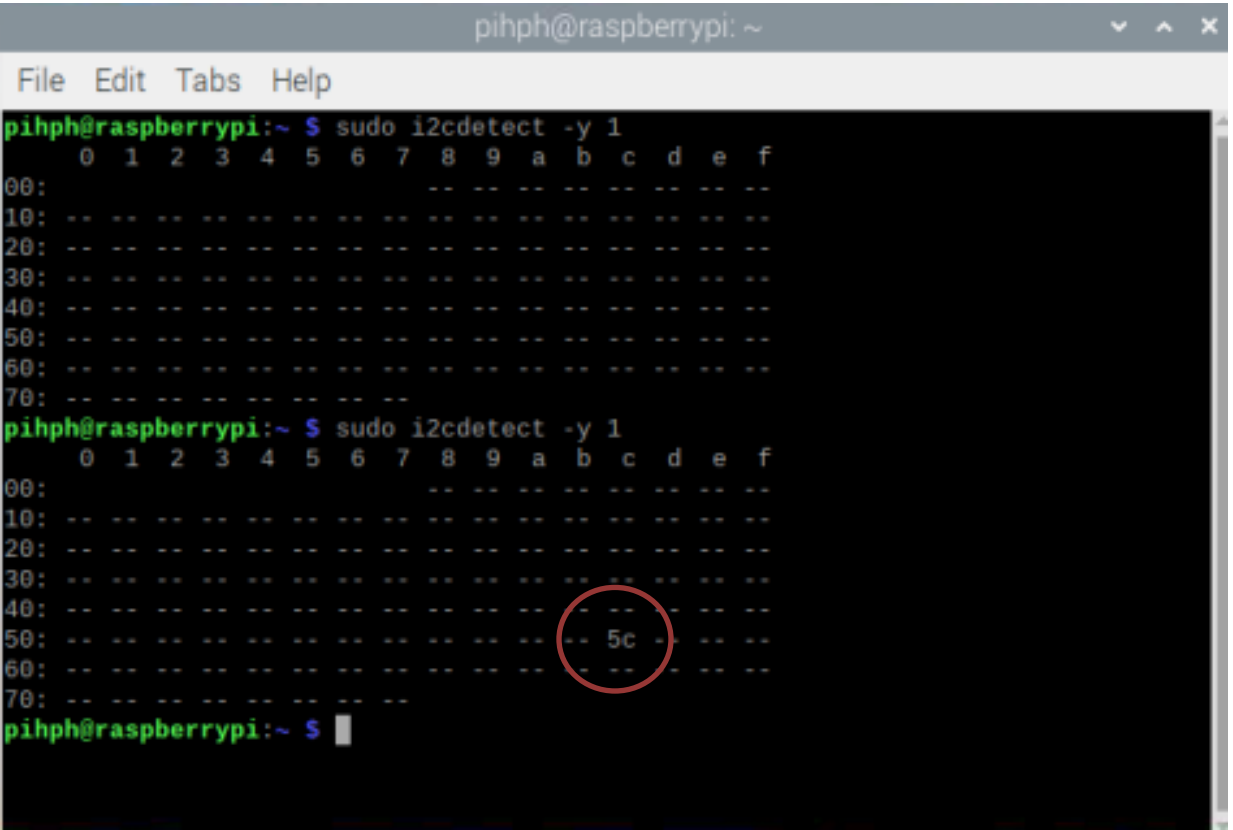
We can read and write its registers using `i2cget`, `i2cset` and `i2cdump`

Example:

```
sudo i2cget -y 1 0x5C
```

Device Address

Detecting I2C Devices




```
sudo i2cdetect -y 1
```

I need to run the command twice because the sensor goes into sleep mode

Python Examples

AM2320 Temperature and Humidity Sensor

Hans-Petter Halvorsen

Table of Contents

Python Examples

- You can create your code from scratch if you read the datasheet for the sensor and know details regarding I2C.
- Typically, it is better to use an existing Python Library or Example
- There exist many different premade Python Libraries and Python Examples for AM2320. Here are some:
  - CircuitPython using Adafruit-Blinka + CircuitPython AM2320 Python Library
    https://pypi.org/project/adafruit-circuitpython-am2320/
  - Gozem/am2320
    https://github.com/Gozem/am2320
  - am2320-driver
    https://pypi.org/project/am2320-driver/

Example 1

CircuitPython and Adafruit-Blinka

Hans-Petter Halvorsen

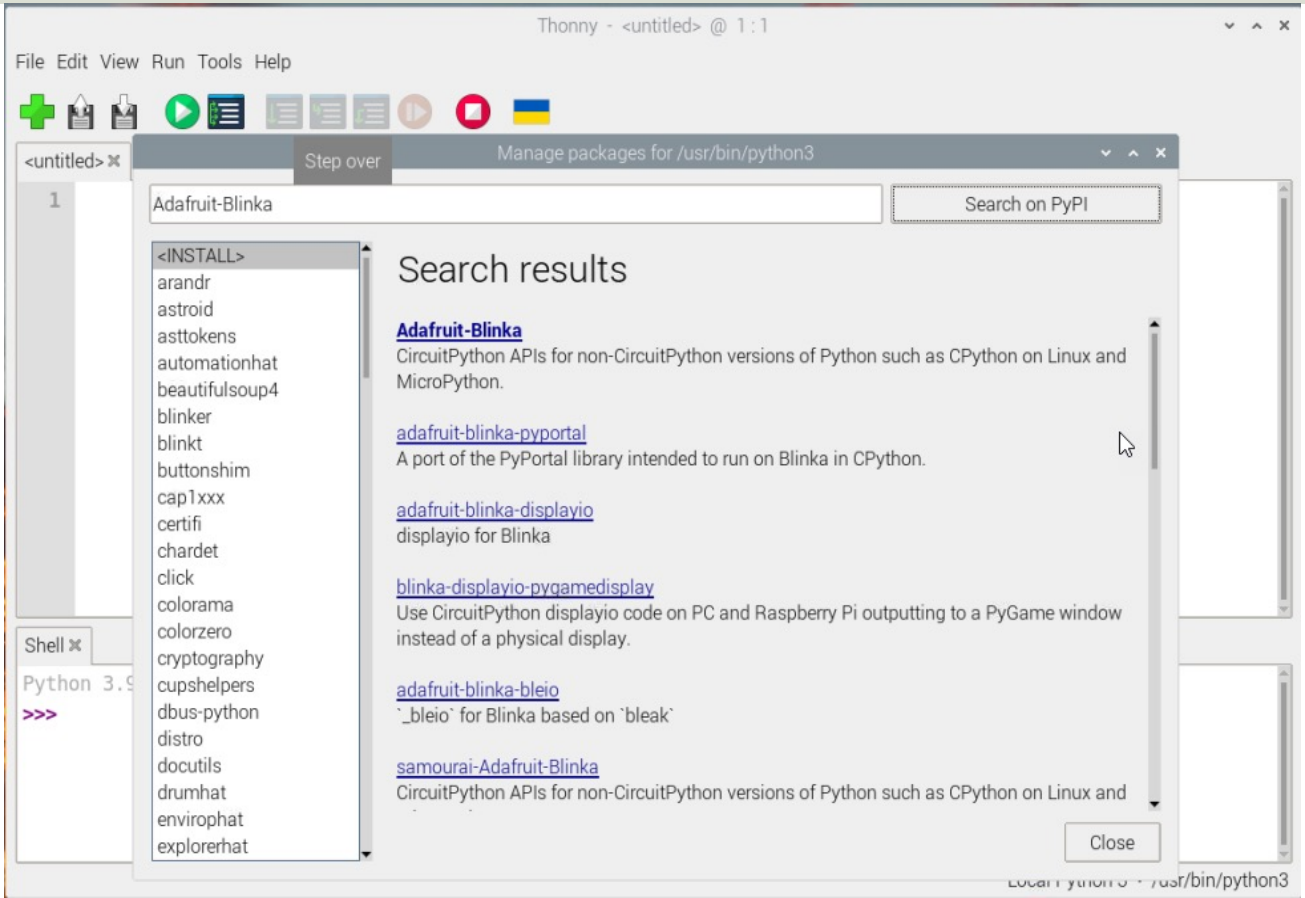
Table of Contents

Install Adafruit-Blinka

- Adafruit-Blinka: https://pypi.org/project/Adafruit-Blinka/
- Install from the Thonny Python Editor (Tools -> Manage packages…). Search for “Adafruit-Blinka“
- or use pip:

```
pip3 install Adafruit-Blinka
```

Install Adafruit-Blinka



Test of Adafruit-Blinka

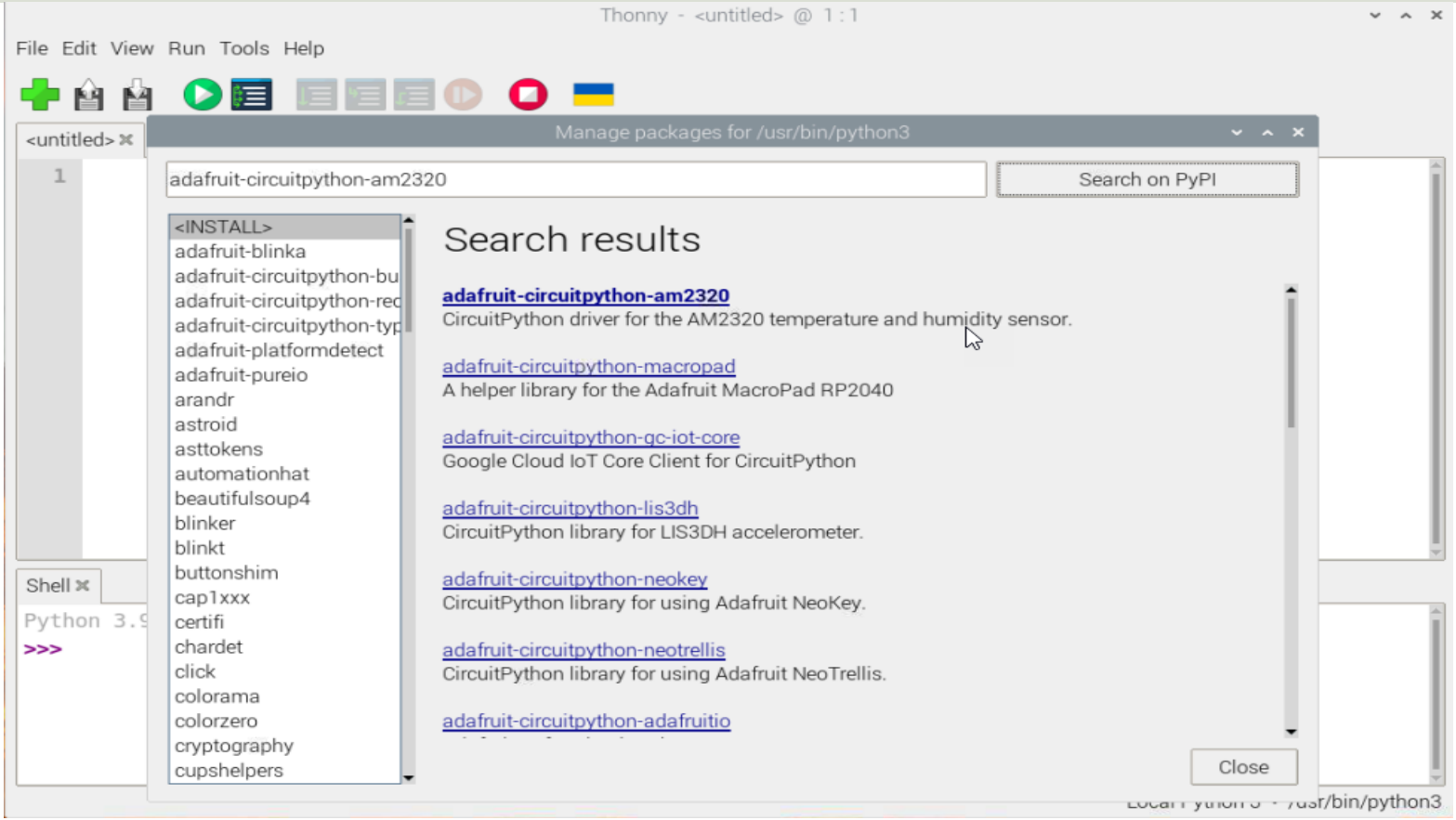
```
import board
import digitalio
import busio

print("Hello blinka!")

# Try to create a Digital input
pin = digitalio.DigitalInOut(board.D4)
print("Digital IO ok!")

# Try to create an I2C device
i2c = busio.I2C(board.SCL, board.SDA)
print("I2C ok!")

# Try to create an SPI device
spi = busio.SPI(board.SCLK, board.MOSI, board.MISO)
print("SPI ok!")

print("done!")
```

CircuitPython AM2320 Python Library

- CircuitPython AM2320 Library: https://pypi.org/project/adafruit-circuitpython-am2320/
- Install from the Thonny Python Editor (Tools -> Manage packages…). Search for “adafruit-circuitpython-am2320”
- or use pip:

```
pip3 install adafruit-circuitpython-am2320
```

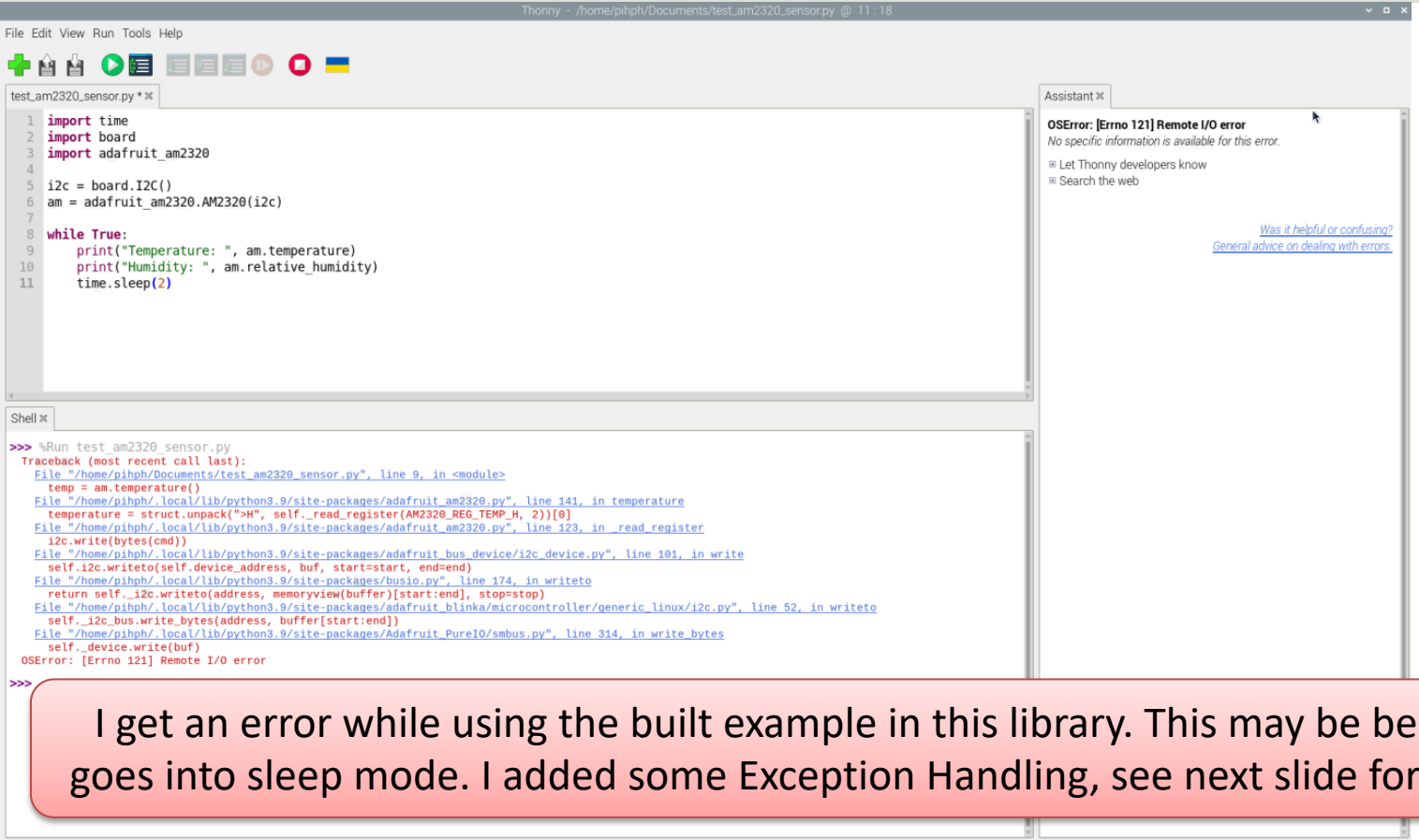
AM2320 Python Library



AM2320 Python Code Example

```
import time
import board
import adafruit_am2320

i2c = board.I2C()
am = adafruit_am2320.AM2320(i2c)

while True:
    print("Temperature: ", am.temperature)
    print("Humidity: ", am.relative_humidity)
    time.sleep(2)
```

Results



I get an error while using the built example in this library. This may be because the sensor goes into sleep mode. I added some Exception Handling, see next slide for updated Example

Updated Python Code

```
import time
import board
import adafruit_am2320

i2c = board.I2C()
am = adafruit_am2320.AM2320(i2c)

while True:
    try:
        print("\nTemperature: ", am.temperature, "°C")
        time.sleep(0.10)
        print("Humidity: ", am.relative_humidity, "%RH")
    except:
        print("Error Reading Sensor Data")
    time.sleep(5)
```

Results


Thonny - /home/pihph/Documents/test_ar
File Edit View Run Tools Help
test_am2320_sensor.py
am2320_sensor_ex_loop.py
1 import time
2 import board
3 import adafruit_am2320
4
5 i2c = board.I2C()
6 am = adafruit_am2320.AM2320(i2c)
7
8 while True:
9 try:
10 print("\nTemperature: ", am.temperature, "°C")
11 time.sleep(0.10)
12 print("Humidity: ", am.relative_humidity, "%RH")
13 except:
14 print("Error Reading Sensor Data")
15 time.sleep(5)
Shell
>>> %Run test_am2320_sensor.py
Error Reading Sensor Data
Temperature: 23.4 °C
Humidity: 26.3 %RH
Temperature: 23.5 °C
Humidity: 26.3 %RH
Temperature: 23.5 °C
Humidity: 26.3 %RH
Temperature: 23.5 °C
Humidity: 26.3 %RH
Temperature: 23.5 °C
Humidity: 26.3 %RH
Temperature: 23.5 °C
Humidity: 26.3 %RH
Temperature: 23.5 °C
Humidity: 26.3 %RH
Temperature: 23.5 °C
Humidity: 26.3 %RH

Example 2

Gozem/am2320

Hans-Petter Halvorsen

Table of Contents

Gozem/am2320

- Gozem/am2320: https://github.com/Gozem/am2320
- This is not a Python Library, it is just a Python Example, that you can copy and use
- The “am2320.py” Example creates and use a Class
- I have modified and put the Class into a separate File

AM2320 Python Class

```
import posix
from fcntl import ioctl
import time

class AM2320:
    I2C_ADDR = 0x5c
    I2C_SLAVE = 0x0703

    def __init__(self, i2cbus = 1):
        self._fd = posix.open("/dev/i2c-%d" % i2cbus, posix.O_RDWR)
        ioctl(self._fd, self.I2C_SLAVE, self.I2C_ADDR)

    def __del__(self):
        posix.close(self._fd)

    @staticmethod
    def _calc_crc16(data):
        crc = 0xFFFF
        for x in data:
            crc = crc ^ x
            for bit in range(0, 8):
                if (crc & 0x0001) == 0x0001:
                    crc >>= 1
                    crc ^= 0xA001
                else:
                    crc >>= 1
        return crc

    @staticmethod
    def _combine_bytes(msb, lsb):
        return msb << 8 | lsb

    def readSensor(self):
        try:
          posix.write(self._fd, b'\0x00')
        except:
          pass
        time.sleep(0.001)  #Wait at least 0.8ms, at most 3ms

        # write at addr 0x03, start reg = 0x00, num regs = 0x04 */
        posix.write(self._fd, b'\x03\x00\x04')
        time.sleep(0.0016) #Wait at least 1.5ms for result

        # Read out 8 bytes of result data
        data = bytearray(posix.read(self._fd, 8))

        # Check data[0] and data[1]
        if data[0] != 0x03 or data[1] != 0x04:
            raise Exception("First two read bytes are a mismatch")

        # CRC check
        if self._calc_crc16(data[0:6]) != self._combine_bytes(data[7], data[6]):
            raise Exception("CRC failed")

        # Temperature resolution is 16Bit
        temp = self._combine_bytes(data[4], data[5])
        if temp & 0x8000:
            temp = -(temp & 0x7FFF)
        temp /= 10.0

        humi = self._combine_bytes(data[2], data[3]) / 10.0

        return (temp, humi)
```

Modified version of Gozem/am2320:
https://github.com/Gozem/am2320

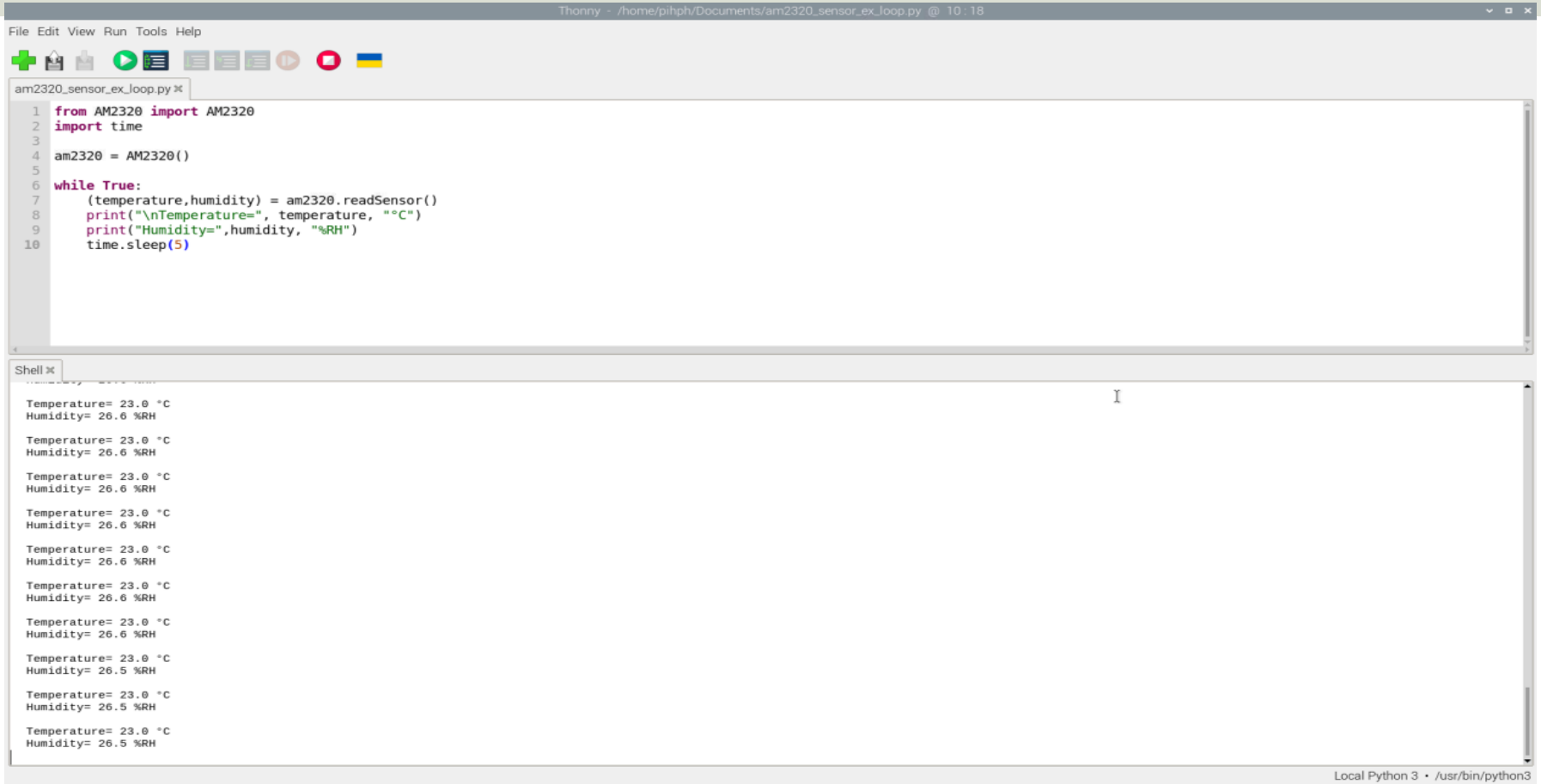
AM2320 Python Code Example

```
from AM2320 import AM2320

am2320 = AM2320()

(temp,humid) = am2320.readSensor()
print(temp)
print(humid)
```

AM2320 Python Code Example

```
from AM2320 import AM2320
import time

am2320 = AM2320()

while True:
    (temp,humid) = am2320.readSensor()
    print("\nTemperature=", temp, "°C")
    print("Humidity=",humid, "%RH")
    time.sleep(5)
```

Results


Thonny - /home/pihph/Documents/am2320_sensor_ex_loop.py @ 10 : 18
File Edit View Run Tools Help
am2320_sensor_ex_loop.py
1 from AM2320 import AM2320
2 import time
3
4 am2320 = AM2320()
5
6 while True:
7 (temperature,humidity) = am2320.readSensor()
8 print("\nTemperature=", temperature, "°C")
9 print("Humidity=",humidity, "%RH")
10 time.sleep(5)
Shell
Temperature= 23.0 °C
Humidity= 26.6 %RH
Temperature= 23.0 °C
Humidity= 26.6 %RH
Temperature= 23.0 °C
Humidity= 26.6 %RH
Temperature= 23.0 °C
Humidity= 26.6 %RH
Temperature= 23.0 °C
Humidity= 26.6 %RH
Temperature= 23.0 °C
Humidity= 26.6 %RH
Temperature= 23.0 °C
Humidity= 26.6 %RH
Temperature= 23.0 °C
Humidity= 26.5 %RH
Temperature= 23.0 °C
Humidity= 26.5 %RH
Temperature= 23.0 °C
Humidity= 26.5 %RH
Local Python 3 • /usr/bin/python3

Example 3

am2320-driver

Hans-Petter Halvorsen

Table of Contents

am2320-driver Python Library

- am2320-driver: https://pypi.org/project/am2320-driver/
- Install from the Thonny Python Editor (Tools -> Manage packages…). Search for “am2320-driver”
- or use pip:

```
pip3 install am2320-driver
```

am2320-driver

This AM2320 Python Library offers 3 functions:

- `get_temp()`
- `get_humi()`
- `get_humi_temp()`

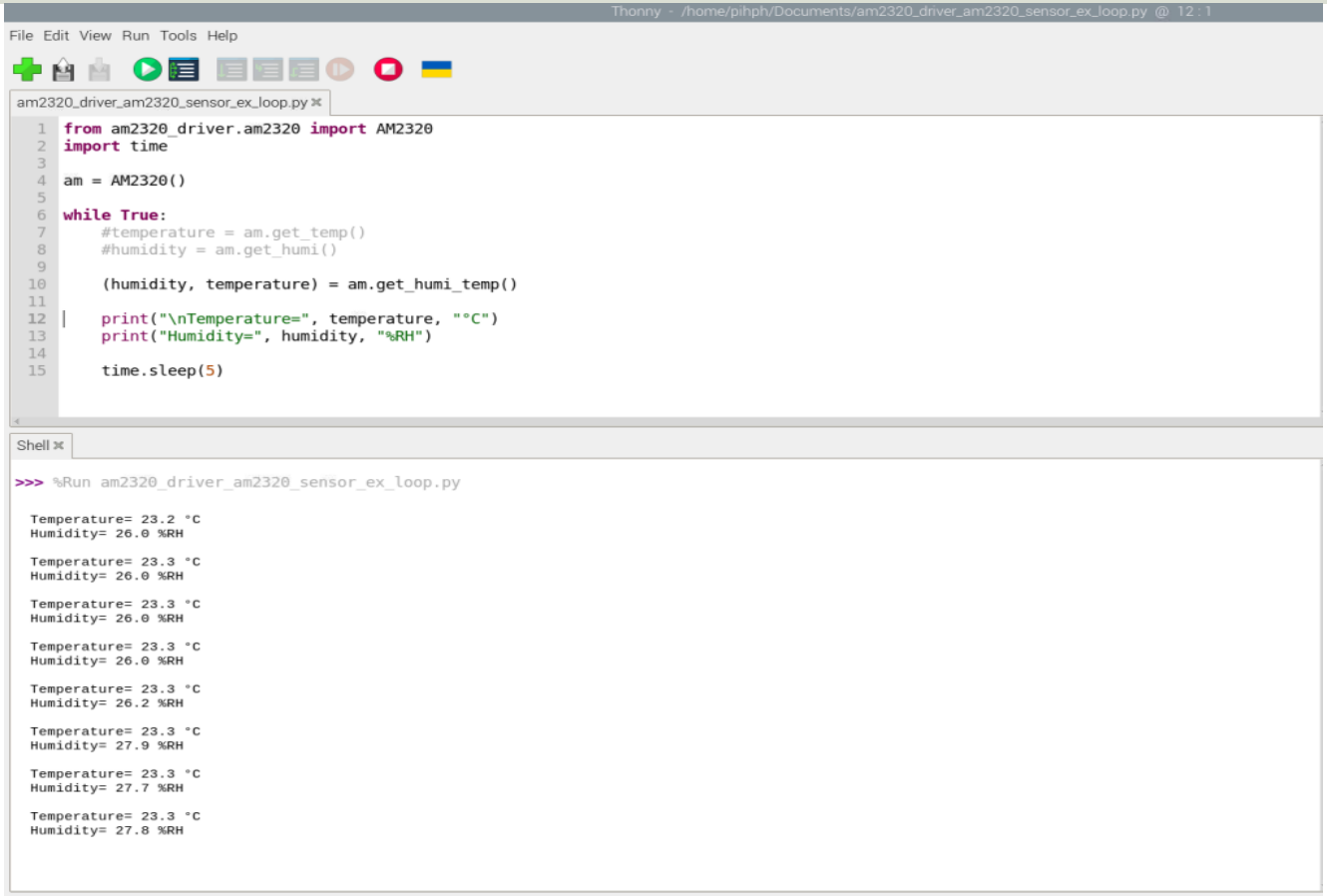
AM2320 Python Code Example

```
from am2320_driver.am2320 import AM2320
import time

am = AM2320()

while True:
    temperature = am.get_temp()
    humidity = am.get_humi()

    print("\nTemperature=", temperature, "°C")
    print("Humidity=", humidity, "%RH")

    time.sleep(5)
```

AM2320 Python Code Example

```
from am2320_driver.am2320 import AM2320
import time

am = AM2320()

while True:
    (humidity, temperature) = am.get_humi_temp()

    print("\nTemperature=", temperature, "°C")
    print("Humidity=", humidity, "%RH")

    time.sleep(5)
```

Results


Thonny - /home/pihph/Documents/am2320_driver_am2320_sensor_ex_loop.py @ 12 : 1
File Edit View Run Tools Help
am2320_driver_am2320_sensor_ex_loop.py
1 from am2320_driver.am2320 import AM2320
2 import time
3
4 am = AM2320()
5
6 while True:
7 #temperature = am.get_temp()
8 #humidity = am.get_humi()
9
10 (humidity, temperature) = am.get_humi_temp()
11
12 print("\nTemperature=", temperature, "°C")
13 print("Humidity=", humidity, "%RH")
14
15 time.sleep(5)
Shell
>>> %Run am2320_driver_am2320_sensor_ex_loop.py
Temperature= 23.2 °C
Humidity= 26.0 %RH
Temperature= 23.3 °C
Humidity= 26.0 %RH
Temperature= 23.3 °C
Humidity= 26.0 %RH
Temperature= 23.3 °C
Humidity= 26.0 %RH
Temperature= 23.3 °C
Humidity= 26.2 %RH
Temperature= 23.3 °C
Humidity= 27.9 %RH
Temperature= 23.3 °C
Humidity= 27.7 %RH
Temperature= 23.3 °C
Humidity= 27.8 %RH

Example 4

smbus I2C Library

Hans-Petter Halvorsen

Table of Contents

Introduction

- 3 Different Libraries has been tested
- Now I will create my own Examples by reading the Datasheet and using the low-level smbus Python Library for I2C Communication
- AM2320 Datasheet: https://cdn-shop.adafruit.com/product-files/3721/AM2320.pdf
- smbus: https://pypi.org/project/smbus/

smbus Python Library

SMBus (System Management Bus) is a subset from the I2C protocol

You can access I2C devices from Python using the smbus library:

```
import smbus
i2cbus = 1 #Default I2C Bus on Raspberry Pi
addr = 0x15 #am2320
bus = smbus.SMBus(i2cbus) # Initialize

#Write Data
bus.write_i2c_block_data(addr,cmd,vals[])
#Read Data
data = bus.read_i2c_block_data(addr,cmd)
```

https://pinout.xyz/pinout/i2c

https://raspberry-projects.com/pi/programming-in-python/i2c-programming-in-python/using-the-i2c-interface-2

AM2320 Datasheet

Reader sample:			
Function	Function Code	Start address	Frame data content
Read the temperature and humidity	0x03	0x00	Send: (SLA+W) +0x03+0x00+0x04
			Return: 0x03 +0 x04 + humidity + high + low temperature and humidity high temperature low + CRC
Read the temperature	0x03	0x02	Send: (SLA+W) +0x03+0x02+0x02
			Return: 0x03+0x02+High temperature + low temperature+ CRC
Read humidity	0x03	0x00	Send: (SLA+W) +0x03+0x00+0x02
			Return: 0x03+0x02+High humidity+ Low humidity + CRC

In the Datasheet for a given sensor you find all information you need. Here is some important excerpts

◎**Temperature output format**

Temperature resolution is 16Bit, temperature highest bit (Bit15) is equal to 1 indicates a negative temperature, the temperature highest bit (Bit15) is equal to 0 indicates a positive temperature; temperature in addition to the most significant bit (Bit14 ~ Bit0) indicates the temperature sensor string value. Temperature sensor value is a string of 10 times the actual temperature value.

AM2320 Datasheet: https://cdn-shop.adafruit.com/product-files/3721/AM2320.pdf

am2320sensor.py

```
import smbus
import time

i2cbus = 1 #Default
address = 0x5C #AM2020 I2C Address
bus = smbus.SMBus(i2cbus)

def WakeSensor():
  ..
def ReadTemperature():
  ..
def ReadHumidity():
  ..
def ReadTemperatureHumidity():
  ..
```

WakeSensor()

```
def WakeSensor():
    while True:
        try:
            bus.write_i2c_block_data(address, 0x00, [])
            break
        except IOError:
            pass

    time.sleep(0.003)
```

ReadTemperature()

```
def ReadTemperature():
    WakeSensor()
    while True:
        try:
            bus.write_i2c_block_data(address, 0x03, [0x02, 0x02])
            break
        except IOError:
            pass

    time.sleep(0.015)

    try:
        block = bus.read_i2c_block_data(address, 0, 4)
    except IOError:
        pass

    temperature = float(block[2] << 8 | block[3]) / 10
    return temperature
```

ReadHumidity()

```
def ReadHumidity():
    WakeSensor()
    while True:
        try:
            bus.write_i2c_block_data(address, 0x03, [0x00, 0x02])
            break
        except IOError:
            pass

    time.sleep(0.015)

    try:
        block = bus.read_i2c_block_data(address, 0, 4)
    except IOError:
        pass

    humidity = float(block[2] << 8 | block[3]) / 10
    return humidity
```

```
def ReadTemperatureHumidity():
    WakeSensor()
    while True:
        try:
            bus.write_i2c_block_data(address, 0x03, [0x00, 0x04])
            break
        except IOError:
            pass

    time.sleep(0.015)

    try:
        block = bus.read_i2c_block_data(address, 0, 6)
    except IOError:
        pass

    humidity = float(block[2] << 8 | block[3]) / 10
    temperature = float(block[4] << 8 | block[5]) / 10
    return temperature, humidity
```

Python Code Example

```
import time
import am2320sensor

while True:
    temperature = am2320sensor.ReadTemperature()
    print(temperature)

    humidity = am2320sensor.ReadHumidity()
    print(humidity)

    time.sleep(5)
```

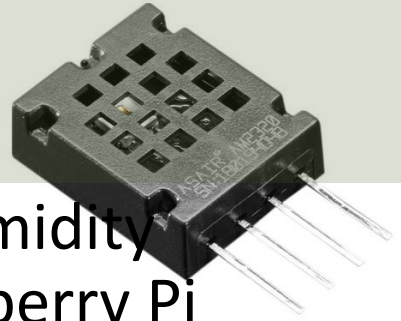
Improved Formatting

```
import time
import am2320sensor

i=1
while True:
    temperature = am2320sensor.ReadTemperature()
    humidity = am2320sensor.ReadHumidity()
    print(i, "Temperature:", temperature, "°C")
    print("Humidity:", humidity, "%RH\n")
    i = i + 1
    time.sleep(5)
```

ReadTemperatureHumidity() Example

```
import time
import am2320sensor

i=1
while True:
    t, h = am2320sensor.ReadTemperatureHumidity()

    print(i, "Temperature:", t, "°C")
    print("Humidity:", h, "%RH\n")

    i = i + 1
    time.sleep(5)
```

Summary



- In this Tutorial an AM2320 Temperature and Humidity Sensor has been used in combination with Raspberry Pi
- Many different Python Libraries and Examples exists
- 3 different libraries/examples were tested and was working after some trial and error and adjustments
- Finally, I made my own Python Code from “scratch” using the low-level smbus Python Library for I2C Communication
  - It has been implemented as a Python Module with functions for reading Temperature and Humidity
  - So far it is not available from https://pypi.org or https://github.com
  - But you can download it for free from my Website/Blog

Resources

- AM2320 Sensor Overview: https://learn.adafruit.com/adafruit-am2320-temperature-humidity-i2c-sensor
- Datasheet: https://cdn-shop.adafruit.com/product-files/3721/AM2320.pdf
- CircuitPython: https://learn.adafruit.com/circuitpython-on-raspberrypi-linux
- Adafruit am2320 Library: https://docs.circuitpython.org/projects/am2320/en/latest/index.html
- Gozem/am2320: https://github.com/Gozem/am2320
- am2320-driver: https://pypi.org/project/am2320-driver/

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog