```
#!/usr/bin/python

import time
import smbus


# HYT221 Class

class HYT221 :

    i2c = None
    messwerte = None

    address = 0x28

    #Constructor
    def __init__(self, address):
        self.i2c = smbus.SMBus(1)

        self.address = address

    def messung(self):      # Initialisiert neue Messung
        try:
            self.i2c.write_quick(self.address)
            time.sleep(0.05)        #kurze Wartezeit für den Sensor
        except IOError, err:
            return self.errMsg()

    def readRawResult(self):        # fragt aktuelle Messwerte ab
        try:
            self.messwerte = self.i2c.read_i2c_block_data(self.address, 0,4)
# Anforderung der Messergebnisse (4 Byte)
        except IOError, err:
            return self.errMsg()

    def readHumidity(self):         # Berechnet aus den letzten Messwerten die
Feuchtigkeit, fuer neue Messwerte ist ein Aufruf von messung() und
readRawResult() erforderlich
        rawHum = (self.messwerte[0]& 63) * 256 + self.messwerte[1]      #
Zusammenfuegen der 2 Messwerte-Bytes, Status-Bits entfernt

        Hum = 100.0 * rawHum / 16384.0          # Berechnung Feuchtigkeit,
siehe Datenblatt
        return Hum

    def readTemperature(self):      # Berechnet aus letzten Messwerten die
Temperatur, fuer neue Messwerte ist ein Aufruf von messung() und
readRawResult() erforderlich
        Temp = 0.0                                      #Initialisierung von
Temp
        rawTemp = ((self.messwerte[2] *256) + (self.messwerte[3] & 252))/4
# Zusammenfuegen der 2 MesswerteBytes, Entfernen der letzten 2 Bytes

        Temp = (165.0 * rawTemp / 16384.0) - 40.0

        return Temp

    def messungKomplett(self):              # Vollstaendiger Messablauf
```

```
        self.messung()
        self.readRawResult()
        Temperatur = self.readTemperature()
        Feuchtigkeit = self.readHumidity()

        return Temperatur, Feuchtigkeit
```