

Technische Hochschule Mittelhessen  
Fachbereich Informationstechnik - Elektrotechnik - Mechatronik

# LoRa based IoT endpoint and gateway

Bachelorarbeit

von

**Silvere Sacker Ngoufack**

Betreuer:

Erster Prüfer Prof. Dr. Hartmut Weber

Zweiter Prüfer Dipl.-Ing ...

Friedberg, den 26. Mai 2020

# Inhaltsverzeichnis

<b>Danksagung</b>	<b>iii</b>
<b>Selbständigkeit Erklärung</b>	<b>iv</b>
<b>1 Abstract</b>	<b>1</b>
<b>2 Einleitung</b>	<b>2</b>
2.1 Motivation . . . . .	2
2.2 Aufgabenstellung und Zielsetzung . . . . .	2
2.3 Gliederung der Arbeit . . . . .	3
<b>3 Komponenten und Software-Entwicklung</b>	<b>4</b>
3.1 STM32L4 Discovery kit . . . . .	4
3.1.1 HTS221 Temperatursensor . . . . .	5
3.1.2 LSM6DSL 3D Gyroscope und 3D Beschleunigungssensor . . . . .	8
3.2 LoRa Node: i-nucleo-lrwan1 . . . . .	8
3.2.1 LoRaWAN Protokoll . . . . .	8
3.2.2 AT Commandos . . . . .	8
<b>4 Gateway und LoRaWAN server</b>	<b>9</b>
4.1 Gateway . . . . .	9
4.2 Server . . . . .	9
<b>5 Fazit</b>	<b>10</b>
5.1 Zusammenfassung . . . . .	10
5.2 Ausblick . . . . .	10

# Danksagung

Ich möchte mich an dieser Stelle bei allen derjenigen bedanken, die mich während der Anfertigung dieser Bachelorarbeit unterstützt und motiviert haben.

Zuerst bedanke ich mich herzlich bei Jonh Martial Madieu mein Chef. Für seine hilfreichen Anregungen und seine konstruktive Kritiken bei der Entwicklung dieser Arbeit möchte ich mich bedanken.

Ich bedanke mich bei meiner Eltern, dafür das sie mich immer unterstützt haben.

Ein besonderer Danke gilt meiner Freundin und meinen Freunden, die durch ihre Fragen und Anmerkungen mein Wissen erweitert haben.

# Selbständigkeit Erklärung

Hiermit versichere ich, Silvere Sacker Ngoufack, die vorliegende Arbeit selbstständig und ausschließlich unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst zu haben. Inhalte dieser Arbeit, die wörtlich oder sinngemäß aus den Quellen entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit oder Teile daraus wurden in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegt und auch nicht veröffentlicht.

Fiedberg, den 26. Mai 2020

---

SILVERE SACKER NGOUFACK

# Abbildungsverzeichnis

2.1	LoRaWAN Netzwerk . . . . .	3
3.1	B-L475E-IOT01A Discovery kit [5] . . . . .	5
3.2	Humidity sensor analog-to-digital flow [3] . . . . .	5
3.3	Linear interpolation to convert LSB to rH % [3] . . . . .	7

# Tabellenverzeichnis

3.1	Kalibrierregister für relative Feuchtigkeit . . . . .	6
-----	---	---

# **1 Abstract**

## 2 Einleitung

Dieses Kapitel soll der Leser auf den Inhalt der Arbeit aufmerksam machen, ihn mit der Aufgabestellung vertraut machen und über die Strukturierung und Zielsetzung der Arbeit Auskunft geben.

### 2.1 Motivation

Objekte werden in der heutigen Tagen immer mehr mit elektronik und Inteligenz versehen. Die Leute wollen auf Grund dieser Entwicklung, dass Prozesse oder bestimmte Aufgaben ohne menschliches Eingreifen erledigt und miteinander vernetzt werden. Das System soll nur überwacht werden und die Ergebnisse zu bestimmte Zwecke benutzt werden.

Das Internet der Dinge (in englisch *internet of things*, Kurzform **IoT**) wird dazu benutzt, um die Interaktion zwischen Menschen und vernetzten elektronishchen Systemen zu vereinfachen.

### 2.2 Aufgabenstellung und Zielsetzung

Diese Bachelorarbeit beschäftigt sich mit der Entwicklung eines vernetzten Systems bestehend aus einem 3D Beschleunigungssesor, einem 3D Gyroskop, einem Temperatur- und Feuchtigkeitssensor. Die Sensoren messen Daten und übergeben diese an den STM32L475 Mikrocontroller.

Der Mikrocontroller soll die Daten verarbeiten und mit Hilfe von einem LoRa Modul drahtlos zu einem Server übertragen. Bevor die übertragung erfolgt, muss das LoRa-modul Zugang zu dem Server durch eine Gateway bekommen. Abbildung 2.1 gibt einen gesamten überblick über den Aufbau des gesamten Systems.



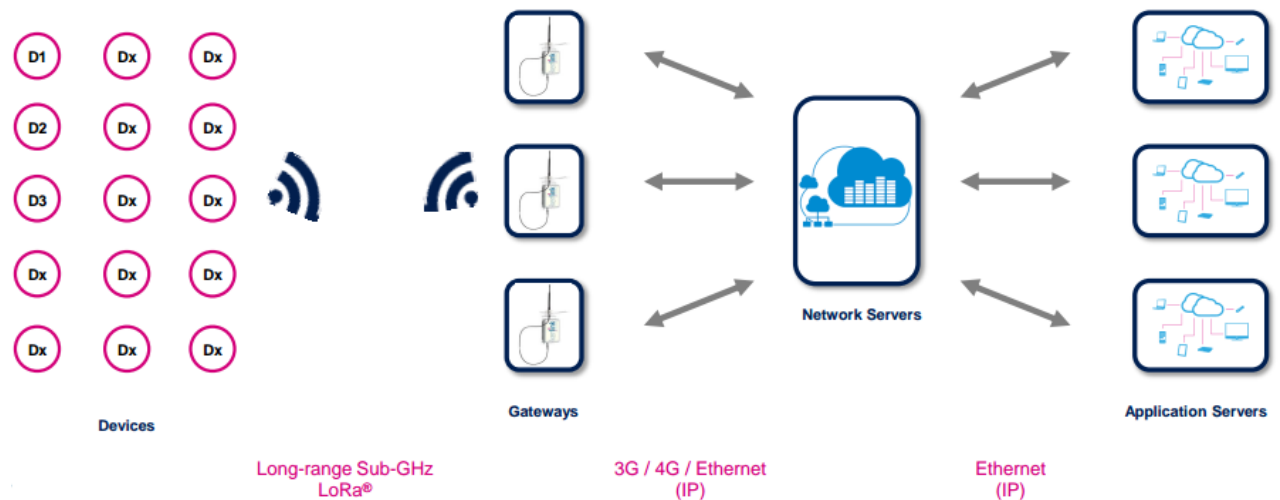


Abbildung 2.1: LoRaWAN Netzwerk

## 2.3 Gliederung der Arbeit

Das Kapitel 3 gibt einen detaillierten Überblick von allen Hardware-Komponenten, die bei der Durchführung dieser Arbeit verwendet worden sind. Noch dazu kommt die Erklärung der Software-Entwicklung, um diese Komponenten für ein vernünftiges Zweck anzusteuern. Als erstes wird auf die Eigenschaften von dem benutzten STM32-Nucleo Board eingegangen. Diesem Kapitel ist auch zu entnehmen, warum genau dieses Board gewählt wurde.

Als nächstes wird das LoRa-Node und das LoRaWAN Protokol beschrieben. Dieses LoRa-Node wird dazu verwendet, um die gesammelte Daten dem Server drahtlos zu übertragen.

Das Kapitel 4 erzählt wie der LoRaWAN-Server und das Gateway funktionieren und zeigt die Konfigurationen, die zu übernehmen sind, um eine Verbindung mit einem LoRa-Node herzustellen.

Anschließend im Kapitel 5 wird eine Zusammenfassung und einen Kleinen Ausblick der Arbeit gegeben.

## 3 Komponenten und Software-Entwicklung

In diesem Kapitel werden die Hardware-komponente, die in dieser auftreten in Einzelheit dargestellt. Noch dazu wird die Funktionsweise und die Ansteuerung dieser Komponenten erklärt.

### 3.1 STM32L4 Discovery kit

Das STM32L4 Discovery kit ist ein IoT Knoten, womit ein Benutzer Anwendungen mit direkter Verbindung zu einem oder mehreren Cloud-Server entwickeln kann. Dieses Discovery Kit ermöglicht eine Vielzahl von Anwendungen, indem es eine Multikink-Kommunikation (Bluetooth Low energie) mit geringerem Stromverbrauch nutzt, Multiway-Erkennung (Erkennung, Umweltbewusstsein) und die kernbasierten Funktionen der STM32L4-Serie von Arm Cortex®-M4 (Siehe Abbildung 3.1).

Das STM32L4 hat einen eingebetteten ST-LINK Debugger/Programmieren, Eingebettete Sensoren und viele andere Eigenschaften, die in dem Datenblatt [5] zu finden sind. Genau wegen der Vielfalt an Eigenschaften wurde dieses Board ausgewählt. Man braucht keine Breadboard im Vergleich zu dem Arduino oder dem Raspberry-pi, um die Sensoren mit den Schnittstellen (USART, SPI oder I2c) des Mikrocontrollers zu verbinden. Noch dazu eignet sich dieses Discovery Kit für ein LoRa-Modul von STMicroelectronics, da es Arduino-Verbinder vorhanden sind. Man soll nur das LoRa-Modul in diesen Verbindern stecken.

Für diese Arbeit werden wir uns auf zwei Sensoren beschränken. Erstmal den HTS221 Temperatur- und Feuchtigkeitssensor und dann den LSM6DSK 3D Gyroscope und 3D Beschleunigungssensor. Diese Daten werden erfasst und drahtlos an dem LoRaWAN-Server übertragen. Die folgenden Unterkapitel beschreiben wie diese Sensoren funktionieren und erklärt wie sie anzusteuern sind, damit die erhaltene Daten näherungsweise die Realität entsprechen.

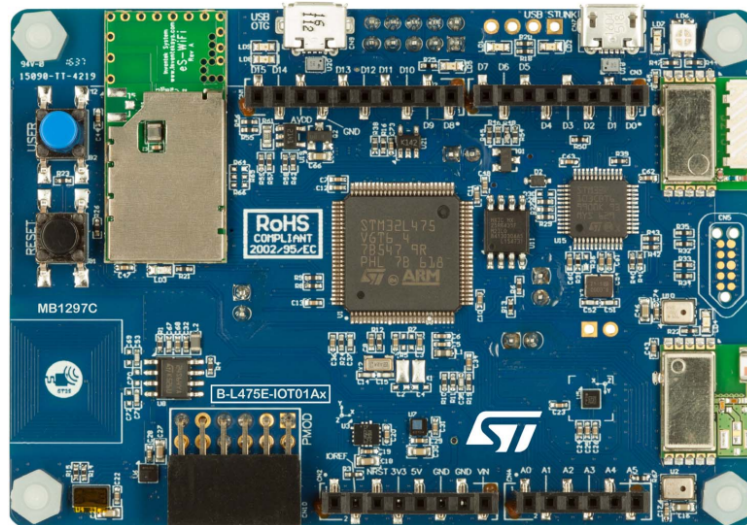


Abbildung 3.1: B-L475E-IOT01A Discovery kit [5]

### 3.1.1 HTS221 Temperatursensor

In diesem Unterkapitel wird der HTS221 Temperatur- und Feuchtigkeitssensor beschrieben und erklärt wie die Temperatur und die Feuchtigkeit zu ermitteln sind.

Der HTS221 Sensor misst die relative Feuchtigkeit und die Temperatur und speichert die Daten (16-Bits von Datentyp Integer) als zweierkomplement. Diese Daten können über I2C- oder SPI-Schnittstelle gelesen werden. Die gespeicherte Daten sind Rohdaten, die am Ausgang von dem Analog-Digital-Converter (ADC) verfügbar sind (Siehe Abbildung 3.2). Um die Temperatur in °C und die relative Feuchtigkeit in % zu bekommen, muss man die Daten aus den Registern konvertieren.

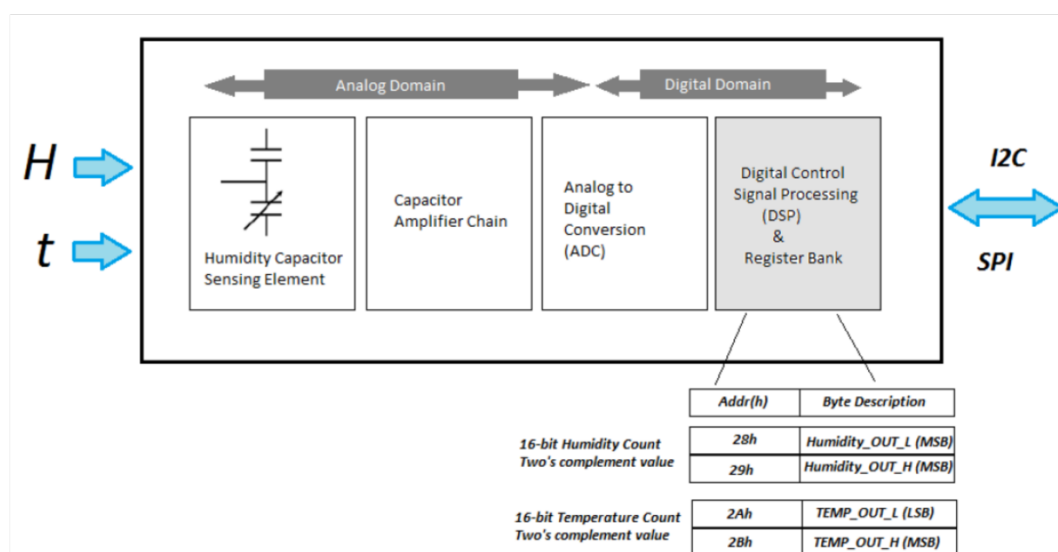


Abbildung 3.2: Humidity sensor analog-to-digital flow [3]

## Feuchtigkeit ermitteln

An dieser Stelle wird erklärt wie die Feuchtigkeit von dem Sensor ermittelt wird. Der HTS221 Sensor speichert den Feuchtigkeitwert in Rohzählungen in zwie 8-Bit-Registern:

- H\_OUT\_H (0x29) (Höchstwertige Bits)
- H\_OUT\_L (0x28) (Niedrigwertige Bits)

Die zwei Bytes werden verkettet, um ein zweierkomplement dargestelltes 16-Bit Wort zu bilden. Der relative Feuchtigkeitwert muss durch lineare Interpolation der Register (HUMIDITY\_OUT\_H & HUMIDITY\_OUT\_L) mit dem Kalibrierregistern berechnet werden.

Der HTS221 Sensor ist bei der Herstellung schon kalibriert und die erforderlichen Koeffizienten sind ADC 16-Bit-Werte, die in den internen register des Sensors zu lesen sind. Eine weitere Kalibrierung durch den Benutzer ist nicht erforderlich.

Die Tabelle 3.1 stellt die register dar, in dem die kalibrierwerte gespeichert sind.

Variable	Adresse	Format <sup>1</sup>
H0_rH_x2	0x30	u(16)
H1_rH_x2	0x31	u(16)
H0_TO_OUT_H	0x36	s(16)
H0_TO_OUT_L	0x37	s(16)
H1_TO_OUT_H	0x3A	s(16)
H1_TO_OUT_L	0x3B	s(16)

Tabelle 3.1: Kalibrierregister für relative Feuchtigkeit

Nun wissen wir welche Register zu lesen sind, damit die relative Feuchtigkeit mit Hilfe der Interpolation berechnet wird. Die folgenden Schritten müssen vor der Berechnung durchgeführt werden:

- Werte von H0\_rH\_x2 und H1\_rH\_x2 aus Registern 0x30 und 0x31 lesen
- H0\_rH\_x2 und H1\_rH\_x2 durch zwei teilen
- Werte von H0\_TO\_OUT aus Registern 0x36 und 0x37 lesen
- Werte von H1\_TO\_OUT aus Registern 0x3A und 0x3B lesen
- Rohdate von H\_T\_OUT aus Registern 0x28 und 0x29 lesen

<sup>1</sup>(u16) 16Bit-Wert ohne Vorzeichen, (s16) 16-Bit-wert mit Vorzeichen

Nach dem diese Register gelesen wurden, kann nun die Berechnung der relative Feuchtigkeit erfolgen.

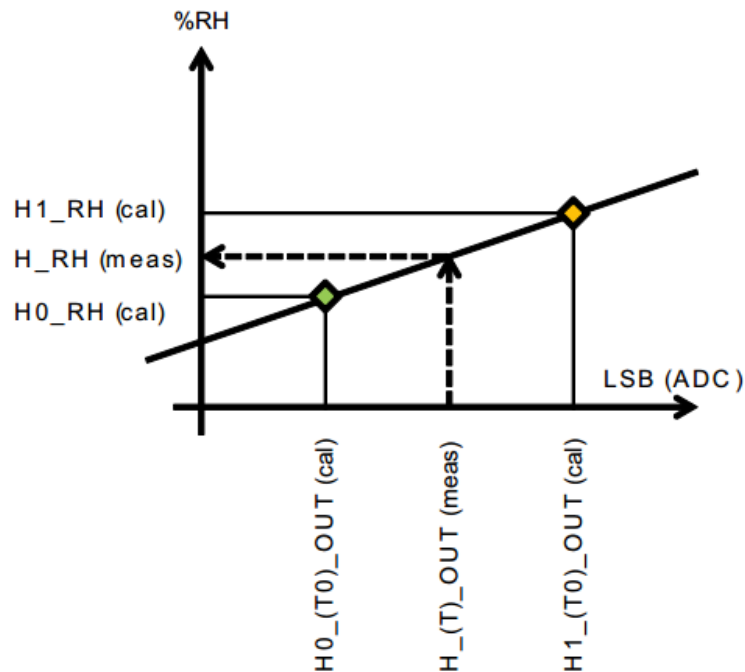


Abbildung 3.3: Linear interpolation to convert LSB to rH % [3]

Aus Abbildung 3.3 bekommt man mit Interpolation die folgende Formel:

$$RH\% = \frac{((H1_{rH} - H0_{rH}) \cdot (H_{T\_OUT} - H0_{T0\_OUT}))}{(H1_{T0\_OUT} - H0_{T0\_OUT})} + H0_{rH}$$

Da die kalibrierwerten bei der Herstellung des Bausteins schon festgesetzt sind, soll man die Kalibrierregister bei der Programmierung nur ein mal auslesen, dies erspart den Rechenaufwand des Mikrocontrollers.

### Temperatur ermitteln

Der HTS221 Sensor speichert die Temperaturwerten in Rohzählungen in zwie 8-Bit-Registern:

- T\_OUT\_H (0x2A) (Höchstwertige Bits)
- T\_OUT\_L (0x2B) (Niedrigwertige Bits)

### **3.1.2 LSM6DSL 3D Gyroscope und 3D Beschleunigungssensor**

## **3.2 LoRa Node: i-nucleo-Irwan1**

### **3.2.1 LoRaWAN Protokol**

### **3.2.2 AT Commandos**

## **4 Gateway und LoRaWAN server**

### **4.1 Gateway**

### **4.2 Server**

## **5 Fazit**

### **5.1 Zusammenfassung**

### **5.2 Ausblick**



# Literaturverzeichnis

- [1] libopencm3. libopencm3 lowlevel open-source library for arm cortex mcus. 2013. <https://github.com/libopencm3>.
- [2] saleae. Debug hardware like the pros with the logic analyzer you will love. 2020. <https://www.saleae.com/>.
- [3] STMicroelectronics. Capacitive digital sensor for relative humidity and temperature. STMicroelectronics, 2018. <https://www.st.com/en/mems-and-sensors/hts221.html#resource>.
- [4] STMicroelectronics. inemo inertial module: always-on 3d accelerometer and 3d gyroscope. 2018. <https://www.st.com/en/mems-and-sensors/lsm6dsl.html#resource>.
- [5] STMicroelectronics. Discovery kit for iot node, multi-channel communication with stm32l4. 2019. <https://www.st.com/en/evaluation-tools/b-l475e-iot01a.html#resource>.
- [6] STMicroelectronics. dm00083560-stm32l47xxx-stm32l48xxx-stm32l49xxx-and-stm32l4axxx-advanced-armbased-32bit-mcus-stmicroelectronics. 2019. <https://www.st.com/en/evaluation-tools/b-l475e-iot01a.html#resource>.
- [7] USI. Wm-sg-sm-42 at command reference manual. 04 2018. [https://github.com/USIWP1Module/USI\\_I-NUCLEO-LRWAN1/tree/master/preloaded\\_firmware](https://github.com/USIWP1Module/USI_I-NUCLEO-LRWAN1/tree/master/preloaded_firmware).