Bachelor Project

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Measurement

LoRa Based Smart Button

Šárka Květoňová

Supervisor: doc. Ing. Radislav Šmíd, Ph.D.
May 2018

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Květoňová Šárka** Personal ID number: **434915**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Measurement**

Study program: **Open Informatics**

Branch of study: **Computer Systems**

II. Bachelor's thesis details

Bachelor's thesis title in English:

LoRa Based Smart Button

Bachelor's thesis title in Czech:

Inteligentní tlačítko s LoRa komunikací

Guidelines:

Design and realize smart button with an e-ink display, additional binary/analog inputs, and LoRa communication interface. Use the NUCLEO-L073RZ development kit with an ultra-low-power 32-bit STM32L0 microcontroller. The device should be equipped with push buttons, e-ink display connected via SPI. Use the Thingsnetwork platform together with Node-RED (or alternative) programming tool for creating a simple demo application.

Bibliography / sources:

[1] STMicroelectronics datasheets, http://www.st.com
[2] The Things Network documentation, https://www.thethingsnetwork.org/
[3] Node-RED documentation, https://nodered.org/
[4] Ganssle, J.: The Art of Designing Embedded Systems, Elsevier Science, 2008.

Name and workplace of bachelor's thesis supervisor:

doc. Ing. Radislav Šmíd, Ph.D., Department of Measurement, FEL

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **09.01.2018** Deadline for bachelor thesis submission: ________

Assignment valid until:
by the end of summer semester 2018/2019

doc. Ing. Radislav Šmíd, Ph.D. Supervisor's signature	Head of department's signature	prof. Ing. Pavel Ripka, CSc. Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt Student's signature

Acknowledgements

I would like to thank my family and friends who have supported me in making this work. I also want to thank the leader of my work, doc. Ing. Radislav Šmíd, Ph.D., for the valuable advice and leadership of this work.

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

Prague, May , 2018

Abstract

This work deals with design and realization of smart button with an e-ink display, additional binary inputs, and LoRa communication interface. The core hardware consists from NUCLEO-L073RZ development kit with an ultra-low-power 32-bit STM32L0 microcontroller and I-NUCLEO-SX1272D RF expansion board.

The device is equipped with two push buttons, e-ink display connected via SPI.

The communication is realized using the Thingsnetwork platform together with Node-RED programming tool.

As a simple demo application was used a printer with two built-in sensors monitoring the presence of paper and toner.

Keywords: smart button, LoRa, Nucleo-L073RZ, SX1272, e-ink

Supervisor: doc. Ing. Radislav Šmíd, Ph.D.

Abstrakt

Práce se zabývá designem a realizací chytrého tlačítka s e-ink displayem, přídavnými binárními vstupy a komunikačním rozhraním LoRa. Základní hardware se skládá z vývojového kitu Nucleo-L073RZ s 32bitovým mikrokontrolérem STM32L0 s ultra nízkým výkonem a rozšiřující deskou I-NUCLEO-SX1272D RF podporující přenos.

Zařízení je vybaveno dvěma tlačítky a e-ink diplayem připojeným skrze SPI.

Komunikace probíhá pomocí platformy Thingsnetwork společně s programovacím nástrojem Node-RED.

Jako jednoduché demo byla použita tiskárna se dvěma zabudovanými senzory sledujícími přítomnost papíru a toneru.

Klíčová slova: chytré tlačítko, LoRa, Nucleo-L073RZ, SX1272, e-ink

Contents

1 Introduction 1

1.1 The current state of smart buttons 1

1.1.1 Dash Button 2

1.1.2 Flic 4

1.1.3 Philips Hue Tap 5

1.1.4 BT.TN 6

1.1.5 Dimple 7

1.2 Work objections 8

2 Introduction to the Internet of Things 9

2.1 Short look at the history of IoT . 9

2.2 Term of IoT 9

2.3 LPWAN technology 10

2.3.1 LoRaWAN 11

2.3.2 A class device 13

2.3.3 B class device 13

2.3.4 C class device 13

3 Realization of smart button 15

3.1 MCU 16

3.1.1 Nucleo-L073RZ 16

3.2 LoRa network interface 20

3.2.1 I-NUCLEO-SX1272D expansion board 20

3.2.2 The Things Network server . . 21

3.3 E-ink display 21

3.3.1 Principles of working 21

3.3.2 eINK click 22

4 Software/firmware 23

4.1 Mbed OS 5 23

4.2 STM32 ST-LINK Utility 24

4.3 Important libraries and files 25

4.3.1 Commissioning.h 25

4.3.2 SSD1606_SPI.cpp 25

4.4 NodeRed 26

5 Results 29

6 Conclusion 33

6.1 Work results 33

6.2 Future development 34

A CD 35

B General block diagram of the STM32L073xx device 37

C Simplified Block Diagram of SX1272-3 39

D Bibliography 41

Figures

1.1 Dash Button layout [2] 2
1.2 Dash Button - front side [8] 3
1.3 Dash Button - back side [8] 3
1.4 Samples of using the flic button [6] 4
1.5 Philips Hue Tap [12] 5
1.6 BT.TN use [26] 6
1.7 Two and four buttons modification of Dimple [3] 7

2.1 A view of a star and a mesh topology [27] 10
2.2 Scheme of communication device A class[1] 13
2.3 Scheme of communication device B class[1] 13
2.4 Scheme of communication device C class[1] 13

3.1 General scheme of interaction user with smart button 15
3.2 General scheme of smart button and its components 16
3.4 Top layout of STM32 board [24] 17
3.3 Hardware block diagram [24] . . . 17
3.5 NUCLEO-L073RZ and NUCLEO-L010RB board pinout [24] 18
3.6 STM32L073xZ circuit diagram [21] 19
3.7 I-NUCLEO-SX1272D expansion board pinouts[23] 20
3.8 MikroBUS pinout[13] 22

4.1 A sketch of a typical Mbed board’s hardware architecture[10] 23
4.2 Example of STM32 ST-LINK Utility GUI[22] 24
4.3 Picture of TTN devide overview[15] 25
4.4 Example of TTN application in NOode-RED[14] 27
4.5 Setting of NodeRED for application[16] 27
4.6 Setting of email on NodeRED[16] 28

5.1 Designed system of smart button with additional components 29
5.2 Designed system of two buttons and e-ink display 30
5.3 Example of NodeRED user interface for maintenance needs . . . 30
5.4 Application data from TTN[15] . 31
5.5 Flow-chart of application on NodeRED 31
5.6 Example of emails alert of printer defects 32

B.1 General block diagram of the STM32L073xx device 37

C.1 Simplified Block Diagram of SX1272-3 39

Tables

5.1 Description of values from buttons on TTN server[15] 31

Chapter 1

Introduction

Nowadays, no one will be surprised that automation is present in every sector of life. Every day we can meet with automatic locking of cars, coffee machines or vending machines for public transport tickets. People are more and more eager to automate even rare occasions such as buying food supplies, hygiene needs, or clothing. Automation is required not only for basic human needs but also in the industrial sector. Perhaps every machine requires regular maintenance. Such maintenance has been carried out by specially trained people to date. However, people are not always available 24 hours a day and can not be reached in all places where the machines are driven. Likewise, cables cannot be routed to all locations, where information about the current status of the product will be transmitted. For this reason, it was very pioneering to set up a wireless connection between the machine and the technical support. Not all the events related to the machine condition can be planned. Components wear out unevenly depending on the quality of the maintenance, the frequency of use and the conditions in which the product is operated. For example, if we introduce an industrial copier, we would like to check the ink value, the power supply or if the internal equipment is not exposed to too high temperatures. The company began to look for sensors that would watch these and many other phenomena. The main idea would be to use wireless information through an interface to a system that would process and process the information.

1.1 The current state of smart buttons

We have been in the market for a few years now with a range of wireless devices that allow us to meet customer requirements with a few clicks. From ordering your favorite lemonade, by starting the last hit of the selected artist to sharing your position via GPS on your mobile. This and many others have already been created to the satisfaction of all smartphone users. These buttons can be divided into several categories. One of the division concerns the method of data transfer. According to the microcontrollertips.com [8] Dash Button developed by Amazon is a Wi-Fi connected device that reorders a specific product with the press of a button. When a customer is running low on his favorite products at home, just press Dash Button to place an order with Amazon. Similarly, another button created by SHORTCUTlabs called Flic works. On the official website of Flic flic.io [CITACE flic.io], we can read that this product communicates, unlike the Dash button through Bluetooth 4.0.

1.1.1 Dash Button

To use the Dash Button from Amazon, the customer must download a special Amazon application and create a user account. In this Amazon account, a user adds a Dash button to his devices and can conveniently shop for the comfort of his home. The company offers two types of buttons. The first is a pre-set button which, when pressed, performs a preset product order, which is in most cases shown on the packaging. The second product has become the possibility for the user to set the button to order the product selected by him.

On the Amazon official website [2] we can find the exact distribution of the function buttons into categories:

- Household
- Snack/Food
- Beverages
- Pets
- Baby & Kids
- Beauty
- Health
- Exclusives & More

The customer can then order food for himself or a dog, but also diapers for his baby or even a washing powder, dishwasher cleaner and clothes.

In the following picture, Amazon.com offers a simple Dash Button layout. It can be seen that the button supports the two features below.

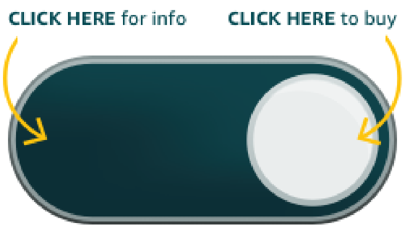


Figure 1.1: Dash Button layout [2]

The microcontrollertips.com website [8] also provides photographic documentation with descriptions of single components.

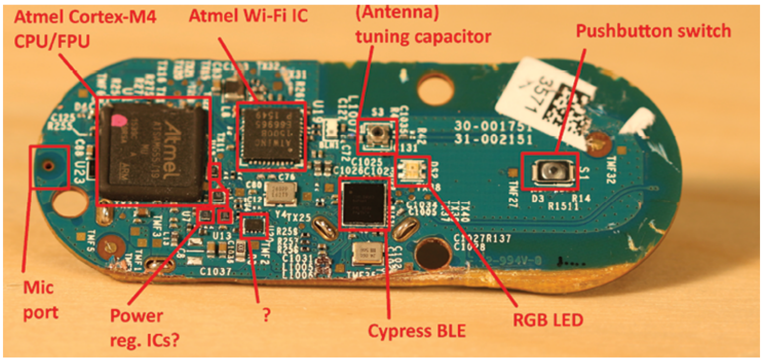


Figure 1.2: Dash Button - front side [8]

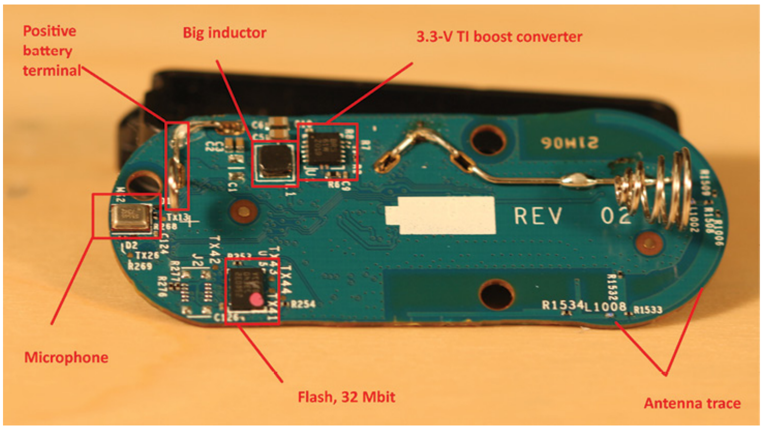


Figure 1.3: Dash Button - back side [8]

According to more information on microcontrollertip.com[8], the power for the Dash Button is powered by one AAA battery. Amazon.com[2] has a button life of up to 1,000 clicks before the battery runs out. According to microcontrollertips.com[8], service life is subject to a frequency of use. The following quote explains how long you can expect the button to work in run mode. *"Those who have analyzed Dash Button circuitry say it draws 200 to 300 mA when on, 2.3 µA when in sleep mode. Thus the Dash Button will last awhile when it only draws hundreds of milliamps during brief button pushes. But if operating as a proxy server or listening device, it would draw this kind of current nearly all the time. From the AAA cell's current versus service hour curves, the Dash Button might last just four hours if run in active mode this way."*, says the article[8].

1.1.2 Flic

As a representative of Bluetooth transfer, here is the product of the Swedish company Shortcut Labs called Flic. [7] Unlike the Dash Button, the Flic button uses Bluetooth technology for signal transfer. In addition to Bluetooth technology, Flic can work through WiFi and IR (the entire name of Infrared Technology). If we were interested in the principle of functioning, Flic works in three modes. Single click, double click and button hold.

The main advantages of Flic buttons [6] include easy setup. Just connect Flic to The Flic App and then choose commands. One Flic button can be connected to one end device. Each button operates according to the manufacturer's data at a distance of up to 50 meters. Thanks to low-cost Bluetooth technology, battery life is up to 2 years. If we are interested in compatibility, Flic is compatible with Android 4.4+, iOS 9+ and MacOS Sierra. Flic can be set to control lamps, start Spotify, take a selfie or for calls to friends. Flic is compatible with IFTTT, Sonos, Spotify and plenty of other smart home systems.

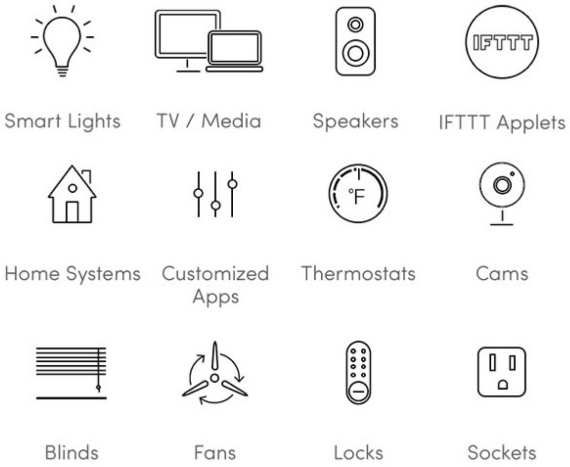


Figure 1.4: Samples of using the flic button [6]

The power supply of the Flic button is solved by a 3V replaceable CR2016 battery. Shortcut Labs has created several videos to help users with basic button setup or battery replacement. The button can be easily dismantled and then folded again. The replaceable back cover gives the user the option to attach the button to a flat surface using a washable rear glue or to attach a strap to the back of the button, which can then be attached to the clothing or backpack to keep the Flic button together.

1.1.3 Philips Hue Tap

Other technologies emerging on the market include ZigBee technology. The representative of this technology was created by Philips. [12] Philips Hue tap is a clever light switch with four buttons. The product does not need power from external batteries, the energy for operation is generated by kinetic energy at every click. The product must be reconnected with pre-installed software, in this case, the Philips Hue app. Here users can preset button functions. More confusingly, the app configures the hub to recognize the incoming tapes and output the appropriate controls to the connected lights. The Philips Hue Tap according to the manufacturer works at a distance of 30 meters. As in the case of Flic, it is also possible to have the product either firmly attached or worn around the building. Disadvantages of Philips Hue Tap are seen by users at a high purchase price.

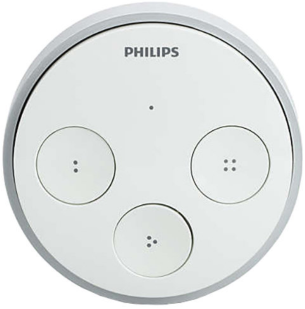


Figure 1.5: Philips Hue Tap [12]

1.1.4 BT.TN

No less important product using wireless technology is BT.TN. The big red button [4], which used either mobile data or Wi-Fi, now offers models that support Sigfox technology. The button is produced in two size versions and was previously intended for use in large companies only. But the use of home use is inexhaustible from mitigating all unpleasant sleeping effects such as lights from nearby streets, turning on the night mode on appliances, and sending messages when your spike arrives safely home. According to the first company project[19] user can set BT.TN to execute controlling almost any Internet-connected device or service but the button has, e.g., many predefined tasks:

- Send messages: email or text messages
- Send social media messages: Tweets or Facebook status updates
- Submit any HTTP get or post request, i.e., control a web page
- Control smart home devices: OpenHome and SmartThings compliant
- Trigger IFTTT recipes or Zapier zaps
- Publish an RSS feed
- Update counters on the web
- Everything else through REST API
- And much more (no kitchen sink, though)...

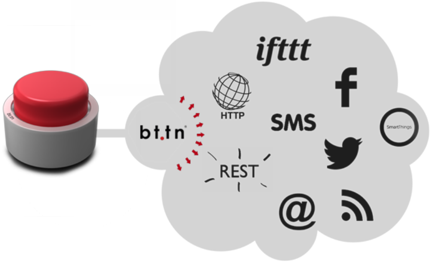


Figure 1.6: BT.TN use [26]

The button implements three main functions [4], namely short press, long press and "not pressed." The bttn actions are run by the bt.tn cloud server. Wi-Fi bttns can also send HTTP requests directly to the local network server. Powering a small red button is powered by a rechargeable Li-Po battery, a large model uses 4x AA alkaline batteries, both models support charging with a micro-USB charger.

1.1.5 Dimple

Dimple [3] was created as a small NFC sticker with buttons for Android devices. It works with Android 4.0+ devices and as other mentioned buttons need an application in a phone for its configuration. For the best function is recommended to put the Dimple sticker on a non-metal phone body.

Use of Dimple can be separated into four different section:

- Settings and Application
  - Change phone settings (Bluetooth, Wi-Fi, ringer/media volume, screen brightness, etc.)
  - Launch an application
  - Join a Wi-Fi Network
  - Show a message
- Communication
  - Make a call
  - Send a text message
  - Start a Google Talk conversation
  - Share a contact or business card
- Location and Web
  - Show an address on a map
  - Open a web page
  - Foursquare or Facebook check-in
- Social
  - Automatic Facebook "Like"
  - Update Facebook status
  - Post a tweet or follow a contact on Twitter
  - Connect on LinkedIn

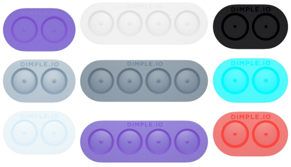


Figure 1.7: Two and four buttons modification of Dimple [3]

Dimple need for its functions no battery. For the successful using of NFC technology is neccessary to stick the sticker near the NFC module on the mobile phone. NFC technology works between devices in maximum distance 5 centimeters.

1.2 Work objections

The aim of this bachelor thesis is to design and implement a smart button system containing the following components:

- E-ink display Mikroe-2659
- Additional binary/analog inputs
- LoRa communication interface

For realization should be used the NUCLEO-L073RZ development kit with an ultra-low-power 32-bit STM32L0 microcontroller. The device should be equipped with push buttons and e-ink display connected via SPI.

Further will be used the Thingsnetwork platform together with Node-RED (or alternative) programming tool for creating a simple demo application.

Chapter 2

Introduction to the Internet of Things

2.1 Short look at the history of IoT

The term Internet of Things (IoT) was first used in 1999 during a presentation introducing the possibility of Radio Frequency Identification (RFID) for Procter & Gamble by Kevin Ashton, one of the founders of research center named AutoID at Massachusetts Institute of Technology (MIT). Ashton spoke for the first time about the possibility of linking everyday needs between each other, Machine to Machine (M2M) technology, through the network. Though his idea describes the base meaning of IoT, it is not appropriate to consider it as a suitable definition of this issue. Alone the idea of linking physical devices to the network as part of their joint communication was established in 1970. However, it was not possible to do for not enough IP addresses to connect so many devices. That is a reason why IoT came into the subconscious to the general public only recently with the advent of IPv6 and could still expand. [5]

2.2 Term of IoT

IoT in IT indicates the connection of built-in devices with the Internet. It was created due to the need for connecting a growing number of end devices. These devices are often spaced apart over long distances. Their location is not constant, they are often found deep beneath the earth's surface, under the water or deep inside the buildings. It is not always possible to connect or implement the Internet using a physical connection, such as a cable. This limitation has created a desire for connecting the device using wireless communication. There have been some technology interconnecting devices every day with a global network. In summary, they were these technologies called M2M, which means communication between similar type devices.[5]

Continuously created several technologies linking everyday needs with a global network. In summary, they were these technologies called M2M (Machine to Machine), which means communication between devices of similar type. [5] This category includes a 2G network, later replaced by LTE networks that allowed to communicate with larger bandwidths, but at the cost of increased energy consumption. No increase in consumption was caused by the price increase technology; There was a need to develop a technology that will work with lower energy consumption but will retain you have the required connection properties. The Low Power Wide Area Network (LPWAN) technology has emerged for this. [9]

2.3 LPWAN technology

Based on the conference in September 2017 [9], LPWAN is a low-power technology. He can also communicate over a long distance. By this time, the prior art used network mesh topology to limit the distance between the nodes. With the advent of LPWAN, which uses the topology of the star, it was already possible to calculate the distance of the end node per kilometer. There are, of course, exceptions, such as IRQF technology, which even today do not communicate at a distance of more than a few dozen meters. The star topology marks the linking of the device to a star shape. This is the most used method linking the device to the network.

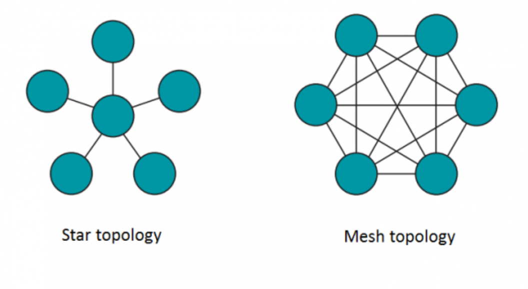


Figure 2.1: A view of a star and a mesh topology [27]

The star topology brings simplification of network management and the reduction of the necessary firmware on the end devices. In LPWAN, star topology is implemented by link gate terminals, the gateway, which is the only common point for all communication participants.

The gateway then sends the received message to the central network server. LPWAN the technology uses a large amount of protocol. The most famous may include SigFox, RPMA, NB-Fi, Weightless, DASH7 or LoRaWAN. Some companies focus on the development of devices targeted for the protocol. This entails the advantage of using licensed frequencies.

2.3.1 LoRaWAN

Main characteristic of LoRaWAN

- 250-byte packets
- Confirmed and unconfirmed messages
- Port-level communication (1 to 223)
- Uplink and downlink communications
- Advanced Encryption Standard (AES) 128
- Unique ID for each end device (EUI64, Extended Unique Identifier 64bit)
- two modes of activation of end devices
    - OTAA (Over-the-Air-Activation)
    - ABP (Activation By Personalization)
- 32-bit address of end devices
- Several security keys (AES-128)
    - Application Key (128 bit)
    - Network Session Key (128 bit)
    - Application Session Key (128 bit)
- Transfer rate from 250 bps to 50 kbps

Detailed view of LoRaWAN

By an article in the ICT Express magazine from 2017 [20], LoRaWAN is a communication protocol as well as system architecture, based on the principle of radio network ALOHA developed at the University of Hawaii in the 1970s. LoRaWAN is designed for systems using a physical Lora layer from the Semtech company based in California, which we will discuss later. LoRaWAN, as well as LPWAN, uses for connecting end nodes to the central network server the star topology. LoRaWAN is predominantly used in Europe and North America. We focus primarily on the application of the protocol in Europe. Here, the bandwidth used for transmission is limited to the range 125 kHz - 250 kHz. There is also access a separate FSK channel for higher data transfer. In Europe, LoRaWAN can use only ten channel for data transmission, in contrast to North America, where 64 channels are reserved for LoRaWAN.

LoRaWAN has to deal with the duty cycle limitation on both continents, but neither at one of these locations is not limited by the channel time limitation. LoRaWAN uses three common 125 kHz channels for a bandwidth of 868 MHz. Specifically, 868.10, 868.30 and 868.50 for connecting terminals to the network. Once a node is connected to a network server in the device can provide an additional channel. In Europe, the same channel is used for uplink (bandwidth 125 or 250 kHz) and downlink (bandwidth of 125 kHz). [20] Both, the uplink and the downlink, communicate with the terminal with each other. However uplink means communication is established by the end node to the server, while the downlink describes the efforts of the connection from the server to the end device. [9]

- Uplink[1]

LoRa packets based on an explicit header with a Cyclic Redundancy Check (CRC) at the end of the packet.

Preamble	PHDR	PHDR_CRC	PHY payload	CRC

- Downlink[1]:

Lora packets based on the explicit header without the CRC at the end of the packet.

Preamble	PHDR	PHDR_CRC	PHY payload

LoRaAlliance states [1], each Gateway implements LoRa, and can also potentially include Ethernet or other cellular connection capabilities to connect to the network server. The transmitted signal from the end node side receives any nearby Gateway, so the given signal can receive multiple Gateways and send it to the network server. If the server receives the message in multiple copies, the server retains only one message and sends the response to the optimal Gateway. LoRaWAN uses the so-called Adaptive Data Rate (ADR) and multichannel multi-mode broadcast.

Unlike some other protocol standards that are protected and therefore inaccessible to the public, the company LoRaAlliance has made the LoRaWAN protocol standard open. During the design of the LoRa specification, different applications were considered for which the network could be used. And because there is a wide variety of applications with different communication and energy-saving requirements, three different device classes have been designed.

2.3.2 A class device

Type A devices have the lowest power consumption of all three classes. The end device establishes communication with the server as required. Shortly after the server receives a message with the end device side, two separate downlink windows are created to communicate with the endpoint on the server side. Communication of this type of device is realized only through unicast[1].

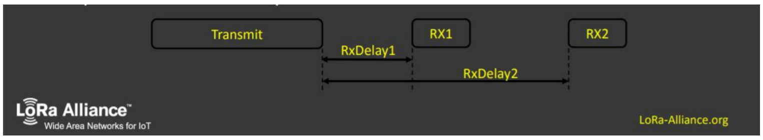


Figure 2.2: Scheme of communication device A class[1]

2.3.3 B class device

Device Class B extends the functionality of the device a lower class, i.e. the class type A. Unlike Class type A, has a class B type higher power consumption and allows communication through multicast. This is due to the fact that the server is establishing communication at fixed intervals and not only at the request from the end device.[1].

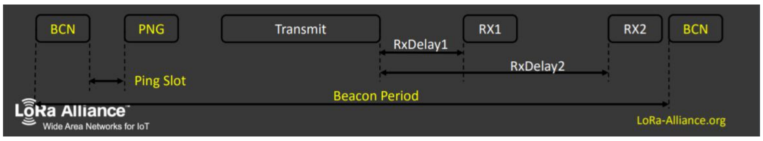


Figure 2.3: Scheme of communication device B class[1]

2.3.4 C class device

Of all three types of devices, C-type devices have the highest power consumption. The C-class extends the functions of the previous classes. However, devices of this type are listening for the whole time they are not broadcasting. As a result, these devices have a zero delay during communication with the server.[1].

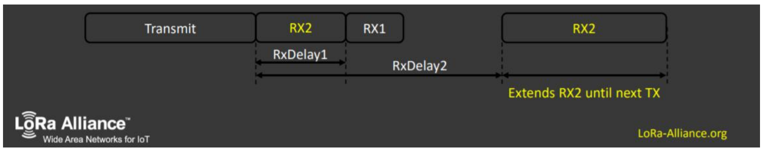


Figure 2.4: Scheme of communication device C class[1]

Physical layer LoRa

The LoRa physical layer is derived from Chirp Spread Spectrum Modulation (CSS) with built-in Forward Error Correction (FEC). In Europe and North America uses the band Industrial, Scientific and Medical (ISM), which is located below 1 GHz. Broadband nature allows better compensation for Signal To Noise Ratio (SNR). Demodulation of the signal is carried to the level of 19.5 dB below the noise floor. CSS modulation allows longer communication range than Frequency Shift Keying (FSK) modulation, even without increasing power consumption. CSS modulation also provides immunity to the Doppler effect.[9].

Thanks to broadcast at higher power levels is achieved by increasing the range of end nodes. In Europe, the maximum transmitting power is +14 dBm. The data transfer rate depends on the location as well as the Spreading Factor (SF). SF is defined for this technology in the range SF7 to SF12. By increasing the SF we can achieve a higher SNR but at the cost of increasing the transmission time. Increasing bandwidth will reduce transmission time, but will also reduce the sensitivity of the receiver. Detection of activity on a channel is mediated by Carrier Activity Detection (CAD). CAD brings an increase in activity detection rate compared to the Received Signal Strength Indicator (RSSI). It also provides better recognition of the difference between the noise and the desired LoRa signal.[20]

Chapter 3

Realization of smart button

For the realization of the smart button was selected development board NUCLEO-L073RZ from STMicroelectronics. The board includes an extension of the I-NUCLEO-SX1272D LoRa® RF expansion board from Semtech and a sub-gigahertz SMA antenna.

Figure 3.1 shows the general idea of user interaction with a smart button and sending an email with a specific error message using LoRa network Interface.

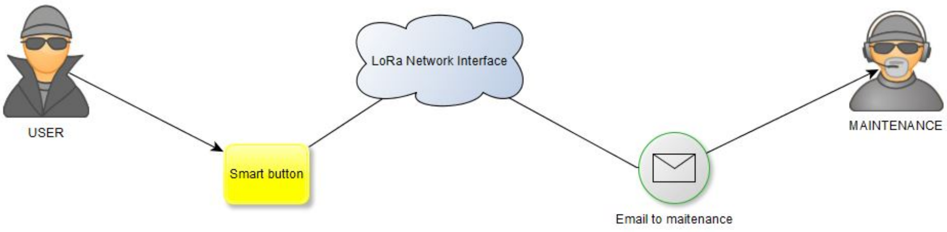


Figure 3.1: General scheme of interaction user with smart button

Further, on the figure 3.2 can be seen a general diagram smart button. The figure shows all important components from which the button is composed. There is also an e-ink display, as well as a demonstration of sending an email using the NodeRED GUI.

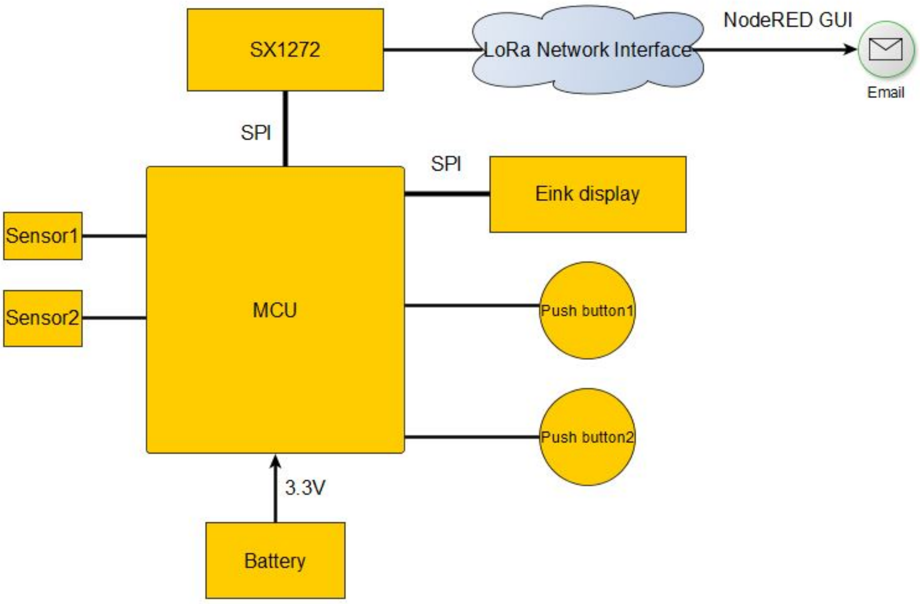


Figure 3.2: General scheme of smart button and its components

3.1 MCU

3.1.1 Nucleo-L073RZ

The NUCLEO-L073RZ [23] contains the MCU STM32L073RZ. It is a 32-bit microcontroller based on the Cortex®-M0 + core, which has 192 Kbyte Flash memory and 20 Kbyte SRAM. The written properties together with the peripheral set enable insertion of the LoRa middleware layer. Among other things, the NUCLEO-L073RZ board contains the ST Morpho extension header for access to STM32 I / O pins. Next development board has an integrated Debian / programmer ST-LINK / V2-1 with an SWD connector that is also capable of controlling serial communication with STM32L073RZ.

The development kit can be connected to a computer via USB. However, the battery power is used to realize the ideal end device. The datasheet contains precise instructions on how to implement this power supply. We connected the battery holder to a 3 x AAA (or AA) battery on the back of the NUCLEO-L073RZ board and connected the +/- terminals to the power pins on the ST morpho connector. We set the JP5 jumper to E5V, so the board takes power from the battery. Now, resistor R32 can be removed to reduce the current consumption of the board. On JP1 we placed a jumper. This enabled the ST-LINK debugger to release the reset pin of the STM32 target device when the ST-LINK USB device is not connected and calculated on the computer. The disconnected junction enabled the firmware STM32 to start firmware using the NUCLEO-L073RZ board connected to the computer via USB.

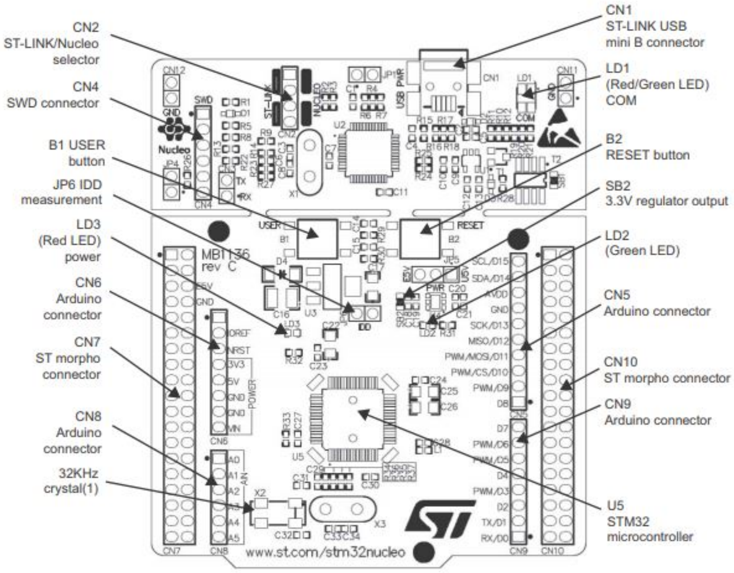


Figure 3.4: Top layout of STM32 board [24]

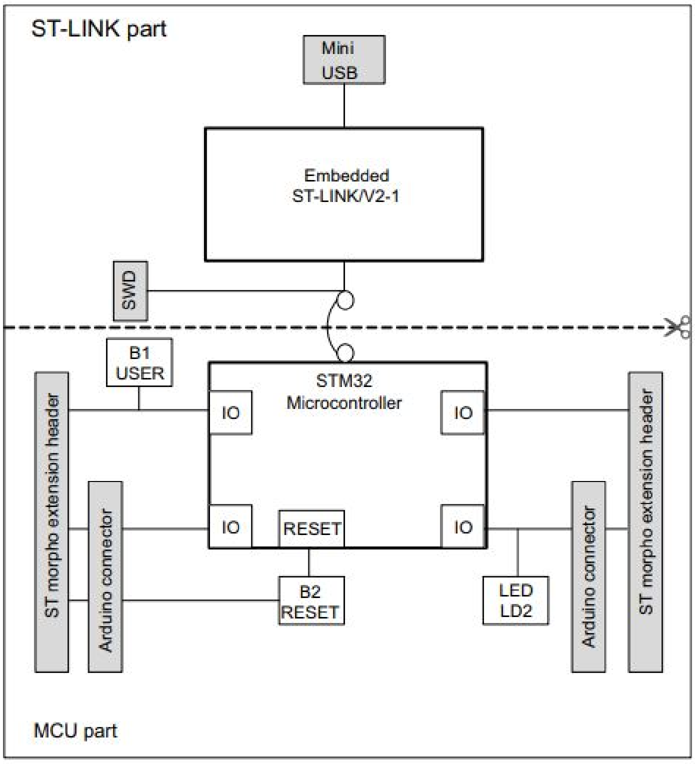


Figure 3.3: Hardware block diagram [24]

The STM32 Nucleo development kit[24] is divided into two parts: ST-LINK and STM32. The division can be seen in the figure3.3. If we want to reduce the size of the plate, the plate PCB ST-LINK can be removed.

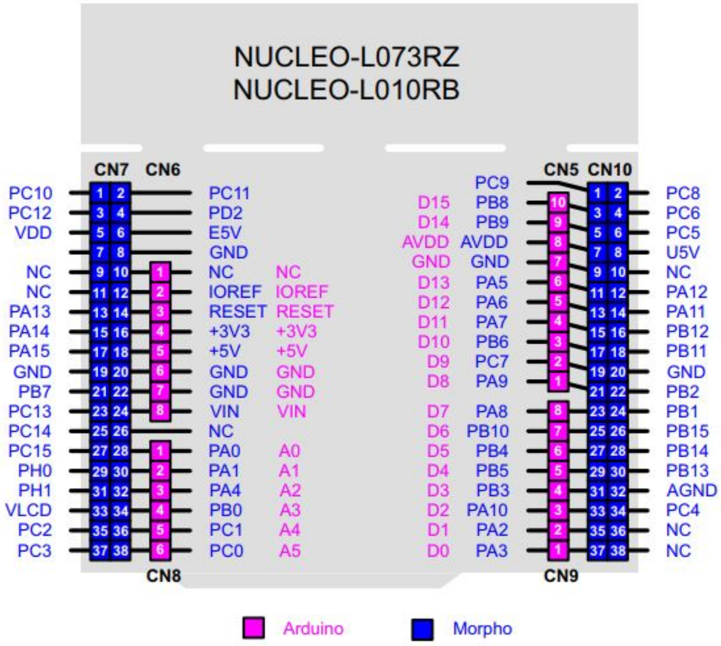


Figure 3.5: NUCLEO-L073RZ and NUCLEO-L010RB board pinout [24]

If the ST-LINK part is removed, the remaining target part of the STM32 must only be powered by VIN, E5V, and 3.3V on the STN of the CN7 or VIN connector and 3.3V on the Arduino CN6 connector. The ST-LINK part is still usable for programming the main part of the STM32 board using the cables between the CN4 and SWD signals available on the ST morpho connector, more specifically SWCLK CN7 pin 15 and SWDIO CN7 pin 13.

Accurate placement of individual function elements on the board is shown in figure 3.4. The figure 3.5 describes the layout of the pins on the board. For more information on the specific pin functions, refer to the official STMicroelectronics manual for Nucleo-L073RZ [24], where the exact pinout is described in the tables in the Hardware layout and configuration section. Another source to explore pinout development kit became the site of MBed company [11], which deals with the board Nucleo-L073RZ and pinout are color-divided according to each function.

Figure 3.6 describes which specific functions are on the board. Here is shown, for example, what types of connections we can expect to use or how many ADC channels are present on the board.

For a more detailed understanding of the interconnection of the individual processor components, a general block diagram of the STM32L073xx device[25] is attached in the Appendix B.1 section.

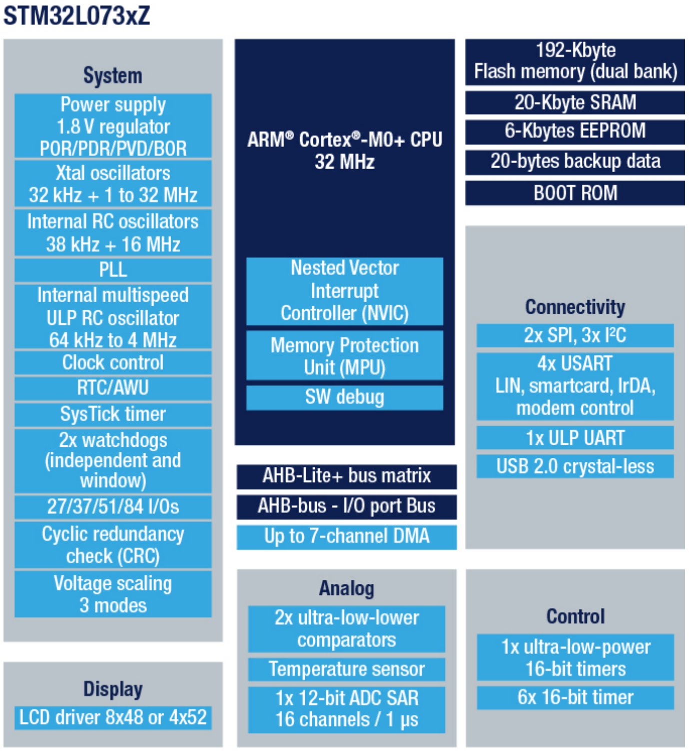


Figure 3.6: STM32L073xZ circuit diagram [21]

To connect other components to the development kit, the pins were mainly used from part CN7. In the right half of the CN7, there are Power and control pins. From there, 3.3V and GND were connected. These were later developed into e-ink display, user buttons, and simulation switches. The specific connection of the display to the board will be described in the section 3.3. User buttons and simulation switches were connected from the CN10 section. More specifically, the buttons from PC_6 and PC_8 pins. Switches, simulating the state of paper or toner, from pins PA_6 and PC_5. Buttons and switches can be connected freely to any free pin, except for pins connected to other components. Such pins are, for example, pin PA_5 on parts CN10 and CN9 or pins PA_2 and PA_3 on parts CN9.

3.2 LoRa network interface

3.2.1 I-NUCLEO-SX1272D expansion board

The I-NUCLEO-SX1272D[23] is the code name for the SX1272MBDAS expansion board used by STMicroelectronics. The extension includes the LoRa RF transmitter SX1272 and all associated RF matching and component filtering to enable a single, 50-ohm antenna port. The kit is wearing a 32MHz crystal.

Connection to other boards is made possible through SPI communication and interrupt lines. In the figure, 3.7 can be seen the distribution of the specific pins of the expansion board. Extensions can only be connected to the development kit in one way. The expansion board pins connect to the Arduino^{TM} UNO V3 pins the development kit.

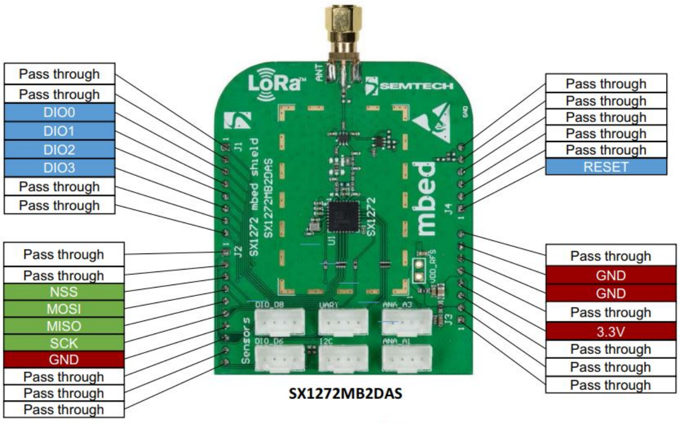


Figure 3.7: I-NUCLEO-SX1272D expansion board pinouts[23]

For a more detailed understanding of the communication inside the SX1272, a simplified block diagram of the SX1272 [18] is attached in the AppendixC.1 section.

Main features of SX1272:[18]

- LoRa Modem
- 157 dB maximum link budget
- +20 dBm at 100 mW constant RF output vs. V supply
- +14 dBm high-efficiency PA
- Programmable bit rate up to 300 kbps
- High sensitivity: down to -137 dBm
- Bullet-proof front end: IIP3 = -12.5 dBm
- 89 dB blocking immunity

- Low RX current of 10 mA, 100 nA register retention
- Fully integrated synthesizer with a resolution of 61 Hz
- FSK, GFSK, MSK, GMSK, LoRa and OOK modulation
- Built-in bit synchronizer for clock recovery
- Preamble detection
- 127 dB Dynamic Range RSSI
- Automatic RF Sense and CAD with ultra-fast AFC
- Packet engine up to 256 bytes with CRC
- Built-in temperature sensor and low battery indicator

3.2.2 The Things Network server

The Things Network[15] creates a network for the Internet of Things by creating rich data connectivity. The technology Things Network uses is the previously mentioned LoRaWAN that allows the device to communicate via the Internet without the use of 3G networks or WiFi. As already mentioned, LoRaWAN operates at low power consumption for long distances and low bandwidth, which is a perfect combination for the transmission of communications hitch IoT. Things Network is an extensive network of communities where individual members around the world use the network server and gateways, discuss the possibilities of use and help each other. There are also many innovative applications and IoT applications available, as well as all documentation for applications, end-products, gateways, and networks.

3.3 E-ink display

3.3.1 Principles of working

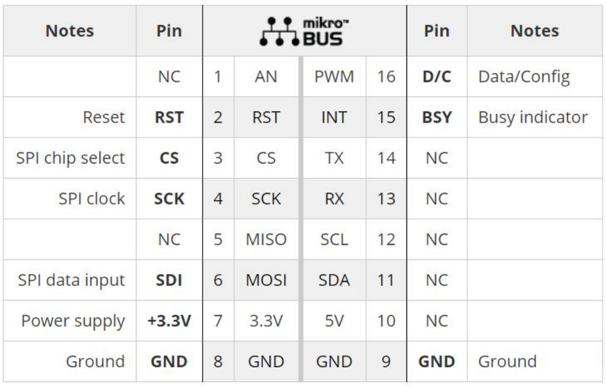
Based on datasheets for our used display [13], the e-ink display can be defined as a set of microcapsules inside which are black and white pigments. The extraordinary display contains a clear liquid in which the pigments move freely between the two electrodes. White pigments are positively charged, while the black pigments are negatively charged. When an electromagnetic field is generated between the electrodes, pigments repel or attract the electrodes, depending on the orientation field. The result of this reaction is the placement of light or dark pigments towards the bottom or top surface of the microcapsules. As black pigments are located in the upper part of the microcapsule, they are shown as black pixels and vice versa. It is also possible to place the dose of pigment so that each color occupy half of the top surface of the microcapsules and thus arose the gray shades.

3.3.2 eINK click

eINK click [13] is the Clickboard ^{TM} adapter that connects the compatible eINK display to the host MCU. The Click ^{TM} board comes with a 172x72 eINK display, powered by the integrated SSD1606 controller, and can display four shades: black, dark gray, light gray and white. This type of impression is used, for example, on Kindle E-readers.

The figure 3.8 shows the pinout that is used to connect the e-ink display to the host MCU. Power supply 3.3V and GND were involved from the MCU Nucleo-L073RZ part CN7. For the connection Reset signal was selected PA_3 pin, SPI chip select (CS) is connected next to, on the PA_2 pin. Both of these pins are located in the left half of the CN7; the other described pins will be found only on the right side of the CN7.

MOSI, or SPI data input, is connected to the PA_4 pin. MOSI represents a data output from master to slave. For communication, it is also important to have connected SPI clock, abbreviated SCK. Thanks to SCK, data can be generated. The SCK is implemented on the PB_0 pin. No less significant signals for communication through the SPI are Data/Config (D/C) connected to pin PC_0 and Busy Indicator (BSY) connected to pin PC_1.



Notes	Pin	mikroBUS				Pin	Notes
	NC	1	AN	PWM	16	**D/C**	Data/Config
Reset	**RST**	2	RST	INT	15	**BSY**	Busy indicator
SPI chip select	**CS**	3	CS	TX	14	NC	
SPI clock	**SCK**	4	SCK	RX	13	NC	
	NC	5	MISO	SCL	12	NC	
SPI data input	**SDI**	6	MOSI	SDA	11	NC	
Power supply	**+3.3V**	7	3.3V	5V	10	NC	
Ground	**GND**	8	GND	GND	9	**GND**	Ground

Figure 3.8: MikroBUS pinout[13]

Chapter 4

Software/firmware

4.1 Mbed OS 5

Arm Mbed OS [10] allows you to create applications for running on built-in devices by providing a layer that interprets the application code in a way that the hardware understands.

Arm Mbed supports writing application code in C language and C ++. It uses the APIs provided by Mbed OS. Using these APIs makes it possible to use the systems on different microcontrollers in a uniform way.

The Arm Mbed hardware architecture is designed to ensure the productivity of all tools and utilities. Most boards have an integrated debugging circuit that helps develop device programming, program execution protocol, and debug access. In the figure 4.1 you can see the basic architecture of a Mbed board.

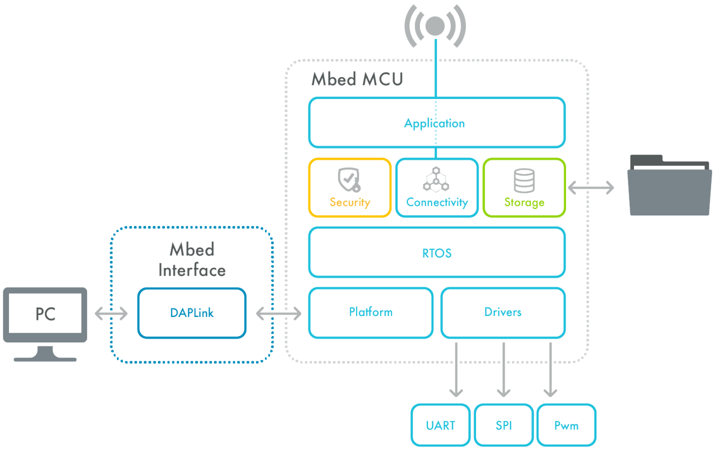


Figure 4.1: A sketch of a typical Mbed board's hardware architecture[10]

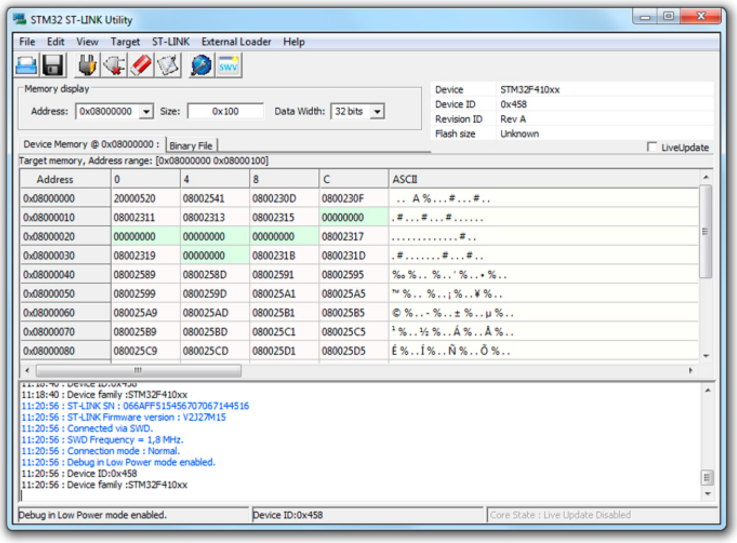
For use in this work, a web version of the Mbed OS compiler was used. The LoRaWAN-demo-72 code from the Semtech company [17] was first loaded into the compiler. LoRaWAN-demo is a ClassA device example project using LoRaWAN-lib and SX1272Lib libraries. This demo application sends a frame every 4 to 6 seconds (randomly) and displays its current status using a serial port as the display (VT100). This file was only the default program for the smart button project, for whose needs were changed.

Thanks to the Mbed OS compiler, a binary file has been obtained and uploaded to the development kit using the STM32 ST-LINK Utility.

4.2 STM32 ST-LINK Utility

STM32 ST-LINK Utility [22] (STSW-LINK004) is a fully functional software interface for programming the STM32 microcontroller. It provides easy to use and efficient environment for reading, writing and verifying memory. The tool offers a wide range of functions for programming the STM32 internal device memory (Flash, RAM, OTP, etc.), external memory for program content verification (checksum verified during and after programming, compared with the file) and automation programming STM32.

The STM32 ST-LINK tool is presented as a graphical user interface (GUI) with command line interface (CLI).



Address: 0x08000000 Size: 0x100 Data Width: 32 bits

Device	STM32F410xx
Device ID	0x458
Revision ID	Rev A
Flash size	Unknown

□ LiveUpdate

Target memory, Address range: [0x08000000 0x08000100]

Address	0	4	8	C	ASCII
0x08000000	20000520	08002541	0800230D	0800230F	.. A%...#...#..
0x08000010	08002311	08002313	08002315	00000000	.#...#...#......
0x08000020	00000000	00000000	00000000	08002317	.............#..
0x08000030	08002319	00000000	0800231B	0800231D	.#.......#...#..
0x08000040	08002589	0800258D	08002591	08002595	‰%.. %..'%..•%..
0x08000050	08002599	0800259D	080025A1	080025A5	™%.. %..¡%..¥%..
0x08000060	080025A9	080025AD	080025B1	080025B5	©%..-%..±%..µ%..
0x08000070	080025B9	080025BD	080025C1	080025C5	¹%..½%..Á%..Å%..
0x08000080	080025C9	080025CD	080025D1	080025D5	É%..Í%..Ñ%..Õ%..

```
11:18:40 : Device ID:0x458
11:18:40 : Device family :STM32F410xx
11:20:56 : ST-LINK SN : 066AFF515456707067144516
11:20:56 : ST-LINK Firmware version : V2J27M15
11:20:56 : Connected via SWD.
11:20:56 : SWD Frequency = 1,8 MHz.
11:20:56 : Connection mode : Normal.
11:20:56 : Debug in Low Power mode enabled.
11:20:56 : Device ID:0x458
11:20:56 : Device family :STM32F410xx
```

Debug in Low Power mode enabled. | Device ID:0x458 | Core State : Live Update Disabled

Figure 4.2: Example of STM32 ST-LINK Utility GUI[22]

For the correct operation of the program, it is only necessary to upload the ST-LINK Utility into the development board using the STM32 ST-LINK Utility. After obtaining a binary file from Mbed OS and open it using the STM32 ST-LINK Utility button "*Open file*," select "*Connect to the target*" for connecting to the development board. By pressing "*Program Verify*," the file will be uploaded to the kit.

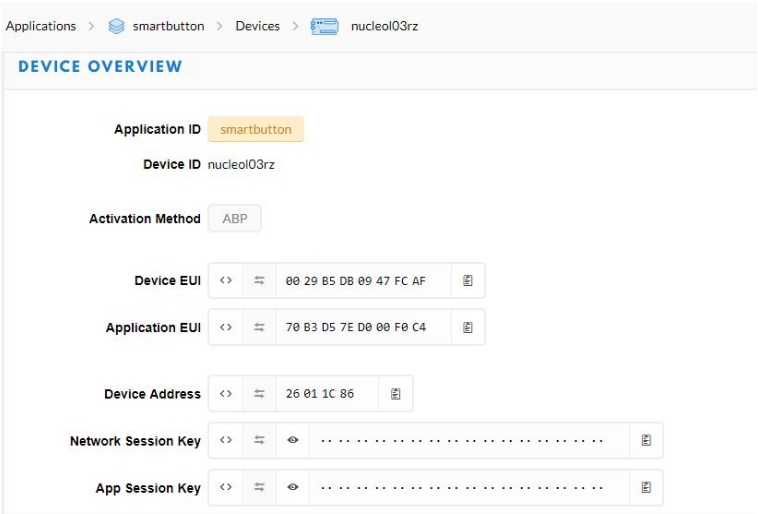
4.3 Important libraries and files

The entire program is written in the main.c file, which is located in the project on the enclosed CD in the app folder. This folder also contains files for listing via serial port to the terminal. The files are kept in case the developer would need to control some outputs applications locally, without access to Lora Network Interface.

4.3.1 Commissioning.h

Commissioning.h declaring end device commissioning parameters. For the successful operation of the project, it is necessary to change the parameters for OVER_THE_AIR_ACTIVATION (OTAA) here. Activation by Personalization (ABP) is used in the project, OTAA is set to 0. It is also necessary to copy LORAWAN_DEVICE_EUI, LORAWAN_APPLICATION_EUI, LORAWAN_APPLICATION_KEY, LORAWAN_DEVICE_ADDRESS, LORAWAN_NWKSKEY, LORAWAN_APPSKEY, which are assigned to the user after logging on to the TTN server.

The figure 4.3 shows the data from thethingsnetwork.org that are entered into the *Commissioning.h* file.



Application ID	smartbutton
Device ID	nucleol03rz
Activation Method	ABP
Device EUI	00 29 B5 DB 09 47 FC AF
Application EUI	70 B3 D5 7E D0 00 F0 C4
Device Address	26 01 1C 86
Network Session Key	.. ..
App Session Key	.. ..

Figure 4.3: Picture of TTN devide overview[15]

4.3.2 SSD1606_SPI.cpp

The SSD1606_SPI is a library for using the SSD1606 e-Paper display driver. To use the library in this project, it was necessary to set the exact height and width of the display in pixels. To initialize the display, the *void SSD1606Init()* function was used, and the background of the display was loaded with white color using the *void SSD1606Fill(unsigned char color)* function, where the function argument is the color expressed in hexadecimal number. The text strings on display are written in black with the command *unsigned char*

*SSD1606String(unsigned char x, unsigned char y, char *pString, int font)*, where x represents the initial pixel in the x-axis, y represents the initial pixel in the y-axis, *pString* is the required text string for the listing, and *font* indicates the font type to be used. The used fonts belong to the Lucida Console, font family. The last command that causes the resulting text listing to display is *SSD1606UpdateScreen()*. Only after executing this command will be the text displayed on display. The *SSD1606Init()* and *SSD1606UpdateScreen()* commands must be performed each time a value is overwritten on the e-ink display.

4.4 NodeRed

Node-RED[16] is a flow-based programming tool, originally developed by IBM's Emerging Technology Services team and now a part of the JS Foundation. It is the programming tool for wiring together hardware devices, APIs, and online services. Node-RED consists of a Node.js-based runtime that point a web browser at to access the flow editor. Within the browser can be created an application by dragging nodes from a palette into a workspace and start to wire them together. The palette of nodes can be easily extended by installing new nodes created by the community.

For the needs of a project communicating through the LoRa network interface, it was necessary to add a palette for TTN applications [14] to the Node-RED environment. Node-RED runs on a command line on Windows. The user is assigned a local Web address on which the Node-RED development environment is running.

In this GUI, simulation has also been developed demonstrating a smart button interacting with the user. To link the Nucleo-L073RZ development board to the Node-RED development environment, it is necessary to fill in data defining a specific board from a TTN server, such as "Device ID."

Node-RED also sent emails reporting defects and defects on the printer demo. For the use of the email sending function, it is necessary to stop the "Mail Shield" in the antivirus protection of the device from which the emails are sent. An email client was created for Google for the project, which required "Enable Less Secure Applications" in your account settings.

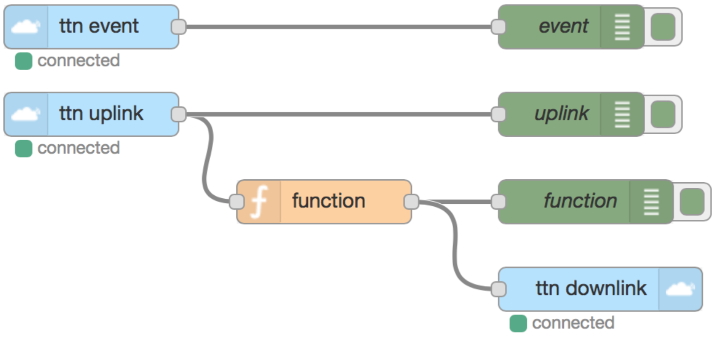
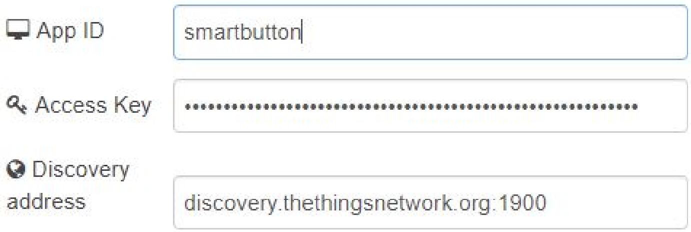


Figure 4.4: Example of TTN application in NOode-RED[14]

For the needs of a project communicating through the LoRa network interface, it was necessary to add a palette for TTN applications to the Node-RED environment. Node-RED runs on a command line on Windows. The user is assigned a local Web address on which the Node-RED development environment is running.

In this GUI, simulation has also been developed demonstrating a smart button interacting with the user. To link the Nucleo-L073RZ development board to the Node-RED development environment, it is necessary to fill in data defining a specific board from a TTN server, such as *Device ID* and *ACCESS KEYS*.



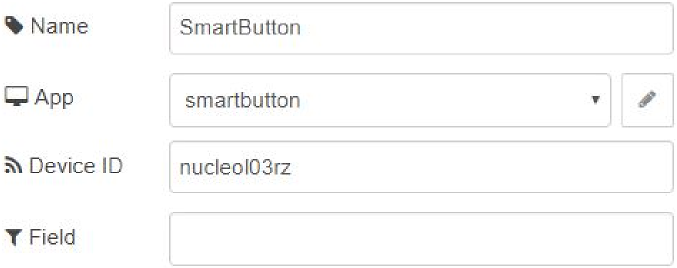
App ID: smartbutton

Access Key: ••

Discovery address: discovery.thethingsnetwork.org:1900

Figure 4.5: Setting of NodeRED for application[16]

Node-RED also sends emails reporting faults and defects on the printer demo. For the use of the email sending function, it is necessary to stop the "Mail Shield" in the antivirus protection of the device from which the emails are sent. For the project was created an e-mail address by Google in which it was necessary to turn on "Allow less secure apps" in the setting of the email client. The figure 4.6 shows the necessary settings for sending an email on the NoreRED GUI page.



Name	SmartButton
App	smartbutton
Device ID	nucleol03rz
Field	

Figure 4.6: Setting of email on NodeRED[16]

Chapter 5

Results

As a result, the two-button system with the e-ink display shown in the figure 5.1 was created. The 5.1 figure also shows the connection of individual components to the Nucleo-L073RZ board, which is powered by 3xAAA batteries.

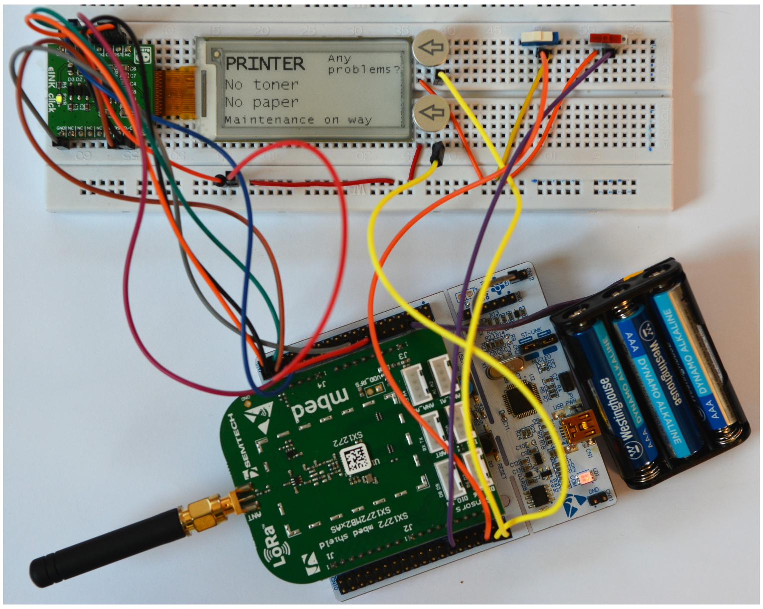


Figure 5.1: Designed system of smart button with additional components

The system consists of a Nucleo-L073RZ development board with a 32-bit ultra-low performance STM32L0 microcontroller and an I-NUCLEO-SX1272D RF expansion board supporting transmission through the LoRa Network Interface. Two push buttons for user interaction and e-ink display are connected to the board to output values from two sensors that simulate missing paper or toner. The sensors are addressed with two sliding binary switches. In reality, these sensors have been built inside the printer, which has become a simulation model for addressing the smart button issue.

The system is designed to display the current toner and paper value in the printer. If the sensor reports a shortage of paper or toner, it will be printed on the appropriate "NO," otherwise "OK" is displayed. In this case, a request is pressed to press the hold the key to call up the bottom edge of the display. When you press this button, using the LoRa Network Interface and the applications created in NodeRED will send an email. In which the state of each component is written. Consequently, the problem may be solved. The user is informed of this step again on the bottom line of the display. This is illustrated in the figure 5.2.

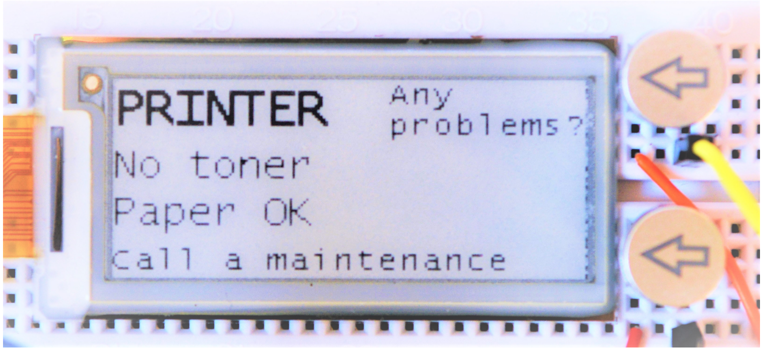
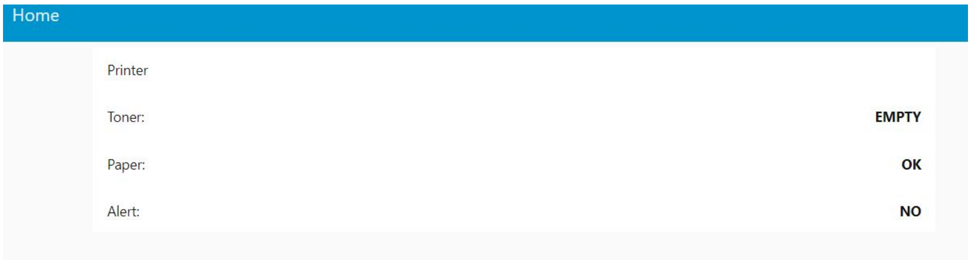


Figure 5.2: Designed system of two buttons and e-ink display

The second system buttons handle unexpected events that are not monitored by sensors. So if the sensors do not report an error, but the user has found the error, press the button "Any problems?" email will be sent again. In case the email was not maintained for some reason, a user interface was created using the NodeRED GUI to display the status of toner, paper, and other issues. An example of this GUI can be seen in the figure 5.3.

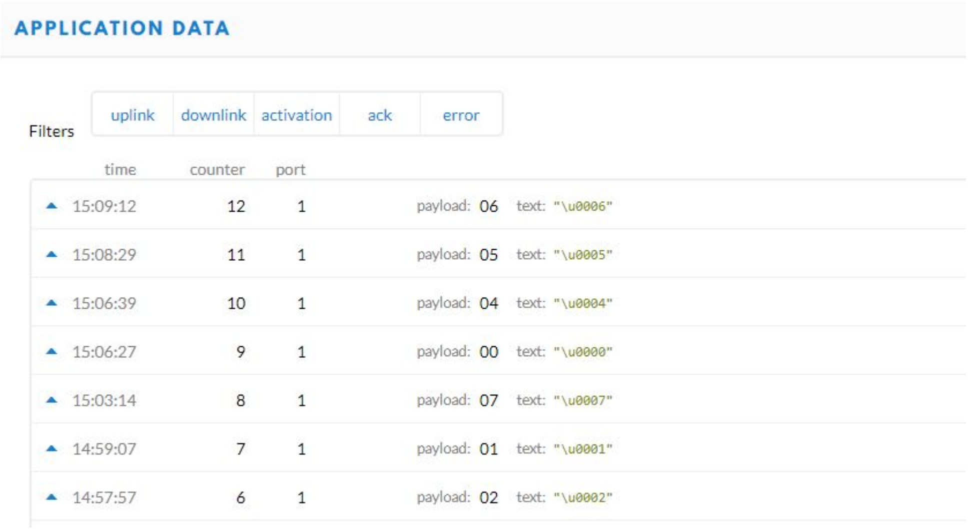

Home

Printer

Toner:	EMPTY
Paper:	OK
Alert:	NO

Figure 5.3: Example of NodeRED user interface for maintenance needs

A sample data transfer at thethingsnetwork.org is shown in the figure 5.4. The 5.1 table was created to describe each message. The individual numbers represent the decimal value of the binary representation of data in the field where the values are stored. The value of the zero position shows the toner, the value of the first position paper state and the second position is the status reporting of other errors. The third position is not used. Value 03 will never be seen, because the bottom button does not send data in case Toner: OK Paper: OK Alert: NO



APPLICATION DATA

Filters: uplink | downlink | activation | ack | error

time	counter	port	
15:09:12	12	1	payload: 06 text: "\u0006"
15:08:29	11	1	payload: 05 text: "\u0005"
15:06:39	10	1	payload: 04 text: "\u0004"
15:06:27	9	1	payload: 00 text: "\u0000"
15:03:14	8	1	payload: 07 text: "\u0007"
14:59:07	7	1	payload: 01 text: "\u0001"
14:57:57	6	1	payload: 02 text: "\u0002"

Figure 5.4: Application data from TTN[15]

Value	Description
00	Toner: NO Paper: NO Alert: NO
01	Toner: OK Paper: NO Alert: NO
02	Toner: OK Paper: OK Alert: NO
04	Toner: NO Paper: NO Alert: YES
05	Toner: OK Paper: NO Alert: YES
06	Toner: NO Paper: OK Alert: YES
07	Toner: OK Paper: OK Alert: YES

Table 5.1: Description of values from buttons on TTN server[15]

The application created on NodeRED is shown in the figure 5.5. The source code for the application is enclosed on the CD in the *NodeREDCode.txt* file.

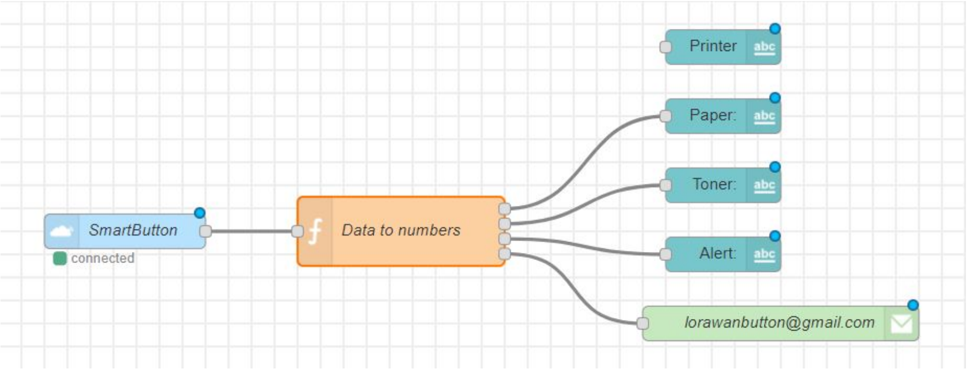
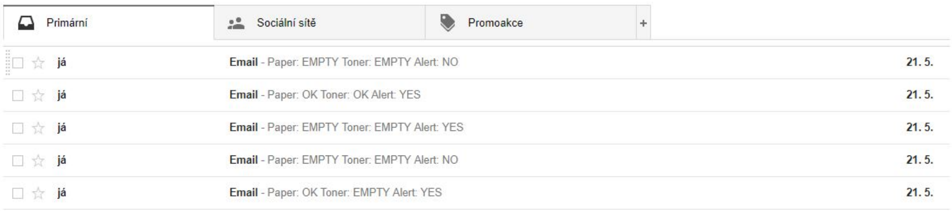


Figure 5.5: Flow-chart of application on NodeRED

An example of the emails sent for different error messages is shown in the figure 5.6.



já	**Email** - Paper: EMPTY Toner: EMPTY Alert: NO	21. 5.
já	**Email** - Paper: OK Toner: OK Alert: YES	21. 5.
já	**Email** - Paper: EMPTY Toner: EMPTY Alert: YES	21. 5.
já	**Email** - Paper: EMPTY Toner: EMPTY Alert: NO	21. 5.
já	**Email** - Paper: OK Toner: EMPTY Alert: YES	21. 5.

Figure 5.6: Example of emails alert of printer defects

Chapter 6
Conclusion

6.1 Work results

This bachelor thesis aimed to design and construct a system of smart buttons with e-ink display communicating through the LoRa communication interface.

The resulting system consists of the NUCLEO-L073RZ development kit with an ultra-low power 32-bit STM32L0 microcontroller and an I-NUCLEO-SX1272D RF expansion board supporting transmission through the LoRa Network Interface. Two buttons for user interaction are connected to the system. Simulation of external inputs is provided by binary switches. For executing text statements was used the Mikroe-2659 e-ink display. The display is connected to the system as well as the expansion board via the SPI communication interface. The LoRa Network Interface was used as the communication interface of the entire system. The whole system is powered by 3xAAA batteries.

Using the connected buttons, the user enters the entire system. Buttons can be configured as desired by the customer. Button configuration is implemented using the NodeRED flow-based programming tool for IoT. Thus, the populated solution is not the only option. The NodeRED flow-based programming tool for IoT provides a service request processing automation system. Text statements are made on the Mikroe-2659 e-ink display, the main advantage of which is to keep the text on the screen even after disconnecting power. System status monitoring is solved using binary inputs. Based on the values of these inputs, data is sent via TTN after pressing the user button.

The result of the work is a simple application of two push-buttons smart button system demonstrating a printer maintenance support. For examples of fault simulation, "No Paper" and "No Toner" was selected. In the case of one, the other or both errors, the user is asked to press the key to call the operator. After pressing the button, the data is sent to the TTN server, after that it is transferred to the NodeRED data handling tool. The created application in the NodeRED data handling tool sends an email with information about reported errors to the maintenance worker or another authorized person. The upper user button is designed for cases where none of the sensors report an error, but the printer user is not satisfied with the printer's functionality. Even in this case, when you press the appropriate button, the email is being mailed, this time because of an unspecified error. The value of error-simulating sensors is as follows. 0 marks the missing paper or toner, and the value 1 marks a sufficient level of paper or toner in the printer.

6.2 Future development

- Creating a suitable case using a 3D printer
- Reducing the entire system by creating a printed link to solve the problem
- Creating feedback from the maintenance side by sending downlink NodeRED using a programming tool.
- Sending the repair status to the display using the downlink via the NodeRED programming tool
- Creating a multi-button system based on the number of sensors

Appendix A

CD

Appendix B

General block diagram of the STM32L073xx device

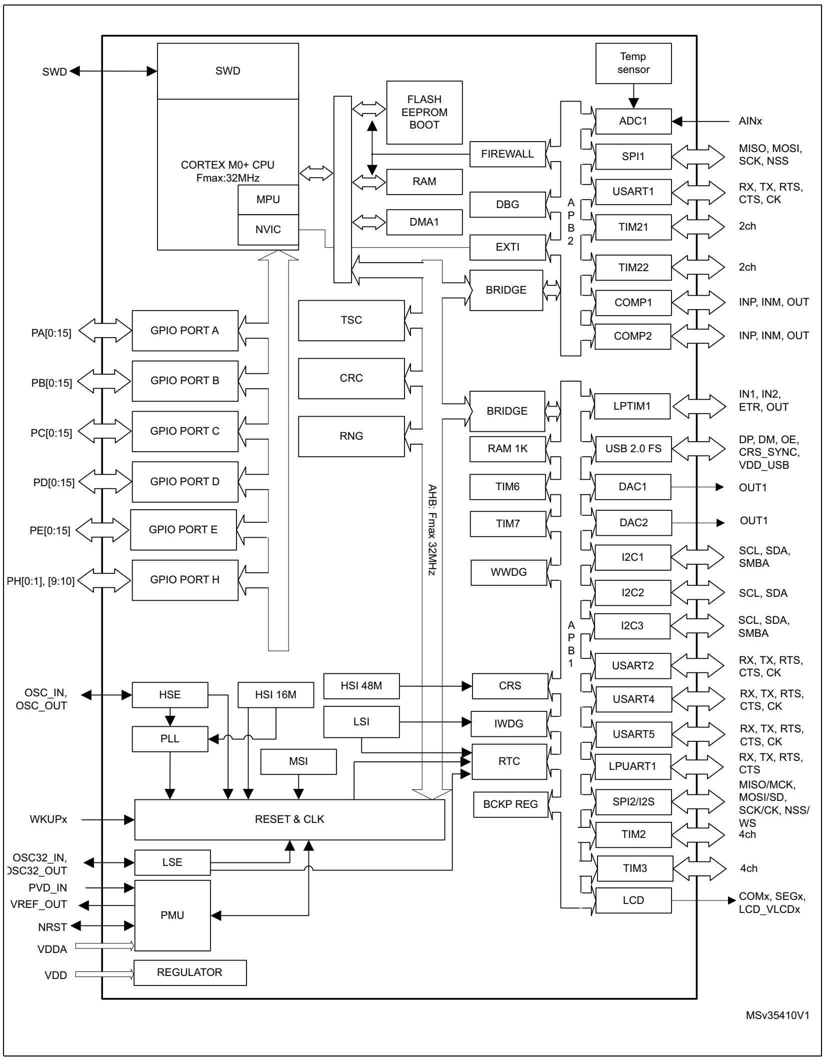


Figure B.1: General block diagram of the STM32L073xx device

Appendix C

Simplified Block Diagram of SX1272-3

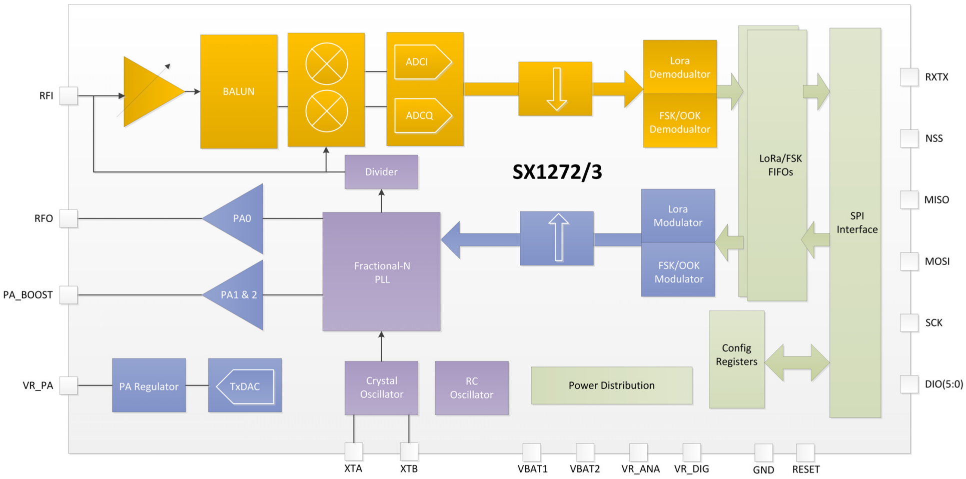


Figure C.1: Simplified Block Diagram of SX1272-3

Appendix D

Bibliography

[1] LoRa Alliance. https://www.lora-alliance.org/technology, 2017. Accessed: 2018-04-14.

[2] Amazon.com. Learn more about dash buttons. https://www.amazon.com/ddb/learn-more. Accessed: 2018-03-21.

[3] Wood Chris. Dimple nfc-enabled buttons customize the android experience. https://newatlas.com/dimple-nfc-android-buttons/32042/. Accessed: 2018-04-01.

[4] The Button Company. Our ioi button models. https://shop.bt.tn/pages/iot-button-models. Accessed: 2018-04-08.

[5] Vojtěch Hauser. Bezpečnost v stítích lpwan/lpn pro aplikace v iot. Master's thesis, České vysoké učení technické v Praze, 5 2016.

[6] Shortcut Labs. Flic - the official website. https://flic.io/. Accessed: 2018-03-25.

[7] Shortcut Labs. Flic hub: Simplify home control with smart buttons. https://www.indiegogo.com/projects/flic-hub-simplify-home-control-with-smart-buttons/. Accessed: 2018-03-25.

[8] Teschler Lee. Teardown: Does the amazon dash set the tone for IoT appliances? https://www.microcontrollertips.com/teardown-amazon-dash-set-tone-iot-appliances/.

[9] Malekian Reza Marais M. Jaco and Adnan M. Abu-Mahfouz. Lora and lorawan testbeds: A review. In *Africon*, pages 1496–1501. University of Pretoria, 9 2017.

[10] Mbed. Mbed os 5. https://os.mbed.com/docs/v5.8/introduction/index.html. Accessed: 2018-05-19.

[11] Mbed. Nucleo-l073rz. https://os.mbed.com/platforms/ST-Nucleo-L073RZ/. Accessed: 2018-05-11.

[12] Campbell Mikey. Review: Philips hue tap reimagines the light switch. https://appleinsider.com/articles/14/07/23/review-philips-hue-tap-reimagines-the-light-switch. Accessed: 2018-03-27.

[13] MikroElektronika. eink click - mikroe-2659. https://www.mikroe.com/eink-click. Accessed: 2018-05-02.

[14] Network Performance Monitor. The things network node-red nodes. https://www.npmjs.com/package/node-red-contrib-ttn. Accessed: 2018-05-20.

[15] The Things Network. The official the things network. https://www.thethingsnetwork.org/. Accessed: 2018-05-12.

[16] Node-RED. Flow-based programming for the internet of things. https://nodered.org. Accessed: 2018-05-20.

[17] Semtech. Lorawan-demo-72. https://os.mbed.com/teams/Semtech /code/LoRaWAN-demo-72/. Accessed: 2018-05-19.

[18] Semtech. Semtech sx1272. https://www.semtech.com/products/wireless-rf/lora-transceivers/SX1272-download-resources. Accessed: 2018-05-12.

[19] Raution Siiri. bttn – the simplest internet user interface in the world. https://www.indiegogo.com/projects/bttn/. Accessed: 2018-04-11.

[20] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. A survey on lpwan technology: Lora and nb-iot. *ICT Express*, 3(1):14–21, 2017.

[21] STMicroelectronics. Stm32l073rz. http://www.st.com/en /microcontrollers/stm32l073rz.html. Accessed: 2018-05-11.

[22] STMicroelectronics. Stsw-link004. http://www.st.com/en/development-tools/stsw-link004.html. Accessed: 2018-05-19.

[23] STMicroelectronics. *Ultra-low-power STM32 and LoRa® Nucleo pack with NUCLEO-L073RZ board and I-NUCLEO-SX1272D RF expansion board*, 2016. Rev 2.

[24] STMicroelectronics. *STM32 Nucleo-64 boards*, 2017. Rev 12.

[25] STMicroelectronics. *STM32L073x8 STM32L073xB STM32L073xZ*, 2017. Rev 4.

[26] Alun Williams. Press the red button, and make it happen! https://www.electronicsweekly.com/blogs/gadget-master/consumer-electronics/press-red-button-make-happen-2014-06/. Accessed: 2018-04-11.

[27] George Zerveas. One to rule them all? https://ict.swisscom.ch/2015/11/one-to-rule-them-all/, 11 2015.