LORA EARLY WARNING FLOOD DETECTION SYSTEM

Technical document

Students:
Joëlle de Vries, Pascal Holthuijsen, Jelmer Surewaard, Arno Nagtzaam
Version: 1.0
Date: 29-06-2021

Introduction

This technical document explains why certain technical decisions have been made in this project and how these decisions were established. The problem which this project solves is shown in Background and problem. The other chapters in this document are based upon the main research question: "How can a long-range flood detection system be implemented in Nepal in a way that people can be evacuated in a timely manner?". The main research question has been split up into several sub questions and goals. Based upon this, several requirements were realised for the creation of this project.

The first chapter explains the background of this project and the problem it is trying to solve. The second chapter contains the expected results of the project. The third chapter shows the requirements. The fourth chapter describes the hardware that is used in this project. The fifth chapter explains the functionality of the Gateway. The sixth chapter shows the research done regarding the microcontroller used in the project. The seventh chapter explains in detail how the network is implemented. The eighth chapter explains what communication protocol is used within the project. The ninth chapter explains what kind encryption can be used for this project. The tenth chapter shows the research done on minimizing energy consumption. The eleventh chapter shows the results obtained from tests. At the end of this document a conclusion is shown alongside a recommendation.

Table of Contents

1. Background and problem

Nepal has to deal with the Monsoon every year. This is a period of six months with a lot of rain. Due to this rainfall, it often happens that rivers flood with catastrophic consequences for communities around rivers. With recent past records of disaster during the Monsoon flood, telemetric systems for data monitoring were realized to be essential and found to be successful to help warn the communities. Everyone is well aware of the large number of casualties that the natural disasters like floods have brought in the past, especially in the far Western and Southern regions of Nepal. Even though it is not possible to challenge the force of nature, it is possible to take preventive measures to reduce these numbers. Different efforts have been forwarded in the past from various authorities. The analysis shows a timely dissemination of the river level reading could warn the people early and save thousands of lives and properties. The early warning system for flood prediction is being developed for this very purpose.

The Robotics Association of Nepal (RAN) has already developed a system to measure water levels and predict floods. The problem they face is that they don't have sufficient technical knowledge to transfer this data to the right community in Nepal. RAN has indicated that they want to develop a LoRa (Long Range) 2.4GHz network to send data from the river to the community.

Diyalo Foundation was co-founded by Jord Drontmann who is committed to the Nepalese population with a focus on helping Nepalese youth to develop technical projects. Jord has made contact with RAN and from this contact the LoRa early warning flood detection project was created. After conceiving this project, Diyalo Foundation went looking for experts in the field of LoRa. Contact was made with KPN as they have set up a LoRa network in the frequency band 863 - 870 MHz in the Netherlands. Due to legislation in Nepal, the LoRa network in Nepal will have to function on the 2.4GHz frequency band. Since KPN has no experience with this, the Diyalo Foundation has searched further and eventually ended up with Maurice Snoeren, a teacher at Avans Hogeschool. Maurice has accepted this project and has appointed four students of computer science to carry out this project.

2. Project result

Based on the research question: “How can a long-range flood detection system be implemented in Nepal in a way that people can be evacuated in a timely manner?” several sub questions and goals have been realised. These were also shown in the project plan document.

Research questions:

- In what kind of landscape should this project be realised?
- What communication protocol should be used?
- How often should data be transferred over the network?
- What is the longest reliable distance between network nodes?
- How can the nodes be protected against weather conditions?
- How can the nodes work with the lowest possible amount of power usage?
- What kind of hardware should be used?

Goals:

- A sensor must be placed at the source of the flooding to know if a flood is going to happen.
- A village must be notified over the 2.4 GHz network using LoRa.
- The desired range between the sensors and the village is 25 km.
- The energy usage needed by the network cannot be higher than 20 Watt.
- All the hardware used in the network must be solar powered.
- The hardware must be protected from all weather conditions like rain and lightning.

3. Project Requirements

Based on goals shown in the project plan the requirements were composed using the MoSCoW method:

Must haves

- LoRa 2.4GHz network
- Network range of 25 km
- LoRa Gateway with data logging
    - Monitoring whether the system is alive and the sensors are working properly in the gateway.
- Usage of Arduino like microcontrollers as repeaters
- Robust and maintainable system
- Understandable documentation
- Use of a small single-board computer as Gateway (for example Raspberry Pi)
- Sensor readings

Should have

- As low as possible energy usage; maximum 20 Watt
- Use of solar panels in proof of concept
- Use of batteries in proof of concept
- Heartbeat of LoRa modules
- Use of antennas for transmitting and receiving radio waves
- Bidirectional communication (short messages)

Could have

- Housing of LoRa repeaters resistant to bad weather conditions in proof of concept
- Lightning protection

Won't have

- High data communication network
- Graphical user interface for network data

4. Network

A node network will be used in this project. If the nodes function over an 8-kilometer radius, RAN has already established a potential layout. However, since this project will be expanded to other parts of Nepal, it must be modular and simple to switch out nodes if one fails. Inside the network, the network must embrace modularity. The layout that RAN has already created is shown in the image below.

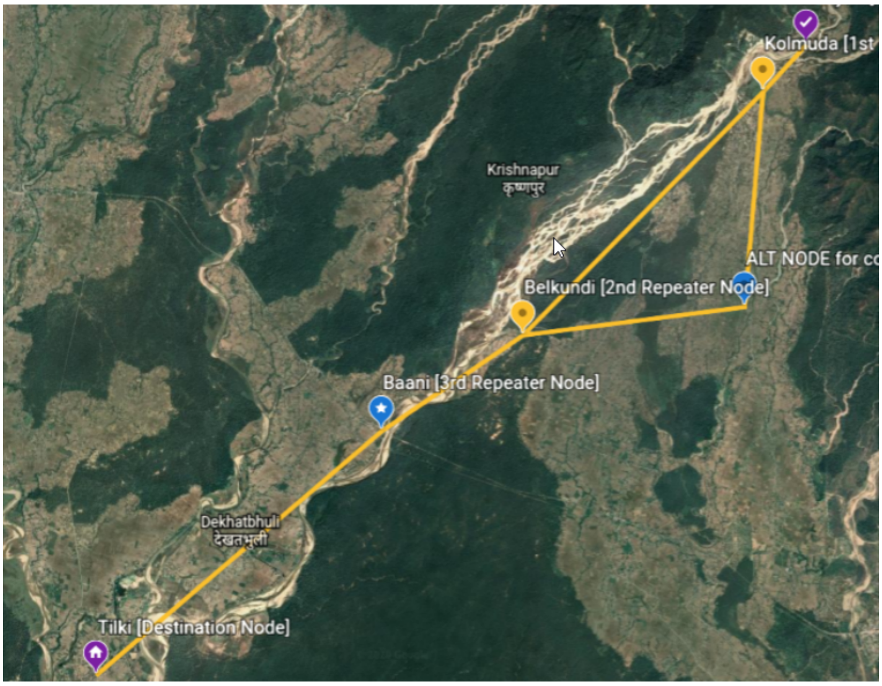

Figure 1. Proposed network layout from RAN

4.1. Network topologies

There are five types of topologies that are commonly used when setting up a mesh network. Following are the five topologies with an image for each topology and a short explanation describing whether it is applicable in this project.

- Mesh Topology
    - Great scalability simple to add nodes

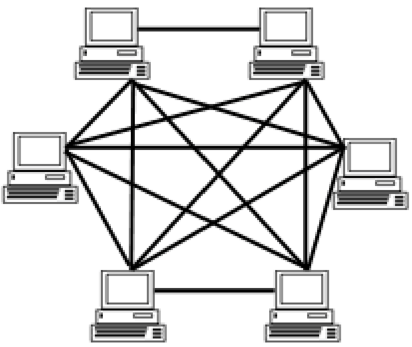

Figure 2. Mesh topology

- Star Topology
    - Not useful since it can’t cover large distance

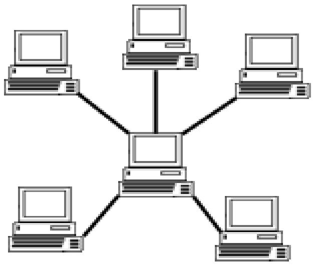

Figure 3. Star topology

- Bus Topology
    - When made wireless great to cover long distance

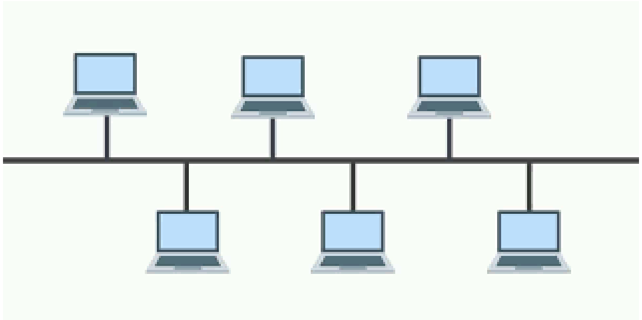

Figure 4. Bus topology

- Ring Topology
    - Not usable in this situation since its meant for a ring and not a line

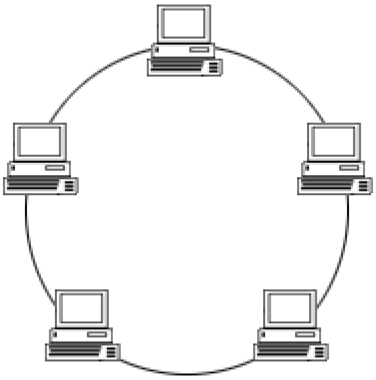

Figure 5. Ring topology

- Hybrid Topology
    - Combination of 2 or more topologies

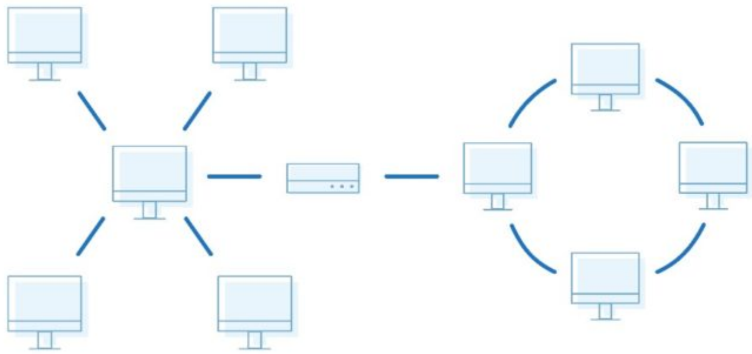

Figure 6. Hybrid topology

A Mesh network and a Bus Network are the two best choices in this case. The main issue with a bus topology is that there are times when the network divides into two separate connections. In this case, a mesh topology would be best suited, as any node within the radius of a different node would be connected to that node. Any node that is identified by other nodes can be added to the network in this way, making it scalable and modular.

4.2. Communication methods

Communication within a mesh network needs to be handled in a correct manner so no loops occur. If data gets sent to a node and then gets sent back to that same node a loop can occur, trapping the data and draining more power in the process. To establish a way of communication without faults a design is necessary. This paragraph will go over multiple ways to communicate within the mesh network.

Requirements:

- Low energy consumption
- Robust
- Understandable
- Fast

Method 1: Mesh network

The sensor node sends sensor data. All nodes that are close enough to receive data will receive the data. Every node that receives the data will store the data and echo it. If a node receives data that is the same as the stored data, it will do nothing. This ensures that data will only once be sent over the network.

The figure below shows a diagram of the network. Every line represents data being sent from one node to another. Every dotted line represents duplicated data that will be received by a node because it already has the data stored in memory.

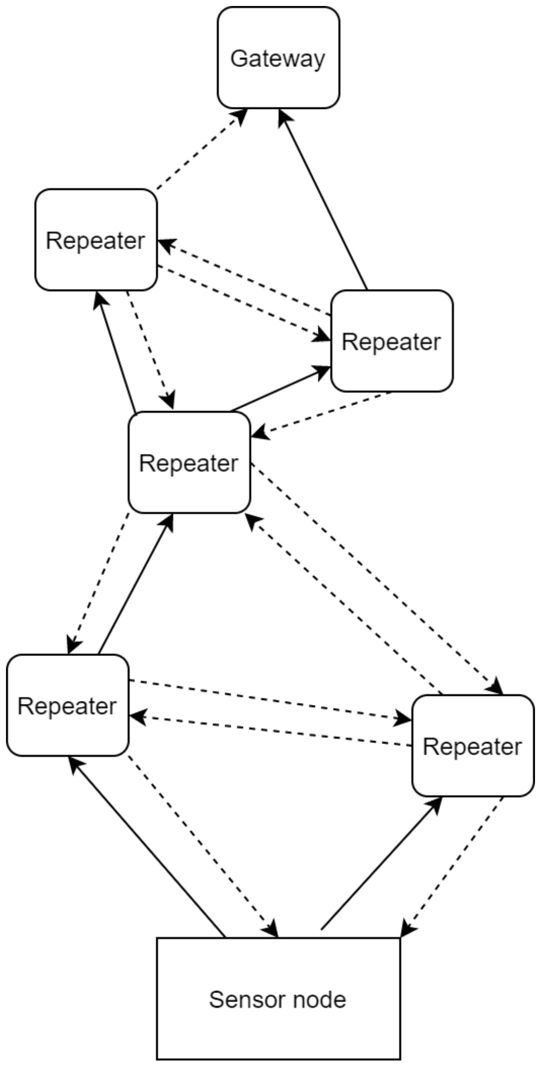

Figure 7. Example of a mesh network

This method is fast and relatively simple to implement because the network does not have to implement a route.

Method 2: Data list

A node sends data to all closest nodes and adds id to a list. If the id does not exist in the list the data should be sent to all closest nodes. If the id already exists in the list the data should not be sent to the next nodes. Thus, the data will not be sent back where it came from.

The data list will be stored on the sensor side according to the time they are received. The data list that was first received is the fastest route and therefore will be used to send the data to the gateway. If that route isn't usable because of a broken node the second fastest route will be used. When a new node is added to the network or a node is repaired, the route will be checked again to see which is the fastest.

Method 3: Graph

This system works with a Graph system. A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.

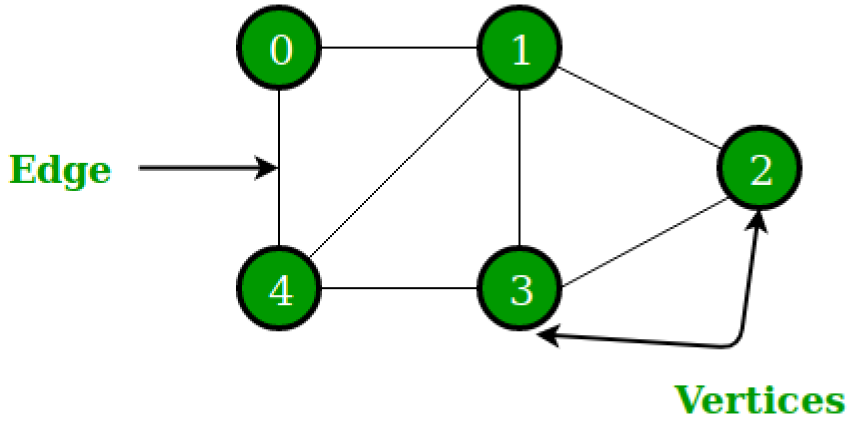

Figure 8. Example of a graph

A graph can have weighted edges, weighted edges can be used to define the distance between the nodes. This method is going to use this technique to define the shortest path between the start and destination.

Every time a node is added to the network it makes a connection with all nearest nodes. The node sends the connection to the gateway. The gateway keeps the graph of the entire system in its memory. After the entire network is set up the gateway will look for the fastest route between the sensor node and the gateway. This will be done using a depth first search algorithm using the graph in the gateway.

5. Hardware

This chapter describes the various hardware components used and researched in this project.

5.1. LoRa module

Research has been done on three different LoRa 2.4GHz chips. The NiceRF LoRa1280-TCXO, The NiceRF LoRa1280F27-TCXO and the Ebyte E28-2G4M27S. Following is a table that gives an overview of the different variables that are important for this project.

	NiceRF LoRa1280-TCXO	NiceRF LoRa1280F27-TCXO	Ebyte E28-2G4M27S
Operating Voltage	3.3V	5V	3.3V
TX Current	28 mA	<19 mA	580 mA
RX Current	<12 mA	<600 mA	14.5 mA
Sleep current	1.0 µA	7.0 µA	2.0 µA
Maximum output power	12.5 dBm	27 dBm	27 dBm
Receiving sensitivity	-132 dBm	-132 dBm	-132 dBm

Pros and cons of the NiceRF LoRa1280-TCXO

The great advantage of the NiceRF LoRa1280-TCXO is the low power consumption. This is an important point for the project, as each module must consume as little power as possible. However, due to the lower power consumption, the maximum output power is also lower. As a result, the maximum achievable distance is considerably smaller than the other two LoRa chips.

Pros and cons of the NiceRF LoRa1280F27-TCXO

The great advantage of the NiceRF LoRa1280F27-TCXO is the long distance that can be reached. Due to the fact that this chip operates on 5V, the power consumption is considerably higher than the NiceRF LoRa1280-TCXO. However, the higher power consumption also brings its advantages, such as the greater distance that can be reached. Which is desirable in a setting where there is no possibility to place modules in short distances of each other.

Pros and cons of the Ebyte E28-2G4M27S

The great advantage of the Ebyte E28-2G4M27S is the long distance that can be achieved with relatively low power consumption. Unlike the NiceRF LoRa1280F27-TCXO, which has the same maximum output power, the Ebyte E28-2G4M27S operates at 3.3V. The Ebyte chip consumes more power when transmitting data than the other two chips.

Conclusion

While testing the three different chips, the discovery has been made that receiving data on the Ebyte chip does not work properly. The data received was corrupt and could not be read. As a result, the decision was made to use the NiceRF LoRa1280F27-TCXO. Since it can reach the greatest distance and the power consumption is not high enough that a battery cannot supply enough power.

5.2. Antenna

Research has been done on two types of antennas that can be used in combination with LoRa.

Parabolic antennas

The first type of antenna that has been studied is a parabolic antenna (Wikipedia, n.d.). The main advantage of a parabolic antenna is that it has high directivity. It functions similarly to a searchlight or flashlight reflector to direct the radio waves in a narrow beam, or receive radio waves from one particular direction only. Parabolic antennas have some of the highest gains, meaning that they can produce the narrowest beamwidths, of any antenna type.

The diagram below shows how a parabolic antenna works. For a reliable connection, the transmitter and receiver need to be precisely lined up with each other.

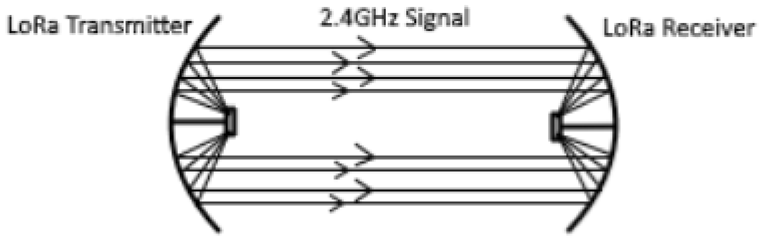

Figure 9. Schematic example of parabolic antennas

Omnidirectional antennas

The second type of antenna that has been researched are omnidirectional antennas (Wikipedia, n.d.). The advantage of an omnidirectional antenna is that it radiates a signal equally in all horizontal directions, while the power radiated drops off with elevation angle so little radio energy is aimed into the sky or down toward the earth and wasted.

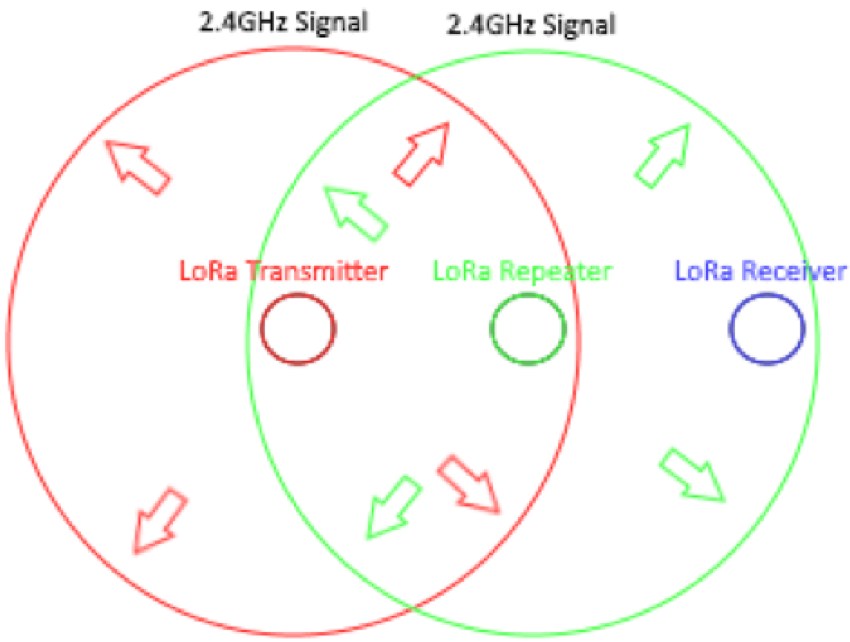

Figure 10. Schematic example of omnidirectional antennas

Conclusion

Parabolic antennas should theoretically be able to bridge the distance of 25 km between sensor and gateway. However, a direct line of sight connection from sensor to gateway is not always possible in the landscape of Nepal. This would mean that there must be repeaters in the network. Due to the fact that parabolic antennas have to point towards each other, it is impossible to use this type of antenna in this project. This would mean that each repeater would need two antennas: one for receiving data and one for sending data to the next repeater. With an omnidirectional antenna it is possible to receive and send data from all directions. This ensures that each repeater only needs one antenna.

5.3. Printed Circuit Board

In order to deliver a complete product, it was decided to develop a printed circuit board (PCB). There are two advantages of a PCB for this project. The first advantage is that it is a product that is robust and sturdy. The second advantage is that it increases the range of the LoRa modules. This is because there are fewer loose connections between components. When prototyping with a breadboard a lot of wires are used which can cause a lot of interference. A PCB can provide a direct connection from the LoRa chip to the antenna which results in less interference.

Design tool

The design tool used for creating the printed circuit board is EasyEDA. EasyEDA is a browser-based circuit board designer. EeasyEDA was chosen because there is no need to install any special software and a big advantage to EasyEDA is their large and updated library of components on the cloud which is created by a large user base.

Design

The design consists of two main components. The first component is the LoRa chip and the second component is a microcontroller. These components have been connected to each other according to the following table:

Table 1. Connection table of LoRa1280F27

LoRa1280F27 Pinout	Arduino Nano Every Pin
VCC	+5V
TCXOEN	+5V
NRESET	D8
BUSY	D7
DIO1	D4
DIO2	D5
DIO3	D6
NSS	D10
SCK	D13/SCK

MOSI	D11/MOSI
MISO	D12/MISO
GND	GND

Pin headers were added to the board so different types of sensors can be connected to the board. The following two images are respectively a schematic of the LoRa modules and a 2D view of the final PCB design.

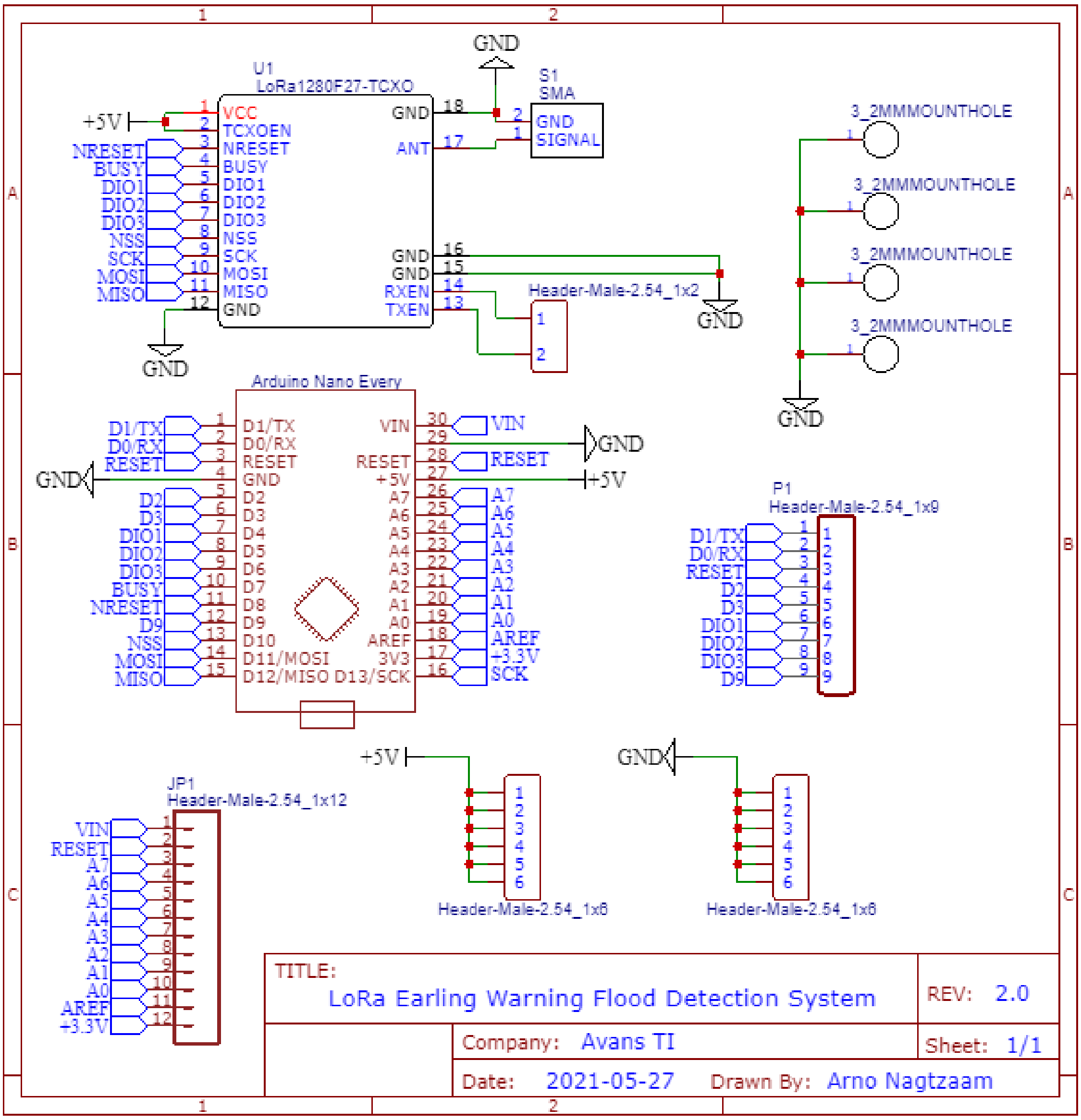

Figure 11. Schematic view of developed PCB

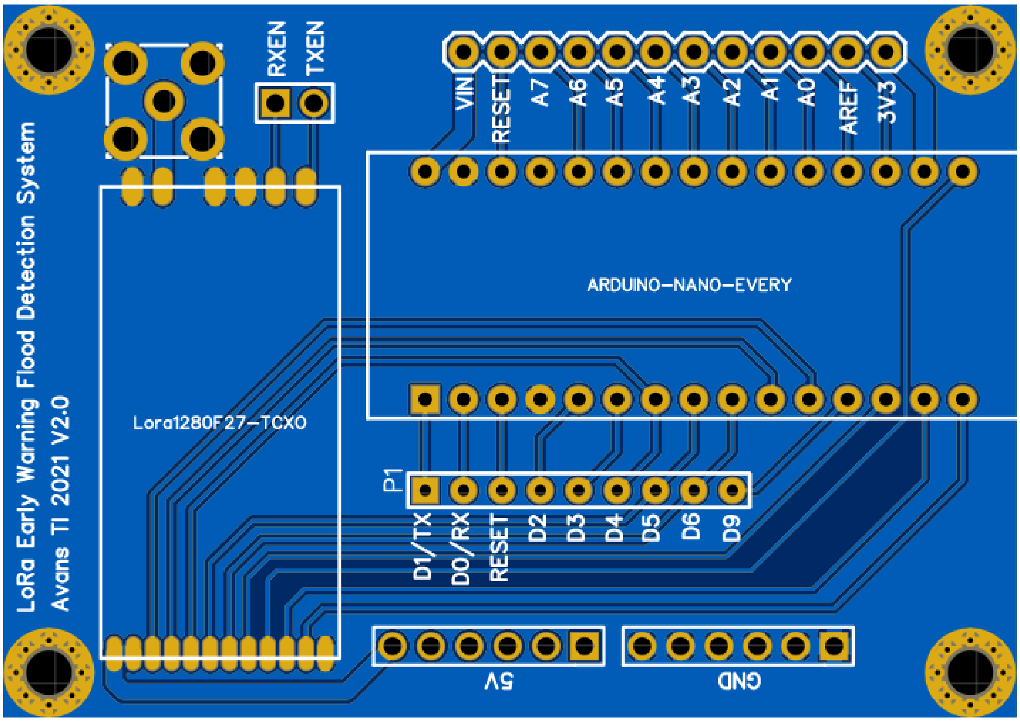

Figure 12. Schematic view of developed PCB

5.4. Final product

Once the design was created, a Gerber file was generated and sent to a PCB manufacturer. After 5 business days the circuit boards arrived and were ready to be soldered. Once everything was soldered and connected to the board the modules were ready for use. The following picture shows six fully assembled circuit boards.

Figure 13. Six fully assembled circuit boards

6. Microcontrollers

To communicate with one another, the LoRa modules need a microcontroller. This microcontroller must be able to use SPI to send and receive data from the module. There are a few other prerequisites for this module.

6.1. Requirements

- Low power consumption
- SPI
- 3.3 volt
- IO pins
- 24+ mA io pin output
- Relatively low cost below 10 euro (1414 Nepalese Rupee)

Boards

- Arduino nano - 19 mA pc (Arduino_store, n.d.)
- Teensy 4.0 - 20 euros (2828 Nepalese Rupee) + high ram and processing capabilities 20 mA output
- ESP 32 - output ampere not high enough limited to 12 mA whilst 24mA is necessary
- ESP 8266 - output ampere not high enough limited to 12 mA whilst 24mA is necessary

Microprocessor without board

ATtiny84A (Rosmianto, 2020) max draws 25 mA on IO pins and it has a power-down current of 100nA and an active current of 210µA at 1MHz. An advantage is that this microcontroller is widely supported and has many libraries available on the internet. A downside is that it is m more complex to program compared to programming an Arduino in the Arduino IDE.

Any of the microchips above can be used in this project, except for the ESP, which does not provide enough power to the LoRa module. Since Arduino and Arduino libraries are used in everything done so far, it's best to use a microchip that can be programmed using the Arduino IDE. Therefore, the usage of Arduino Nano is advised.

6.2. Arduino Nano Every in Depth

In the following table an overview of the technical details of the Arduino Nano Every is shown.

Table 2. Technical details of Arduino Nano Every

Microcontroller	ATMega4809 (datasheet)
Operating Voltage	5V
VIN min-MAX	7-21V
DC Current per I/O Pin	20 mA

DC Current for 3.3V Pin	50 mA
Clock Speed	20MHz
CPU Flash Memory	48KB (ATMega4809)
SRAM	6KB (ATMega4809)
EEPROM	256byte (ATMega4809)
PWM Pins	5 (D3, D5, D6, D9, D10)
UART	1
SPI	1
I2C	1
Analog Input Pins	8 (ADC 10 bit)
Analog Output Pins	Only through PWM (no DAC)

The Arduino nano Every uses the ATMega4809 chip as its processor. The datasheet of this chip is available on this link:
https://content.arduino.cc/assets/Nano-Every_processor-48-pin-Data-Sheet-megaAVR-0-series-DS40002016B.pdf
Even though the power consumption of a development board is relatively high with 19 mA compared to some stand alone chips running on 100nA like the ATtiny84A. The benefit in using an Arduino board is the widely available libraries and community knowledge.

This is the only chip that could be found which is compact, works with low power consumption and provides enough power to work with the LoRa module. So, it is recommended that Arduino Nano Every is used. If the preference lies with another chip, the chip needs to have an output current via the I/O pins of at least 24mA. Otherwise, the LoRa network will not work with the recommended LoRa chips.

7. Proof of concept

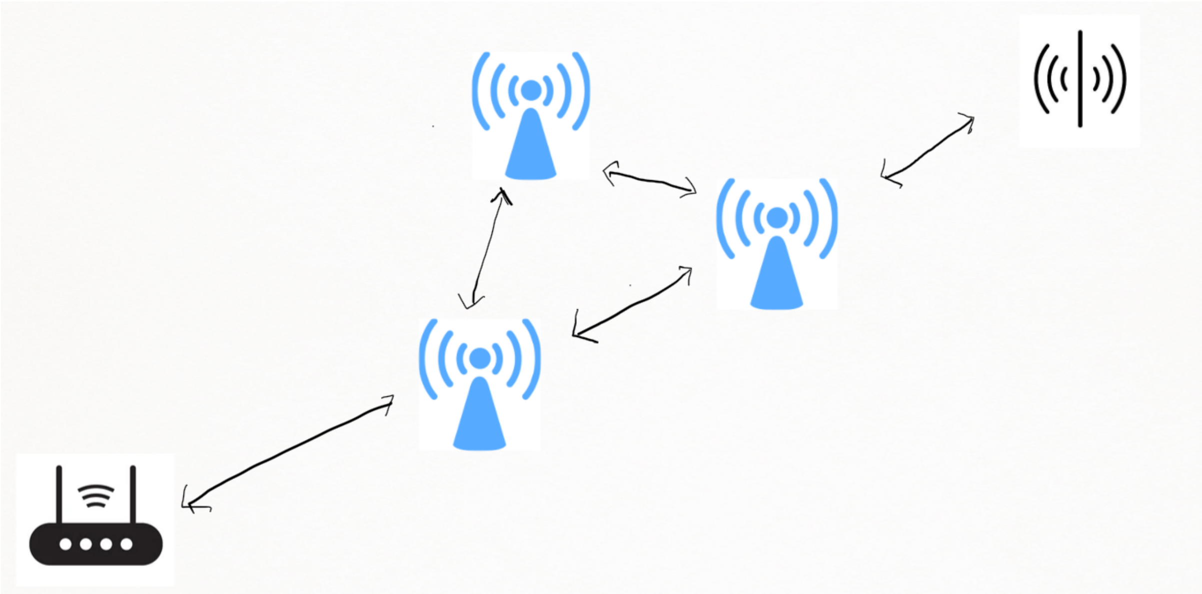

Figure 14. Example of mesh network

The proof-of-concept mesh network looks like the image shown above. The network contains three different parts: the gateway, the repeater nodes and the sensor node.

7.1. Repeater nodes

The repeaters send and receive data to move data across the network. When receiving data each repeater checks if it has received this message before by checking the identifier. If the message has been received before, the repeater does nothing. If the message has not been received before the repeater saves the message identifier and sends it to the next repeater. The repeaters use an Arduino nano microcontroller.

Repeater nodes are capable of the following actions:

- Send data
- Receive data
- Save data identity
- Delete saved data after a certain period

7.2. Gateway

The gateway is a crucial part of the entire network. The gateway is responsible for receiving and processing any data that will be transferred over the network.

Raspberry pi

Because data logging is a crucial part of the gateway, a microcontroller is not sufficient to do all the tasks. Therefore, the decision has been made to use a single board computer to handle this part. The gateway needs to be able to easily connect to the internet. A Raspberry Pi was an obvious choice as these modules are readily available in the whole world, and lots of support is found online.

The used Raspberry Pi runs on a Raspbian 32-bit operating system. On this Raspberry Pi, it is advised to install Arduino IDE. Not only is it possible to easily alter the source code, if necessary. Also, this IDE can be used to export binary files. This is the important part to run Arduino scripts on a Raspberry Pi.

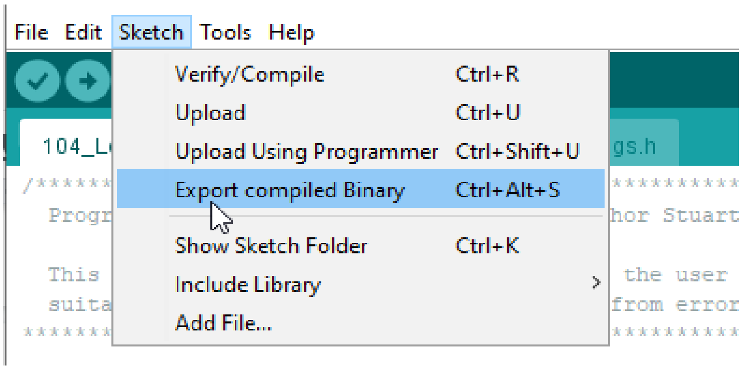

Figure 15. Shortcut for exporting compiled binary file in Arduino IDE

This will compile the entire project into a binary file, which is now executable. First, the right permissions have to be assigned to the binary file. This can be done with the following command:

chmod 755 ./[PATH_OF_BINARY_FILE_HERE]

Once the permissions have been correctly set, the binary file can be executed using the following command:

sudo ./[PATH_OF_BINARY_FILE_HERE]

Functionality

The gateway will use the communication protocol as described later in this document. The only added functionality the gateway module will contain, is data logging. Therefore, currently the data that is received on the gateway is logged on the console.

It is desired that the logged data is sent to a web server. This however is outside the scope of this project.

Libraries

The Stuarts library used in the rest of the system will be used in the gateway. Only crucial parts from this library will be used in the compiling of the code. These files are:

- SX128XLT.cpp
- SX128XLT.h
- SX128XLT_Definitions.h

7.3. The sensor node

The sensor collects data from the water level sensor. Once this data is received the node will send the data to the repeaters in the network. The sensor node uses an Arduino nano microcontroller.

The sensor node is capable of the following actions:

- Sending data
- Receiving data
- Collecting data from sensors

8. Communication Protocol

The Communication protocol that this project will use is a binary protocol. The choice for a binary protocol has been made because it is the smallest way of combining a lot of data in a single object.

The protocol is based on the following data:

- ID
- Voltage Node
- Amperage Node
- Battery level Node
- Sensor data
- Checksum

ID	Node: Volt	Node: Amperage	Node: Batery Level	Sensor data	Checksum

The data is structured in the way shown above. Starting with the ID followed by the electronic data from the sending node, then the sensor data and ending with the checksum to check if all data is sent.

Every part needs a set amount of bits assigned to it, this ensures that it is programmable and has a standard that does not vary. Bits are assign from the left to the right so on the left is bit 0

Table 3. Detail of standard message protocol fields

Type	Length	Bits	Unit	Range
ID	8 Bits	0 - 7	-	0-128 (first bit reserved)
Volt	8 bits	8 - 15	mV	0-255
Amp	8 bit	16 - 23	mA	0 - 255
Battery Level	16 bits	24 - 39	mV	0 - 65536
Sensor data	16 bits	40 - 55	-	0 - 65536
Checksum	7 bits	56 - 62	-	0 - 127
Parity Bit	1 bit	63	-	0 - 1

The parity bit is a bit which is used to check if the data has not been changed. It is far from fail safe but an extra check will help to check if the data is correct. When the parity bit is 0 the amount of ones in the message is even. When the parity bit is 1 the amount of ones in the message is odd.

There is a second message that can be sent over and that is a heartbeat message. This message tells the gateway that a certain repeater is alive this message contains the following parts:

- ID
- RepeaterID
- Checksum

Table 4. Detail of heartbeat message protocol fields

Type	Length	Bits	Unit	Range
ID	8 Bits	0 - 7	-	0-128 (first bit reserved)
Repeater ID	8 Bits	8 - 15	-	0-255
Checksum	7 bits	16-22	-	0 - 127
Parity Bit	1 bit	23	-	0 - 1

The first bit of the ID will be set to 1 for heartbeat messages and set to 0 for normal messages.

9. Security

Encryption within the LoRa network will ensure that the data is still intact. Encryption will make the data more difficult to tamper with, since it needs to be decrypted before anyone can read it. Encryption will make sure that the data arrives untouched at its destination. Because encryption is not without its flaws, it is not guaranteed that the data is safe. However, it is definitely worthwhile to implement.

KPN has informed that AES128 encryption is used within their LoRa network. Since LoRa is used on microcontrollers, a lightweight encryption method is a must-have to ensure a low amount of power consumption.

The 3 commonly used types of encryption are: AES, DES and RSA. Of these 3 algorithms, DES is the most outdated. DES will not be compared in this research because AES is newer and safer and replaced DES as the encryption standard. That leaves AES and RSA.

RSA encryption is very usable when there are two physically or geographically different end-points. It however does not fit well in a broadcasting environment like this project. Communication in this project is mostly broadcasting, as there can be multiple receiving and/or sending devices. RSA is asymmetric and thus, the public key needs to be sent over. Therefore RSA is not practical for this use case.

AES, on the other hand, does not work as well when there are two physically or geographically different end-points as it is a symmetric algorithm. But, this is a positive thing for a broadcasting scenario: the key does not have to be sent over the network, thus no back and forth communication is necessary.

Encryption is not realised in the proof of concept because constructing a functioning network was a higher priority. Implementing encryption could not be realised, because it took more time and resources than initially planned

10. Energy usage

In order to calculate the amount of energy which is needed to power a repeater module, certain things need to be calculated first.

Firstly, the amount of energy generated by a solar panel should be greater than the amount of energy consumed by the repeater module. This will guarantee that the repeater module always has enough power to operate.

Secondly, in this research the Arduino Nano will be used as the microcontroller in order to calculate the amount of energy that will be used.

Lastly, the consideration of a Wake-Up-Beacon (Faycal Ait Aoudia, 2016) (WUB) and a Wake-Up-Receiver (WURx) needs to be taken into consideration. This research includes the calculation for both with and without WUB and WURx. Comparing the power consumption for both methods are vital to ensure the lowest possible energy consumption (Gammon, 2012).

Table 5. Comparison between two LoRa modules

LoRa module	LoRa1280	LoRa1280F27
Power	12.5dBm	27.0dBm
Sensitivity	-132dBm@LoRa	-132dBm@LoRa
Receiving current	<10mA	<17mA
Emission current	<60mA	<600mA
Sleep current	<1uA	<6uA
Transmission rate	0.476~202Kbps@LoRa	0.476~202Kbps@LoRa
Transmission distance	1.7KM	4KM

The decision was made to shift the priority from researching energy usage to researching the network communication protocol. Thus, no conclusion can be made from this research.

The team believes that the system will not be consuming enough power to drain a solar-powered battery that could be used with the LoRa modules.

11. Results

11.1. Range test

To test the full range of a single module, a range test has been conducted.

Distance: **2.60km** (2 points)

Due to the flat terrain of the Netherlands, it was quite difficult to conduct long line of sight range tests. A range of 2.6 kilometres has been achieved between 2 modules, though through optimization a longer distance is a possibility.

This range test was done with the following settings:

- Bandwidth: 200
- Spreading Factor: 12
- Power: 10
- Coding rate: 4/8
- Frequency: 2445000000 Hz

11.2. Network test

To test the entire network, a network test was conducted. This test was performed to test the functionality of the sensor, repeater and gateway nodes. In this test, a single sensor, 2 repeaters and a single gateway module were used. Because only the network protocol needed to be tested, and not the range, the antennas were demounted from the modules to decrease the communication distance.

Each node was placed inside the range of the next node. The nodes were placed behind corners inside a building, and an extra check was done to ensure that each node could only receive data from 1 other node.

Sensor data was generated from the sensor node and transferred over the network. In the end, the sensor data was successfully received at the gateway.

```
Sensor: 10
10dBm Packet> ʕ□□□
  BytesSent,8   CRC,B3AE   TransmitTime,818mS   PacketsSent,34
98s   Z□□□
10dBm Packet> Z□□□
  BytesSent,8   CRC,EB3   TransmitTime,818mS   PacketsSent,35
```

```
10dBm Packet> g
                   BytesSent,8   CRC,1E0B   TransmitTime,819mS   PacketsSent,231
10
```

11.3. Research questions

In the beginning of this project, certain research questions were proposed. These questions are answered and shown in the following paragraphs. The proposed research question are as follows:

- In what kind of landscape should this project be realised?
- What communication protocol should be used?
- How often should data be transferred over the network?
- What is the longest reliable distance between network nodes?
- How can the nodes be protected against weather conditions?
- How can the nodes work with the lowest possible amount of power usage?
- What kind of hardware should be used?

Landscape

The landscape this proof of concept will be realised in is a mountainous landscape. There are forest areas which can cause interference in the connection between LoRa modules. When realising the proof-of-concept RAN has to make sure there is a line-of-sight connection between the LoRa modules.

Communication protocol

The communication protocol needs to contain certain data. Each message needs an id and the data that it sends. The messages look different depending on what kind of data is sent. If heartbeat data is sent then only the id of the sending repeater is shown in the message, if sensor data is sent then a lot more data about the sensor is shown in the message.

Data transfer times

The number of times data has to be sent over the network is variable. During the Monsoon it is advised to increase the number of times data will be sent over the network. Because during these times the occurrence of a flood is more likely than in the off-season. The number of times is up to RAN to decide.

Repeater Distance

Thorough testing has been done to find the maximum distance between two repeaters while still being able to keep a reliable connection. The results of these tests as shown earlier in this document is that each LoRa module is able to reach a minimum distance of 2.6 km.

Weather protection

When the research phase of this project was completed, it was decided that weather protection will be outside of the scope of the project. This is because the team does not have sufficient technical knowledge about creating weather-proof housings.

However, mounting the LoRa modules inside a case has been taken into account during the PCB designing process. Mounting holes have been added to the PCB so it could easily be mounted inside a custom weather-proof case.

Power usage

The power usage research was not completed within this proof of concept project, there is however documentation on the power usage as shown in the chapter energy usage. Also the choice for hardware was made considering low power consumption. The only missing point is that there is not a full test showing the power usage of the hardware during testing.

Hardware

The choice of hardware was hugely made upon low power consumption and distance. The LoRa module was chosen because it could reach a distance of at least 2.5 km and the Arduino Nano Every microcontroller was chosen because of its low power consumption.

11.4. Goals

The goals that have been set at the beginning of this project are as follows:

- A sensor must be placed at the source of the flooding to know if a flood is going to happen.
- A village must be notified over the 2.4 GHz network using LoRa.
- The desired range between the sensors and the village is 25 km.
- The energy usage needed by the network cannot be higher than 20 Watt.
- All the hardware used in the network must be solar powered.
- The hardware must be protected from all weather conditions like rain and lightning.

The sensor

For this proof of concept an example sensor was used to act like the sensor RAN uses for their water level readings. The sensor used for testing purposes was an ultrasonic sensor. The data from this sensor was able to travel across the network from the sensor module to the gateway.

Notifying villages

The ability to send data from a sensor over a LoRa 2.4GHz network has been realised. Thus, once the data is received in the gateway it is up to RAN to realise a way to warn the inhabitants of villages once a flood is detected by the system.

25km range

Each repeater in the realised project is able to reach a distance of 2.6 km. Using multiple of these repeaters, the desired range of 25km can be reached. This range can also be extended as needed by adding new repeaters to the network.

Energy usage

Because the energy usage research has not been finished it is not certain if the power usage is below 20 Watt. However because all the hardware chosen does have low power consumption the change is that it does not go higher than 20 Watt.

Since the priority has been shifted from the energy consumption to the working of the network this part of the research has been dropped midway

Solar power

The usage of solar panels was not specifically tested in this proof of concept. However, a battery with a solar panel can easily be connected to the LoRa modules via a micro-USB connection.

Casing

Mounting the LoRa modules inside a case has been taken into account during the PCB designing process. Mounting holes have been added to the PCB so it could easily be mounted inside a custom weather-proof case.

However, the team does not have sufficient technical knowledge about developing casings. The decision has been made to place this goal outside of the scope of the project.

12. Conclusion and recommendation

This chapter answers the research question identified in the chapter project result. This is discussed by means of a conclusion and recommendation.

12.1. Conclusion

The research question to be answered by means of this report is: “How can a long-range flood detection system be implemented in Nepal in a way that people can be evacuated in a timely manner?”. This paragraph will give a short summary of the problem, the implementation and the result of the project.

The problem this project was trying to solve was that the people in Nepalese villages were not warned about an incoming flood which caused many casualties. However, building a network in Nepal to warn the Nepalese villages of an incoming flood proved difficult for the Robotics Organization of Nepal (RAN). The reason for this was because RAN did not have the appropriate technical knowledge to realise the project. Because of restrictions within Nepal, the network had to be built within the 2.4GHz bandwidth. Also, because there is no main power grid in Nepal the system must work on batteries and solar panels.

To help RAN, students of the Avans University of Applied Sciences have helped RAN to build a proof-of-concept. For the proof-of-concept, a mesh network had to be realised. To do that, research has been done into hardware like the microcontroller and the LoRa module. As well as research into software like the working of a mesh network, communication protocol and power usage. These different parts combined form the LoRa early warning flood detection system project.

The finished project can now send sensor data across the network from one point to another. To make sure that each repeater in the network can reach the next repeater, thorough testing has been done on the range of the repeaters. These results can be found in the chapter results. With this we have answered the following research question: “How can a long-range flood detection system be implemented in Nepal in a way that people can be evacuated in a timely manner?”

12.2. Recommendation

The recommendation to RAN is to use this proof-of-concept project to make a fully working early warning flood detection system. The sensor node code currently operates using an ultrasonic sensor. The correct water level sensor will need to be implemented in Nepal. Currently, the only type of data the gateway logs, is sensor data received by the network. The connection to the internet still needs to be implemented. Further development to the gateway can be done by RAN.

The communication protocol is optimized to send data in a 16-bit format. This is done to minimize the data transfer over the network in order to reduce energy consumption. In case this format is not big enough, the format can be enlarged by changing the communication protocol and changing the size of the Sensor Data block. Note: The checksum and parity bits will also change positions.

The research regarding energy usage, casing, solar panels and batteries were not completed. Therefore, it is advisable to continue this research.

It is also recommended that a form of AES is used for encryption. The reason for this is because it works better in a broadcasting environment. The preference is for a less heavy form of AES, like AES128.

This proof of concept may not be in itself used as the final because the students of Avans cannot be in any way directly responsible for the lives of the people in the Nepalese villages. The students will still be available for any future questions from RAN. However, it is true that after this proof of concept, the students will not make any further extensions to this project.

Bibliography

Arduino_store. (n.d.). Retrieved from https://store.arduino.cc/arduino-nano

E.E. Bronkhorst, D. K. (2018, July 2). Retrieved from https://repository.tudelft.nl/islandora/object/uuid:d0e15e21-3a61-4e7b-996b-1cecace6f83d/datastream/OBJ/

Faycal Ait Aoudia, M. M. (2016, August). Retrieved from https://www.researchgate.net/publication/309498865_A_Low_Latency_and_Energy_Efficient_Communication_Architecture_for_Heterogeneous_Long-Short_Range_Communication

Gammon, N. (2012, January 13). Retrieved from http://www.gammon.com.au/forum/?id=11497

Rosmianto. (2020, June). Retrieved from https://electronics.stackexchange.com/questions/506354/teensy-4-x-imxrt1062-max-output-current

Wikipedia. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Omnidirectional_antenna

Wikipedia. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Parabolic_antenna