TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Elektrotechnik und Informationstechnik

Professur für Mess- und Sensortechnik

Research Project

Documentation of an Open Source Internet of Things Gateway and Integration of PanStamp Architecture

Author:

Muhammad Aashir Ayaz

Matriculation No: 448046

Supervisors:

Prof. Dr. Ing- Olfa Kanoun

M. Sci. Dip. -Ing.Sabrine Kheriji

Dip. -Ing Martin Götz

Information and Communication Systems

ABSTRACT

The presence of documentation helps keep track of all aspects of an application and it improves on the quality of a software product. Successful documentation will make information easily accessible, provide a limited number of user entry points, help new users understand quickly, simplify the product and reduce costs. For a programmer, it is always essential to have a reliable documentation**.** The aims of the project are the documentation of existing systems i.e. MSP430 and Arduino modules which involve point to point wireless communication and implementation of novel panStamp module in the MST-Gateway system and also document this new implemented panStamp module accordingly.

In this project, two PanStamp modules share information i.e. Received Signal Strength power Indicator (RSSI) and Link Quality Indication (LQI) values, through wireless communication and the receiver module is connected a Raspberry-Pie through serial communication. The run time values of RSSI and LQI are arranged and stored in database and uploaded to online server. The remote PanStamp transmits data to the other PanStamp set-up connected to the server. The visualization of real time data is done in MATLAB to observe and facilitate the user.

To provide detailed and clear source of written information for new users and developers, it is required to have easily and user friendly approach for documentation. For this purpose, Dokuwiki plays role for MST-Gateway structure documentation including hardware and software detailed information but dokuwiki cannot support the source code documentation. Therefore Doxygen is use for documenting source codes with more advance graphs and diagrams. All aspect of project are achieved successfully and real time evaluation of implementation is done through visualization.

ACKNOWLEDGEMENT

It is my pleasure to present the project report on “Documentation of an Open Source Internet of Things Gateway and Integration of PanStamp Architecture”.

There are a number of people whose contributions have helped make this project possible. Firstly, I would like to thank my supervisors, Sabrine Kheriji and Martin Götz who have spent time with me discussing various aspects of the project and providing ideas and alternatives to problems I encountered along the way. Also, I would like to thank Prof. Olfa Kanoun for giving me the opportunity to do my research project in the Chair of Measurement and Sensor Technology. Finally, I would like to express my thanks to all those who were involved directly or indirectly in completion of my project.

November 16, 2018, Chemnitz

Date, Place

Muhammad Aashir Ayaz

Table of Contents

List of Figures

Chapter No. 1

1 Introduction

1.1 Overview

This research project has two main objectives: The first objective of this project was to set up a wireless gateway node using PanStamp [1] modules and doing serial communication with RassberryPI. Two PanStamp modules are used for this purpose. The PanStamp modules are used to transfer data between two points wirelessly. The RassberryPI microcontroller is used to process the data and help in transferring it to the server. The project further can be divided into the following sections:

- Design of protocol to transfer data (RSSI and LQI) between the PanStamp modules
- Implementation of the protocol by programming with the Arduino
- Development of python application to work as a gateway between the PanStamp and RasberryPI. Its further doing visualization and storage in database
- Visualization of the transmitted data in Matlab
- Storage of PanStamp related information into a database

The second objective do the documentation of the existing gateway which consist on a MSP-EXP430G2 [2] and Arduino Uno communicate wirelessly with Xbee as well as new module PanStamp after implementation. For this purposed, two documentation software's Doxygen [3] for firmware documentation and Dokuwiki [4] for structure documentation are used. Documentation has two parts: Internal and external documentation. Internal documentation is written in a program as comments. External documentation is written in a place where people who need to use the software and hardware can read about them how to use the software and hardware.

Each section is described in detail later. A wireless network system such as this has applications in the fields of healthcare monitoring, process management, environmental sensing, and industrial monitoring and in the entertainment industry.

1.2 Motivation

For a programmer, it is always important to have a detail, clear and reliable source of information for better perspective**.** The presence of documentation helps new developers and users to keep track of all aspects of an application, better understanding of its different features and it improves on the quality of a software product. Its main objectives are development, maintenance and providing knowledge to other developers. Successful documentation will make information easily accessible, help new users understand quickly and reduce support costs. For this project a detailed documentation of an existing MST_Gateway with its structure and endpoint firmware is given.

Environmental monitoring systems have increased in importance over the years. Wireless sensor networks (WSNs) have a high impact in modern research. WSNs are the backbone for digitalization in industry and many application. Main challenges of the sensor network are the wireless communication, the power supply and energy management. Within this project, a PanStamp development board is used to build up a bidirectional communication as system nodes.

1.3 Project Aims

The first aim is to add a new wireless sensor network such as PanStamp which is not implemented yet in the MST-Gateway. Moreover, two PanStamp modules extracted RSSI (receiver signal strength indication) and LQI (link quality index) values which are useful for quality of service measurements or location approximation algorithms are used serial communication of PanStamp coordinator with RasberryPi is established gateway. Then a firmware stores the values into database and results are visualized on Matlab.

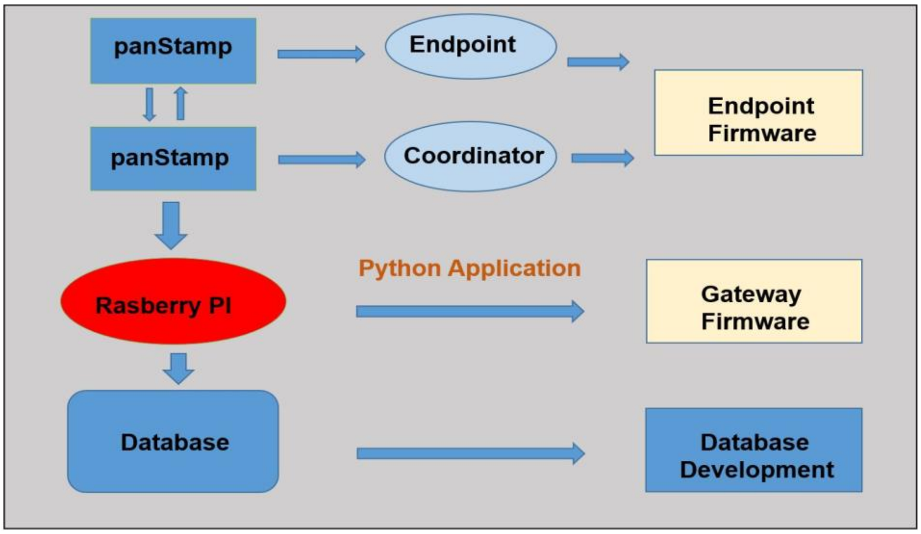


Figure 1.1 Implementation on PanStamp

The second objective of the research is to provide a detailed and clear documentation of an existing MST-gateway including each step of gateway structure and firmware. For this purpose, two software's are used named as Dokuwiki [4] for gateway structure and Doxygen [3] for source code comment.

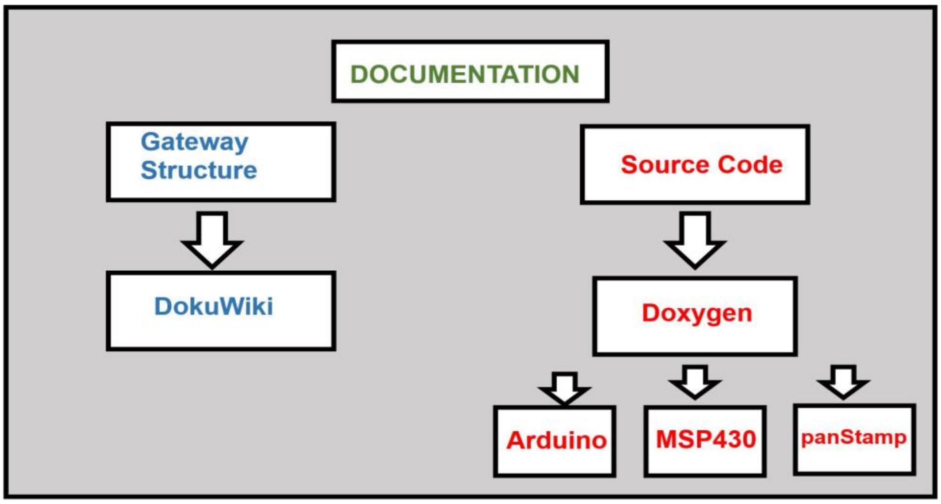


Figure 1.2: Documentation of MST- Gateway structure

Chapter No. 2

2 Technical Description

2.1 Hardware Components

The hardware components used in the system are:

- PanStamp Board
- MSP430
- RasberryPI
- Arduino

2.1.1 PanStamp Board

PanStamp [1] is an autonomous low-power wireless modules programmable from the Arduino IDE and made for telemetry and control projects. These modules communicate over the free 868-900-915 MHz bands available worldwide and are designed to last for months and even years when powered from simple alkaline batteries, depending on the duty cycle and transmission interval programmed [1].

In addition, PanStamps form complete wireless ecosystems with direct connectivity to the Internet. The company provided software applications for configuring wireless networks and turning any computer into an automation server. PanStamp also provides customized Raspbian images for Raspberry PI giving this popular computer platform a key role in any automation project as a wireless-IP gateway with stand-alone capabilities [1].

PanStamps include an on-board microcontroller so they can be run autonomously without any external processor. There are two different models of PanStamp, each using a different microcontroller:

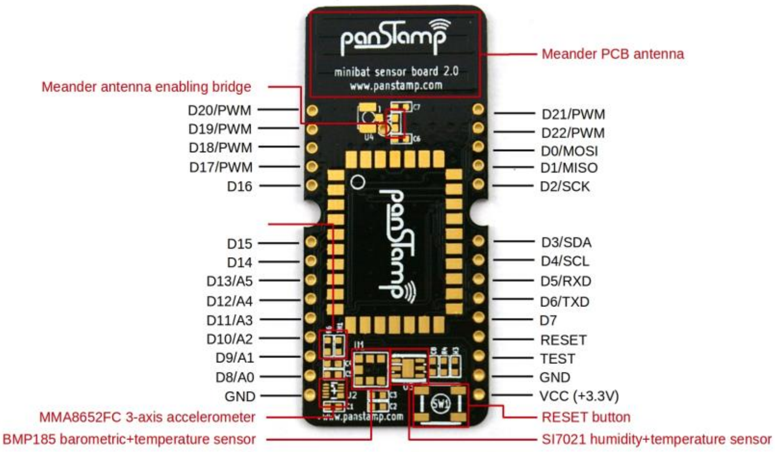


Figure 2.1: PanStamp NRG 2.0 [1]

- **PanStamp AVR**

This model hosts an Atmega328p MCU made by Atmel. This is the same MCU available in all Arduino UNO's so they are fully compatible with every version of the Arduino IDE. PanStamp (the company) provides an Arduino library for these modules, featuring a full communication stack and protocol that takes less than 8 Kbytes in Flash [1].

- **PanStamp NRG**

NRG modules share the same footprint and similar pin out as the AVR model but use a MSP430 core instead of an Atmega MCU. These modules were specially designed to be programmed from Arduino 1.6.

2.1.2 MSP430

The MSP430 [2] is a mixed-signal microcontroller. Built around a 16-bit CPU, the MSP430 is designed for low cost and specifically, low power consumption embedded applications [2].

The best way of entering into embedded system learning and development platform is using Texas Instrument's microcontroller: MSP430 Launch Pad, which was introduced in July 2010. The Launchpad is fabricated in supportive way to use the different input/output ports more easily and precisely. The Launchpad MSP430 of G series microcontroller is operated for the implementation. The key features are:

- Low to ultra-low power consumption
- Several programmable power saving modes
- On-chip/on-board emulation and debugging features

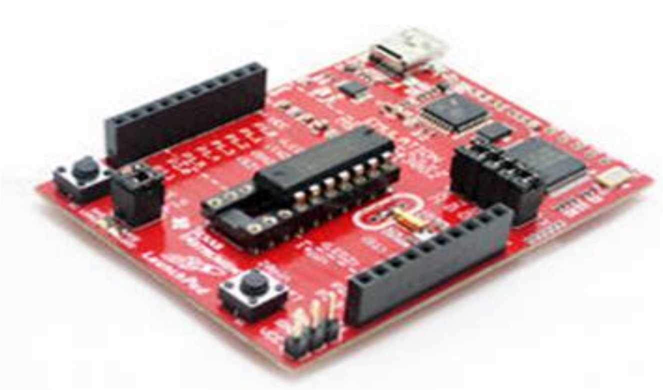


Figure 2.2: MSP430 development board [2]

2.1.3 Raspberry Pi

The Raspberry Pi [5] is a single-board computer developed by the British company. The calculator contains a one-chip system from Broadcom with an ARM microprocessor , the footprint of the board is about the size of a credit card . The Raspberry Pi came on the market in early 2012; it has great market success for programming and experimenting. It has very simple built-up computer was developed by the foundation with the aim to facilitate the acquisition of programming and hardware knowledge for young people.

There is a large range of accessories and software for numerous applications. For example, the use as a media centre is wide spread because the computer can decode video data with full HD resolution (1080p) and output via the HDMI interface. As operating system are primarily adapted Linux - distributions with graphical user interface for use; for the latest model, Windows 10 also exists in a special Internet of Things version without graphical user interface. The starting procedure is usually done by a removable SD memory card as an internal boot media . With the newer generation with the BCM2837, it is also possible to boot from a USB mass storage device or network. A native interface for hard disk drives is not available, additional mass storage can be connected via USB interface , for example external hard disks or SSDs or USB memory sticks [5].

2.1.4 Arduino

Arduino [6] is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package [6].

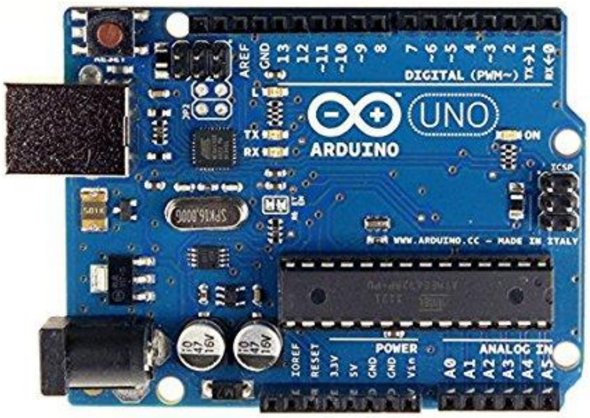


Figure 2.3: Arduino development board [6]

2.2 Software Components

The corresponding software used were:

- Arduino IDE
- Python
- Visual Studio Code
- Doxygen
- Dokuwiki

2.2.1 Ardiuno IDE

The Arduino comes with a simple software called the Arduino software. It is also used for PanStamp. It can be downloaded from Arduino's official website. It is a simple software with a text editor for writing the code, a message area, a text console that displays the progress, a toolbar with buttons for common functions, and a series of menus. The code can be typed in the text editor compiled. The Arduino software checks the code for any errors before flashing it to PanStamp. Also, it has the provision to send simple textual data to and from the board using a serial monitor. For programming the microcontroller, Arduino's IDE provides support for C and C++. The board has RX and TX LEDs that flash while data is being transmitted to and from between an Arduino and the computer. All these features keep track of the data flow [7].

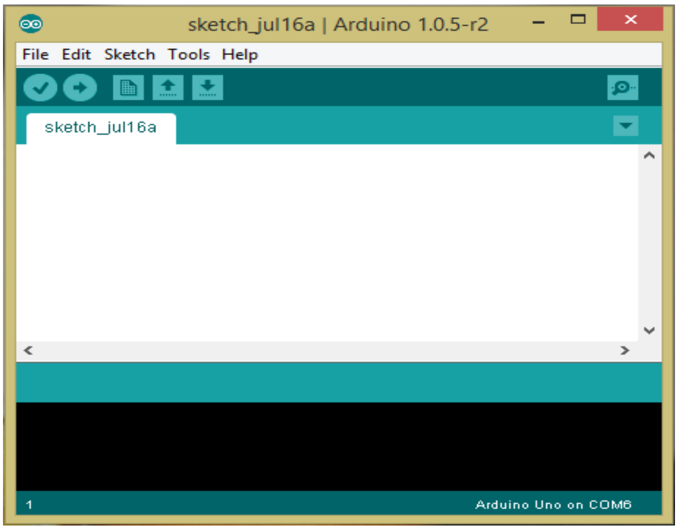


Figure 2.4: Arduino IDE [7]

2.2.2 Python

Python [8] was designed with the aim of simplicity and clarity. This is achieved mainly by two measures. For one thing, the language can handle relatively few keywords . On the other hand, the syntax is reduced and optimized for clarity. This makes python-based scripts much tighter than in other languages.

Python is a universal, commonly interpreted higher-level programming language . It claims to promote a readable, concise programming style. For example, blocks are not structured by curly braces, but by indentations. Because of its clear and concise syntax, Python is easy to learn [8].

2.2.3 Visual studio code IDE

Visual Studio Code [9] is a free source code editor from Microsoft . Visual Studio Code is cross-platform available for the operating systems Windows , mac OS and Linux . Visual Studio Code is based on the Electron framework and enables u. a. Syntax highlighting , code folding , debugging , auto completion and versioning. Visual Studio Code has nothing in common with Visual Studio . Unlike Visual Studio, Visual Studio Code does not work on project files, but on source code files and folders.

2.2.4 Doxygen

Doxygen [3] is a documentation generator, a tool for writing software reference documentation. The documentation is written within code, and is thus relatively easy to keep up to date. Doxygen can cross reference documentation and code, so that the reader of a document can easily refer to the actual code.

2.2.5 Dokuwiki

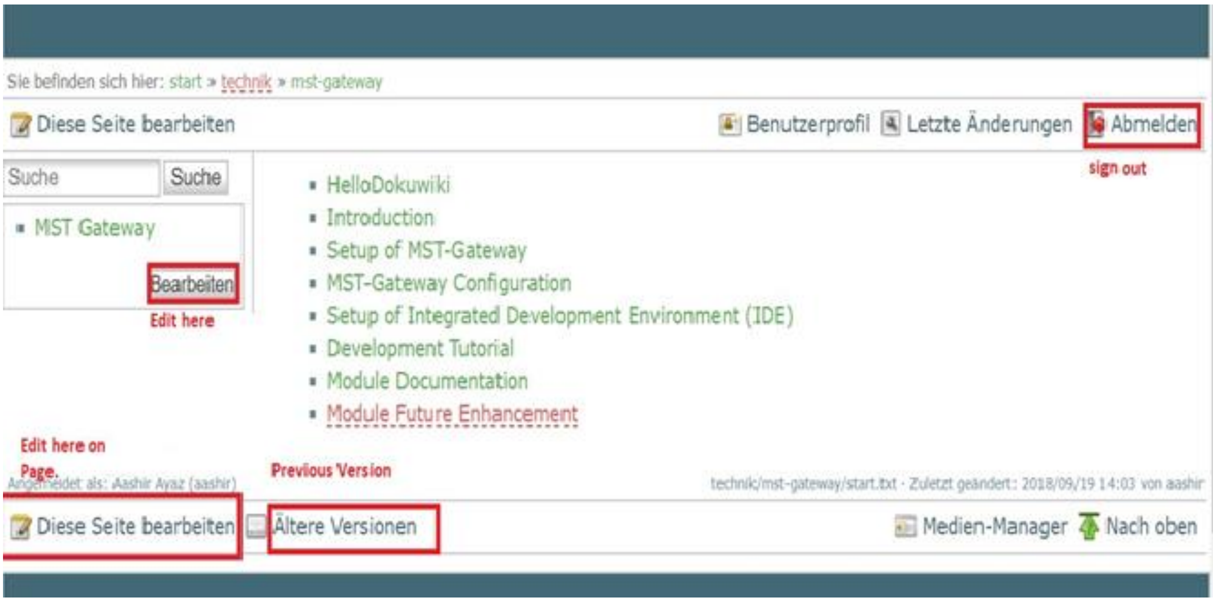
DokuWiki [4] is a wiki application licensed under GPLv2 and written in the PHP programming language. It works on plain text files and thus does not need a database. Its syntax is similar to the one used by MediaWiki.

Chapter No. 3

3 Documentation of the MST-Gateway

3.1 Dokuwiki

DokuWiki was designed by Andreas Gohr in June 2004. It is usable for bigger documentation projects. DokuWiki is a simple to use and highly versatile Open Source wiki software that doesn't require a database. It is preferable by users for its clean and readable syntax. The ease of maintenance, backup and integration makes it an administrator's friendly. Built in access controls and authentication connectors make DokuWiki especially useful in the enterprise context and the large number of plugins beyond a traditional wiki [4].



Sie befinden sich hier: start » technik » mst-gateway

Angemeldet als: Aashir Ayaz (aashir)

technik/mst-gateway/start.txt · Zuletzt geändert: 2018/09/19 14:03 von aashir

sign out

Edit here

Edit here on Page.

Previous Version

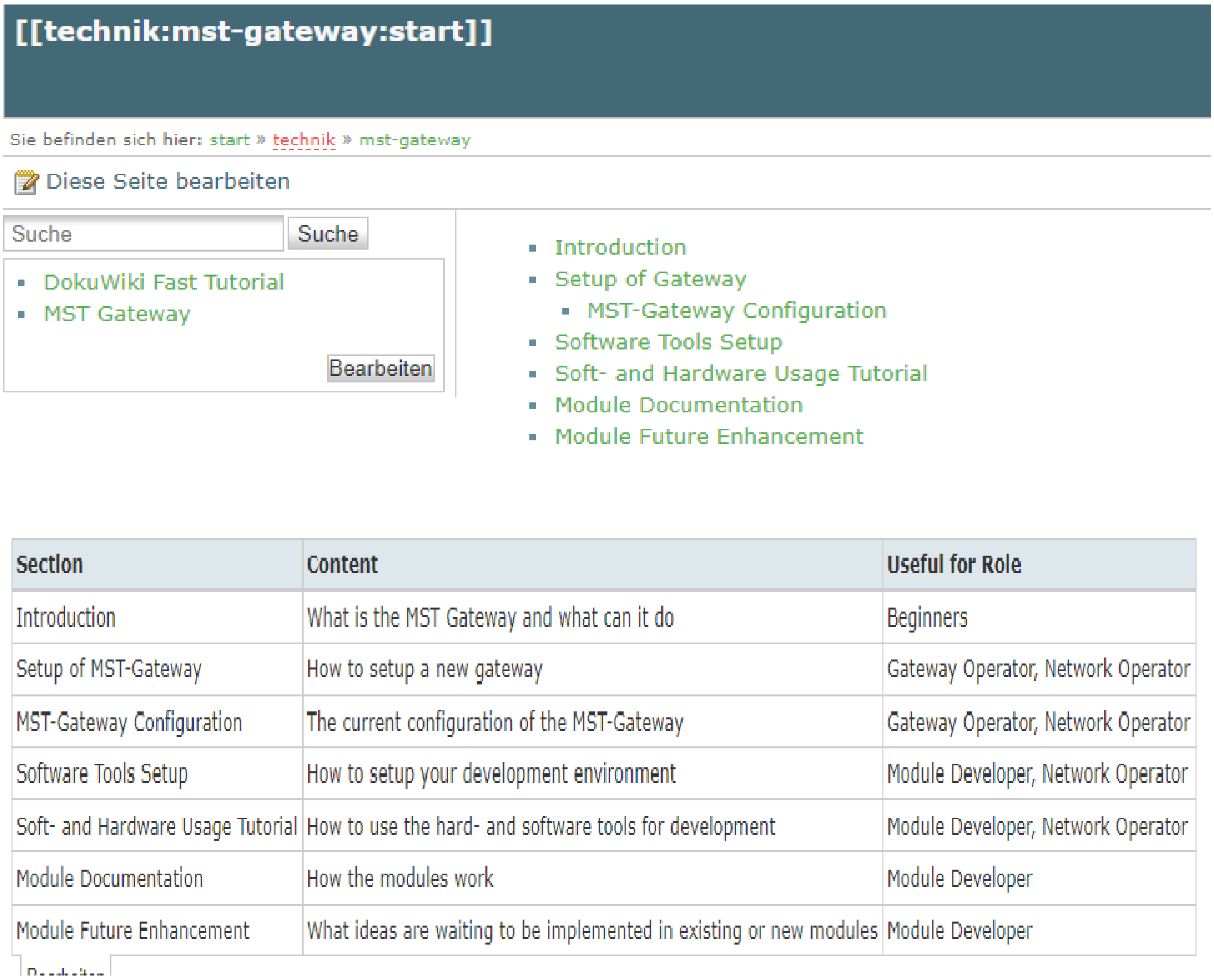
Figure 3.1: Dokuwiki front page

3.1.1 Uses

- Easy to install and use
- Low system requirements
- Large variety of extensions
- Open source

3.1.2 Gateway Structure

- Its look like

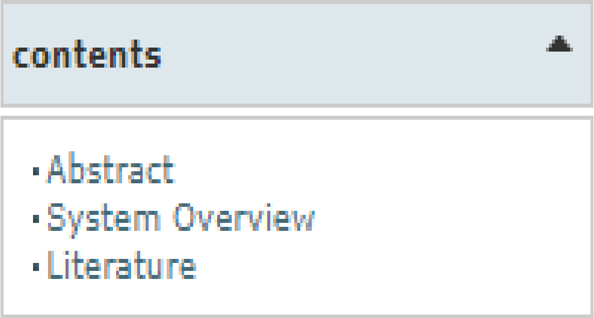

[[technik:mst-gateway:start]]
Sie befinden sich hier: start » technik » mst-gateway
Diese Seite bearbeiten
Suche
Suche
DokuWiki Fast Tutorial
MST Gateway
Bearbeiten

- Introduction
- Setup of Gateway
  - MST-Gateway Configuration
- Software Tools Setup
- Soft- and Hardware Usage Tutorial
- Module Documentation
- Module Future Enhancement

Section	Content	Useful for Role
Introduction	What is the MST Gateway and what can it do	Beginners
Setup of MST-Gateway	How to setup a new gateway	Gateway Operator, Network Operator
MST-Gateway Configuration	The current configuration of the MST-Gateway	Gateway Operator, Network Operator
Software Tools Setup	How to setup your development environment	Module Developer, Network Operator
Soft- and Hardware Usage Tutorial	How to use the hard- and software tools for development	Module Developer, Network Operator
Module Documentation	How the modules work	Module Developer
Module Future Enhancement	What ideas are waiting to be implemented in existing or new modules	Module Developer

Figure 3.2: Front View of MST Gateway structure using Dokuwiki

- Introduction has following contents

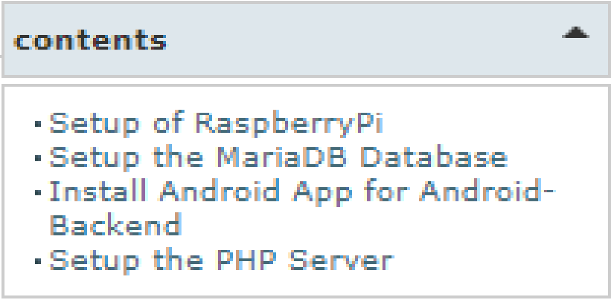


contents

- Abstract
- System Overview
- Literature

Figure 3.3: Introduction content of Dokuwiki

- Setup of MST-Gateway

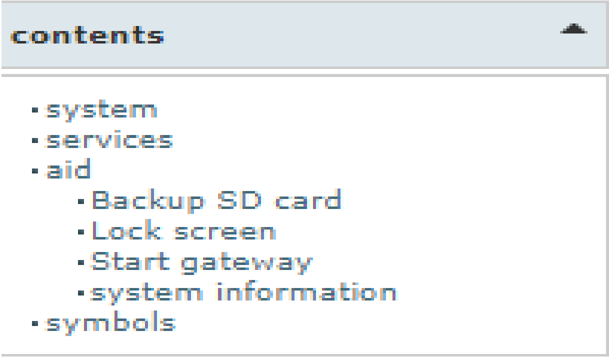


contents

- Setup of RaspberryPi
- Setup the MariaDB Database
- Install Android App for Android-Backend
- Setup the PHP Server

Figure 3.4: Setup in Dokuwiki

- MST-Gateway configuration

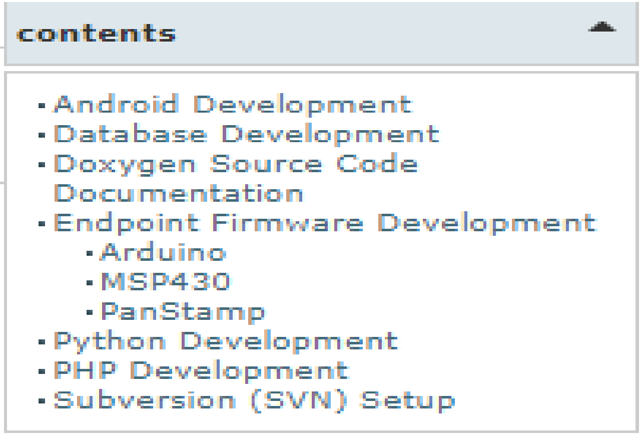


contents

- system
- services
- aid
  - Backup SD card
  - Lock screen
  - Start gateway
  - system information
- symbols

Figure 3.5: MST-Gateway configuration

- Software tool setup

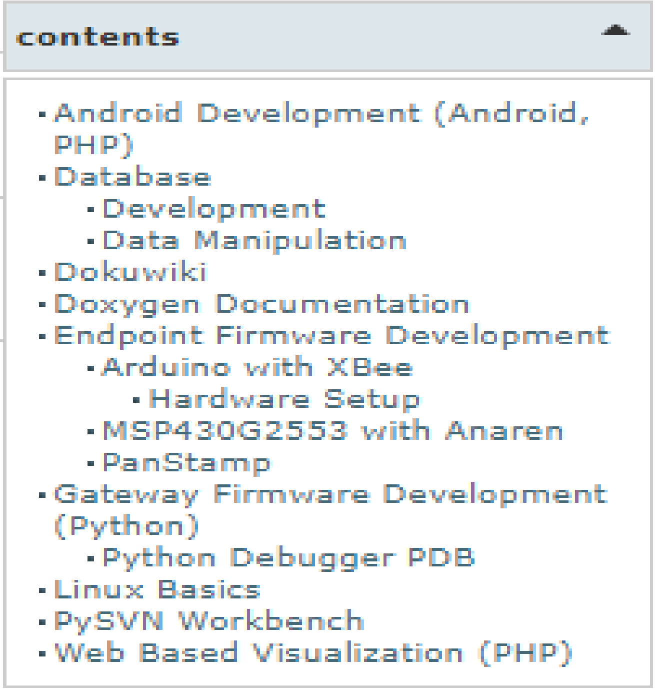


contents

- Android Development
- Database Development
- Doxygen Source Code Documentation
- Endpoint Firmware Development
  - Arduino
  - MSP430
  - PanStamp
- Python Development
- PHP Development
- Subversion (SVN) Setup

Figure 3.6: Software tool setup

- Software and hardware usage tutorial



contents

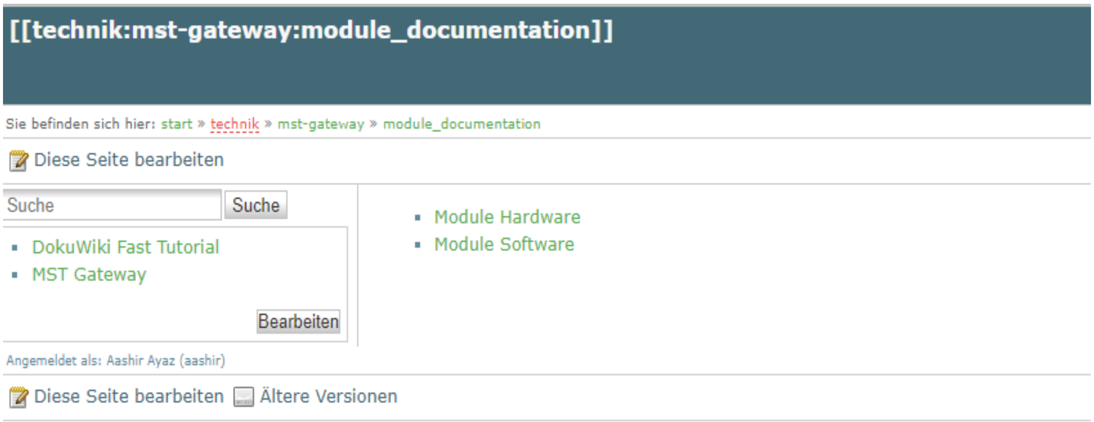
- Android Development (Android, PHP)
- Database
  - Development
  - Data Manipulation
- Dokuwiki
- Doxygen Documentation
- Endpoint Firmware Development
  - Arduino with XBee
    - Hardware Setup
  - MSP430G2553 with Anaren
  - PanStamp
- Gateway Firmware Development (Python)
  - Python Debugger PDB
- Linux Basics
- PySVN Workbench
- Web Based Visualization (PHP)

Figure 3.7: Software and hardware usage tutorial

- Module Documentation

  It is divided into two sections:

  1. Module Hardware
  2. Module Software



[[technik:mst-gateway:module_documentation]]

Sie befinden sich hier: start » technik » mst-gateway » module_documentation

Diese Seite bearbeiten

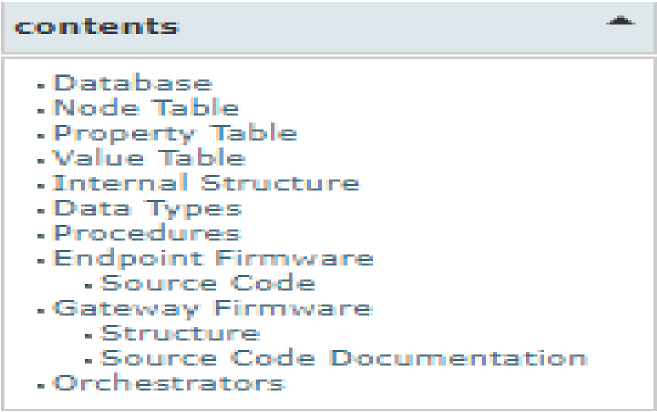
- DokuWiki Fast Tutorial
- MST Gateway

- Module Hardware
- Module Software

Angemeldet als: Aashir Ayaz (aashir)

Figure 3.8: Module documentation view in Dokuwiki

- Module software



contents

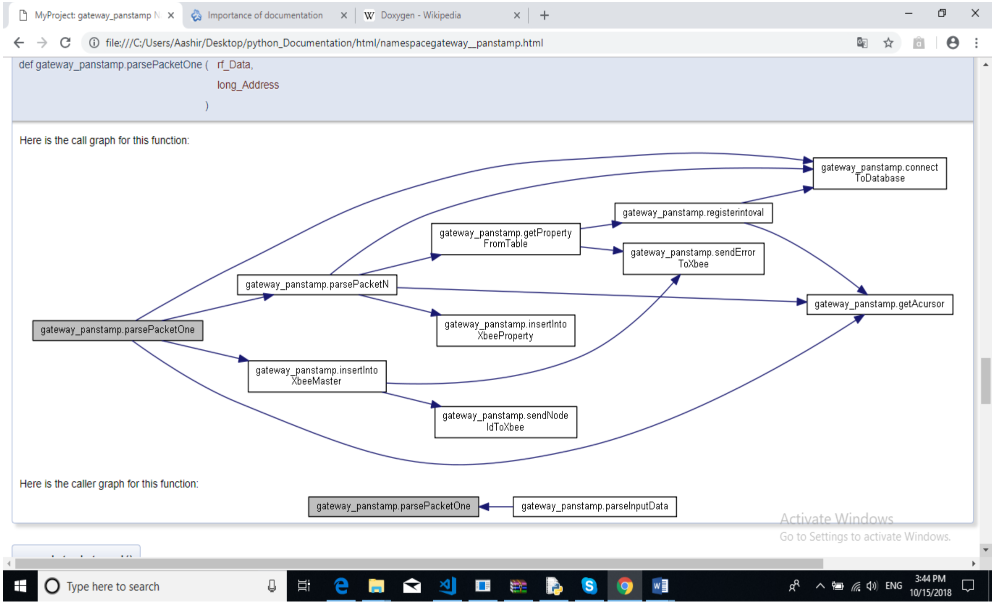
- Database
- Node Table
- Property Table
- Value Table
- Internal Structure
- Data Types
- Procedures
- Endpoint Firmware
  - Source Code
- Gateway Firmware
  - Structure
  - Source Code Documentation
- Orchestrators

Figure 3.9: Software view

3.2 Doxygen

3.2.1 Overview

Doxygen is free software. It has more options and easy to use because it has good markdown support [3].



```
def gateway_panstamp.parsePacketOne (  rf_Data,
                                       long_Address
                                     )
```

Here is the call graph for this function:

gateway_panstamp.parsePacketOne
gateway_panstamp.parsePacketN
gateway_panstamp.getProperty FromTable
gateway_panstamp.registerintoval
gateway_panstamp.connect ToDatabase
gateway_panstamp.sendError ToXbee
gateway_panstamp.getAcursor
gateway_panstamp.insertInto XbeeProperty
gateway_panstamp.insertInto XbeeMaster
gateway_panstamp.sendNode IdToXbee

Here is the caller graph for this function:

gateway_panstamp.parsePacketOne
gateway_panstamp.parseInputData
MyProject: gateway_panstamp
Importance of documentation
Doxygen - Wikipedia
file:///C:/Users/Aashir/Desktop/python_Documentation/html/namespacegateway__panstamp.html
Activate Windows
Go to Settings to activate Windows.
Type here to search
ENG
3:44 PM
10/15/2018

Figure 3.10: Doxygen view

3.2.2 Design

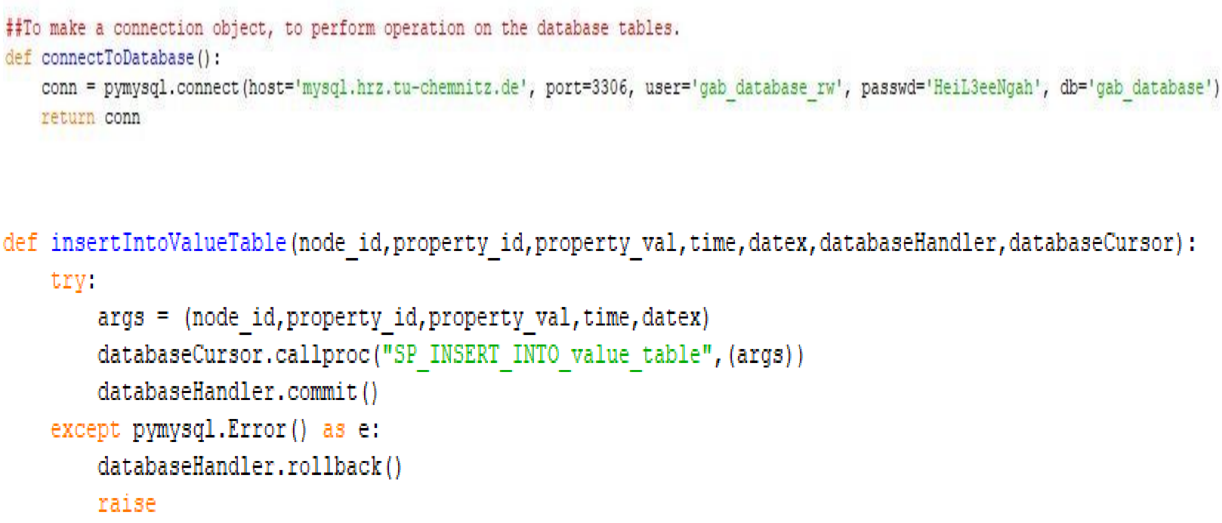
Doxygen extracts documentation from source file comments and it supports the documentation tags used in the Qt toolkit and can generate output in HyperText Markup Language (HTML) as well as in Microsoft Compiled HTML Help (CHM), Rich Text Format (RTF), Portable Document Format (PDF), LaTeX, PostScript or main pages.

3.2.3 Uses

- More programming languages such as C, C++, C#, PHP, Python and VHDL etc.
- Easy to use
- More options e.g. advanced diagrams and graphs
- Markdown support

3.2.4 Source Code

- PanStamp IDLE code

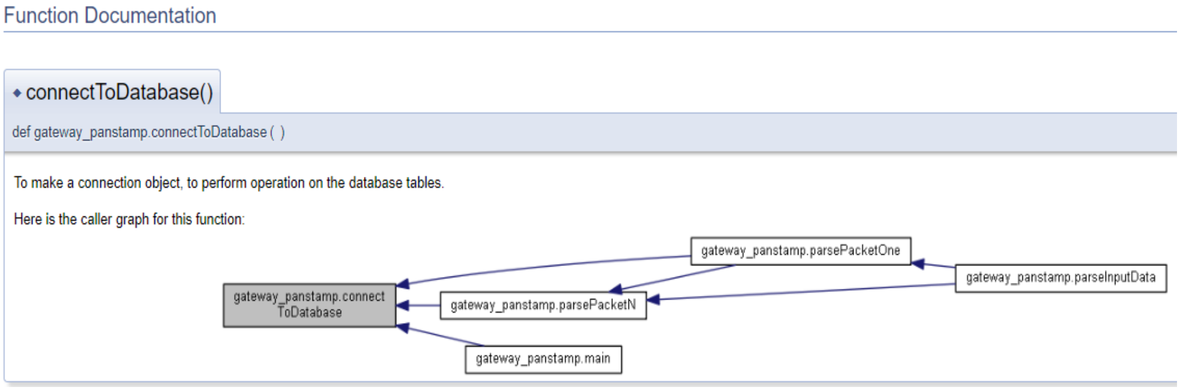


```
##To make a connection object, to perform operation on the database tables.
def connectToDatabase():
    conn = pymysql.connect(host='mysql.hrz.tu-chemnitz.de', port=3306, user='gab_database_rw', passwd='HeiL3eeNgah', db='gab_database')
    return conn


def insertIntoValueTable(node_id,property_id,property_val,time,datex,databaseHandler,databaseCursor):
    try:
        args = (node_id,property_id,property_val,time,datex)
        databaseCursor.callproc("SP_INSERT_INTO_value_table",(args))
        databaseHandler.commit()
    except pymysql.Error() as e:
        databaseHandler.rollback()
        raise
```

Figure 3.11: PanStamp IDLE code view

- PanStamp code view after Doxygen



Function Documentation

◆ connectToDatabase()

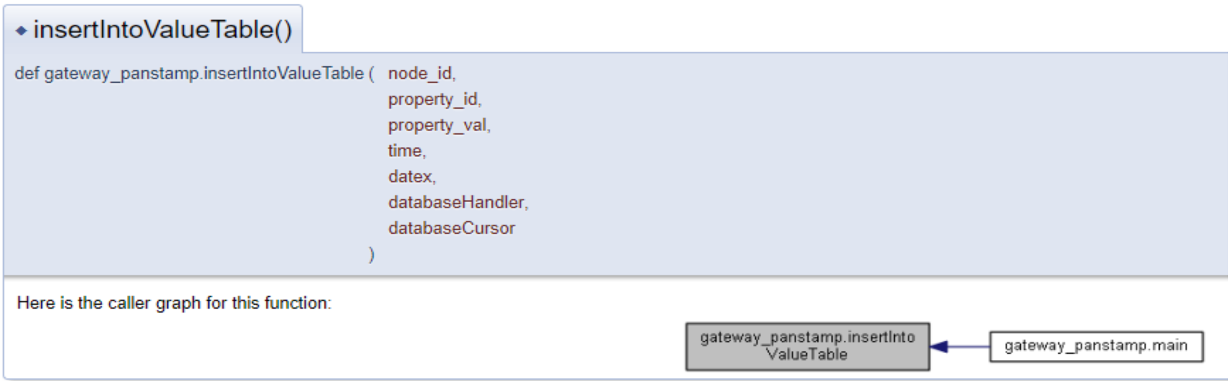
def gateway_panstamp.connectToDatabase ( )

To make a connection object, to perform operation on the database tables.

Here is the caller graph for this function:

gateway_panstamp.connect ToDatabase
gateway_panstamp.parsePacketOne
gateway_panstamp.parsePacketN
gateway_panstamp.parseInputData
gateway_panstamp.main

Figure 3.12: Doxygen function documentation 1



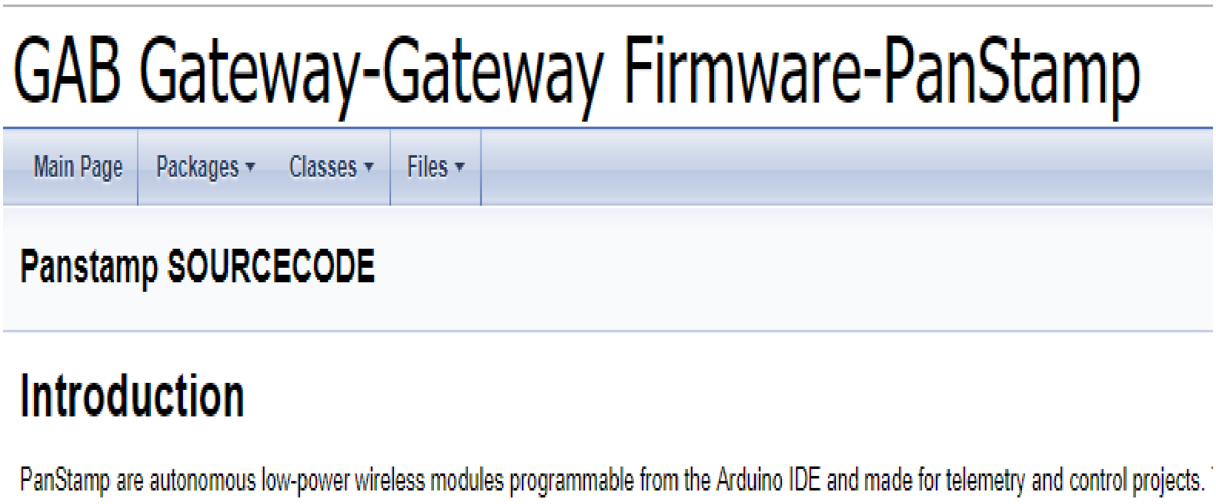
◆ insertIntoValueTable()

```
def gateway_panstamp.insertIntoValueTable ( node_id,
                                            property_id,
                                            property_val,
                                            time,
                                            datex,
                                            databaseHandler,
                                            databaseCursor
                                          )
```

Here is the caller graph for this function:

Figure 3.13: Doxygen function doumentation 2

- Doxygen main page view



GAB Gateway-Gateway Firmware-PanStamp

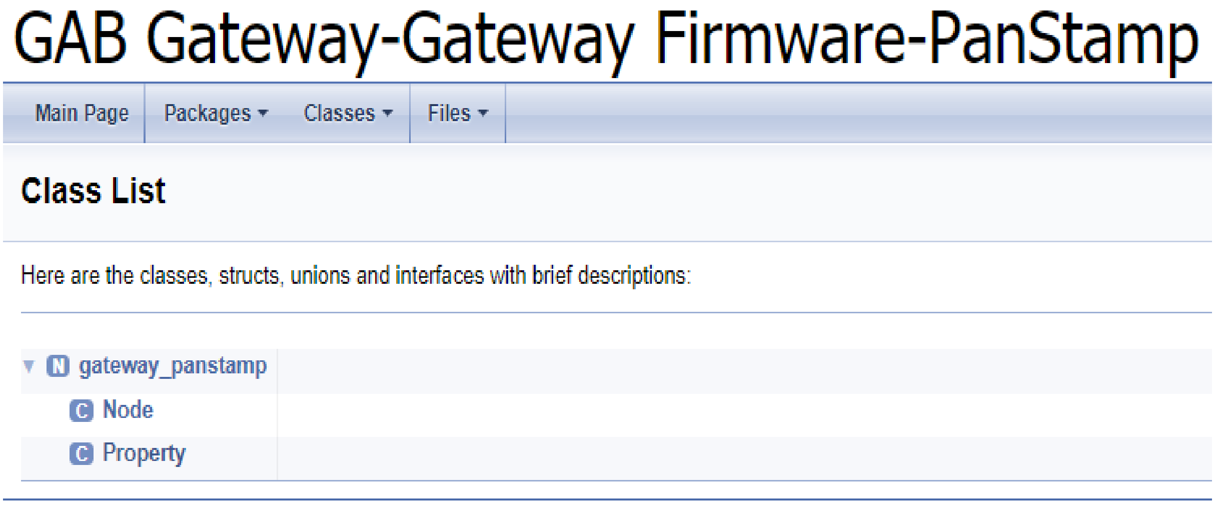
Panstamp SOURCECODE

Introduction

PanStamp are autonomous low-power wireless modules programmable from the Arduino IDE and made for telemetry and control projects.

Figure 3.14: Main page

- Doxygen class list



GAB Gateway-Gateway Firmware-PanStamp

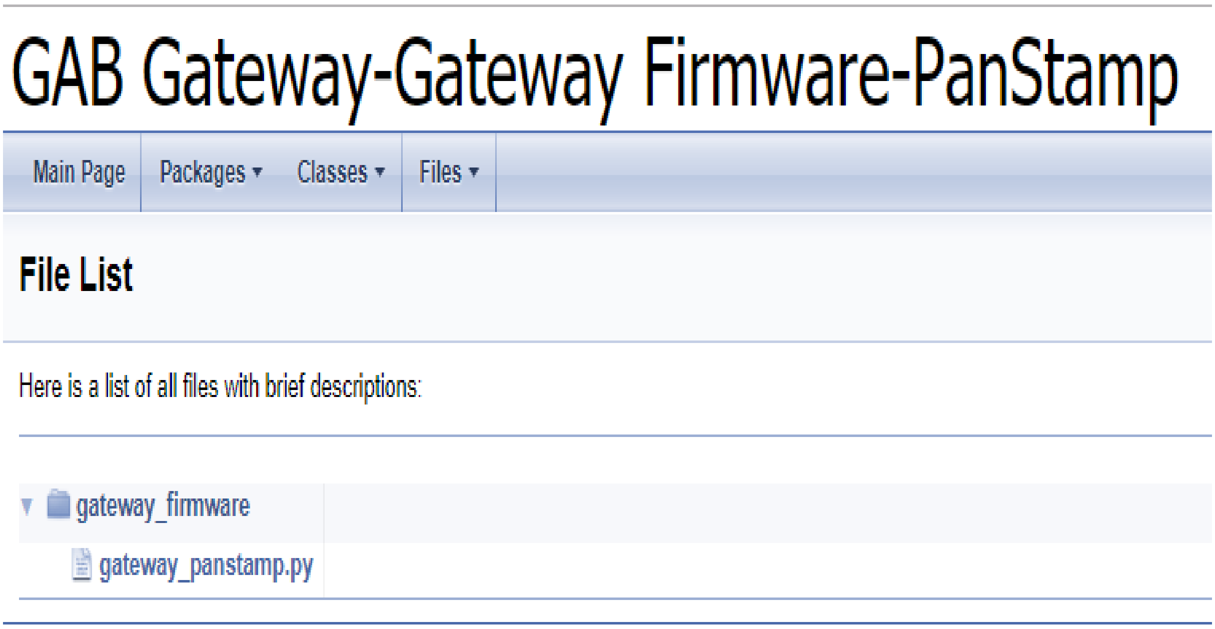
Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

▼ N gateway_panstamp
C Node
C Property

Figure 3.15: Class list

- Doxygen File list



GAB Gateway-Gateway Firmware-PanStamp

File List

Here is a list of all files with brief descriptions:

▼ gateway_firmware
gateway_panstamp.py

Figure 3.16: File list

Chapter No. 4

4 Functional Description

4.1 Design of the System

As seen in Figure 4.1, the system can be partitioned into the following sub systems.

- A couple of PanStamp systems that can communicate wirelessly
- Python TCP/IP socket to act as a gateway between the serial port and other applications
- Python application that inserts values into a database
- Matlab application that receives data from the python socket and displays it as a graph in real time
- Database that stores certain information after receiving it from the python socket

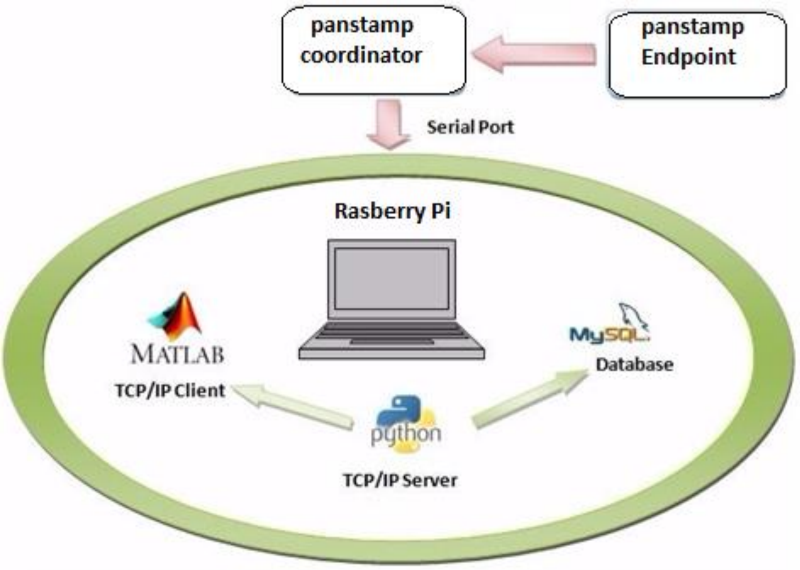


Figure 4.1: Illustration of the sub-components of the system and their interactions

The hardware consists of mainly two modules, the coordinator and the endpoint.

The endpoint continuously sends data to the coordinator, which can be visualized in the PC on the coordinator side. The data is also stored in the database for future reference.

The data sent is in the form of packets and the frame structures have been followed. Both the coordinator and endpoint PanStamp modules have been configured with Arduino Uno software sketches.

The Coordinator PanStamp communicates with the PC via the serial port and the data received by the Coordinator PanStamp is passed on to the PC for visualization. However, the serial port can interact with one application at a time. Therefore, we have to use Python to create a gateway between the serial port, the database and Matlab which is the visualization software.

4.2 Design of the Gateway Application

The gateway application is to be implemented using python. The main functions of this application are:

- Receive the data (RSSI and LQI) values from PanStamp modules.
- Analyze the data
- Categorize the data
- Based on the category, process the data
- Pass on the pansSamp details to the database
- Pass the values and the plot labels to Matlab

4.3 Design of the TCP/IP Server-Client Socket

When the data is passed on from the coordinator system to the serial port, we need a gateway to efficiently manage and redirect the data the way we want. For this reason, a TCP/IP socket was designed. The server side of the socket was decided to be made with python. It is an interpreted and interactive language with object oriented programming. Python is remarkably powerful and extremely simple in its syntax. So, it was chosen to perform this task.

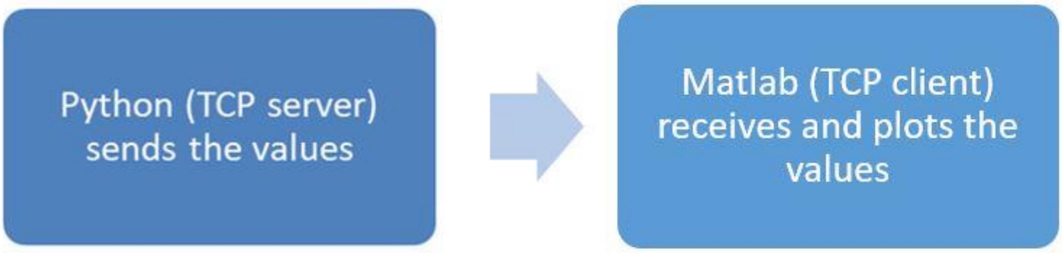

Python (TCP server) sends the values
Matlab (TCP client) receives and plots the values

Figure 4.2: TCP Server and client

The TCP/IP client was set up using Matlab. So, the data would pass from the python application to Matlab where it would be converted to a real-time graph. The reasons for choosing Matlab for the task were that it can evaluate large amount of data that is acquired automatically and it has plenty of useful inbuilt functions. The fact that it is highly user-friendly as well made it a good choice for the visualization. Python comes with a library called matplotlib which mimics Matlab in many ways and can be used for graphical representation of data. However, it was not as user-friendly as required. Hence, we decided to go with Matlab.

4.4 Design of the database and tables

For storing the PanStamp data, a database is created. We need two tables to efficiently manage the information. One master table that would store all the PanStamp related details such as its address, its title and how many capabilities it possesses and another table which would hold all the information related to individual capability. The first table is called ‘PanStamp_master_table’ and the second one is called ‘PanStamp_property_table’. One entry in the PanStamp_master_table can have ‘n’ corresponding entries in the PanStamp_property_table.

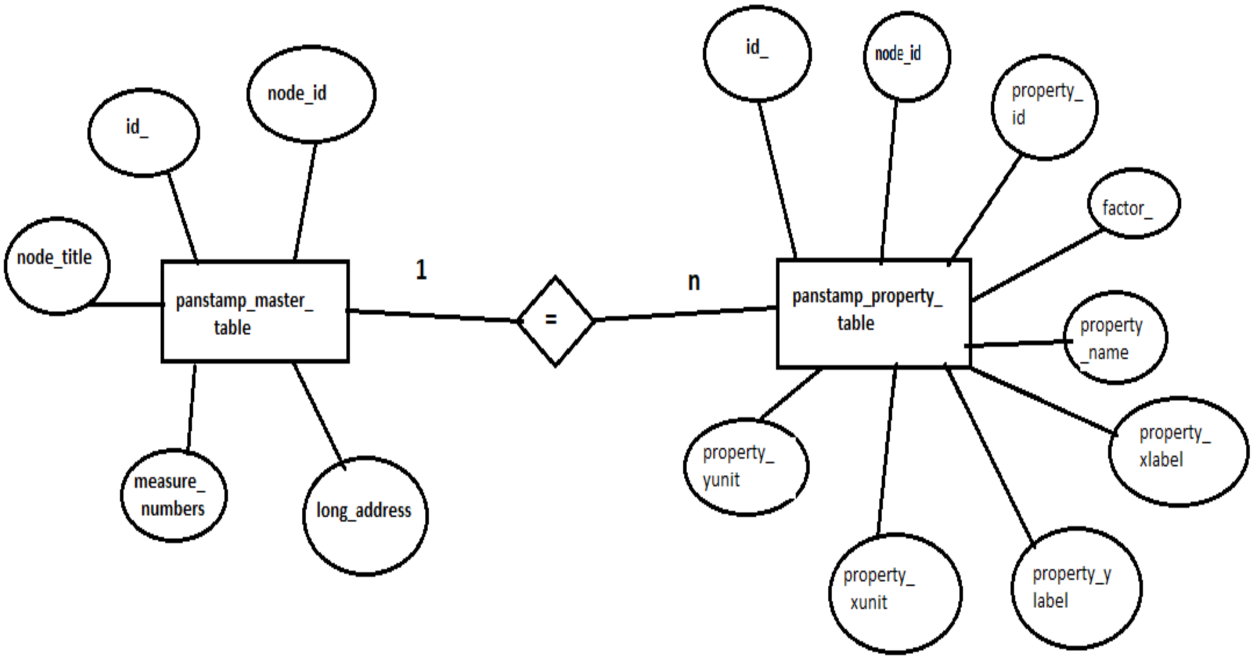
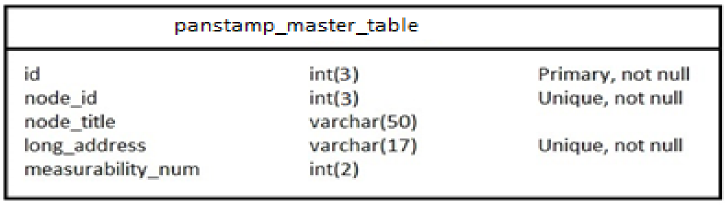


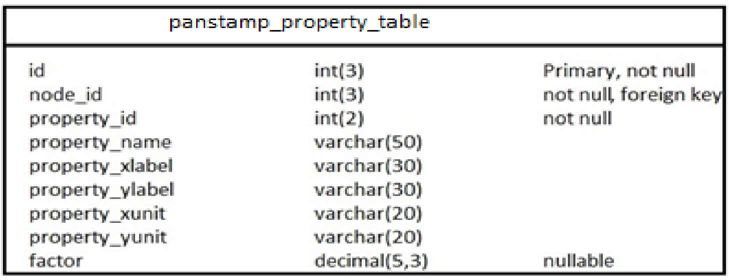
Figure 4.3: Database entity relationship diagram

In the above figure 4.2, one can see the relationship between the two tables along with the column names of each table.

Table.1 Database Scheme



panstamp_master_table		
id	int(3)	Primary, not null
node_id	int(3)	Unique, not null
node_title	varchar(50)	
long_address	varchar(17)	Unique, not null
measurability_num	int(2)	



panstamp_property_table		
id	int(3)	Primary, not null
node_id	int(3)	not null, foreign key
property_id	int(2)	not null
property_name	varchar(50)	
property_xlabel	varchar(30)	
property_ylabel	varchar(30)	
property_xunit	varchar(20)	
property_yunit	varchar(20)	
factor	decimal(5,3)	nullable

Above table.1 represents the database scheme for the project. The database is called PanStamp_database and it has 2 tables' PanStamp_master_table and PanStamp_property_table. The diagram clearly shows the number of columns in each table, the names of the columns and the size and type of the columns. If a column has any special conditions on it, those are shown in this diagram as well.

It is to be observed that the node_id column of the PanStamp_property_table is a foreign key that references the node_id column of the PanStamp_master_table. This feature will ensure that no PanStamp data will be added to the PanStamp_property_table without having an entry in the PanStamp_master_table.

Chapter No. 5

5 Implementation of the system

The implementation of the system consists of the following parts:

Hardware setup

Configuration of PanStamp modules

Sending and receiving data

Visualization

5.1 Hardware Setup

The hardware setup of this project was to set up a wireless gateway node using two PanStamp modules and doing serial communication with RassberryPI. Two PanStamp modules are used for this purpose. One PanStamp as Endpoint (Tx) and one PanStamp as a Coordinator (Rx). The PanStamp modules were used to transfer data between two points wirelessly. The RassberryPie microcontroller were used to process the data and help in transferring it to the server. The project further can be broken down into the following sections:

- Design of protocol to transfer data between the PanStamp modules
- Implementation of the protocol by programming a class to use with the Arduino
- Development of python application to work as a gateway between the PanStamp and RasberryPI.
- Visualization of the transmitted data in Matlab
- Storage of PanStamp related information in to a database

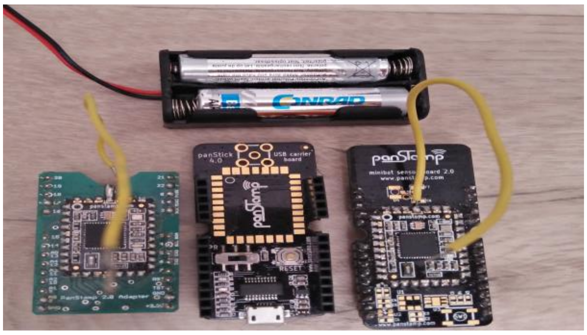


Figure 5.1: Hardware components

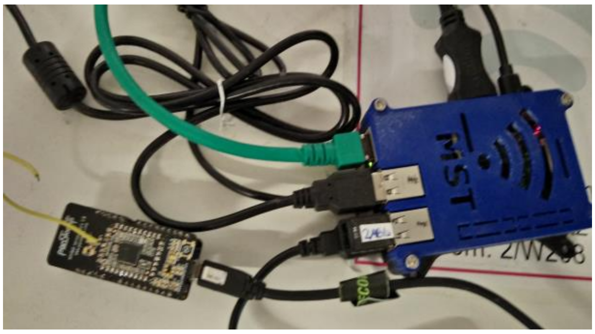


Figure 5.2: Hardware setup

5.2 Configuration of PanStamp Modules

For configuration of two PanStamp modules is need to use Arduino IDE software environment. Arduino IDE provides different sketch packages for PanStamp wireless communication. In this project were used modem.ino sketch as a coordinator configuration and basicRadio.ino sketch as an endpoint configuration. After configuration of two PanStamp modules their results were seen on Arduino IDE serial monitor screen and after that on raspberry Pi command line with the help of python gateway firmware.

5.3 Sending and Receiving Data

The PanStamp endpoint sends the data which passes through the coordinator PanStamp and is analyzed by the python gateway application before being routed to the database or to Matlab. So, this part of the system can be broken down into four sub-parts:

1. Transmitter sketch
2. Receiver sketch
3. MySQL database
4. Python gateway application

5.3.1 Transmitter Sketch

Using the Arduino IDE, the Arduino sketch, basicRadio.ino is flashed on to the PanStamp endpoint (Tx). The version used for this project is Arduino IDE 1.6.4. This software can be downloaded for free from the official website for Arduino.. The sketch has minimum number of code lines required to communicate. This ensures that the system is easily configurable. A user who needs to add a new PanStamp to the network needs to just pass the parameters via the sketch. The rest is handled by the Arduino library class.

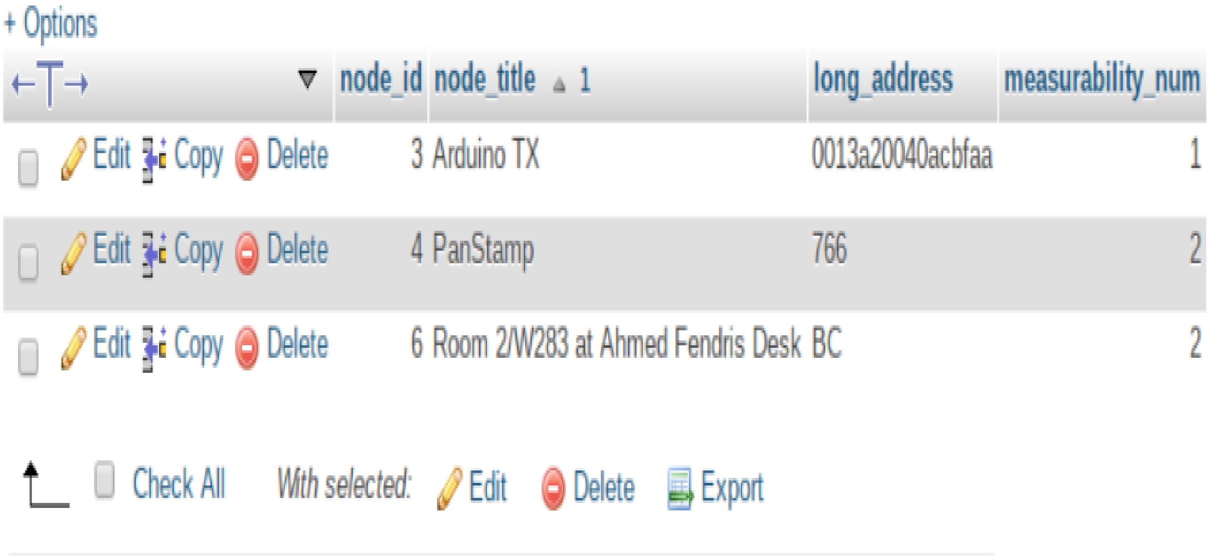
5.3.2 Receiver Sketch

Using the Arduino IDE, the Arduino sketch, modem.ino is flashed on to the PanStamp coordinator (Rx). The sketch also has minimum number of code lines required to communicate. This ensures that the system is easily configurable. A user who needs to add a new PanStamp to the network needs to just pass the parameters via the sketch.

5.3.3 MySQL Database

To store RSSI and LQI values into database. There are three existing tables

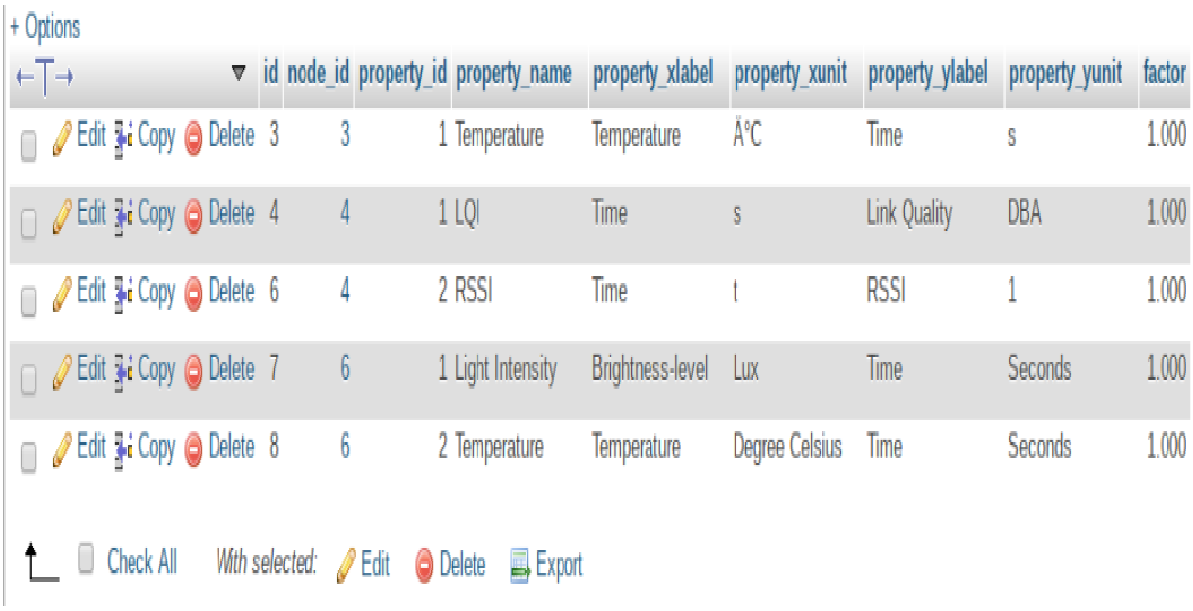
- Node_table for store node_Id for PanStamp



node_id	node_title ▲ 1	long_address	measurability_num
3	Arduino TX	0013a20040acbfaa	1
4	PanStamp	766	2
6	Room 2/W283 at Ahmed Fendris Desk	BC	2

Figure 5.3: Node_table in database

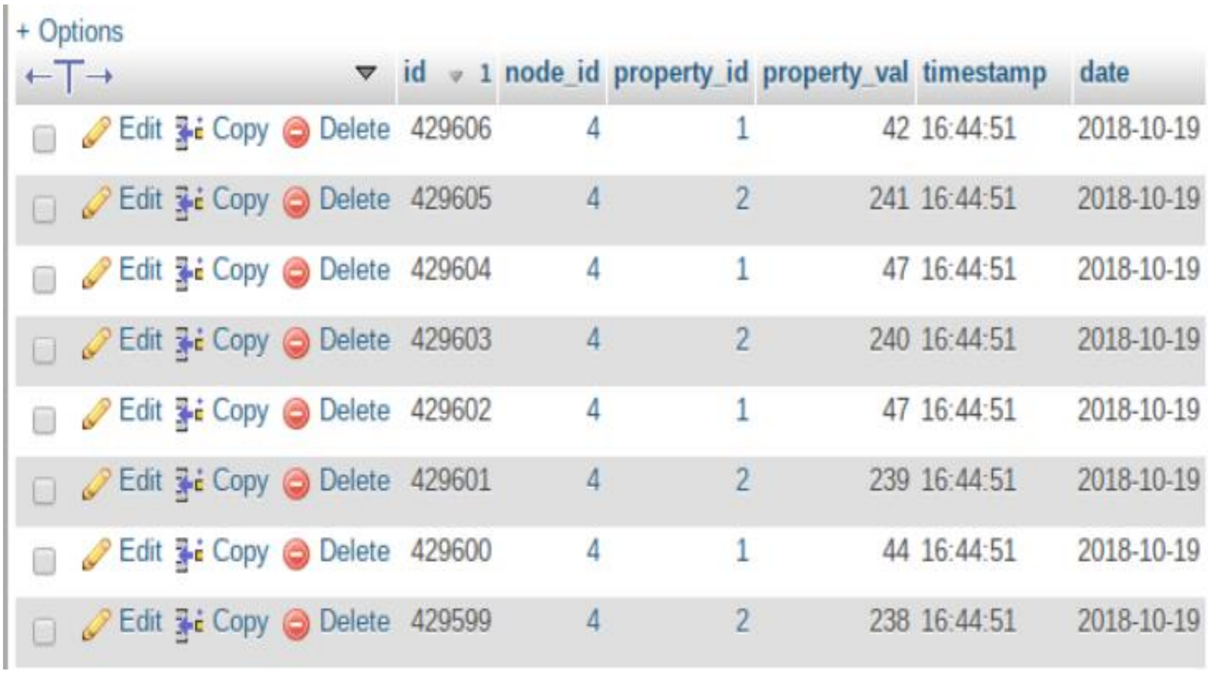
- Property_table to store RSSI and LQI units, label etc



←T→	id	node_id	property_id	property_name	property_xlabel	property_xunit	property_ylabel	property_yunit	factor
□ Edit Copy Delete	3	3	1	Temperature	Temperature	Â°C	Time	s	1.000
□ Edit Copy Delete	4	4	1	LQI	Time	s	Link Quality	DBA	1.000
□ Edit Copy Delete	6	4	2	RSSI	Time	t	RSSI	1	1.000
□ Edit Copy Delete	7	6	1	Light Intensity	Brightness-level	Lux	Time	Seconds	1.000
□ Edit Copy Delete	8	6	2	Temperature	Temperature	Degree Celsius	Time	Seconds	1.000

Figure 5.4: Property_table in database

- To store RSSI and LQI upcoming values into Value_table in database



←T→	id 1	node_id	property_id	property_val	timestamp	date
□ Edit Copy Delete	429606	4	1	42	16:44:51	2018-10-19
□ Edit Copy Delete	429605	4	2	241	16:44:51	2018-10-19
□ Edit Copy Delete	429604	4	1	47	16:44:51	2018-10-19
□ Edit Copy Delete	429603	4	2	240	16:44:51	2018-10-19
□ Edit Copy Delete	429602	4	1	47	16:44:51	2018-10-19
□ Edit Copy Delete	429601	4	2	239	16:44:51	2018-10-19
□ Edit Copy Delete	429600	4	1	44	16:44:51	2018-10-19
□ Edit Copy Delete	429599	4	2	238	16:44:51	2018-10-19

Figure 5.5: Value_table in database

5.3.4 Python Gateway Application

The gateway application, receives the data from the PanStamp module and responds appropriately. A number of python libraries have been used for this purpose.
For installation, in windows command line, run ‘*python -m pip install SomePackage’* for each python library where ‘*SomePackage’* must be replaced by the name of the python library. In the case of the libraries required for this project, it can be replaced by pyserial and PanStamp.py.

For upgrade, run ‘*python -m pip install --upgrade SomePackage’* for each python library where ‘*SomePackage’* must be replaced by the name of the python library.

After all the libraries are successfully installed, PanStamp.py file imported into a new project using visual studio code IDE. The version used for this project visual studio code IDE Community edition 3.7. The community version of the IDE is available for free on the internet.

5.4 Visualization

When a registered PanStamp transmits a status value, the gateway application transmits the status value and the plot labels via the TCP server-client socket connection to Matlab. The status values are plotted live along the xy-axis and the plot is labeled appropriately using the plot labels passed to Matlab once via the gateway application. A single script has been written to implement the mentioned features. The script is called PlotDataLive.m and it should be run in Matlab.

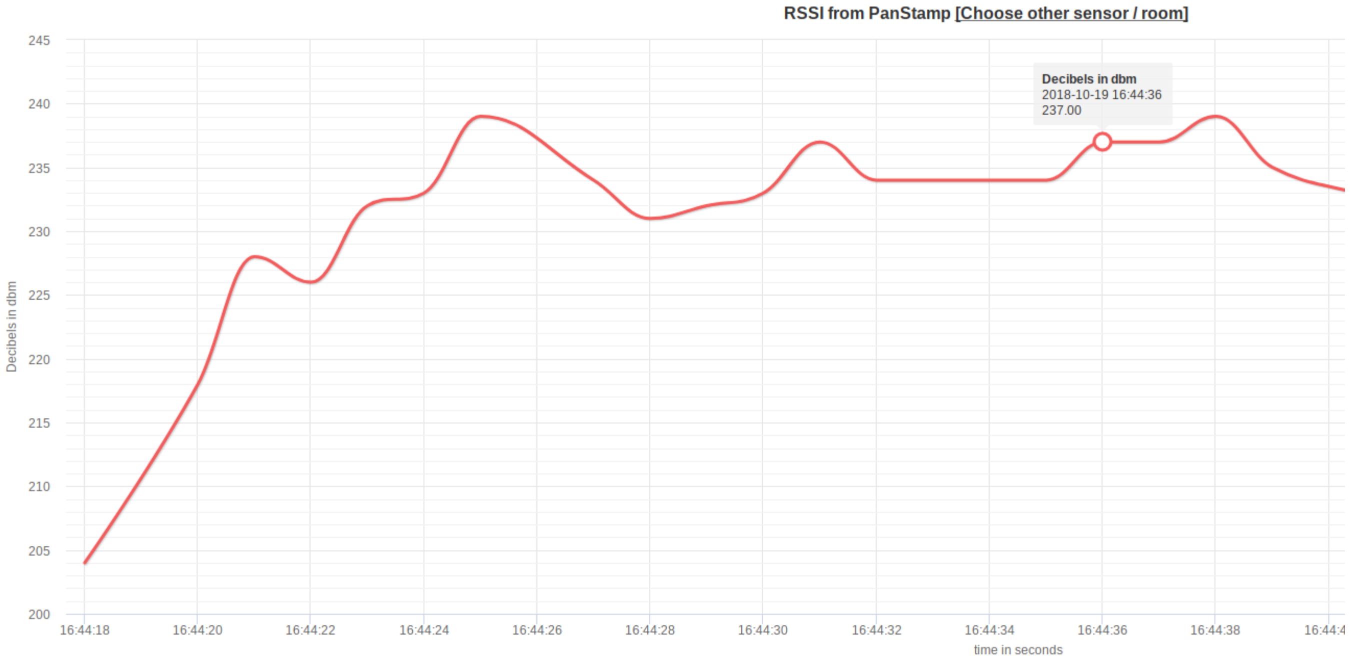


Figure 5.6: Graphical representation of RSSI

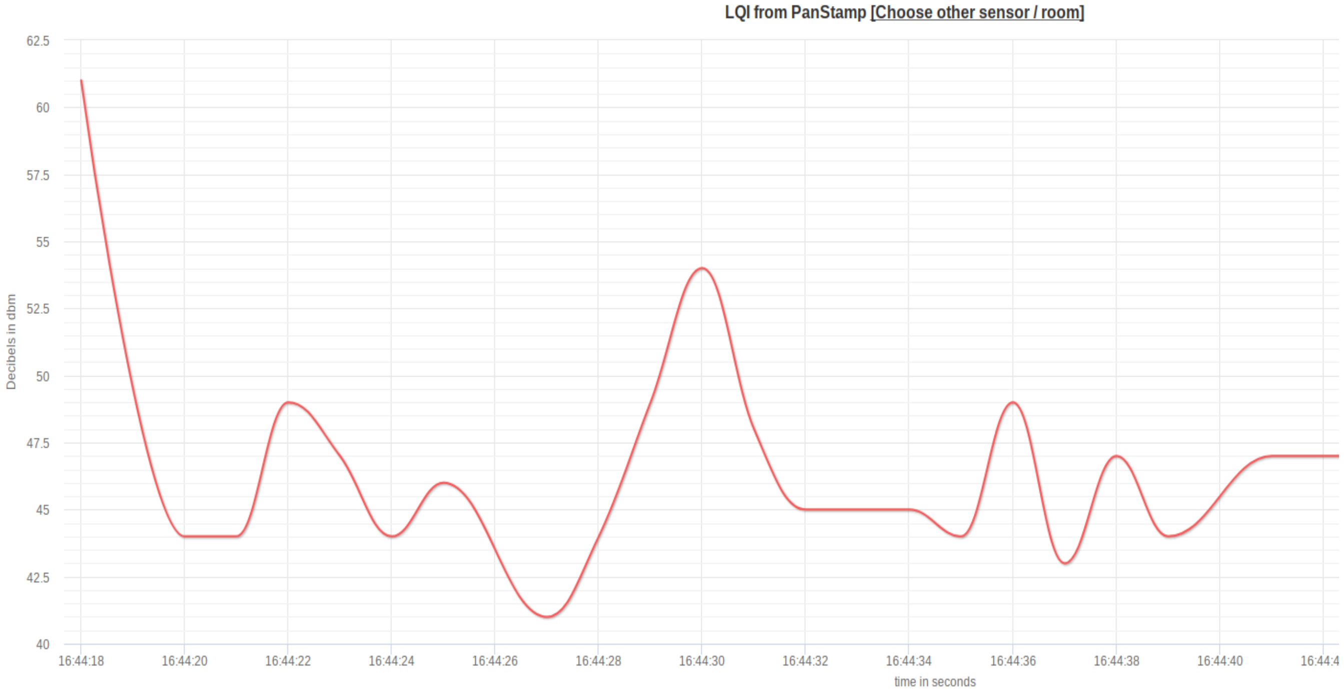


Figure 5.7: Graphical representation of LQI

Chapter No. 6

6 Summary

6.1 Conclusion

For a programmer, it is always essential to have a reliable documentation**.** The presence of documentation helps keep track of all aspects of an application and it improves on the quality of a software product. The main focuses are development, maintenance and knowledge transfer to other developers. Successful documentation will make information easily accessible, provide a limited number of user entry points, help new users understand quickly, simplify the product and reduce costs.

One of the objective in this topic is documenting the existing MST-Gateway structure and its firmware, by using “DokuWiki”. The Dokuwiki doesn’t support source code documentation that is why Doxygen is used to comment the whole firmware of the project. Doxygen runs on most of the operating systems i.e. Linux, Mac, and Windows etc. and also supports many programming languages. It also has capability to generate more advanced diagrams and graphs which can be achieved through “dot" tool from Graphviz.

The other objective is the software development for PanStamp module, which improve wireless sensor network. The wireless communication between the endpoint and coordinator was implemented and retrieved the RSSI and LQI values to display sensor signal strength. For run time plot for these value, a gateway application is also designed using python to make serial interface between PanStamp module and RaspberryPI and store live data to the database. All these objectives have been achieved successfully.

6.2 Future Enhancement

The protocol has been implemented for a system with 2 PanStamp modules and can be further extended to multiple PanStamp modules. Also, the gateway application can be extended by adding a part that allows the PanStamp to be controlled via a web page on a secure network. Another extension can be made to the PanStamp protocol to include a maintenance phase where the PanStamp modules can be regularly checked on by the gateway application. Finally, the gateway application can be changed so that the data can be visualized on multiple computer.

7 References

[1] "GitHub," [Online]. Available: https://github.com/panStamp/panstamp/wiki/First-steps. [Accessed 01 08 2018].

[2] "msp430 controller," [Online]. Available: https://www.mikrocontroller.net/articles/MSP430. [Accessed 01 08 2018].

[3] "doxygen," [Online]. Available: http://www.stack.nl/~dimitri/doxygen/download.html. [Accessed 01 09 2018].

[4] "dokuwiki," [Online]. Available: https://www.dokuwiki.org/dokuwiki. [Accessed 10 09 2018].

[5] "rasberryi Pi," [Online]. Available: https://de.wikipedia.org/wiki/Raspberry_Pi. [Accessed 20 07 2018].

[6] "arduino," [Online]. Available: https://www.arduino.cc/. [Accessed 23 07 2018].

[7] "installing sketches," [Online]. Available: https://github.com/panStamp/panstamp/wiki/Installing-panStamp-cores-and-libraries-for-Arduino. [Accessed 20 09 2018].

[8] "python," [Online]. Available: https://de.wikipedia.org/wiki/Python_(Programmiersprache).

[9] "visual studio code," [Online]. Available: https://de.wikipedia.org/wiki/Visual_Studio_Code.

[10] "mst-gateway," [Online]. Available: https://www-user.tu-chemnitz.de/~gomart/wiki/doku.php?id=technik:mst-gateway:development_tutorial:start#dokuwiki.

[11] D. E. Houssaini, Z. Mohamed, S. Khriji, K. Besbes and O. Kanoun, "A Filtered RSSI Model Based on Hardware Characteristic for Localization Algorithm in Wireless Sensor Networks," 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 118 - 123, 2018.

8 Appendix

8.1 Description of files

Implementation

8.1.1 Arduino sketch

1. Endpoint.h
2. Coordinator.h

8.1.2 Python application

1. gateway_PanStamp.py

Documentation

8.1.3 Endpoint Firmware

Arduino

- Endpoint
- Coordinator

Msp430

- Endpoint
- coordinator

PanStamp

- Endpoint
- Coordinator

8.1.4 Gateway Firmware

- Arduino.py
- MSP430.py
- PanStamp.py

8.1.5 Report

- Master_Research_Project_Muhammad Aashir_Ayaz.pdf