Energy Management in USB-C Based Modular Solar Home Systems

Implementation of USB-C Power Delivery and Data Acquisition Firmware on Multi-port Configuration

by
Faik Nizam

Student Number: 4757270

Thesis committee members:

Prof Dr. J. A. Ferreira (chairman)
Dr. Laura Ramirez Elizondo
Dr. Milos Cvetkovic
Dr. Laurens Mackay (supervisor)

Delft, January 2020

Abstract

Despite of the technological advancements on renewable energies and integration of those into the grid, there are still around 800 million people who lack access to the electricity. In order to tackle this problem, solar home systems are being deployed especially in rural areas to create units for decentralized energy generation, based on DC microgrid premises. Integration of solar home systems, especially in developing countries, has great positive effects on communities' economic and social conditions.

In line with transition to cleaner energy resources and utilization of these, this thesis has been carried at *DC Opportunities RD* to facilitate the rural electrification project. Within this project, the thesis focuses on implementation of a general adaptable firmware based on USB Type-C power delivery protocol on several components of modular solar home system.

Firstly, the report analyses the existing solar home systems with their characteristics on power delivery and energy management. The limitations on the existing projects are studied and the main motivation on deploying the firmware is pointed out. The main objective of the thesis is hence is formulated as achieving and implementing the USB Type-C power delivery protocol with data acquisition layer on prototypes of charging station, power banks and charging hub, that is tailored to rural applications. The firmware design topology is followed to take into account the modularity, scalability and applicable to energy management.

The USB Type-C power delivery firmware is developed by studying and adapting two open source packages. Several adaptation layers are developed and the firmware is implemented using micro controllers, from STM32, that are capable of running operating system with no faults, deploying several layers of tasks and state machines. Power delivery library functions are tackled with layers of software architectures and data monitoring and transmission layer is deployed, in order to validate the system as well as the hardware.

Implemented firmware is developed on the prototypes and tests are run with the current editions of the prototype. The energy management layer and formulated test case will be run and tested on March 2020, in the field test in Ethiopia. This field test will provide results and data, thanks to the developed firmware and acquisition system, in order to achieve a step closer to the ultimate goal of the project, which is implementing low voltage DC microgrid in rural areas.

Preface

This master thesis and project is the concluding phase of my studies in Msc Sustainable Energy Technology in TU Delft. This programme had been a great step for a person like me to explore the opportunities in the green energy area, to gain broad knowledge on Solar systems, power electronics and economics and integration side of the renewable systems. Likewise, the thesis topic had been another great challenge yet a new door that introduced me vast knowledge on embedded software design and power electronics. I would like to firstly thank Prof. dr. J. A. Ferreira for giving his persistent support on the project.

Besides studies, I had been part of a great D:DreamTeam, MOR Team. This experience was also another step of pushing my limits but also giving me opportunities to be involved in industry, have practical experience in electrical and PV system design and development.

Apart from knowledge, during this past 2.5 years, I have gained many friendships, a great supervisor, Dr. Laurens Mackay and supportive colleagues. I would like to thank to all. Of course this would not be possible with my friends and especially my family. I appreciate each and everyday with them and count my blessings on all the achievements. Yet, many more to come!

Contents

List of Figures

List of Tables

1
Introduction

1.1 Problem Definition

1.1.1 Rural Electrification in Developing Countries

In the last century, electricity has become an integral part of the daily life. Besides the improved education and public health, energy access plays a vital role on the communities and the nation's economic growth and social equity. Hence electricity needs to be utilized resourcefully. In the search for that, renewable energy sources have become prominent focus for electricity generation. This has led to the technological improvements of sources, solar and wind being the most prominent.

Despite the technological advancements on renewable energy resources and the decreasing costs of energy generation, there is still a considerable amount of people who lack access to energy. According to the World Bank, there are still roughly 860 million people, that constitute to 11% of global population that has lack of access. Furthermore, it is another challenge to bring to such envelope of population clean and reliable energy [1]. This challenge has become more evident due to the increasing global warming and recent energy shortages resulted. In developing countries, the rural areas where there is substantial amount of population lives has hurdles of access to the grid energy. This challenge is quite prominent in the Sub-Saharan and East African countries, where 600 million people still live in the dark. Figure 1.1 shows the amount of people still without access to electricity in the past decade and the projected years in the future, where the data is retrieved from International Energy Agency Outlook [2].

According to the IEA, Ethiopia hosts the greatest number of population in East Africa without access to electricity, which is around 60 million [2]. According to the Ethiopian government, 60% of the population living in rural areas does not have access to electricity [3]. Therefore, in order to take measures and tackle this problem, with the aids of international organizations, the National Electrification Program is constituted. It aims to achieve nationwide access to electricity by 2025. The purpose of the program is to deploy solar/hydro power which is regarded as the realistic solution, especially for the rural areas, where the grid is not accessible. This increases the demand for the micro gird or the nano grid applications, which in turn also affects the stability of the grid.

1.1.2 Modular Solar Home Systems

With the emerging renewable sources and its technologies, new business models and energy generation implementation models arose. Countries have given importance to the subsidies and

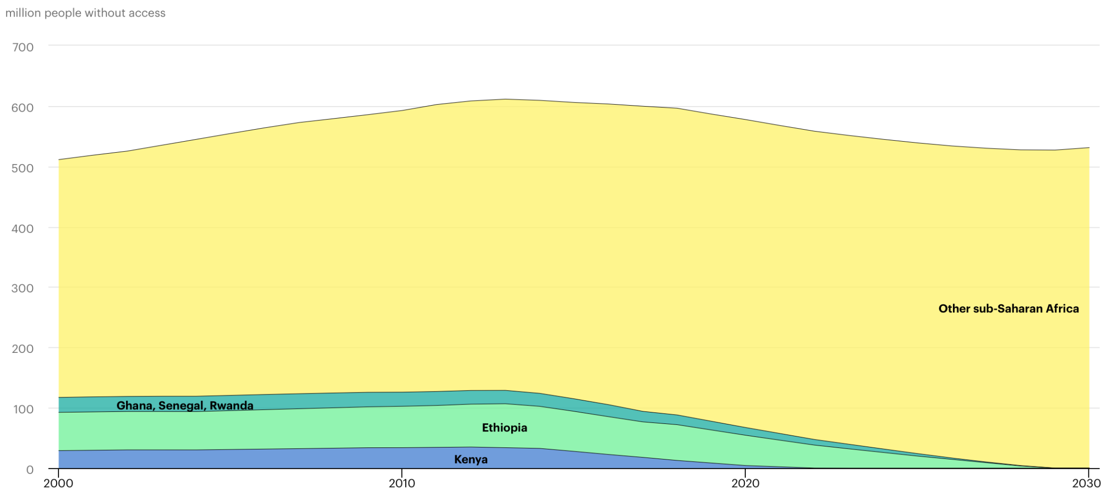


Figure 1.1: Number of people without access to electricity in Sub-Saharan African countries. Ethiopia is one of the Sub-Saharan African countries with the lowest electrification rate. Although the non-electrification rate is expected to decrease in the next decade, it is still a considerable amount [2]

policy changes to increase the generation and accessibility to clean energy resources and thus to the decentralized energy distribution. Decentralized generation, especially for the rural electrification, can be the long term solution, tackling the problem of infrastructure and reliability in remote areas.

Solar home systems are stand-alone systems that are emerged as off-grid solutions for households. They incorporate photovoltaics as the main energy source and optionally battery banks as the main storage as well as DC or AC Loads. Figure 2.5 illustrates an example Solar Home System from a company "BBoxx" operating in East African countries, with a PV module, battery, electronics and DC Loads. In rural areas, the solar home systems are designed to provide energy for the loads of a household, where the grid is inaccessible. Hence the solar home systems can be seen as one of the integrating units of the decentralized energy generation. With the advancing technology, many smart components can be part of the solar home systems that bring an innovative approach to the energy management for high efficiency and reliability.

Figure 1.2: Example SHS from BBoxx LTD. operating in East African countries [4]

In the past decade, usage of solar home systems has become a complementary way of extending the grid for targeted rural areas. Off-grid solar based systems have benefited over 100 million people to supply electrification, where around 24 million people have used solar home systems [5]. Besides access to electricity, solar home systems have proved to improve the living conditions of the users in terms of improved economic and social conditions; creating job opportunities, enhanced business revenue, increased monthly income and reasonable working hours, among other health related benefits [6].

In a stand-alone system, efficiency of the system is quite crucial for the robustness and reliability. Since the current generation and storage options in solar home systems run on Direct Current (DC), using electrical devices and loads operating on DC is one of the main focus on improving the efficiency, by saving on conversion losses. There are many electronic devices such as smartphones, battery banks, Laptops and TVs that run on DC. Indeed, there is still a challenge especially in the area of solar home systems, to integrate such devices into system with the minimum possible power loss. Furthermore, having smart measures on the system is another challenge. Nevertheless, with the emerging technology of USB Type-C interface, DC loads can directly be connected to the system without conversion into alternative current (AC). Moreover, the USB Type-C interface allows bidirectional power flow and data communication, which is a great means of achieving energy management in such systems.

1.1.3 SHS Limitations and Energy Management Challenges

Even though solar home systems are perceived as the emerging off-grid solution for electrification problem, they bear limitations and challenges, which slow down the process of achieving fully accessible electrification. The main limitations are listed down followed by explanations.

- **Cost**. Even though the sun's energy is freely accessible, utilizing this energy via photovoltaics and bringing it into useful electricity has significant upfront costs. The current systems available in East African market have initial costs of ranging from 4-15 $\$/W_p$ [7]. The initial costs of the system can be equal to the yearly income of a low-income rural family in East Africa [8].
- **Power Capacity**. In comparison to the grid utilities, solar home systems are limited in terms of their power availability. Even though there is significant amount DC appliances and home electronics that can be powered by SHS, the high power appliances such as water pumps can not be powered by the existing SHS.
- **Battery Sizing**. The battery is the crucial element of the SHS where the excess energy is stored and used when the solar energy is not available. However, the battery is the element with the highest initial costs and affects the system lifetime. Hence, it is crucial to make the system sizing and battery sizing accurate in order to decrease the effect of this challenge.
- **Load Demand**. In line with the battery sizing and power availability, load demand is a crucial input for the system sizing, as well as capturing the electricity demand of a household. However, for rural electrification, where the households have very limited electrical footprint, there is a lack of load data. This makes the system design and management challenging for rural applications. It is required to take into account the load consumption and the user preferences with the available loads, preferably DC appliances.

The aforementioned limitations incur challenges for the development of the solar home systems. Furthermore, these limitations induce barriers on the energy management of such systems. The aim of the energy management on SHS is to monitor, control and optimize the generation, power flow and consumption of energy. With the embedded control systems, such ambitions are possible to achieve. However, this increases the cost of the system as well as it is limited by the load demand and power capacity of the solar home system.

1.2 Motivation

In a solar home system, with PV generation and battery storage, it is generally aimed to maximise the utilization of energy generation by PV through the system to the loads, while keeping the efficiency of the system as high as possible. Minimization the usage of batteries improves the system efficiency, which in turn affects the reliability of that system. The demand side management can be used to mitigate the use of batteries and maximise the photovoltaics utilization with load scheduling [9]. Furthermore, as per the bottom-up approach, energy management systems have been utilized in residential buildings to change the use of the load quantitatively by monitoring and communicating with the so-called "smart loads/appliances". In the utility or grid level, the demand side management practices are used to yield aimed changes in load profile of consumers by providing incentives [10]. Hence, in a SHS, monitoring, communication and control of the loads and sources is critical for the the improvement of the system's reliability and efficiency.

1.3 Research Objective

In line with the aforementioned problem statement, this thesis aims to put forward a reliable power delivery protocol and data monitoring system for a modular charging station, that is adaptive to the load cases with LED and power banks connected on four ports. Hence, the main objective of the thesis is ***to achieve and to implement the USB Type-C power delivery protocol and data acquisition on the prototypes of charging station, power banks and the main hub tailored to rural applications, that can be extended with Energy management promises***.

In order to achieve the project objective and to have a throughout investigation on the process of application from design to implementation, research questions are formulated. The driving questions that the project aims to answer are as follows:

(1) *How can power delivery over USB-C be implemented on the solar charging station for multi-port configuration?*

The goal is to investigate, to design and to implement the USB Type-C power delivery protocol for the rural electrification application. Considering the fact that current emerging technologies only allow maximum of two ports of type-C charging/discharging, the investigated protocol and the base projects will be adjusted for multi-port (more than two ports) configuration. It is important for the implemented system to be adaptable and robust. Thus two main power delivery embedded software packages are investigated, in order to understand the architecture, protocols, applied layers and important steps of implementing the layers within the firmware. Accordingly, the project evolved on implementing the firmware based on one of the packages that gives possibility of using up to eight ports for power delivery along with data.

(2) *What are the structures of data that are important to monitor, to be communicated over the ports for direct battery bank charging and energy management mechanism?*

Besides the power negotiation, the power delivery firmware acts as a side-band channel enabling communication between the connected ports. The communication can consist of discovering the capabilities of both sides, discovery and entering of modes supported by ports, and interchange of vendor defined data. The focus on this question is to implement the data communication among the ports beyond the power negotiation, capture and categorize the important parameters to be communicated and to form and to store a structural architecture of data, that can be buffered and monitored. By answering this question, charging/discharging sequences can be implemented, decision making mechanisms will be dealt and user friendly data monitoring interface can be developed.

(3) *What are the user preferences tailored to rural use in Ethiopia and how to implement the demand response on the system?*

This question deals with the demand response and conclusion of the thesis. The implemented system and data monitoring mechanism is aimed to be tested in rural village in Ethiopia. The implementation and testing will give feedback, with the acquired data of the user preferences. The answers to this question will be used in the future for improving the system in terms of robustness, reliability and applicability to tailored case.

1.4 Thesis outline

The thesis follows a step-wise structure, mainly based on the order of the research objectives realised. Chapter 1, identifies the problem within the context of rural electrification, investigates the challenges of technologies in the field and proposes the main objective of the project as well as the research questions. Besides the introductory chapter, the thesis has the following structure:

- Chapter 2 involves the literature study carried out about the current energy management implementation on modular systems, relevant algorithms used in the context of demand response, parameters and characteristics of the USB Type-C power delivery protocol. The analysis on the following topics will establish a throughout overview on the ideal way of implementation of proposed system as well as the design limitations and parameters.

- Chapter 3 presents a physical overview of the applied prototypes or devices within the system. The devices are elaborated in terms of the important parameters, electrical circuits and characteristics that work in line with the power delivery software. In later stages, these devices will be referred and be used as part of the test case.

- Chapter 4 presents the two embedded software packages tailored for power delivery applications. The architectural topology is investigated, the adjusted software layers are presented within this chapter. Furthermore, this chapter gives more insight into the challenges, design limitations and how such challenges been tackled.

- Chapter 5 assesses the implemented modular system, focusing on the power delivery firmware. This chapter includes the state flow diagram of important decision mechanisms and messaging sequences and that of the communication implementation. Furthermore, the data objects that are designed to be communicated within the negotiation are presented within this chapter.

- Chapter 6 elaborates the experimental results of the implemented system. The results are based on the laboratory tests. This chapter firstly presents a framework for the collection of data, presents the key outcomes with measurements and observations. Secondly, the implementation of test case and energy management is presented. Finally the experiment set up is explained and results are discussed.

- In Chapter 7 the conclusions are drawn based on the results acquired from the implemented system. The research questions are evaluated and the further discussions give room for the future work improvement.

2
Literature Review

This chapter evaluates the necessary background information to realize the project. Firstly, a thorough development research on the current solar home systems, with a focus on the energy management aspect is performed. Secondly, the steps taken into a test case are assessed, with load profile determination, the operational constraints and system limitations which will evolve into a load control state machine. Lastly, USB Type-C protocol characteristics are evaluated.

2.1 Solar Home Systems

As mentioned in Section 1.1.2, solar home systems are PV based stand-alone systems, where the generation power is solely based on solar panel and backed up by a storage system, such as Lithium-ion power banks. Figure 2.4 illustrates block diagram of a SHS with the main components.

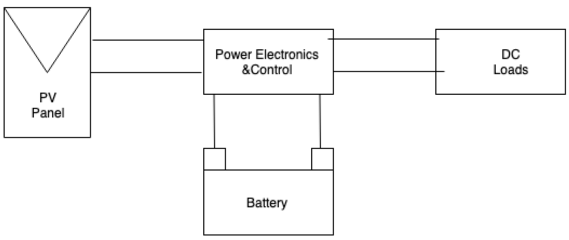


Figure 2.1: Block diagram representing Solar Home System.

In 2017, Ethiopian government introduced standards for all the operative Solar Home System in the country along with United Nations Development Programme (UNDP). According to the standards, the system should be plug-and-play, in other words, no extra technical tools needed for operation of the system as well as no technical experience for the installation of the system. Furthermore, the SHS should come with "pay-as-you-go" (PAYG) business model, where the system is rented to customer on a monthly or similar basis [11]. Such system is required to come with a battery, a charge controller, a solar panel, LED bulbs and a mobile charger and should have basic power to charge such load. The output and loads covered are required to be functioning only in DC. Moreover, the Solar home system's bus voltage should not exceed 35 V according to the same standards [11].

2.1.1 Existing Systems in Rural Electrification

Since the introduction and adaptation of Sustainable Development Goals (SDGs), there have been many incentives and subsidies that lead to practices worldwide for sustainable development. As mentioned before in Section 1.1.2, the solar home systems play a pillar role in rural electrification and bringing clean energy in undeveloped countries. Consequently, there are many existing projects, mainly taking place in Africa, deploying PV based stand-alone systems.

A solar lantern is the most basic and foremost merging technology in rural electrification, bringing an alternative solution to existing kerosene lamps. A solar lantern can be categorised as pico-solar product, where its generation is limited to around 11 Wp and to a load of LED or a mobile charging unit. Such system does not have complicated energy management system, since the generated energy is directly stored and used for LED lighting. An example market-available product is a "d.light" S30 model, that has 0.3 W PV panel, 60 lumens bright LED and 12 hours maximum run-time [12].

Figure 2.2: d.light S30 Solar Lantern, the most basic necessity and electronics in the rural electrification [12]

Another example system from the same developer is "d.light x850" DC solar home system that comes with 40 Wp Panel, 5 LED lights of maximum brightness of 1280 lumens, a battery pack and other electronics. Plug-and-play characteristics are deployed on the system and business model of pay-as-you-go is also one of the options. Such system has a charge controller for the PV panel and battery charging and does not include further complex energy management system.

Figure 2.3: d.light X850 Solar home system including a PV panel, LED lights, battery pack and set of electronics coupled [13]

A competitor is "Azuri" PAYG solar home system, that also has been operating in African countries. The system deploys a 50 W solar panel, DC loads such as television, mobile charging, 4 LEDs and a radio. The system has a main controller and a management system, which monitors the customer's consumption pattern and adapts its energy output accordingly, via machine learning. Furthermore, the energy management system has input of weather data and can adapt to climatic conditions for reliable operation.

Figure 2.4: Azuri Technologies, solar home system, coupled system that comes with set of DC loads supported by PV panel and main controller. The system has a data management system for monitoring and managing the power delivery. [14]

Another existing system for rural electrification is the aforementioned "BBOX bPower 50", which is a solar home system offered in 3-year payment plan. The system comes with a 50 Wp solar panel and a power controller that integrates the connection to PV panel, 6 DC output connections, 2 USB connections and a data monitoring and communication system. Such monitoring system is able to offer information on voltage, current and temperature data which helps maintenance and timely support. Furthermore, it is able to remotely connect via 2G mobile network. The system is illustrated in Figure 2.5. The system's data acquisition and connectivity to DC appliances increase the system's efficiency, reliability as well as enhanced costumer satisfaction.

Figure 2.5: Example SHS from BBoxx LTD. operating in East African countries. A complete DC system that comes with a payment plan and implemented with energy management system [4]

2.1.2 Energy Management Systems Under Research

Besides the existing systems and projects available, there are systems that are under research and development. These aim to accomplish control and energy management, more specifically in a built environment.

In [9], Chauhan et. al. configured a prototype of an autonomous DC microgrid in a residential building with a proposed demand-side control algorithm. The system is composed of an array set of PV, charge controllers, batteries as energy storage, set of DC loads connected to the automatic switches. There are current sensors on the generation and consumption side connected and monitored by micro controller at a minute-interval. The control objective of the demand side management algorithm on the micro-controller is to maximise the direct use of PV and to maintain the state of charge level(Soc) of battery banks for the robustness and redundancy. The loads are classified as deferrable or controllable. The deferrable loads are scheduled within this control schedule in order to reduce the cycle of charging/discharging of batteries, by shifting the utilization into sunny hours.

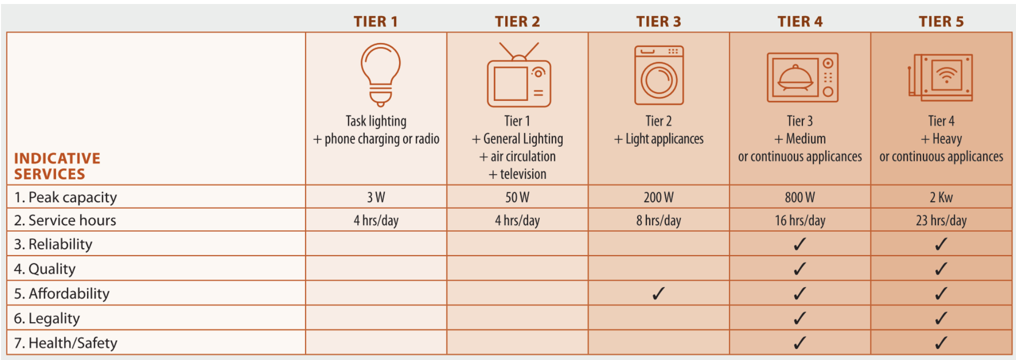
In [15], the author investigates different types of energy management systems implemented in the laboratory scale for testing. One approach taken is the device-level energy management system, in which the household devices are remotely monitored and controlled. The system uses ZigBee protocol in a mesh-network, where each node consists of a smart plug attached to the loads. Each set of data is transmitted via serial communication and processed with Python scripts as part of the interface. A priority control technique is implemented on the device nodes, which made the dynamical control feasible within the test.

ZigBee protocol is one of the network architectures that are adapted and used in the energy management field within built environment. Besides ZigBee (250 kbps) protocol, Ethernet (10 Mbps) and WiFi (54 Mbps) are the wireless solutions. The significant factors in such architecture of networks are the cost-effectiveness, reliability of data access and competency of network technology [16]. ZigBee wireless protocol is proved to be a simpler and less expensive technology in relation to WiFi. On the other hand, ZigBee protocol has lower data rate, low processing capabilities and high chance of interference in the same frequency band in comparison to WiFi [17].

2.1.3 Load Classification

Even though the solar home systems have emerged into market and rural microgrids have faced a rapid growth, there is still lack of load profile. Within this section, a framework is assessed and several loads are evaluated to create a load profile which will be the test case for the project.

Recently, policy makers have developed a framework to assess the electrification according to seven indicative services. The so-called approach is Multi-Tier Framework, a survey-based approach and classifies the electrification in five tiers [18]. The multi-tier framework indicative services and five categories are represented in Table 2.6. A solar home system generally lies into category 2 as well as it can extend up to category 3. These tiers have peak capacity of up to 200 W and households eventually move to the next tier as they climb up the energy ladder.



INDICATIVE SERVICES	TIER 1	TIER 2	TIER 3	TIER 4	TIER 5
	Task lighting + phone charging or radio	Tier 1 + General Lighting + air circulation + television	Tier 2 + Light applicances	Tier 3 + Medium or continuous applicances	Tier 4 + Heavy or continuous applicances
1. Peak capacity	3 W	50 W	200 W	800 W	2 Kw
2. Service hours	4 hrs/day	4 hrs/day	8 hrs/day	16 hrs/day	23 hrs/day
3. Reliability				✓	✓
4. Quality				✓	✓
5. Affordability			✓	✓	✓
6. Legality				✓	✓
7. Health/Safety				✓	✓

Figure 2.6: Multi-tier framework classification of electricity access [18]. The solar home systems generally lie into Tier 2 or extendable to Tier 3, according to their peak capacity.

The analysed systems lie in either Tier 2 or Tier 3, dependent on the peak power capacity, where they support low-power home electronics as well as being affordable and reliable. The systems support off-grid appliances, which can be categorised as efficient. The appliances are generally tailor-made for the low-income rural households. A dedicated survey is done across the continents aiming at creating a quantitative ranking on the load demand [19] of consumers in rural areas. The survey also categorizes the appliances according to their power rating, in Watts and the "needs", from a-f, the appliances can meet. The survey results is illustrated in Table 2.1 and these can be useful for realising the priority and the usage of the loads connected to a solar home system.

Table 2.1: Load Categorisation according to the survey [19]

Appliance	Typical Rating [W]	Needs
LED	1-5	a
Mobile charging	3-20	b
TV	10-50	b
Radio	2-5	b
Fridge	40-400	c
Fan	15-100	c
Laptop	30-100	b
Water Pump	40-800	d
Rice cooker	200-250	e
Iron	150-2000	d
Kettle	100-800	e
Speaker System	10-50	b
Hair clippers	15-50	d

In line with the multi-tier framework and listed appliances, the project test case will involve high-efficiency DC appliances. The appliances with power rating no more than 100 W and with matching needs of up to "c" level will be considered and used.

2.2 Energy Management Systems Applications

Energy management system in a household is a device or a unit that monitors, analyzes and controls the house energy usage and reflects the data back to the user. Such unit is also capable of integrating renewable energy sources like solar or wind to match the needs of the users. Energy management systems evolve with the emerging technologies and according to the applications. Household energy management systems (HEMS) can be applied in four major areas: customer-, network-, market- and service-based applications.

- **Customer-Based Applications:** This type of HEMS aims to let the users manage energy use while adapting the daily activities and needs. This application places the users as the center point of decision-making process. According to the [15], the example projects resulted in positive behavioral changes and conscious consumption reduction. Power monitoring and feedback, remote device control and personalisation or goal setting (such as yearly consumption) are the main functionalities of this type of application.
- **Network-Based Applications:** This application is applicable for micro grid and grid-tied occasions where the power flow between generation and loads is optimized to reduce peak loading. It aims to improve the grid reliability. Demand response is used within this application, in which the users are encouraged to have short-term reductions in their energy demand [15]. Furthermore, load shifting and congestion management are ways of supporting distribution network within this application.
- **Market and Service Based Applications:** These types of applications focus on the energy management in distribution network level. Electricity tariff system is an example of

market-based application. On the other hand, service based applications are the contracts established via legal entities, such as service providers, that make sure the energy is delivered at the lowest cost [15].

For rural electrification application, customer-based application is foreseen to be the most prominent type of application, where the user needs are accommodated while decreasing the energy use as much as possible. Data acquisition and feedback system will work in line with this application. Besides, demand response or load shifting functions of the network-based application can also be integrated to this application. The control signal that is normally used in demand response algorithms can be adjusted to a different system parameter such that the PV generated energy is utilized or storage lifetime is maximized.

Multi-agent framework is an intelligent energy management system for households, that aims to meet constraints of multiple actors. Agents in such system are pack of software and hardware tools that can intelligently operate, act towards a goal and adapt autonomously. Hence, a multi-agent framework in EMS, is a community of autonomous, intelligent and goal-oriented agents who cooperate and coordinate their decisions making to reach a global goal. For rural electrification with smart DC loads, PV panel, battery bank and energy management unit, the agents can solely represent each unit, or each division of control areas.

2.3 USB Type-C Power Delivery Protocol

Universal Serial Bus (USB) is the most common serial interface that is deployed in almost all the electronic devices. The latest version is the USB Type-C interface. This latest specification has brought multi functionality within a single cable. The distinctive functionalities can be listed as:

- Power role negotiation.
- Bidirectional power flow and dynamical alteration of the data and power flow.
- Up to 100 W of power capabilities, with USB Power Delivery protocol.
- High-band video capabilities with alternate mode negotiation, allowing different data communication protocols.
- Reversible connector, enhancing user experience.

The revolutionary USB Type-C interface maintains the host-to-device relationship, unlike the previous specifications, thanks to the Configuration and Communication Channel (CC) lines. Furthermore, *Configuration and Communication Channel* is used to negotiate and to set up power as well as settings for the proprietary peripherals such as alternate modes. Besides the CC lines, the USB Type-C receptacle hosts 22 other pin connections that ensure pin-out reversibility while providing USB 2.0 interface and USB 3.1 high speed communication. The pin out view of USB Type-C plug is represented in Figure 2.7.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12
GND	TX1+	TX1−	VBUS	CC1	D+	D−	SBU1	VBUS	RX2−	RX2+	GND
GND	RX1+	RX1−	VBUS	SBU2	D−	D+	CC2	VBUS	TX2−	TX2+	GND
B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1

Figure 2.7: USB Type-C pin out view [frontal][20]

Any device that comply with the USB Type-C specifications is able to incorporate a configuration process to detect and to establish the source-sink connection. Source is the power supplier with

USB Power Delivery specifications and also acts as a Downstream Facing Port (DFP). DFP is the host role in the data negotiation. On the other hand, sink is the device that consumes power from the Vbus in power negotiation and acts as Upstream Facing port. Upstream Facing port (UFP) is the device role in the data negotiation. Moreover a device can support and operate as both source and sink power roles, which is called as Dual-Role power.

2.3.1 Cable Detection and orientation

The device power roles are initially specified according to the resistor types and values connected to the CC lines. A device with the pull up resistors connected on the CC lines is configured as a source whereas a device with pull down resistors connected on the CC lines is determined as a sink. The termination requirements are dependent according to the port supplying capability. The CC lines are hence used to determine the cable attachment/detachment and the orientation of the attachment.

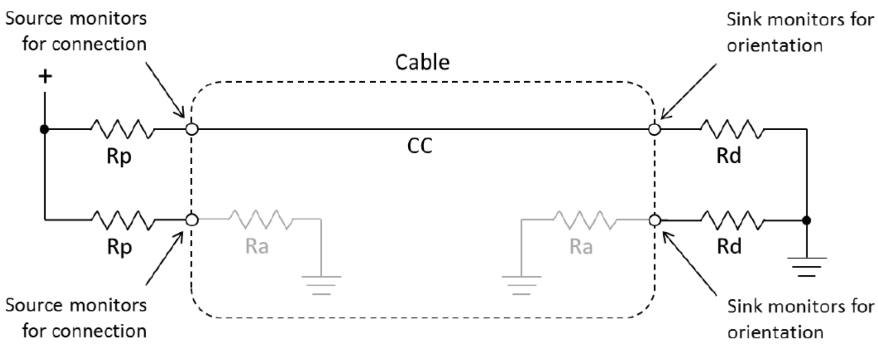


Figure 2.8: USB Type-C cable detection and orientation electronic circuit illustration, for port partners:source and sink side [20]

2.3.2 Power Levels

Any device with USB Type-C connector is able to support 5V with default values of current that were specified in the previous USB specifications. Indeed a source is also able to implement higher current over Vbus if it complies with USB Type-C power delivery. The current levels, which are available, are advertised or negotiated via the CC pins. These current levels are:

- Default values: 500 mA (USB 2.0) and 900 mA (USB 3.1)
- 1.5 A
- 3 A
- 5 A (for only Electronically Marked Cables - EMC)

The current levels can be dynamically negotiated and scaled according to the capabilities of the source or requirements of the sink. This capability enhances the ability of energy management in the devices and usage of the power resourcefully. Besides the current level negotiation, a device that supports PD is able to manage loads with high power, thanks to voltage railing. Unlike the previous USB modes of operations that only source nominal 5 V, Type-C PD capable device is able to offer a range of 5-20 V of nominal voltage. Table 2.2 summarizes the device power options.

Table 2.2: Hierarchical order of USB protocols and their power delivery capabilities.

Mode of Operation	Nominal Voltage (V)	Max. Current (A)	Max. Power (W)
USB 2.0	5	0.5	2.5
USB 3.1	5	0.9	4.5
USB Type-C current at 1.5 A	5	1.5	7.5
USB Type-C current at 3 A	5	3	15
USB PD	up to 20	up to 5	100 W

According to the USB PD Type-C specification [20], there are power rules that the device as a source needs to follow. The normative power levels are introduced with the minimum current and responding voltage levels, as illustrated in Figure 2.9. Besides the fixed voltage, a PD capable device can also act as a Programmable Power Supply, where the voltage level varies. The power rules, hence, introduce permitted voltage levels and programmable power supply selections as function of cable current ratings. Furthermore, referring to Figure 2.9, the source needs to be able to provide lower power profiles than the negotiated one. Hence, it is made sure that any high capability source is able to support low power capability devices.

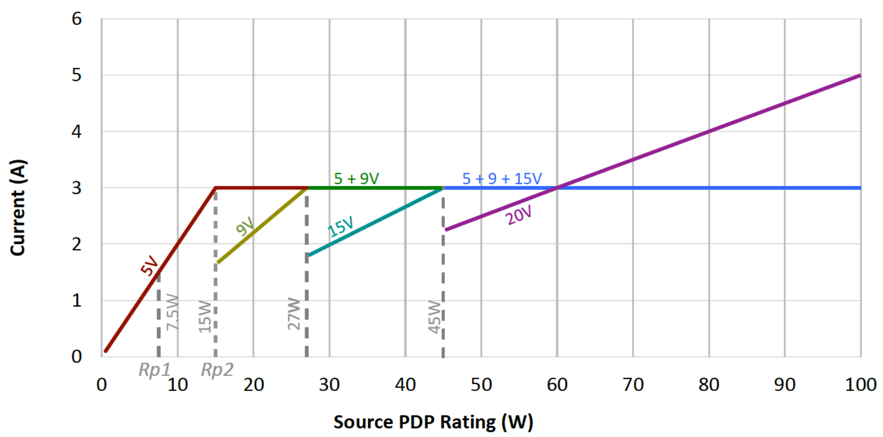


Figure 2.9: Power levels with the minimum current and corresponding voltage levels. A PD capable device is required to follow these power rail rules, voltage levels and current ratings.

For instance, an adapter with 50 W power capability rating is required to support 20 V at 2.5 A, 15 V at 3 A, 9 V at 3 A and 5 V at 3 A. These capabilities are advertised via Power Data Objects as 32-bit data messages. The data messages contain information on the current, peak current, voltage level and other source capabilities. Power data objects differ according to the source/adapter electrical capabilities. According to the USB Power Delivery Specifications [20], a fixed PDO represents a capability of a well regulated voltage supply. On the other hand, augmented/programmable PDO represents the capability of a supply with output voltage that can be programmable and adjusted over a voltage range. Accordingly, in order to meet the conditions of the user without compromising of the specifications, there are rules to be followed by a source. These can be highlighted as:

- Maximum current for Normative and Optional Fixed supply PDOs Shall be either RoundUp (x/Voltage) or RoundDown (x/Voltage) to the nearest 10mA.
- Maximum current for Programmable Power Supply APDOs shall be RoundDown (x/Max Voltage) to the nearest 50mA.

- The Programmable Power Supply voltage ranges map to the Fixed Supply Voltages. For each Fixed Voltage there is a defined voltage range for the matching Programmable Power Supply APDO.

Similarly, the devices configured or designed as sinks are required to comply the rules for the power as well. A sink is allowed to use power of a source with x Watts power rating;

- Either operate or charge from Sources that have a PDP Rating more than x Watts.
- Either operate, charge or indicate a capability mismatch from Sources that have a PDP Rating less than x Watts and more than 0.5 Watts.

2.3.3 Data Buffering and Communication

Data transmission and communication in USB Type-C connectors and cables is achieved across a single signal wire, which is one of the CC lines. The signaling is achieved by a pair of transmitter and a receiver. Collision avoidance is practiced for error minimization on communications.

The data packet buffered is a set of formatted data. It consists of a preamble (to train the receiver), an indication of start of packet (SOP), data buffer that includes message header and a CRC (for data and header protection), and an indication of End of Packet (EOP) [20].

The PD messages follow a specific signalling scheme, called Bi-phase Mark Coding (BMC) [20]. The dedicated DC connection, which is one of the CC Line, is used for transmission of PD messages. Table 2.3 lists the electronic characteristics of the receiver and transmitter required as well as some of the normative requirements of BMC [20].

Table 2.3: USB-C PD data transmitter and receiver characteristics for Biphase Marked Coding signalling in normative operation.

Parameter	Description	Nominal Value	Units
fBitRate	Bit rate	300	kbps
zDriver	Transmitter output impedance	33-75	Ω
cREceiver	CC Receiver capacitance	200-600	pF
zBmcRx	REceiver Input Impedance	1	MΩ

2.4 Findings

Throughout the background research, current existing technologies and systems have been evaluated. It has been found that, although there are existing and emerging technologies in the field of rural electrification, there is still room for development. The current systems are observed to be limited to a specific set of loads. Despite the fact that this increases the reliability of the system, the adaptability and scalability of the system is omitted. These parameters will be taken as basis in the design of the software architecture.

Furthermore, the load profile and devices that make up the loads in such systems are evaluated with the help of the framework. Accordingly, a set of loads, which operate in DC and possess high-efficiency is chosen for the later test cases. The energy management systems in households are identified according to their applications and functions. Consequently, a household energy management system that involves the customers as the decision-makers and adapts to such conditions, is recognised as the most prominent type of application. The loads and devices in such system can also be treated as agents that can cooperate with each other for achieving a common goal.

3

Solar Home System Components

This section takes a look into the each pillar of the proposed system. The proposed system components are classified as basic loads with no adapter, loads with adapter, power bank for storage, charging station or charging hub with multiple ports. Each device will be evaluated in terms of the electronics and microchips used for implementing the required software. Such evaluation will aid to understand how the system works and reacts to the software states.

3.1 Transition into a DC Microgrid

The ultimate aim in rural electrification project is to achieve and to utilize a reliable low-level DC microgrid. However, the trajectory path to achieve this ultimate goal is a gradual one. Indeed, for a fully operating and reliable DC distribution or a microgrid serves challenges in terms of technology [21]. Therefore, several trajectory steps should be taken. In [22], the criterion to design the SHS is listed which leads a path to achieve a low-voltage microgrid.

In [22], the author specifiee the very first step of services that is provided by a solar home system, which is the mobile charging point. The adaptability and modularity feature is then introduced with integration of powerbanks into the system. At this stage users can use charging points during the day to charge the powerbanks and remotely use these devices for powering up several electronics. Figure 3.2 illustrates the first two stages in this transition. It is noted that the introduction of USB Type-C protocol also adds value to modularity at this stage. The capability of charging with high power is expected to let the users use the devices more mobile and with higher usage period.

The next stage is installation of the PV array and coupling it to the system with a DC-DC adapter. Furthermore, with the DC-DC adapter, higher power rating loads can be introduced to the system. In this stage, the power hub will be used as central device for power management, data monitoring and central device of solar home system. The complete system in this stage is represented via block diagram in Figure 3.2.

The final stage of the transition to low voltage DC microgrid within this project is a long-term goal, where the various household systems are coupled together within the community. It is expected to have a boost converter with protection devices to form a low voltage grid.

Consequently, the system components that form the modular solar home system are elaborated, in order to have a clear perspective on how the system will be coupled together and how it will

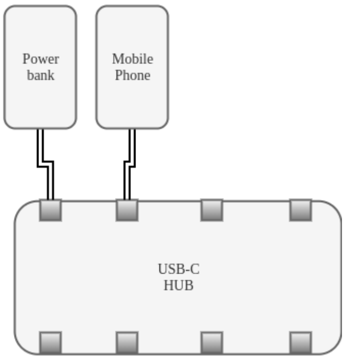


Figure 3.1: Block diagram of the first two stages of the solar home system, where the charging point is utilized for mobile and powerbank charging.

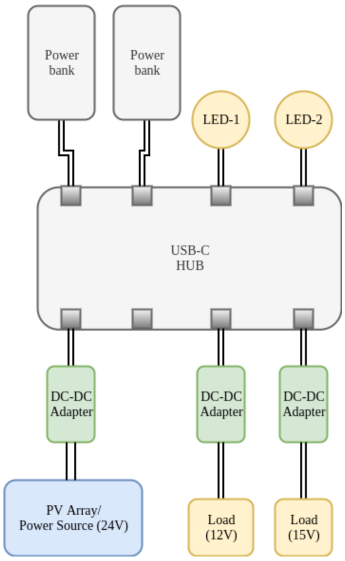


Figure 3.2: Block diagram of the proposed system in modular SHS implementation

function.

3.2 Powerbank

The powerbank is the modular component of the solar home systemt that can be used as the buffer storage and be used remotely with home electronics. The users utilize these devices charging the cells in the charging station and they can transport the energy to be discharged for home electronics in another location.

3.2.1 General Characteristics

The power bank consists of battery cells that are connected in series, rather than conventional parallel configuration. Cells connected in series lead to higher bus voltage and consequently lower current and less thermal losses. Besides this configuration, the powerbank consists of four USB Type-C ports, which adds another novelty. One of the four ports acts as the high power and bi-

Figure 3.3: Powerbank prototype.

directional port feeding the battery cells, under current control. The direct charging line provided to the battery cells from this port can be reversed with a P-type MOSFET switch, so that the port can deliver high-power output directly from battery cells (50 W- 16.8 V). On the other hand, the other ports are only able to deliver 5V on the output and can not perform charging to the battery cells.

3.2.2 Power Delivery Characteristics

In a four cell configuration, the powerbank has a storage capacity of 41.61 Wh with a nominal voltage of 14.6 V. This system voltage is stepped down by a buck converter and delivered to all four ports. The microcontroller used is the STM32G071CBTx, which has integrated USB-PD PHY interface on two channels [23]. Hence two ports (in this configuration; port 0 and port 1) use this MCU as TCPC. This saves space and cost since no additional port controller is implemented on these two ports. On the other hand, port 2 and 3 is connected to an external port controller (PTN5110) [24]. Eventually, the power path of all four ports is controlled by high power PD switch, NX20P5090 [25]. In ports 0 and 1, this switch is controlled directly by the MCU whereas for ports 2 and 3, this is achieved by PTN5110.

Table 3.1: Power Bank Port Power Ratings

Port	Max/Nominal Voltage (V)	Max. Current (A)	Max. Power (W)
0	16.8	3	50.4
1	5	3	15
2	5	3	15
3	5	3	15

3.2.3 Cable Attachment Detection

Cable attachment and detachment detection is achieved by reading the voltage level signals on the CC lines of the port and port partner. As described in Section 2.3, the cable detection is dependent on the resistor values on the lines. Initially, on the port side the lines are pulled-up internally by either the MCU or the PTN5110 and hence the voltage level on each CC line gives out 3.3 V. This is when the source is looking for connection and the CC lines are open. When a sink (a device with pull down resistors) is connected, the voltage level decreases according to the resistor values. Additionally, there are 22 pF capacitors connected on CC lines in order to cancel the noise for a clean data transfer.

On ports 2 and 3, with the PTN5110 TCPC, a detected change in voltage levels in CC Lines sets the corresponding bit high in Alert register of the controller. This manipulated bit in the register rises an Interrupt event on the $ALERT_N$ line, which is pulled up by the MCU 's internal 10 kΩ pull-up resistor. The interrupt event is a falling edge trigger event and as soon as this interrupt is called, it pulls down and triggers an event on the Type-C port controller state machine.

Likewise, for ports 0 and 1, the physical layer is integrated within the MCU STM32G071CBTx [23], the cable attachment detection is done by monitoring the voltage levels on the CC lines. This MCU has specific internal registers for power delivery. By monitoring and writing to these registers, the port controller functions can be achieved. The CC events are read through the status register and any change on the corresponding bits fire an interrupt event on the corresponding interrupt line, connected to the microcontroller. This event wakes up the Type-C Port Controller state machine.

3.3 Charging Hub

3.3.1 General Characteristics

Charging or the power hub is another main component of the system. It consists of 8 ports each capable of delivering power up to 60 W (20 V and 3 A). One of the dedicated ports allows direct charging on the input side and voltage control on the output side. Hence a bi-directional power flow is achieved and powerbank can be connected on this port for direct charging. Besides, another dedicated port can be used for PV panel MPPT interface with the its dedicated boost converter. Charging hub also contains a Wi-Fi module, that can be used for remote control, debugging and data monitoring purposes.

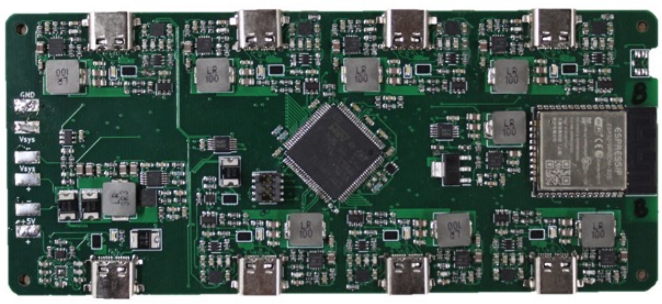


Figure 3.4: Charging Hub prototype.

3.3.2 Power Delivery Characteristics

The microcontroller used is the STM32F091VCTx with increased Random-Access-Memory [26]. The increased RAM leads to higher stack ability and to run tasks for each port. Hence, the power delivery architecture can be implemented without any memory issues. This architecture uses two I^2C buses, one bus for each group of 4 ports. Consequently, the TCPCs on the ports and the Digital-Analog Converters dedicated to each port, communicate with the TCPM over their own I^2C bus. Power path control on the sourcing ports are achieved by dedicated power switches controlled by the port controllers. These dedicated switches also implement over-voltage and over-current protection and report back to the port controller over FAULT line.

3.3.3 Cable Attachment Detection

Likewise in powerbank with the PTN5110 TCPC connected ports, cable attachment/detachment detection is achieved by monitoring the voltage levels on the CC lines. The CC lines are internally pulled-up by the PTN5110 and trigger a detection when the voltage level is pulled down by a connected Sink. This, eventually triggers an event on the EXTI Interrupt line, which wakes up the Type-C port Controller state machine.

3.4 Charging Station

The solar charging station is the main and central component of the solar home system. The solar charging station has a direct PV panel input, that is connected to Maximum Power Point Tracking converter. Furthermore, it possesses an external battery connection via buck/boost converter. It consists of 4 USB Type-C ports for connection of loads and delivering power up to 60 W via power delivery protocol. The charging station also features Raspberry Pi connection for data monitoring and communication over serial bus. Furthermore, this feature can be extended via wireless connection to identify the connected devices and perform energy management applications remotely.

3.4.1 General Characteristics

The ideal function of the solar charging station is when the PV generation is induced from the connected PV panel. This connection is through MPPT interface and its functioning is controlled via MCU. The power generated here is fed into the 24 V bus voltage. This bus voltage is further stepped down as the source for the USB Type-C ports. If no loads are connected or the demand is smaller than the PV generation, this surplus generation is stored within the batteries. Similarly, when the PV source is insufficient to cover the load at a given time, the battery will discharge to supply power to the ports.

3.4.2 Power Delivery Characteristics

The charging station uses the STM32F105RCTx MCU [27] as the TCPM. It also controls the other peripherals such as the buck/boost converter, MPPT controller and peripherals such as Analog-Digital Converter (ADC), Universal Asynchronous Receiver/Transmitter (UART) and Direct-Memory-Access (DMA) for the data monitoring and communication. 24 V bus voltage is further stepped down via converter and the resulting voltage can be fed to the ports. Each port is integrated with the port controller PTN5110 that controls the high power PD switch for enabling/disabling the bus voltage on the ports [24] [25]. The TCPC communicates to the MCU over I^2C bus, which is the bus 1 for all four ports. The switches implement over-voltage and

Figure 3.5: Charging station prototype.

over-current protection, as implemented in the charging hub.

3.4.3 Cable Attachment Detection

Cable attachment/detachment detection is implemented in the same way as the charging hub. The TCPC is responsible for monitoring and detecting connection over voltage drop in CC lines. Since each port has dedicated port controller, each port can generate an individual alert and trigger the corresponding interrupt lines. Consequently, this will wake up the Type-C port controller state machine for that port.

3.5 Load with no adapter

In this section, a fixed DC load will be presented. For this purpose, a LED circuit that acts always as a fixed sink object is chosen. LED lights are the pillar of rural electrification and the basic load of a household. It is one of the most prioritised loads in the load profile since it is a necessity for the households without lighting. The presented LED circuit in Figure 3.6 incorporates 3 homogeneously distributed LED lights.

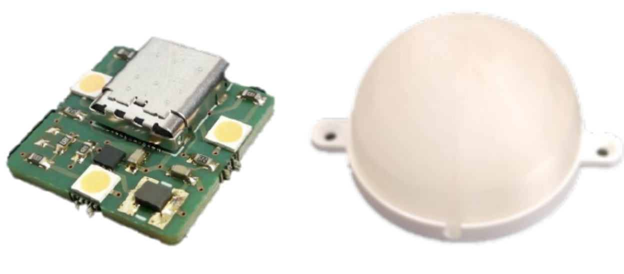


Figure 3.6: LED, fixed DC Load.

3.5.1 General Characteristics

The illustrated LED circuit consumes 2 W while it provides 312 lumens. The light is diffused in all the facing angles, thanks to a custom designed LED casing.

3.5.2 Power Delivery Characteristics

The LED light acts as the fixed DC load. It does not entail any smart controller or chip. Hence it always asks for fixed DC 5 V, 0.4 A.

3.5.3 Cable Attachment Detection

Since this device is configured and used as a sink (consumer), it incorporates pull-down resistors on the CC lines that are connected to the port. The pull down resistor values are 5.1 kΩ. As soon as the it is connected to a sourcing port, the resistors pull down the voltage level lower. This is monitored on the port partner side (source) where the connection is detected. As soon as there is detection and bus voltage is enabled, LED starts receiving voltage over the Vbus, which is the start of the operation.

3.6 Load with DC-DC adapter

The DC-DC adapter with the STM32G0 MCU is used for adapting simple DC loads into the system [23]. Thanks to the electronics on the adapter, different voltage levels can be negotiated over the USB Type-C protocol for different range of loads and other data for energy management can be transmitted/ received. These adapters will be used for the aforementioned loads in Section 2.1.3, to integrate the loads into the solar home system and to implement energy management application.

3.6.1 General Characteristics

The adapter has nominal functionalities, with supply voltage ranging from 5 V to 20 V with USB-C Power delivery connection and capability. On the output, two different supply programmes can be performed. One is constant current and the other one is constant voltage (being programmable), with output range of 0-20 V and 0-5 A. Programmable output is controlled by the main micro controller unit , STM32G071GBx series [23]. The MCU has dedicated pins connected to the Digital to Analog Converter (DAC) on board. By writing data to the registers associated to these pins over the Hardware Arbitrary Layer (HAL) on DAC channels 1 and 2, the output voltage and current limiting is achieved.

3.6.2 Power Delivery Characteristics

The adapter acts as a sink on its USB Type-C port. Hence the adapter when connected, waits to receive the source capabilities data object message from the port partner and responds back with the requested voltage levels. There are four voltage levels that the adapter is able to deliver on its output side. The voltage levels are 5 V, 12V, 15 V and 20 V. According to the load connected on the output side, the output of the converter is adjusted. These sink capabilities are sent to the port partner. If the supplied voltage level on the Vbus has reached the desired voltage level of the connected load, then the high power switch gets enabled. Otherwise, the converter is enabled to

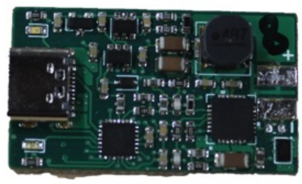


Figure 3.7: DC-DC adapter prototype.

maintain the output voltage on the desired level of the load connected.

3.6.3 Cable Attachment Detection

Cable attachment/detachment in this adapter is achieved via the USB Type-C port and port partner monitoring the CC line voltage levels. Since this adapter is always acts as a sink (consumer), the CC lines are connected to the internal pull down resistors with 5.1 kΩ. When there is no connection, these pins are grounded and eventually on connection with the port partner, the pull down resistors bring the voltage levels down on the port partner side. The change in the voltage levels of the CC lines is detected on the PHY layer of the port partners, achieving the cable detection. Likewise, the Type-C port controller state machine regularly monitors these voltage levels for detection of detachment.

3.7 System Information Flow

In this section, the data and information flow between the prototype components and their entailing peripherals are represented.

Besides the power flow, which is the current flowing in the direction from the source to the sink, there is also data flow on the sideband channels on the USB Type-C receptacle and cables. The information flow is dependent on the configuration of the ports according to their roles, either as Upstream Facing Port (UFP) or Downstream Facing Port (DFP). Initially, the sources are configured as DFP whereas the sinks as the UFP. In the Figure 3.8 the flow of the information throughout the system (hub and a sink) is illustrated on the prototype components. System high level components are taken as basis on the representation of the information flow.

Blue arrows represent the set of data buffer transmitted from the device policy managers of the devices, which are main MCUs. On the one hand, red arrows illustrate the set of information received and to be processed by MCU. The green arrows illustrate the external event, in this

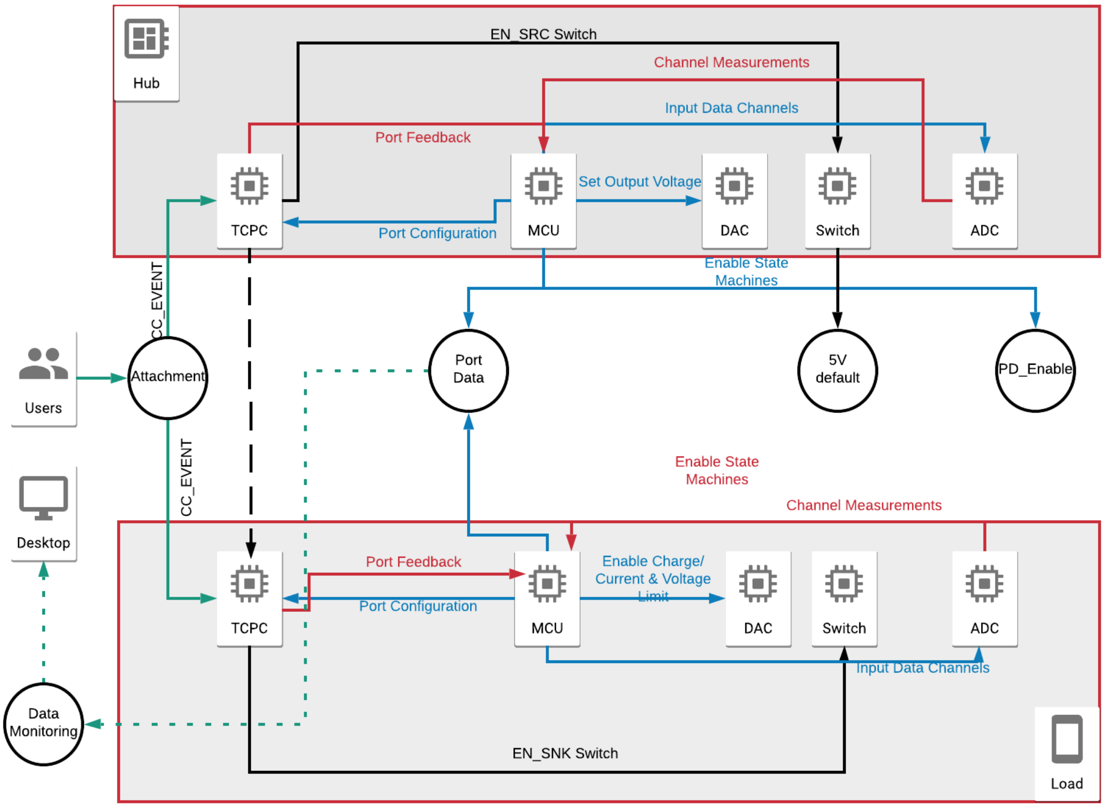


Figure 3.8: Information flow between hub and load

case attachment/detachment. Peripherals and converters are controlled and data information is exchanged by MCUs on the devices whereas the TCPCs (port controllers) have the control on the port switches, data transmission between partners and reporting back to the MCUs. Data gathered about the port power measurements by Analog-to-Digital converter, system information and state changes are buffered to the monitoring layer via serial communication. This set of data is fed to the developer.

4

Firmware Design Topology

There are two open source available embedded software packages based on USB Type-C specifications. This chapter studies these by looking into the architectural topology among layers of application. Two packages are compared and ultimately used with several adaptation layers. The adaptation layers are also presented within this chapter.

4.1 General Architectural Overview

The high-level logical architecture of the USB Type-C power delivery package is presented within this chapter. The general overview can be used to indicate the logical blocks and possible links between the several communication stack. A power delivery capable device should have one of the listed characteristics;

- to be made up of at least one port,
- to be able to sink or source power,
- to be able to toggle between roles (dual role power),
- to be able to communicate using SOP packets.

Furthermore, power delivery capable USB devices consist of four communication stacks.

- Device Policy Manager (DPM): DPM's role within the architecture is to manage the power used for a local port. Device policy manager can also be adjusted to handle the policies for multiple ports. Hence it has the highest decision making mechanism within the Power Delivery stack. It is the main manager of the resources within the port, which is able to talk to the communication stack, Source or sink and port controller block. DPM is also the main and final decision maker within this stack.
- Policy Engine (PE): PE implements the local policy,that is directed from DPM, for that port. Hence, the Policy engine layer is implemented for each port within the device. Negotiation and status mechanisms are carried out within this layer. PE layer is responsible for allowing the power negotiation and establishing the explicit contract. Furthermore, it also handles the Vendor Defined message flows by driving the message sequences.
- Protocol Layer(PRL): PRL forms the messages, in order to communicate the information between the paired ports. PRL receives input from PE indicating the type and data of the message to be sent and indicates the responses back to PE. Hence its main role is to drive message construction, transmission and reception.

- Physical Layer (PHY): PHY layer handles the message transmission and reception of bits across the CC wires of the USB Type-C cable. It also handles the transmission errors, by collision avoidance and detection using a CRC.

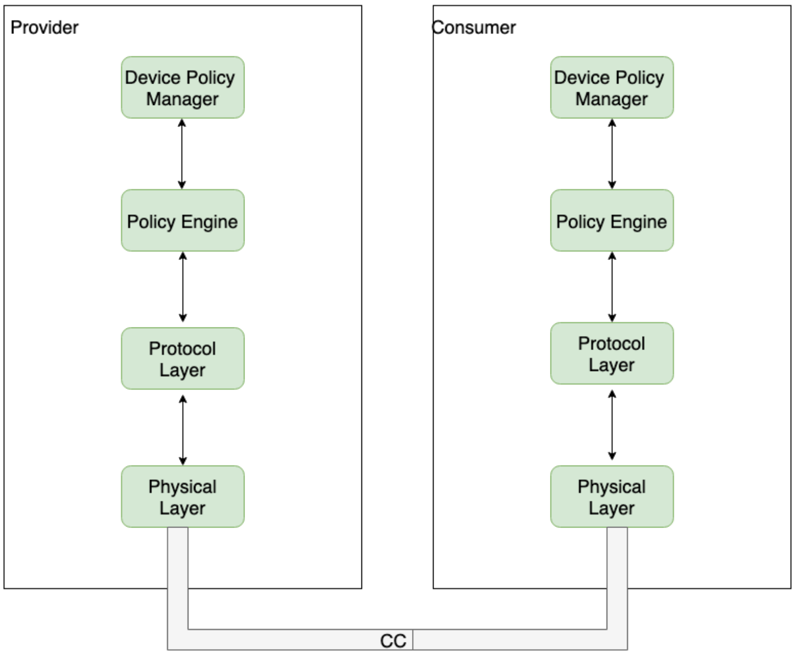


Figure 4.1: USB Power delivery stack architecture

In this high level architecture, policy engine drives the message sequences. Data flow starts from the PE layer and ends down at the PHY layer. For USB Type-C specified applications, Atomic message sequencing (AMS) is being established. AMS is initiated when the policy engine gives a Ready state. These states will be investigated in the following Section 5.2. AMS is initiated either by a source or sink, depending on the configurations directed by device policy manager for that specific port. Figure 4.2 represents the message flow through the layers between a source and a sink. Within the AMS, there are two different types of messages transmitted.

- Control Messages manage the message flow between the port partners. Control messages do not include any additional data within the packet and consist of 16 bits.
- Data Messages are used for exchanging information between the port partners. Data messages can be up to 240 bits.

The messages are constructed within the PRL layer and they are composed of a message header and optionally data objects. Message header consists of 16 bits and can be either a standalone as a control message or as the first part of the data message. It contains the basic information about the message type, port power and data roles. On the other hand, each data object consists of 32 bits and include information to be exchanged either for power negotiation or vendor defined messages. Message construction and types of data messages are further dealt in detail in Section 5.3.

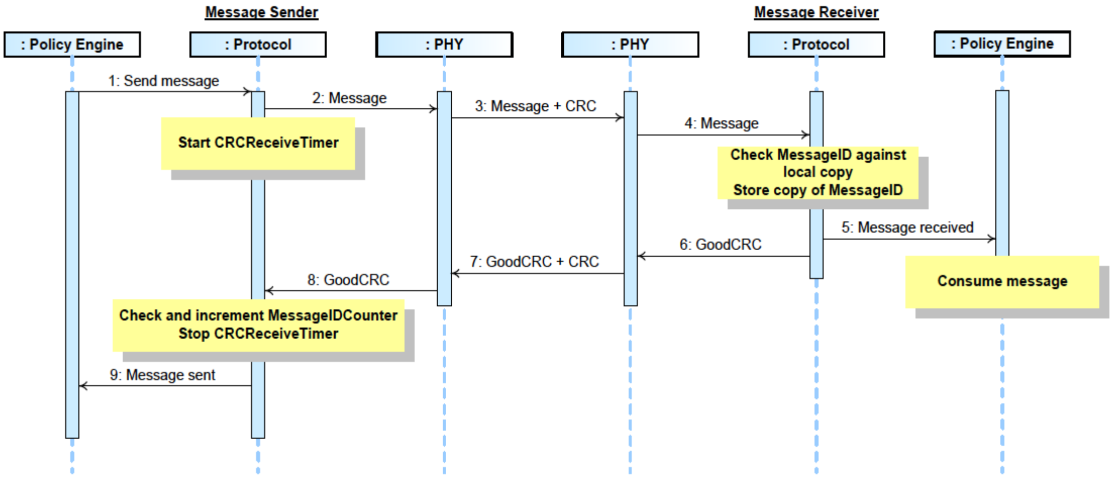


Figure 4.2: USB Power delivery atomic message sequence [20].

4.2 TCPM/TCPC Architecture

In a lower level application, USB power delivery firmware implementation can be achieved in a Type-C Port Manager (TCPM) and a Type-C Port Controller (TCPC) standardized architecture. Rather than using one component as the whole implementation of power delivery architecture layers, several components can be used in order to ease the implementation. The hardware and the software partitioning in a multi-port application is hence one of the main advantages of this architecture.

In this architecture, there is also a TCPC interface (TCPCi) that uses I^2C link for communication between two entities. I^2C is a synchronous serial bus communication used in low-speed communication between ICs and peripherals. It can host multi-master and multi-slave configuration, over Serial Data Access (SDA) line with configured clock Serial Clock (SCL) line. The TCPM is the master in the I^2C application and is able to drive multiple slaves. Moreover, it handles the Policy Engine state machine. On the one hand, the TCPC is the slave in I^2C link and drives the Physical layer. Within this configuration, alert and fault lines can be implemented and are fedback to the TCPM, for troubleshooting and robustness The TCPC interface specifications have been followed to acquire the set of registers for such architecture. Nevertheless, the protocol (PRL) specifications and functions are shared between TCPM and TCPC. The TCPC, in this case, is responsible for the re-transmission of the messages and GoodCRC management.

In the cases of TCPM/TCPC architecture implementation, STM32 MCUs have been used as the TCPM and the PTN5110 TCPC IC is used as the slave TCPC Phy interface controller. Type-C port controller driver has been developed, guided by the device programming manual and the port controller interface specification [24],[28]. The block diagram of involved electronics for this architecture is represented with the power electronics, in Figure 4.4.

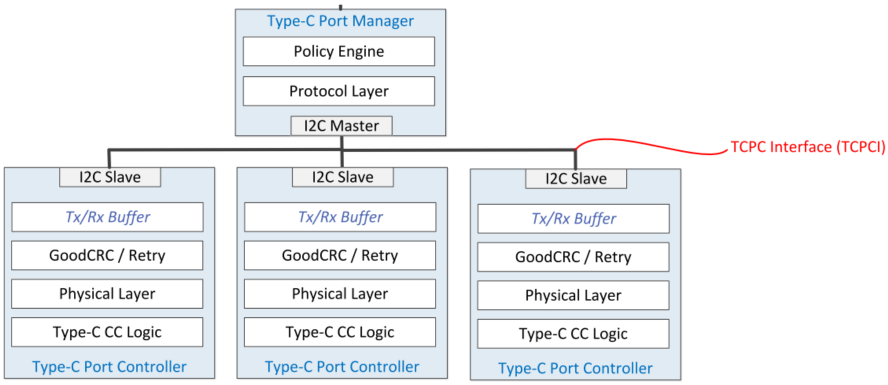


Figure 4.3: TCPM-TCPC via controller interface, retrieved from [28]

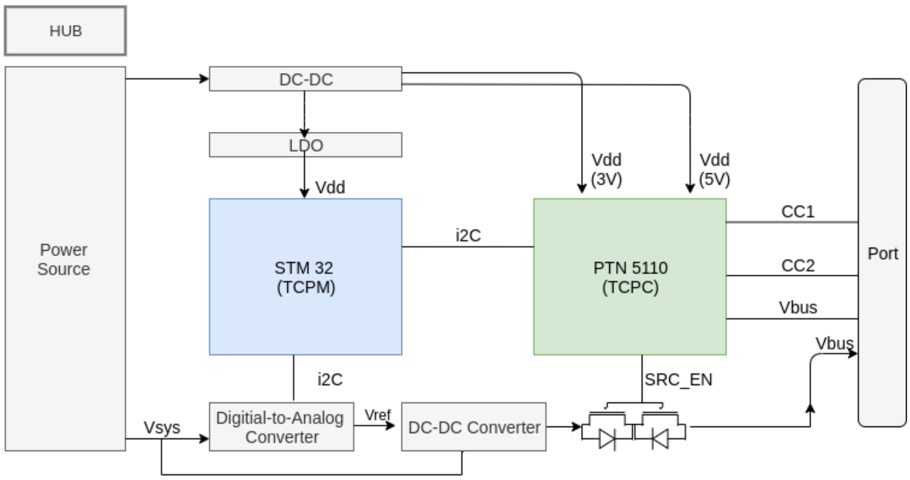


Figure 4.4: Block diagram of the TCPM-TCPC architecture in a Source Device, such as hub.

4.3 STM32 Power Delivery Expansion Package

Originally called, X-CUBE-USB-PD is a certified USB-IF expansion package, that supports many functionalities of USB Type-C specifications. The expansion package aims to run on the STM32F/G0 MCUs acting as the TCPM and optionally as the TCPC. The package consists of the stack libraries, drivers and sources, to make use of applications on USB-PD.

The final version of the expansion package is compliant with the USB Type-C 1.3 and USB PD 3.0 Standards [20]. Hence the package supports functionalities such as dual role port support, PD communication on both sides, cable attachment and orientation detection, message transmission or reception and structured Vendor Defined Messages support for alternate modes. On the other hand, the software package has brought some limitations when used under development. One of the main disadvantages of the software expansion package is that the core stack library was inaccessible and included "protected" configurations and defines that limited the capabilities of the application. In order to investigate the capabilities and development features of the expansion package, core stack library is explored.

4.3.1 Core Stack Library

The package includes the core stack provided in the binary format and the components in source code and header files. Within this library, the TCPM/TCPC architecture is chosen. By such choice, the memory footprint of the Cortex M MCU is decreased. Furthermore, using a USB-PD compliant TCPC allows implementation of *Vbus* voltage monitoring and measurement, *Vbus* forced/auto discharge features upon disconnection as well as allowing two output controls (enabling/ disabling) for controlling the load switches on *Vbus*. Consequently, this architecture offers flexibility on the voltage rails with wide range of input voltages.

Core stack is delivered as library. The library is built as the TCPM/TCPC architecture, following the USBPD 3.0 specifications [20], with DRP and VDM functional capabilities. Furthermore, the library is built on Keil ARM CM4 core with the STM32CubeIDE target compiler, based on Atollic TrueStudio. USBPD core stack and the USB PD capable devices are hence fully managed by this library. On the other hand, user application and hardware part is adjusted. For the TCPC, device driver is developed. This device driver is specifically created for the TCPC component and developed in such a way that it can be adjusted to different drivers. Moreover, device policy manager files are created to manage the logical flows and the system with user applications.

Core stack architecture represented is built on top of the STM32 hardware abstraction layer (HAL) and low-level (LL) layers. HAL drivers provide set of Application programming interfaces, allowing the user to use set of functions for several peripheral features (DMA, ADC, I^2C, UART, clocks, interrupts). These drivers use call-back mechanism for giving feedback on system level initialization, interrupt events and error events. Furthermore, USB-PD on STM32 core library developed on middle-ware component of Real Time Operating System, *FreeRTOS*. Real-time operating system is an operating system, specific for embedded and real time applications, that aims to ensure deterministic response to events. Within this stack,*CMSIS-RTOS* is used as the wrapping layer with its functions for the *FreeRTOS* middle-ware, that manages the high level layers. This allows the developer to manage the clock and timing as well as running several applications on real-time. Hence, the applications can be managed as threads. Any interrupts, events within the core and user applications are handled as messages and put into the queues of that specific thread/ application.

4.3.2 Library Functions and Characteristics

- In the main program file, besides the initialization of the peripherals and MCU clock, *USBPD DPM Init* is called to initialize the USB-PD related communication layers.
- *USBPD DPM User* includes the callback functions for interrupt events, management of power board profiles and functions for fulfilling the core stack API.
- *USBPD PWR IF* defines and development of power profiles, power objects for implementing a power delivery protocol between port partners.
- *FreeRTOS Config* configuration of real time management and definitions, in order to run the operating system reliably and robustly.

4.4 Chromium Embedded Controller Package

Chromium OS project is an open source software development package for embedded controllers, or so called MCUs, specifically built for Chromebooks. Within this package, power delivery and alternate modes are implemented. Here the USB-PD is considered as stack of several state machines. The open source project is acquired and adjusted with only the required state machines, files and functions in order to implement the power delivery and alternate mode negotiations.

It is found out that this package creates more redundancy and accessibility for the core files. Furthermore, the package can be expanded and adjusted to many devices and applications. Besides these positive aspects, the basic power delivery firmware is not limited by the number of ports, within the core library.

On the other hand, ChromiumOS project is not based on the middle-ware sources of Hardware abstraction layer (*HAL*) and *FreeRTOS* as peripherals and run-time. Hence only the power delivery firmware with associated state machine files, header files are acquired. This created an opportunity to develop the adaptation files and drivers, based on the peripheral libraries from the STM32 firmware. Additionally, the programming guides of port controller and power path controllers are also taken basis for custom configuration and adaptation drivers or files.

4.4.1 Core Stack and State Machines

USB Type-C power delivery within this stack is implemented as real time tasks/ threads within *FreeRTOS* that run parallel for each port. Each power delivery task of the corresponding port implements four main state machines. These are;

- A framework file, *usb_sm.c*, that calls entry and exit functions of current states of the stack.
- Policy engine state machine file, *usb_pe_drp_sm.c*, that calls the necessary state for the policy engine implementation of the source and sink applications.
- Protocol engine state machine file, *usb_prl_sm.c*, that implements the message transmission and reception states as well as the hard reset state machine.
- Type-C physical layer file, *usb_tc_drp_acc_trysrc_sm.c*, which implements the port controller state machine for sink and source applications.

The core stack and state machines are developed according to the USB-PD 3.0 Specifications, with DRP and VDM facilitation. The listed state machines are also implemented in a way to communicate with the necessary drivers such as power path controller or port controller drivers. Hence the power delivery task requires a larger task stack size in the RAM, which can store the required data, chunked buffer for reception and transmission. Additionally, there is an Alert task within the core stack that receives and handles the interrupts arose from the port controller and micro controller interrupt lines. This task is implemented in real-time so that it is quick enough to report the events to the power delivery state machines.

4.4.2 Library Functions and Characteristics

- Implemented state machines and power delivery stack is called by *USBPD DPM Init* function within the *main.c* file along with peripheral and clock initialization. This function initializes the interrupt lines, I^2C communication peripheral with FreeRTOS semaphores and eventually creates and calls the tasks that run the alert and state machines.
- Configuration files and header files include all the defines and supporting functions for the USB Type-C power delivery package.
- Policy interface file for power delivery and vendor defined data objects as well as the functions to be called for power transition, dual role swap or port power role swap.
- FreeRTOS Config file, configuration of FreeRTOS application, in order to run the operating system reliably and robust.

4.5 Adaptation Interfaces and Drivers

Although both packages offer wide area of application, they still require adaptation interface and drivers tailored for the user and application. For the rural electrification case, with the required specifications sought, the adaptation interfaces are created mainly for communication between the port controller and port manager, communication for the user/ developer. Furthermore, the drivers are developed for the port controller physical interface, Digital-to-Analog converter and power path controller in order to manage the power and electrical conditions on each port.

4.5.1 Port Configuration File

Dynamic management of the ports are achieved by this file. It includes structure of data that is stored for each port, global defines for the port management, port driver inclusion and Digital-to-Analog converter communication function that is called by device policy manager for each ports actions.

```
/* Includes ------------------------------------------------------------------*/
#include "usbpd_def.h"
#include "string.h"
#include "alert.h"
#include "i2c.h"
#include "stm32f0xx_hal_i2c.h"
#include "tcpc.h"
#include "cmsis_os.h"
#include "tcpc.h"
#include "stdbool.h"

#if !defined(USBPD_PORT_COUNT)
#define USBPD_PORT_COUNT 8
#endif


/**
  * @brief  struct for the ports
  *
  */
typedef struct
{
    TCPC_DrvTypeDef* tcpc_driver; //pointer to the Port controller driver functions
    I2C_HandleTypeDef* i2c_bus; //Port controller i2c communication bus selection
    uint8_t i2c_addr;       //shifted to the left by on bit
    I2C_HandleTypeDef* dac_i2c_bus; //Digital-Analog converter i2c bus selection
    osSemaphoreId semaphore_id; //assigned in uspd_tcpci.c for i2c semaphores
    uint8_t dac_i2c_addr;   //shifted to the left by one bit
    uint8_t dac_channel;    //DAC communication channel, each assigned for specific port
    GPIO_TypeDef* enable_port;  //enabling the dedicated GPIO port of the PD power switch
    bool port_state;            //state of the port
    bool disabled;              //if the port has any hardware issues
    uint16_t enable_pin;        //enabling the dedicated GPIO pin of the PD power switch
    GPIO_TypeDef* led_port;     //port specific LED GPIO port
    uint16_t led_pin;           //port specific LED GPIO Pin
    IRQn_Type alert_interupt;   //Alert interrupt EXTI Line specific to port
    uint16_t alert_interupt_pin;    //Alert line pin dedicated on MCU for that port
    GPIO_TypeDef* alert_port;       //Alert line port dedicated on MCU for that port
    uint8_t port_identifier;
    osThreadDef_t os_thread_def;    //Operating System Task definition for PD app of port
    char os_thread_name[5];         //Task name for PD app of port
} USBPD_Ports;

USBPD_Ports ports_list[USBPD_PORT_COUNT];
TCPC_DrvTypeDef* DevicesDrivers[USBPD_PORT_COUNT];

void initThreadNames();
void init_USBPD_Ports();
HAL_StatusTypeDef  DAC_Set_Value(I2C_HandleTypeDef *hi2c, uint8_t address,uint8_t dac_channel,uint16_t dac_value);
```

Figure 4.5: Port Configuration header file, defining the struct elements.

4.5.2 Type-C Port Controller Driver

The TCPC has three functions that are dedicated to the PHY layer of the PD architecture:

- USB Type-C Port Power Control for VBUS and VCONN (required)
- USB Type-C CC Control and sensing (required)
- USB PD Message delivery

The TCPC uses I^{2}C to communicate with the TCPM. The TCPC is an I^{2}C slave with Alert signal for requesting attention. Port controller driver, with necessary functions and handlers are developed in order to control and feedback to the port manager. Set of functions are illustrated in Figure 4.6.

```
const struct tcpm_drv tcpci_tcpm_drv = {
    .init               = &tcpci_tcpm_init,
    .release            = &tcpci_tcpm_release,
    .get_cc             = &tcpci_tcpm_get_cc,
#ifdef CONFIG_USB_PD_VBUS_DETECT_TCPC
    .get_vbus_level         = &tcpci_tcpm_get_vbus_level,
#endif
    .select_rp_value        = &tcpci_tcpm_select_rp_value,
    .set_cc             = &tcpci_tcpm_set_cc,
    .set_polarity           = &tcpci_tcpm_set_polarity,
    .set_vconn          = &tcpci_tcpm_set_vconn,
    .set_msg_header         = &tcpci_tcpm_set_msg_header,
    .set_rx_enable          = &tcpci_tcpm_set_rx_enable,
    .get_message_raw        = &tcpci_tcpm_get_message_raw,
    .transmit           = &tcpci_tcpm_transmit,
    .tcpc_alert         = &tcpci_tcpc_alert,
#ifdef CONFIG_USB_PD_DISCHARGE_TCPC
    .tcpc_discharge_vbus        = &tcpci_tcpc_discharge_vbus,
#endif
#ifdef CONFIG_USB_PD_DUAL_ROLE_AUTO_TOGGLE
    .drp_toggle         = &tcpci_tcpc_drp_toggle,
#endif
    .get_chip_info          = &tcpci_get_chip_info,
#ifdef CONFIG_USBC_PPC
    .set_snk_ctrl           = &tcpci_tcpm_set_snk_ctrl,
    .set_src_ctrl           = &tcpci_tcpm_set_src_ctrl,
#endif
#ifdef CONFIG_USB_PD_TCPC_LOW_POWER
    .enter_low_power_mode       = &tcpci_enter_low_power_mode,
#endif
};
```

Figure 4.6: Set of functions and linker script for handling the port controller registers.

4.5.3 Type-C Port Controller Interface Link

The USB Type-C Port Controller Interface, TCPCi, is the interface between a USB Type-C Port Manager and a USB Type-C Port Controller. The goal of the USB Type-C Port Controller Interface (TCPCi) is to provide a defined interface between a TCPC and a TCPM in order to standardize and simplify USB Type-C Port Manager implementations. Interface implements wrapper functions to manage the communication between these two entities over serial bus, I^{2}C. Figure 4.7 represents some of the main functions that communicates with the considering port controller's address, with the required register address. Functions handle the data input as 8-bit.

4.5.4 Power Path Controller Driver

Driver includes functions for initializing the driver at the startup, enabling source/sink vbus, implementation and current and voltage limits and discharging the vbus when disconnected. Furthermore, the dedicated DAC is communicated and desired output voltage for sourcing applications set within this driver. Additionally, the dedicated buck converter is enabled or disabled within this driver functions, depending on the application. Figure 4.8 shows the functions used to enable/ disable the dedicated buck converter of the port and the function that sets the value on the output of the DAC that is connected to that port.

```c
/* I2C wrapper functions - get I2C port / slave addr from config struct. */
static inline int tcpc_addr_write(int port, int i2c_addr, int reg, int val)
{
    LOCK_I2C_RESOURCE();
    uint8_t value =val;
    status = HAL_I2C_Mem_Write(ports_list[port].i2c_bus, ports_list[port].i2c_addr, reg, 1,&value, 1, 100);
    UNLOCK_I2C_RESOURCE();
    return status;
}

static inline int tcpc_write16(int port, int reg, int val)
{
    LOCK_I2C_RESOURCE();
    uint16_t value = val;
    status = HAL_I2C_Mem_Write(ports_list[port].i2c_bus, ports_list[port].i2c_addr, reg, 1,(uint8_t*) &value, 2, 100);

    UNLOCK_I2C_RESOURCE();
    return status;
}

static inline int tcpc_addr_read(int port, int reg, int *val)
{
    LOCK_I2C_RESOURCE();
    uint8_t *value;
    status = HAL_I2C_Mem_Read(ports_list[port].i2c_bus, ports_list[port].i2c_addr, reg, 1, value,1, 100);
    *val = *value;
//  return i2c_read8(tcpc_config[port].i2c_info.port,
//              i2c_addr, reg, val);
    UNLOCK_I2C_RESOURCE();
    return status;
}

static inline int tcpc_read16(int port, int reg, int *val)
{
    LOCK_I2C_RESOURCE();
    status = HAL_I2C_Mem_Read(ports_list[port].i2c_bus, ports_list[port].i2c_addr, reg, 1, (uint8_t*)val, 2, 100);

    UNLOCK_I2C_RESOURCE();
    return status;
}
```

Figure 4.7: Wrapper functions achieving TCPM-TCPC communication over serial bus.

```c
 * @Converter Port & Pin
 * Sets High*/
void Enable_Converter(uint8_t PortNum){
    //For the HUB, enable the switch by pulling it HIGH
    HAL_GPIO_WritePin(ports_list[PortNum].enable_port,ports_list[PortNum].enable_pin,GPIO_PIN_SET);
}
/*Disable the Buck COnverter
 * Set Low*/
void Disable_Converter(uint8_t PortNum){
    //FOr the HUB, disable the switch by pulling it LOW
    HAL_GPIO_WritePin(ports_list[PortNum].enable_port,ports_list[PortNum].enable_pin,GPIO_PIN_RESET);

}
/*Sets the voltage output via Digital-Analog Converter
 * @PortNum and @Voltage (mV)*/
void Set_Vout(uint8_t PortNum,uint16_t mVolts){
    //formula
    float dac_value;
    HAL_StatusTypeDef result;
    /**
     *For HUB;
     *For m = -1023/18.07
     *b= -1023*20.8/18.07
     */
    dac_value =((-1023.0f/(1000.0f*18.07f))*(float)mVolts+(1023.0f*20.8f)/18.07f);
    if(dac_value > 1023){
        dac_value=1023;
    }else if(dac_value <0){
        dac_value=0;
    }
    ports_list[PortNum].port_voltage=mVolts;

    result=DAC_Set_Value(ports_list[PortNum].dac_i2c_bus, ports_list[PortNum].dac_i2c_addr, ports_list[PortNum].dac_channel,
            (uint16_t)dac_value);
    if(result){
        led_blink(PortNum, 5);
    }else{
    }
    if(ports_list[PortNum].enable_port != 0){
            Enable_Converter(PortNum);
        }
}
```

Figure 4.8: Buck converter and Digital-to-Analog converter functions.

5

Firmware and System Assessment

The implemented modular system, with adaptation layers are assessed within this chapter. This chapter includes the the state flow diagram of important decision mechanisms,of messaging sequences and that of the communication implementation. Furthermore, the data objects that are designed to be communicated within the negotiation are presented within this chapter.

5.1 Implemented Power Delivery Protocol

Implemented firmware is combination of STM32 Power delivery package and Chromium Embedded controller package, with the presented adaptation files. Power delivery core library and state machines are developed based on the chromium project and adapted to the TCPM/TCPC architecture with the STM32 hardware abstraction (HAL) and low-level layers (LL). The operating system is based on the FreeRTOS as the middle-ware component.

Within the FreeRTOS, several threads/tasks, message queues are created and managed to run in parallel with no run-time or memory issues. The threads in the operating system is handled according to their priorities. The "IdleTask" is the lowest priority within the operating system. The listed tasks are prioritized according to their expected run-time frequency and the event handling speed. Each port runs on its Power Delivery own task with 5 milliseconds intervals. These tasks run the state machines for the Type-C port controller, policy engine and protocol layers. Furthermore, the Alert task is the main task entry point handling the power delivery interrupts for each port. It interprets the input, which is the port number, as pointer into integer. This avoids tiresome memory issues. Table 5.1 lists the tasks that run on each device with their corresponding operating system priorities and stack sizes.

Table 5.1: Firmware Running Tasks

Task	osPriority	Stack Size (bytes)
Alert	Realtime	256
Power Delivery	High	256
LED Blink	AboveNormal	128
Data Logging	AboveNormal	256
Data Transmission	Normal	256

Alert task starts when there is a connection detected by the port controller. The interrupt callback puts a message into the Alert message queue, specifying the port. Any alert event like this is interpreted within the port controller and wakes up the power delivery task by putting an event message into the corresponding port's message queue. Furthermore, any events or state changes within the state machines are communicated between each other by transmission and interpretation of messages with the power delivery queue. An alert message queue is created with such a size that can handle considerable amount of alerts or interrupts. Hence the size of the queue is configured as 8 times the number of ports of any device. Likewise, the power delivery message queue for each port is configured with 16 items with 32-bit sizes.

LED blink task runs each 200 milliseconds and is a physical feedback mechanism on the devices for the user input. The functions within this task can be called from any state without any interruption of other tasks. Hence, user experience is improved. Data logging task handles the ADC for the port and the system data measurement and logging. This ADC converted values are transmitted within the data transmission task via serial UART connections. The data logging and transmission layer is elaborated further in Section 6.1.

Another important aspect of the operating system in the developed firmware is the memory management. The created tasks, queue and the semaphores within the operating system needs RAM [29]. The operating system is configured to allow the operating objects to be created dynamically. This created benefits of simplicity, automatic memory allocation and re-usage of the RAM footprint when the object is deleted [29]. Furthermore, the operating system is allocated to have a total heap size that can handle aforementioned tasks in a multi-port configuration. The heap size for the devices is hence configured as 12 kilobytes.

5.2 State Machines

In this section, the state machines which drive the application with specific decision mechanisms and state flows are represented. State flow diagram involves the transition of states in the system with the components and layers involved. The components involved are represented with the physical and transitional events that occur. Since the protocol and the system involve three layers of components, the transition between each state is shown with the interrupts events/ actions and dependent on the logic conditions.

Since the system involves many components and layers, representing the whole system as a flow diagram is rather complex. Hence it is essential to start the system representation from basic event steps. The physical scenarios are formulated from the perspective of the user hence creating a basis for the user preferences and illustration of real life scenarios. Scenarios are interpreted as the physical events that can occur on the system.

5.2.1 Base Scenario

The base scenario is when the system turns on with a photovoltaic and a charge controller or a power supply connected. In this transition, the system moves from idle state into ready state where the initialization, powering the hardware components and calibration is carried out.

At the initialization, the I^2C link between the port manager and the port controller is set. Furthermore, general-purpose input/output (GPIO) pins, alert pins with interrupt handlers, power switches are configured according to the library and project specifications. At this state, device policy manager configures the port controller device drivers and registers. Besides, for the data communication and transmission to the user and for debugging purposes, the Universal Asynchronous Receiver/Transmitter (UART) serial communication is configured. Additionally, for the real-time on board measurements and monitoring, ADC as well as DMA peripherals are configured and initialized. Furthermore, these two peripherals are calibrated for accurate readings. Port manager handles are configured with the call back functions.

When the initialization is complete, the Device policy manager sends a State ready message to the port controllers. Upon the state transitions, the events are represented to the user with messaging over the UART line and LED blinking on the system itself. Any failure on the initialization will result in interrupt pins or handlers to be called, which will put the system into the idle restart state again. The initialization and system start-up state flow is represented in Figure 5.1.

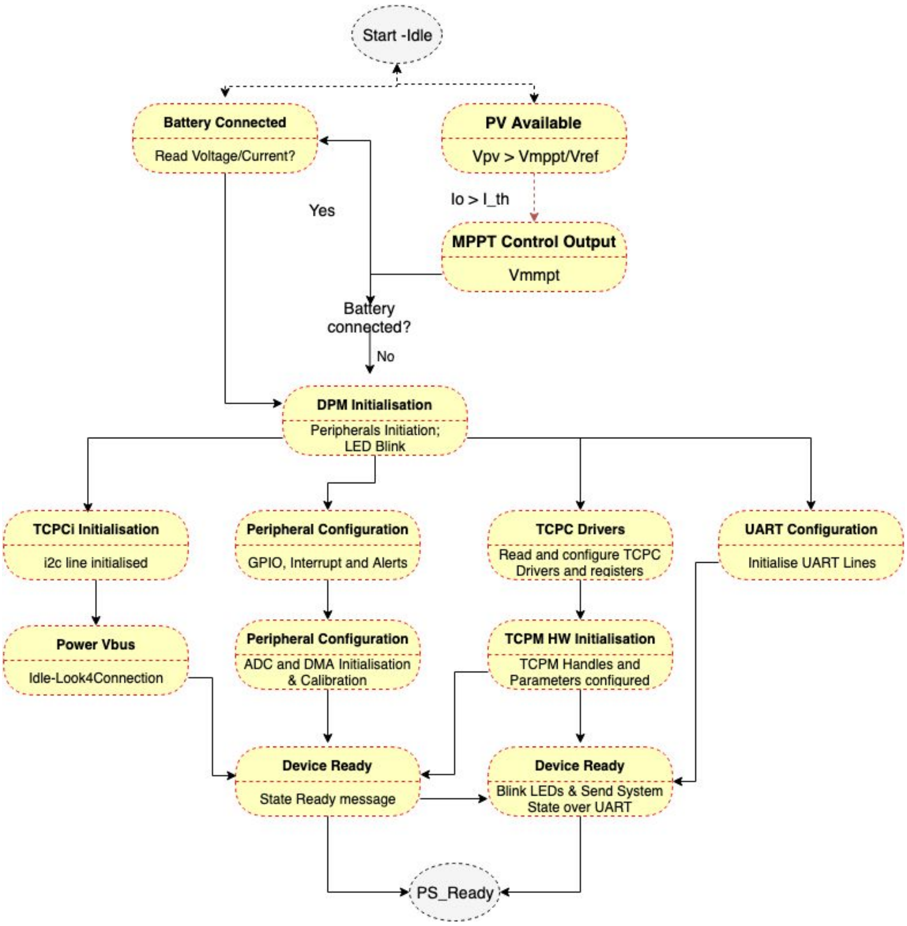


Figure 5.1: System start-up from idle state state flow diagram

5.2.2 Mode1: Fixed Supply PDO, LED Light Connection

In this scenario, a fixed voltage load is attached to one of the ports. With the cable attachment detection, power delivery is negotiated across the CC lines over the PHY Layer.

In order to start the power delivery process, the cable attachment and orientation is monitored across the CC lines. The attachment/ detachment is detected by the port controller, monitoring the voltage levels on the CC lines. When there is a LED/ a sink attached, the voltage level on one of the CC lines is pulled-down, due to the sink port partner pull down resistors. As soon as the cable detection is triggered, the power delivery is enabled. This enables the state machines for the power delivery messaging. Policy engine prepares the source capabilities; the voltage level the source can offer. This message is passed to the protocol layer to be prepared as message buffer and transmitted via physical layer. Hence, the source sends the capabilities. The source capabilities are sent as Power Data Objects (PDOs). Upon the request and acceptance, there is an explicit contract. The power negotiation is represented in Figure 5.4.

Cable attachment is detected when there is a bit wise manipulation on the CC status of the alert register of the port controller. This triggers an interrupt event, sending interrupt handlers to the device policy manager over PHY layer. After checking the status of alert register and reading the specific bits for CC and port status, the port manager clears the alert and sends an attachment message and LED blinking on the specified port. The message is delivered over UART when the state is accepted over the one of the UART lines. Device policy manager sets the associated LED of the port on, according to the source of alert and interrupt. With the accepted connection, port controller adjusts the CC line with updated pull-up resistor value, if necessary. This is followed by a CC.Status alert to port manager. Port manager reads the CC.Status and disables the Looking4Connection over the command register. Consequently, the bus voltage is set at the power status register. As soon as the connection is secured and the CC debounce time is passed, the port power controller is directed to enable the converter. This port power patch controller also adjusts the voltage output of the DAC and enables the sourcing switch . This process results in the required voltage being delivered on the Vbus. Since the LED is not capable of further power delivery negotiation, the LED load will be turned on with the default output voltage, 5V fixed. Furthermore, the port LED is turned on as feedback for the user. Continuous bus voltage monitoring is achieved via port controller dedicated register reading. Likewise, port current monitoring is done over the ADC peripherals with DMA. This data will be sent to user over the UART. The state flow and decision mechanism is represented in Figure 5.2.

5.2.3 Mode2: Fixed suply PDO, LED Disconnection

Upon disconnection, it is expected that the Source Port controller senses the disconnection over the CC register. Type-C state machine within the operating system monitors the status of these registers each run time period (5 milliseconds). Once there is a disconnection, port controller sets an alert about the CC.State and CC.Status. It also sends a message packet to port manager. The port manager reads the alert with the CC.Status and determines the disconnection. This changes the state of the Type-C state machine and directs the power path controller and port controller to change their own states too. The power path controller disables the switch and port controller enables discharge on Vbus. Furthermore, the DC-DC converter is disabled and the voltage output is set to reference voltage value, resulting at 0V. Another Alert is sent to port manager about the Vbus disconnection, that disables the Vbus over the Command register. Port manager sets the corresponding LED of the port to blink, via GPIO. Furthermore, port manager disables the monitoring. Consequently, the status of the system is updated on the port controller. The state is shifted back into looking for CC connection. The disconnection state flow of the system is illustrated in Figure 5.3.

This state flow is handled in the same manner for all the ports, when they are configured as source (SRC). Moreover, the state flow and decision mechanism is same for all the fixed supply power delivery objects, with 5V and 3A. In the rural electrification field test case, this manner would be expected in the system when a LED or a DC Fan is connected or disconnected.

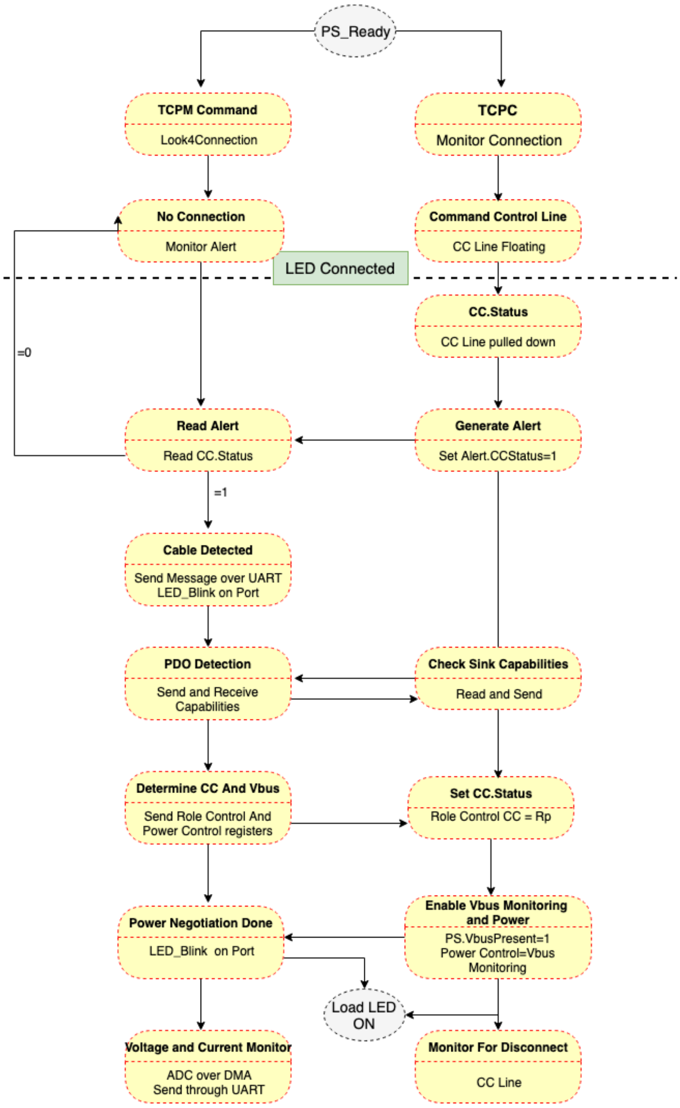


Figure 5.2: LED connection and powering state diagram

5.2.4 Mode3: Powerbank, Fixed PDO Power Negotiation

In this scenario, the powerbank that acts as a fixed PDO sink is connected to one of the source ports on the charging station. Both ports are capable of the USB Type-C Power delivery firmware and hence able to negotiate different voltage level. Powerbank port 0 is the high power port with bi directional power flow. Hence, this port will act as a sink (default) and will request voltage level in order to establish charging either directly or via boost converter. The state flow diagram implemented for this scenario includes the state machines for the Type-C port controller and the policy engine state machine from the source side.

Firstly, the power negotiation state flow is represented between sink and source. Power delivery negotiation is carried out exchanging messages between the source and sink physical and protocol layers. The message flow is illustrated in the Figure 5.4. As illustrated, SRC with the data role of DFP is always the initiator of the power delivery messaging. Power delivery messaging starts with the source sending the capabilities of power data objects to the sink. If the sink does not receive this, then it asks for hard reset. When the source capabilities are received by the sink, it evaluetes the capabilities, chooses the one accordingly and requests it as request data object. The source evaluates this request and if it is one of the offered power level, it switches to the required power. As soon as this happens, the source sends control message of "PS RDY", stating that the explicit contract is set.

On the other hand, in a source, the Type-C state machine before the connection is in TC Unattached SRC state. This state is the following state after the initialization process is complete. In this state, port controller monitors the attachment. As soon as a pull down resistor is connected on one of the CC lines, the state is changed into TC AttachWait SRC. Here the cc state is set to UFP attached. Once a permanent connection is achieved, in other words the tick count on the timer is more than the debounce time of CC lines, state is changed to TC Attached SRC. In this state, data role and polarities are set. Furthermore, the power delivery negotiation is enabled by setting and sending a flag to the policy engine state machine.

As soon as the policy engine receives a pd enabled flag, it sets its state into SRC StartUp. Within this state, the protocol layer is reset to make sure the timers, counter and CRC management are done clearly. The flags are cleared and once again the power role is checked with the data role. The next state is send capabilities, where the source send the power data objects, depicting the power levels it can supply. As soon as the goodCRC is received and checked by port controller, next state is transited. The next state is negotiate capabilities state, in which stating that the power delivery connection is established as soon as a request is received from the sink. If the requested object matches one of the power data objects the source sent, the state is transited into transition to supply. Within this state, the new power level is achieved by setting the voltage level through DAC over the path controller driver. Meanwhile state machine sends ACCEPT message. Once sink acknowledges the sequence with GoodCRC, the source sends a PS RDY control message, signifying the explicit contract is achieved. This concludes the power negotiation unless there is a new power level requested by sink or if there is role swaps or disconnection. Policy engine gets into SRC Ready state to listen for incoming messages or any requests from the device policy manager. The state flow of this mode is illustrated in Figure 5.5.

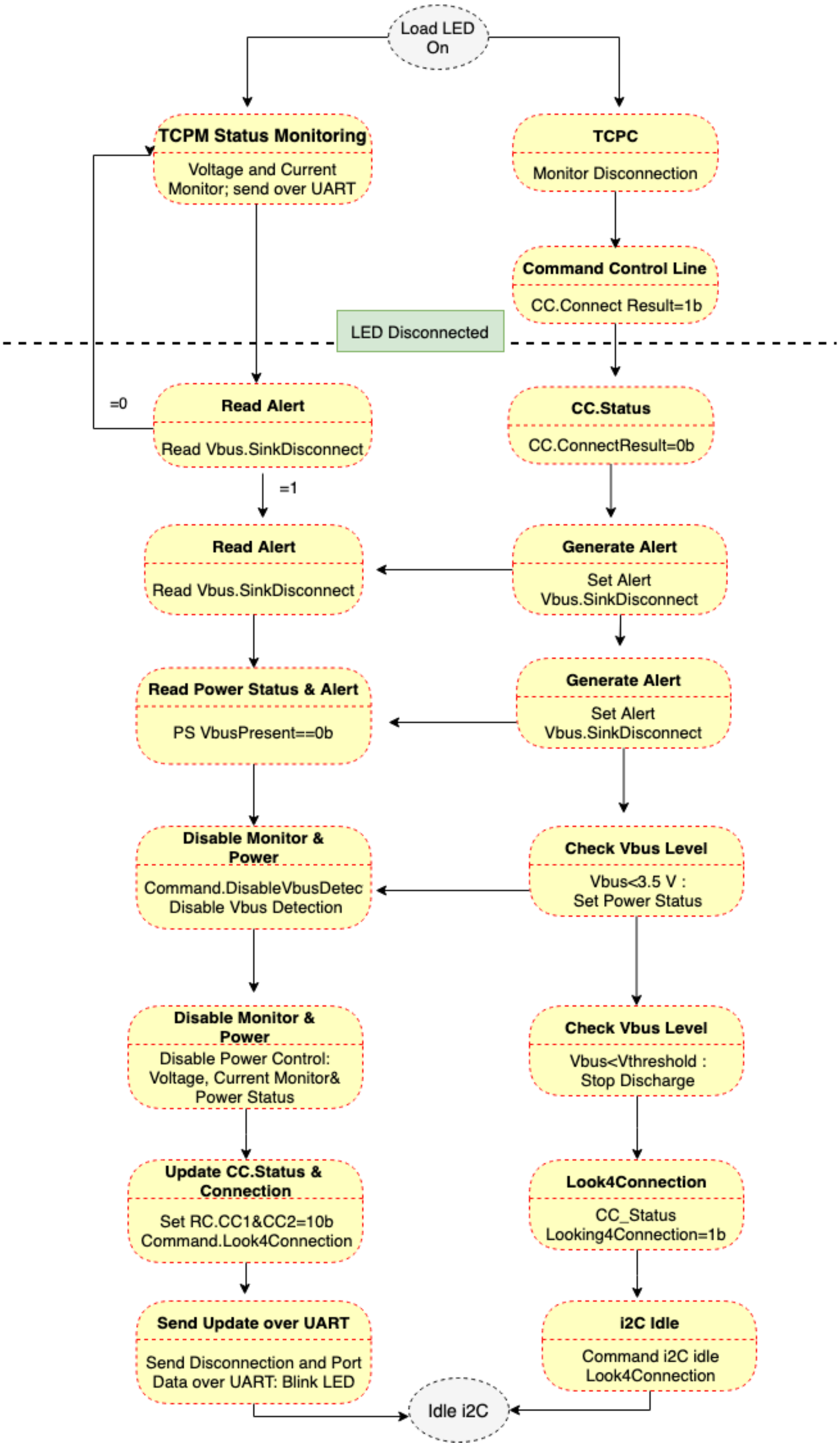


Figure 5.3: LED disconnection state flow diagram

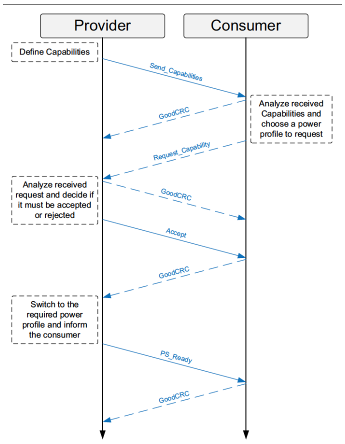


Figure 5.4: Power delivery message sequence[20]

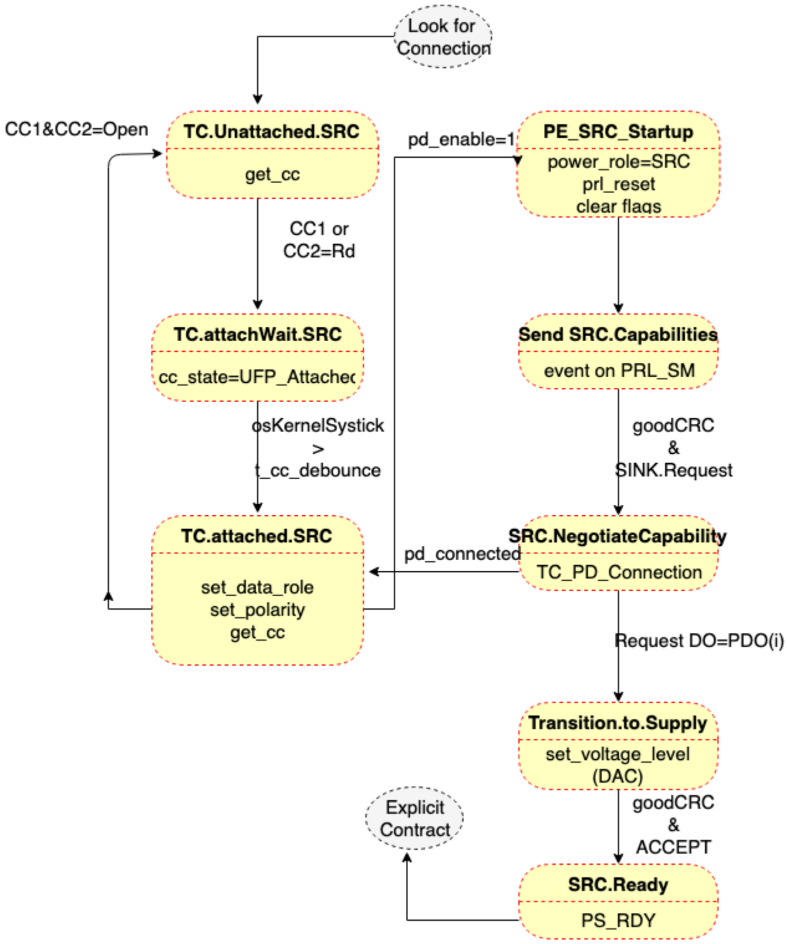


Figure 5.5: State flow diagram of a Source when battery connected, illustrating port controller and policy engine states.

5.3 Data Objects and Data Messaging

In this section, the buffer messages sent between port partners are evaluated. The buffer message packet consist of header and data objects. Data objects can be classified as power data object, request data objects, vendor defined messages. Vendor defined messages are either implemented as structured vendor defined messages or unstructured vendor defined messages. Unstructured vendor defined messages are the self-implemented messages that assign characteristics to the ports, such as their priority, desired voltage levels and time of charging. These will be used after the explicit contract to implement desired energy management application.

5.3.1 Data headers

Data headers are 16-bit message packets that contain basic information about the message and the power delivery port capabilities. It can be used as a standalone control message in the PD atomic messaging sequence when number of data objects field is set to zero or as the first part of the data message when the number of data objects field is non-zero. Table 5.2 illustrates the defined header with correlated bit definitions for a Source and a Sink. In this example, the header for Source represents the header of a Source capabilities message whereas for the Sink it represent the header of a request message.

The *"Extended"* bit is always set to Zero in this application, that defines the message either as control message or data message. The *"No of Data Objects"* represents the number of 32-bit data objects within the package. Control messages that are used to manage the message flow have the number of data objects field set to zero. On the other hand, if the message involves the data messages and the data objects, this field is set to number of corresponding data objects within the buffer. *"MessageID"* field is automatically generated value by a counter from the Physical layer. This field is used to check the successful transmission/reception of the message. For this example, both values are One since both are the first messages initiated from the port's protocol layer. Power role field indicates if the port is a Sink (Zero) or Source (One). *"Specification Rev."* field represents the port's supporting capabilities of the USB Power Delivery specifications. The devices and software is implemented in such a way that they support the latest specification Revision 3.0, which is represented as *10b* in 2-bit field. *"Data Role"* field presents the port's data role, either Zero for UFP/Sink or One for DFP/Source.Finally, *"Message Type"* presents the type of corresponding message. The message type varies for control and data messages. In this example, the message type for the source capabilities message is set to *0001* and for the source request message it is set to *0010*.

Table 5.2: Message Header

Bit(s)	Field Name	SRC Capabilities	SNK Request
15	Extended	0	0
14..12	No.of Data Objects	1	1
11..9	MessageID	1	1
8	Power Role	1	0
7..6	Specification Rev.	2	2
5	Data Role	1	0
4..0	Message Type	1	2

5.3.2 Power and Request Data Objects

The power and the request data objects are the data messages in which the message header sets the number of data objects field to non-zero. These messages represent the source power capabilities or the sink's power requirements.

An example of a fixed power data object that conveys a well-regulated fixed power supply is represented in Table 5.3. The source presents that it is capable of supplying 12 V. *"Fixed Supply"* PDO is represented as 00b. For the implemented case, the bits 29-25 are set to Zero, conveying the static capabilities. *Peak Current* field represents the overload capabilities of the source in case there is a short burst of power activity required. The current provided is set to meet the Operating Current I_{OC} field requested by Sink. In this application, this field is set to Zero regardless of the devices that will be connected. This is due to the fact that no overload or short bursts of activity is expected. *Voltage* field is the bit-wise representation of the advertised voltage level in *50 milli Volts units*. Likewise, *Maximum Current* field is the bit-wise representation of maximum current that port can supply in *10 milli Ampere units*.

Table 5.3: Fixed supply power data object

Bit(s)	Field Name	12 V Fixed Supply
B31..30	Fixed Supply	0
B29	Dual-Role Power	0
B28	USB Suspended Power	0
B27	Unconstrained Power	0
B26	USB Communications Capable	0
B25	Dual-Role Data	0
B24	Unchunked Extended Messages Supported	0
B23...22	Reserved	0
B21...20	Peak Current	0
B19...10	Voltage (50mV units)	240
B9...0	Maximum Current (10mA units)	150

In response to the source capabilities message, the sink responds with a request message. The request data object is formed within the message with the data header. This request data object should correspond to the chosen source power data object and should include the requested power level. An example request data object is represented in Table 5.4 with the required field names.

Table 5.4: Request data object

Bit(s)	Field Name	Requested 12 V
B31	Reserved	0
B30..28	Object Position	3
B27	Giveback flag	1
B26	Capability Mismatch	0
B25	USB Communications Capable	0
B24	No USB Suspend	0
B23	Unchunked Extended Messages Supported	0
B22...20	Reserved	0
B19...10	Operating Current (10mA units)	50
B9...0	Maximum Current (10mA units)	150

The *Object Position* indicate the chosen PDO advertised from Source. The advertised PDOs are listed in the order of lowest voltage level to the highest, with 5V fixed supply always being the first PDO. In this case, assuming that the source advertised PDOs with voltage levels of 5V, 3V, 12V, 15V and 20V, the field is set to Three, representing 12V PDO. *Giveback Flag* field presents the capability of Sink; responding to the request of Source and reducing its load to minimum operating current. The bits 26-20 are set statically to Zero for the implemented application. *Operating Current* presents the bit-wise value of the actual amount of current that Sink needs to operate in whereas *Maximum Current* represents bit-wise value of the highest current that Sink can require during operation, in *10 milli Ampere units*.

5.3.3 Vendor Defined Messages

Vendor defined messages convey additional information about the devices and allow the devices to use alternate modes that are supported by USB Type-C specification, such as display port. Vendor defined messages are implemented, if needed, after the power delivery negotiation. Hence, they are initiated in the condition of an explicit contract is established. In implemented application, vendor defined messages are used to exchange information on the specified Vendor ID, Product ID of the devices. These vendor defined messages are structured VDMs with commands to enter the alternate mode. The alternate mode in this case is custom mode defined for the test case application. Custom mode is specified to convey information on the priorities of the loads.

5.3.4 Unstructured Vendor Defined Messages

Unstructured VDMs are self-developed VDMs in order to exchange extra information about the loads connected. Unstructured VDMs are custom developed and initiated when the partners agree to enter an alternate mode. The U-VDMs are initiated by DFP/source port, requesting the priority levels of the sink. Unstructured VDM header is a 16 bit data packet and is illustrated in Table 5.5. *"Vendor ID"* is the specified ID by the vendors. In this case, since the devices are in the prototyping phase, the STM Microcontroller Vendor IDs are used. *"VDM Type"* presents if the VDM is structured (ONE) or unstructured (ZERO). The bits 14-2 are kept reserved for future use.*"Priority Level"* field presents the bit-wise priority value for that specific load.

The DC-DC adapters are implemented with their own specified operating voltage, current and maximum current, that depend on the connected load. Consequently, according to the connected load, the priority value is specified initially and exchanged within this data buffer.

Table 5.5: Unstructured VDM Header

Bit(s)	Field Name	Load Response
B31..16	Vendor ID	0x0483
B15	VDM Type	0
B14..2	Reserved	0
B1..0	Priority Level	0-4

The state flow diagram of ports with agreed explicit contract that enter to the custom mode is illustrated in Figure 5.6. When the explicit contract is achieved, the source device policy manager commands the policy engine to initiate the vendor defined messages. After the successful exchange of Vendor IDs, the source asks for the supported modes of the sink. If the custom demand response alternate mode is one of the modes that are supported by the sink, the source requests to enter this mode. Upon the reception of acknowledgment with valid SVID and matching mode, alternate mode of demand response is entered. Next, the DFP/source initiates the process of this mode by requesting the priority level. That is by sending the unstructured VDM header with VDM Type=0 and priority level NULL. The UFP/sink responds with the same structure of header with its specific VID and the priority level. The policy engine of DFP passes this information to device policy manager for processing and waits for the next command. Likewise, the UFP/sink policy engine shifts into SNKReady state for waiting the next message from port partner or from its device policy manager.

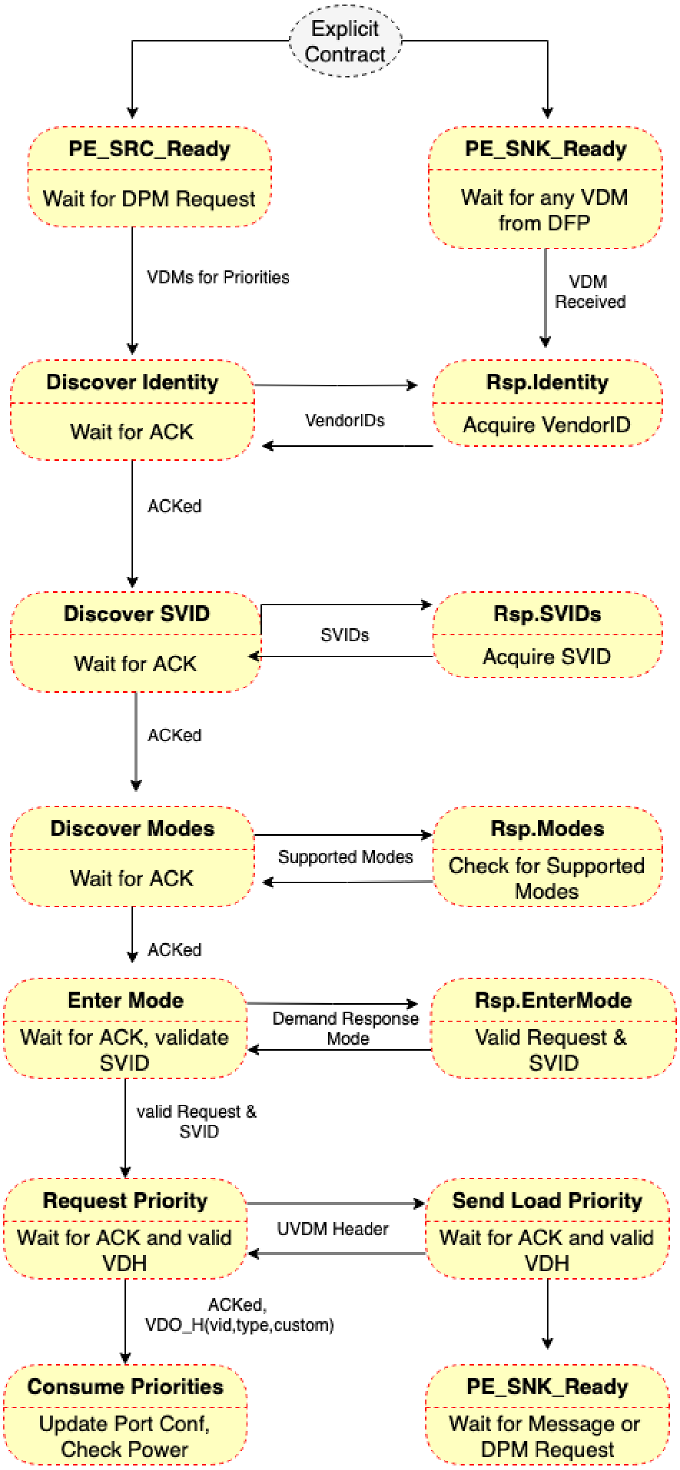


Figure 5.6: Port partners Policy Engines Entering Alternate Mode

6

Test Case and Experimental Results

The data monitoring layer is firstly elaborated within this chapter. This presents the methodology behind the data acquisition. Secondly, the implementation of test case and energy management is discussed. Finally, this chapter elaborates and discusses the experimental results of the test cases formulated. The results are solely based on the laboratory tests. The key outcomes with measurements and observations are presented within this chapter.

6.1 Data Monitoring Layer

Data monitoring layer is implemented within the device policy manager and deployed as one of the tasks that run in parallel within the operating system. The monitored data is divided into two, dependent on the application and the end-use of it. One is the data that is monitored and transmitted continuously. It is the set of parameters acquired from the system and indicates information on electrical and physical properties. Second buffer is the data about that indicates the information on current state of the ports. This includes the information about initialization process, any conditional changes on faults and possible alerts. This is the other set of data which is monitored continuously, however transmitted as message buffer once they are called.

Continuous data acquisition is achieved by the ADC peripheral, that is linked with DMA peripheral. Port current, system voltage and MCU temperature are continuously monitored with specified sampling time. ADC is configured to run on asynchronous clock mode with 12-bit resolution, where it handles the conversion continuously. DMA peripheral linked to ADC runs on circular mode, increments the memory address each time. The data width saved into memory and peripheral is both half-word (16-bit). The sampling time for all the channels to be converted is taken as 160.5 clock cycles. Hence the conversion time is the sum of processing time (12.5 clock cycles) and sampling time (160.5 clock cycles), resulting in 173 clock cycles. With the ADC clock running on 16 MHz, the time of conversion is 11 microseconds. This, in the hub with 12 configured channels, total into 132 microseconds or 7.57 kHz.

The configuration and initialization of the channels are achieved at the startup of the system. This is followed by the conversion within the data logging task. The converted values are saved into the *ADCBuffer*, that is updated at each conversion cycle. The digital values are converted into voltage, temperature or current values following the calculation script, illustrated in Figure 6.1.

```c
#define RANGE_12BITS                    ((uint32_t) 4095)    /* Max digital value with a full range of 12 bits */
/*ADC Parameter*/
#define ADC_BUFFER_SIZE                  12  /* Size of array containing ADC converted values: set to ADC sequencer number

/*Current MEasurement*/
#define RSense 0.001f
#define Gain 50.0f

/*Vsys measurement voltage divider value*/
#define V_DIVIDER ((float)  11.0)

/* Internal temperature sensor: constants data used for indicative values in  */
#define INTERNAL_TEMPSENSOR_V30        ((int32_t) 1430)         /* Internal temperature sensor, parameter V30 (unit: mV).
#define INTERNAL_TEMPSENSOR_AVGSLOPE   ((int32_t) 4300)         /* Internal temperature sensor, parameter Avg_Slope (unit
#define TEMP30_CAL_ADDR   ((uint16_t*) ((uint32_t) 0x1FFFF7B8)) /* Internal temperature sensor, parameter TS_CAL1: TS ADC
#define TEMP110_CAL_ADDR  ((uint16_t*) ((uint32_t) 0x1FFFF7C2)) /* Internal temperature sensor, parameter TS_CAL2: TS ADC
#define VDDA_TEMP_CAL                  ((uint32_t) 3300)        /* Vdda value with which temperature sensor has been cali
/*Internal Voltage REference*/
#define VREFINT_CAL     ((uint16_t *) ( 0x1FFFF7BA))

  * @brief  Computation of temperature (unit: degree Celsius) from the internal
#define COMPUTATION_TEMPERATURE_STD_PARAMS_AVGSLOPE_V30(TS_ADC_DATA)          \
  ((( ((int32_t)((INTERNAL_TEMPSENSOR_V30 * VDDA_TEMP_CAL) / VDDA)      \
       - (int32_t)(((TS_ADC_DATA) * VDDA) / RANGE_12BITS)               \
      ) * 1000                                                           \
    ) / INTERNAL_TEMPSENSOR_AVGSLOPE                                     \
   ) + 25                                                                \
  )

  * @brief  Computation of temperature (unit: degree Celsius) from the internal
#define COMPUTATION_TEMPERATURE_TEMP30_TEMP110(TS_ADC_DATA)                   \
  (((( ((int32_t)((TS_ADC_DATA * VDDA) / VDDA_TEMP_CAL)                 \
        - (int32_t) *TEMP30_CAL_ADDR)                                    \
     ) * (int32_t)(110 - 30)                                             \
    ) / (int32_t)(*TEMP110_CAL_ADDR - *TEMP30_CAL_ADDR)                  \
   ) + 30                                                                \
  )

  * @brief  Computation of voltage (unit: mV) from ADC measurement digital
#define COMPUTATION_DIGITAL_12BITS_TO_VOLTAGE(ADC_DATA)                       \
  ( ((ADC_DATA) * VDDA) / RANGE_12BITS)


/*v_sys Calculation (mV)*from ADC measurement digital value on 12 bit range
#define COMPUTATION_DIGITAL_12BITS_TO_VSYS(ADC_DATA)          \
((((ADC_DATA) * VDDA) / RANGE_12BITS) * (uint32_t)V_DIVIDER)
```

Figure 6.1: Manipulation of digital values read by ADC.

The converted values are saved into the structure as the corresponding parameters within the data logging task. This continuous measured and saved data is passed on to the data transmission layer. The data transmission handles the continuous transmission of data to the developer via Universal Asynchronous Receiver/Transmitter (UART) layer within the micro controllers. This transmission/reception is configured to work with baud-rate of 115200 bits per second. UART transmission and reception is achieved with the JTAG programmer connector over the serial data lines. In addition to the continuous data buffered, the buffer data from the events and the state changes is transmitted via UART layer. The UART transmission is run on the data transmission task that runs with a second intervals. Hence, each second the ADC converted port and system measurement data as well as the state, event or fault information is feedback to the developer. The monitoring of this data is achieved by configuring the reception with baud-rate and printing it on the terminal screen.

Besides the system parameter measurements and states information, power delivery atomic message sequences are monitored. The monitoring of the messaging of power delivery and vendor defined messages are acquired with the STM32 USB Type-C PD Monitoring tool [30].

6.2 Case Study Formulation

This section presents the initial configuration and steps taken to formulate the case study to be presented within the project. The case study is taken as a means to prove the firmware topology designed and showcase the behaviour of the system working with the power delivery characteristics, messaging and data monitoring.

As part of the case study, the charging hub is used as the main source of the system. The charging station at this point could not be involved within the case study due to the hardware issues. The hardware issues resulted in unsuccessful implementation of the software. Hence, the hub is taken as the power and data transmitter within the system. Along with the hub, the designed power banks, LED lights and DC-DC adapters with connected loads are utilized.

6.2.1 Selection and Prioritising of DC Loads

The loads that are utilized in the system are prioritized according to the studies made in the Section 2.1.3. It is found out that the LED is the most necessary load in rural areas for the citizens. Even though the LED lights do not possess MCUs to be programmed, they are chosen to have the highest priorities within the energy management decision making. Hence the device policy manager of the hub assigns the highest priority to the ports which are connected to a port partner without power delivery negotiation.

Furthermore, the second highest need in the electrification and usage of loads is found out to be mobile charging. In the study case, the power banks are taken as the representative of mobile charging since they also involve batteries and can be used remotely in mobile conditions for several electronics charging/utilization. Thus the power banks are configured to have the second highest priority and transmit this information to the source port when requested.

Thirdly, low power consuming electronic device is chosen to be utilized with the DC-DC adapter. According to the load categorisation survey, a TV, a speaker or a radio that functions on DC and has low power ratings can be taken as the next priority load. For this purpose, a speaker that functions with 5 V rated characteristics is utilized and connected to the DC-DC adapter. The micro controller is configured to store the information of these characteristics in addition to the priority level chosen. When requested, the priority level is transmitted via PHY layer to the hub as unstructured vendor defined messages.

6.2.2 Energy Management Implementation

Energy management layer is implemented in the hub within the device policy manager. It is implemented in such a way that, the connected port partners first initiate the power delivery negotiation with the agreed voltage level. Once the alternate mode is entered and the hub acquires the priority of the connected load, the policy engine passes it to the device policy manager to be saved in the port configuration structure. The state flow diagram of the decision making process, illustrating the states of the policy engine, device policy manager and Type-C state machine of the port to be switched off is represented in Figure 6.2.

The data logging layer that runs in parallel with other tasks gets the current in that specific port. Furthermore, within the power delivery task, the Type-C state machine monitors the bus voltage delivered on the port and passes it to the device policy manager. These measurements get updated. The "demand response" function handles the port switching. In order to manage that, the data buffered from data logging task on the port power and total power consumed within the hub is accessed. The maximum power that the hub can deliver is also acquired.

The maximum power the hub can deliver is dependent on the power rating of the power supply in the test case. If the total power that is/will be supplied is higher than the power that the hub can deliver, the load switching will be active. The ports that have the priority data with the highest value in range of 0-4, will be switched off. The device policy manager passes this information to the Type-c state machine of that port. The port power path controller is directed to disable the switch so that that port's bus supply is switched off. The device policy manager checks the total power consumed once again to check if it is still excessive. If that is the case, the next port with the lowest priority, in other words the highest priority value, is switched off. Otherwise, the port with the highest priority or the recent connected port partner is offered with the port capabilities.

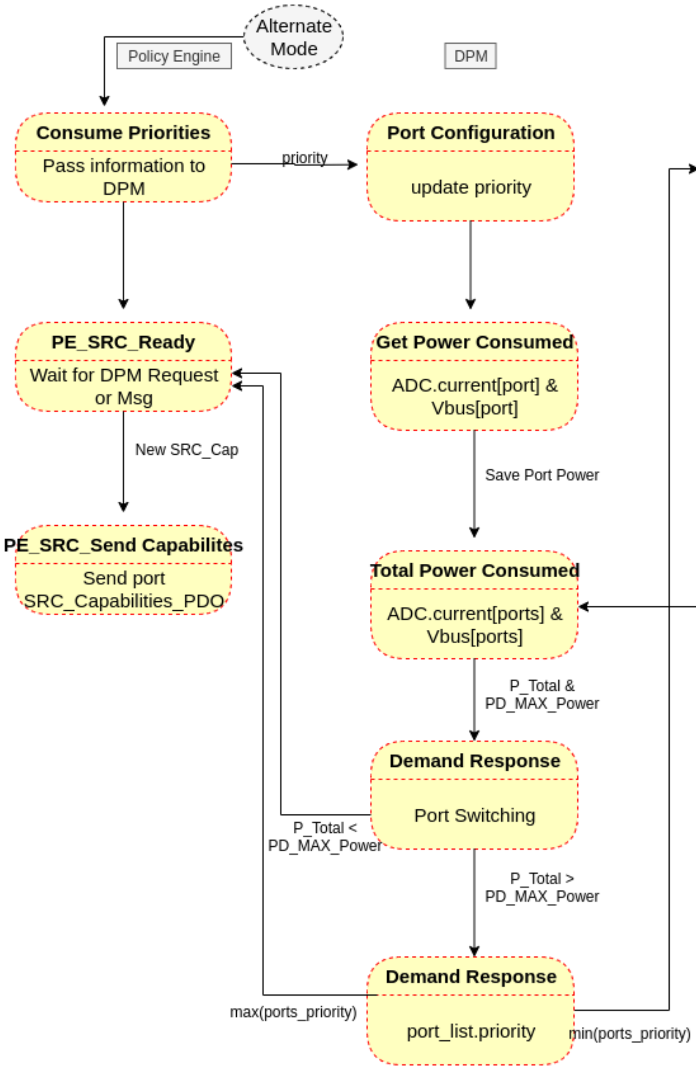


Figure 6.2: Load switching decision pattern and state flow.

6.2.3 Pilot Project and Case Study

In line with the project goals, the system setup with the designed and developed firmware will be tested in the field. Although it was planned to implement the pilot project in a rural village in Tigray region, close to Mekele city in Ethiopia, during the summer period of 2019, the pilot project is delayed to March 2020. The planned location of field test is shown in Figure 6.3 This was mainly due to project management and prototype debugging. Since the prototypes had been manually assembled, designed by young engineers/ students, this process had been longer than expected.

The pilot project and field test will give considerably valuable information on many aspects of the future of the project. From technical perspective, during the field test, data collection on user

Figure 6.3: Rural village in Tigray region, Ethiopia, where the field test will be taken.

preferences on electronics will be acquired. This will create a good basis for creation of a load profile and priorities, that will be useful for reliable and scalable energy management firmware implemented on USB Type-C based systems. Furthermore, the technical performance of devices, mainly the powerbanks and the hub will be assessed. The data acquired during field test on the actual time taken for charging the powerbanks, the period of usage, the energy performance of these prototypes will shed a basis for validating the system.

The case study implying the expected system behaviour in the field test, was planned to be deployed in laboratory conditions. However this could not be implemented neither. The planned testing deployment on the hub could not be achieved since not any more than one prototype port of the hub gave satisfactory results on neither the cable attachment, power delivery nor the atomic messaging sequence. The main reason behind this was found out to be hardware related issues. Mainly the first iterated prototype and power circuit board of the hub had some wrong footprints such as the port connectors. Besides, the excessive amount of both hardware and software debugging resulted in deformation of the components.

Additionally, the prototypes of the powerbank had similar issues on hardware. The issues are resulted from the boost converter not being able to be functioning nor enabled for charging from the hub. Hence the batteries could not be used in coupled to the hub for case study testing.

These unforeseen issues led to a change in the format of the case study. Rather than testing the components as a whole system, the prototypes and the power delivery sequence have been tested individually. This resulted in inability of testing or showcasing the energy management layer within the firmware. This layer is left to be tested for the future iterations of the prototype hardware. This hardware will be implemented by outsourcing firms, so that the assembly and soldering will not create any hardware issues. Hence, a reliable test environment in which the software performance can be monitored, will be achieved.

6.3 Laboratory Test Setup

In order to validate the software and the objectives set for the thesis, the altered test case in the laboratory is set up. Although this verification and validation process is not the definite one, it is a former and preliminary step for the later design and development iterations of the system.

In accordance with the aforementioned issues on the case study, the system formation had to be adjusted for testing. Hence the laboratory test set up was implemented in several steps. The first step has been testing the system performance with the charging hub as a source and powering up a load with steps of voltage. The load was the DC-DC adapter that required several voltage levels. Secondly, the charging and discharging sequence, that is deployed simultaneously on the power bank, has been tested. As a load, a mobile speaker with 5V input voltage has been coupled with DC-DC adapter for power negotiation. The power supply for both cases has been DC power supply. In the first case, the hub has been supplied with 24 V and 1.5 V, with constant voltage, delivering maximum of 36 W. That was observed to be adequate for this test setup since only one port could have been used. On the one hand, for direct charging of the power bank, the output of power supply has been set to constant current of 1.5 A and maximum voltage of 16.8 V.

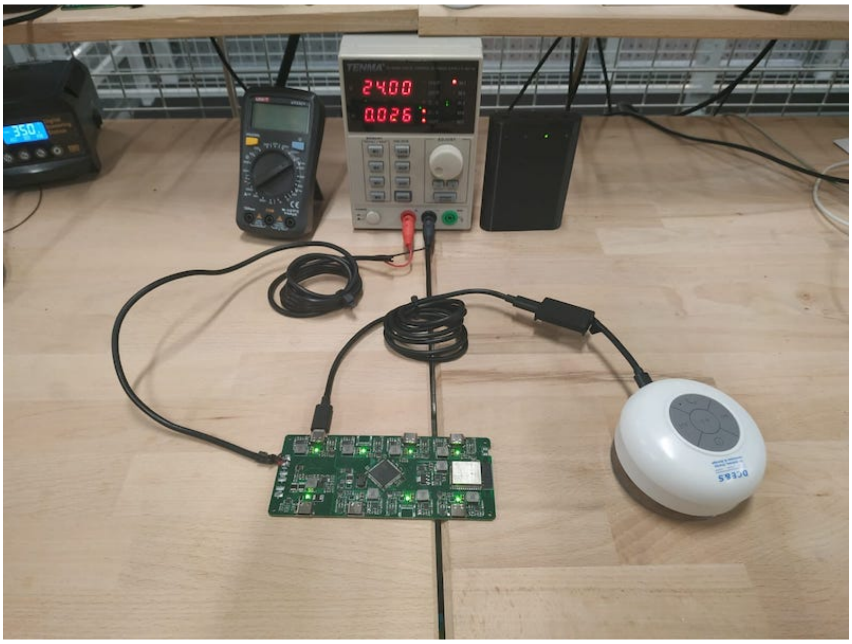


Figure 6.4: Test setup illustration

6.3.1 Power Negotiation with DC-DC Adapter

First step in the test has been checking the power output capabilities of the DC-DC adapter when connected to the hub source port. For this test, the DC-DC adapter has been configured to request input voltage of 19 V and give output of 19 V. Likewise, the hub port is configured to offer source capabilities ranging from 5V to 20V with 1.5A. The fixed source capabilities power delivery objects are used for the power delivery negotiation.

The measurements are acquired from the data logging and data transmission layer of the DC-DC adapter. Th transmission of measured data has been with 100 milliseconds time intervals. The logged data of input voltage and output voltage is illustrated in Figure 6.5.

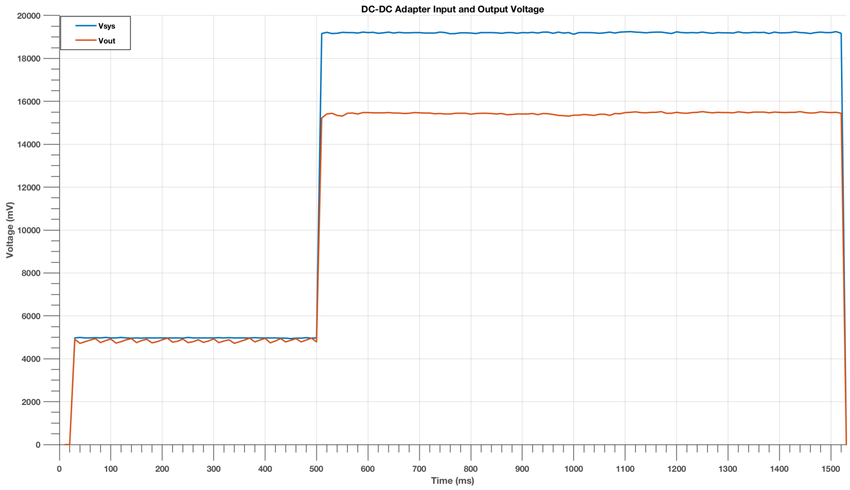


Figure 6.5: Input (blue) and Output (red) Voltage of the DC-DC Adaper. Within 50ms, the adapter is powered with 5 V. This enables the converter and hence the output voltage is set to 5 V. After 350 milliseconds of delay, power negotiation starts and port partners agrees on 19V. This is observed on the system input voltage of the DC-DC adapter,(blue line).

When the DC-DC adapter connected, the default 5V voltage is provided on the bus voltage of the USB Type-C port. In order to get a clear observation of voltage switching pattern, there was a delay of around 350 milliseconds after the PD switch is turned on. As soon as the policy engine of the devices have started, the negotiation has started and DC-DC adapter requested 19 V from hub port. The DAC on the corresponding port has changed its output to 19 V. As soon as the input voltage of the DC-DC adapter stepped up, its output voltage has stepped up as well. Even though, the DAC output of the DC-DC adapter was configured to give its maximum output (within the range 256 bits), it resulted on output voltage level of around 19.2 Volts. This limitation was found out due to the switching sequence of the buck converter within the circuit. In the buck converter, there were two switches involved and they are continuously switching, one being low side and one being high side. At this test, the buck converter switch was not enabled hence, it resulted a substantial offset.

The next step is to capture the bus voltage on the hub sourcing port. The power delivery negotiation has also been captured during this process, that illustrates the message exchange between two port partners. Resulting from the negotiation, the explicit contract has formed in the agreement of providing/ acquiring 12 V with 1.5 A maximum. The bus voltage measurement has been carried out by the Oscilloscope probes. The resulting voltage measurement during the connection and negotiation is illustrated in Figure 6.6.

Furthermore, once the connection is established and the policy engine and protocol layer state machines are enabled, the power negotiation is started. Figure 6.7 shows the messaging sequence for this test. It is noted that the power negotiation occurs and established within tens of milliseconds. The voltage switching occurs with the DAC changing its output voltage as soon as the Accept message is sent by the sourcing port. However, there is an anomaly observed in the bus voltage pattern in Figure 6.6. Just before the voltage switching and the negotiation started, there is a drop of around 2 V in the bus voltage within 50 milliseconds. It is found out that the source policy engine state machine implemented a hard reset. Hard reset is implemented since there was no reception of message from port partner, and that is carried out in order to reset the electronic components for both devices. However, it is observed that the converter did not get disabled within this considerable amount of time and it resulted with its minimum output voltage on the bus, which is around 2.7 V. After this reset, the default 5V is initiated, which follows by the power negotiation and switch to 12V on bus voltage.

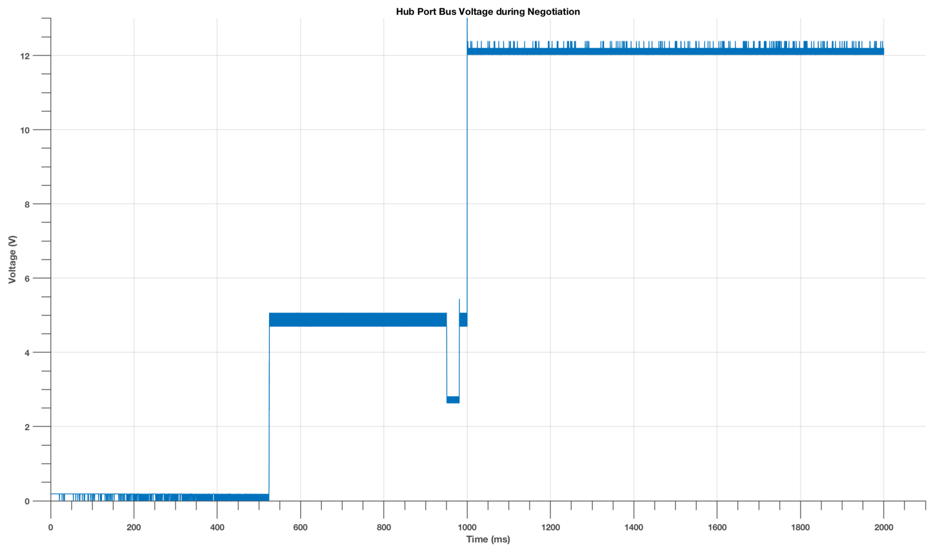


Figure 6.6: Port bus voltage during connection and power negotiation of the Hub. As soon as the hub is powered up and connection is detected , there is 5V delivered on the bus voltage. Apart from the anomaly of sudden drop on bus voltage, the power negotiation results in output of 12 V on the bus voltage once the port partner agrees.

The final step on this test case has been monitoring the CC lines during the connection and power negotiation. The bus voltage, CC channel 1 and CC channel 2 voltage levels are illustrated in Figure 6.8. Considering that the power negotiation and explicit contract happened within around 15 to 20 milliseconds, it is expected to capture the CC line voltage changes accordingly. In unconnected conditions, the CC lines (when configured as pull up resistors) result in 3.3 V in voltage level. The connection is detected on the port controller of source side when this voltage level drops down. With the sink pull down resistors, the pulled-down value is expected to be around 1.1 V on one of the CC Lines. On the other line, since the Vconn is not enabled, it will stay at 3.3 V. In this connection, the CC line 1 (Blue) is pulled down and that is the one is also used for power delivery atomic message sequencing.

AMS in power delivery over CC lines in this case has followed 8 packets of data exchanged. According to the voltage fluctuations on corresponding Configuration Channel Line, the same packets of data has been found out to be 8. However, it is observed that there has been many noise. It is noted that All PD messages are transmitted at 300KHz +/- 10 over the CC line. Furthermore, this communication uses Bi-phase mark coded signalling. According to this coding scheme, the cable inductance and source output impedance should be within a range as shown in Table 2.3. Source output impedance is determined by the driver resistance and the shunt capacitance of the source [20]. Thus it is a frequency dependent term. Indeed, this parameter impacts the noise ingression in the cable. This can be one of the main reasons behind the noise observed on CC line.

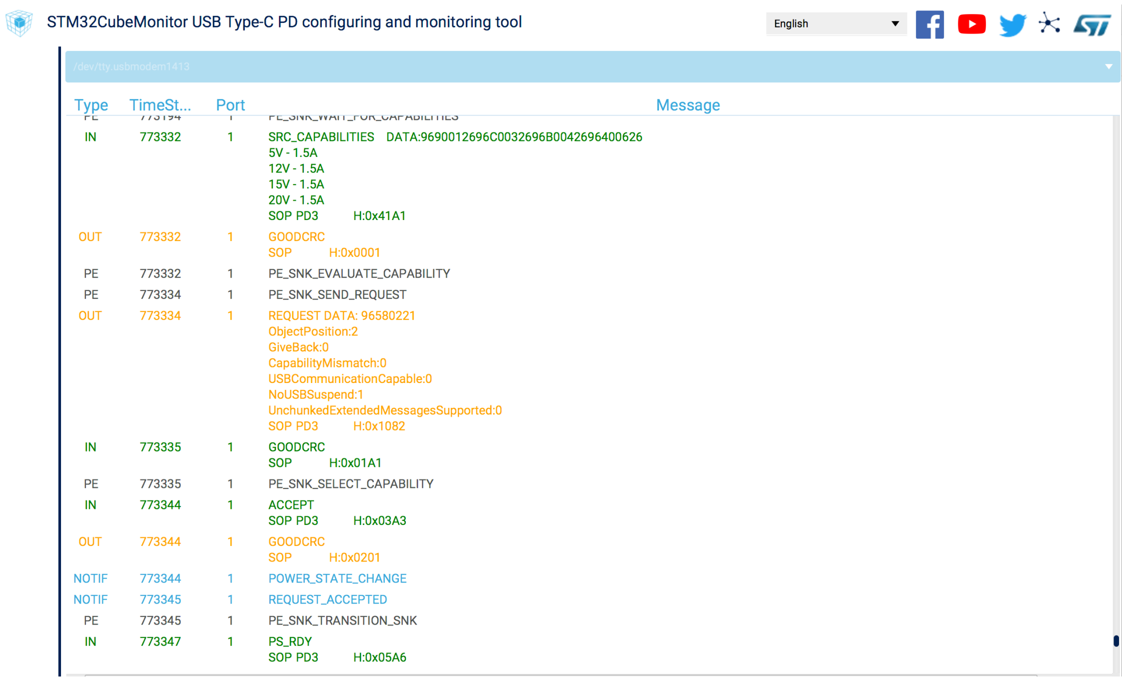


Figure 6.7: Power negotiation with explicit contract established, captured by the tool STM32Cube MonitoringTool; the data packets and timeseries match the CC line voltage changes when compared.

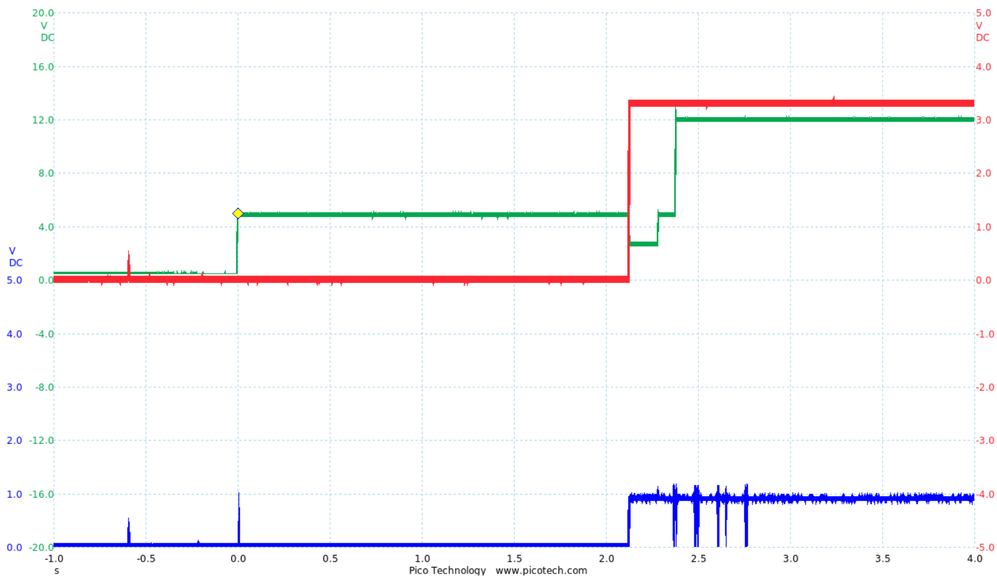


Figure 6.8: Port bus voltage and CC Line voltages.

6.3.2 Power bank Direct Charging and Discharging on Connected Loads

The second laboratory test case is implemented with the battery bank and DC-DC adapter coupled to a load. The load choice is made according to the selection of loads that was foreseen for the case study. Besides, considering the maximum output that DAC of the DC-DC adapter can deliver, the speaker with 5V input rating is chosen. Hence the DC-DC adapter output voltage value from DAC is configured accordingly. The DC-DC adapter, hence, is configured to negotiate and requests 5V fixed request data object. On the one hand, the battery bank is directly charged with the power supply, in constant voltage mode. The output sourcing ports in power bank are already only capable to deliver bus voltage up to 5 V. Hence no further configuration needed for power bank.

Battery voltage level is acquired by measuring the opposite poles in the battery connection with a minute time interval. At the same time, the input voltage of DC-DC adapter system, output voltage and output current data is logged and monitored within the same time period.

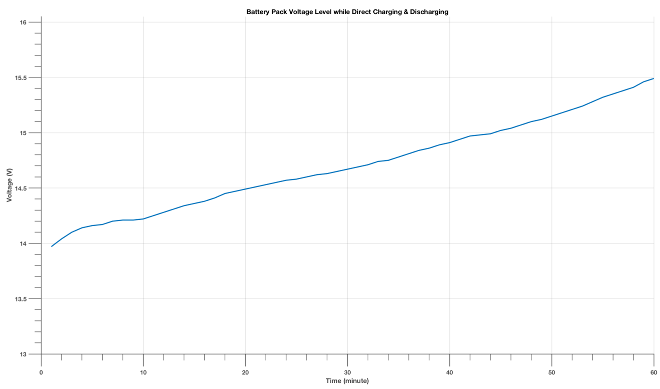


Figure 6.9: Battery voltage level during direct charging and discharging

Figure 6.9 illustrates the voltage level tested with a minute time interval during direct charging from the power supply and discharging through one of the USB Type-C ports. The discharge is implemented with 5V when the DC-DC adapter is connected to a basic load, a speaker. The charging of batteries have started when the battery level was around 13.97 V. The test took for an hour and the resulting battery voltage level was noted as around 15.5 V. This voltage level is noted down to be around 80% of its capacity, since the power flow was about to slow down.

Figure 6.10 illustrates the voltage characteristics of the batteries when it is being directly charged, without any load connections. The test took around 2.5 hours and the measurements were taken in each 5 minutes intervals. It is observed that the lithium ion battery charging rate got slower when they reached around 15.5 V, that is when the charging sequence got closer to constant voltage charging. In order to compare the two charging sequences, the current should be looked at. The current during the charging sequences will give insight into the charging power and rate of the batteries. Figure 6.11 illustrates the voltage delivery profile to a fixed DC load, through DC-DC adapter. The measurements acquired from data logging and transmission layer of DC-DC adapter.

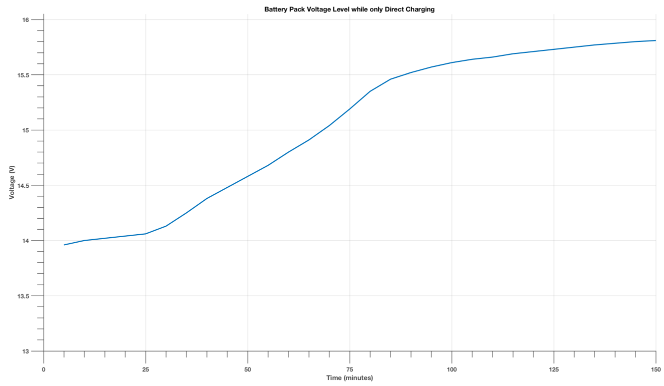


Figure 6.10: Battery voltage level during only direct charging

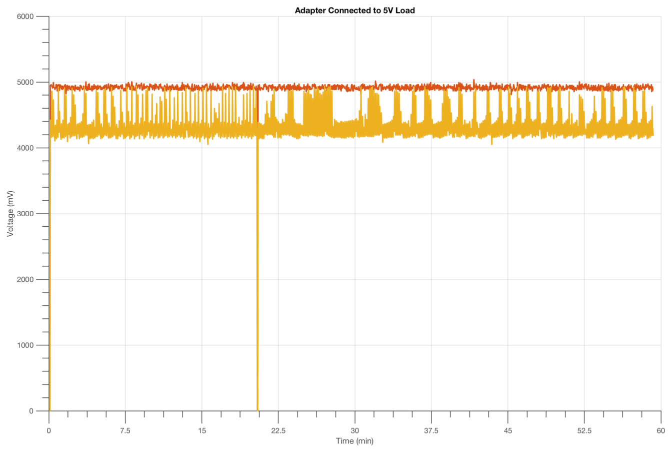


Figure 6.11: DC-DC Adapter powered by battery, Input voltage at constant 5V (red) and output voltage connected to a fixed DC load (yellow).

6.3.3 Laptop Charging

The final laboratory test case is high power laptop charging using USB-C charging hub. The test is carried out in laboratory. The setup is prepared in order to capture the voltage levels on bus of the port and CC lines. The picoscope is used for monitoring the voltage levels on these lines. Furthermore, the STM32G0 monitoring tool is used in order to capture the power delivery negotiation atomic message sequencing. The test set up is illustrated in Figure 6.12.

As soon as the laptop is connected to the charging hub, the policy engines get enabled and the power delivery negotiation starts. Before the negotiation, there is default 5 V and 1.5 A being provided on the port bus line. The charging hub port offers several voltage levels and current levels to the laptop, that would make the high power charging possible. During the power delivery negotiation, these capabilities are sent. The laptop evaluates and requests the most suitable voltage and current level. As soon as this is accepted by the policy engine of the sourcing hub port, the voltage and current levels are adjusted. The power supply ready message concludes the power negotiation. Consequently, the power delivery negotiation is achieved. The captured atomic message sequence is illustrated in Table 6.1.

During the aforementioned power delivery AMS, the voltage level on the CC line where the

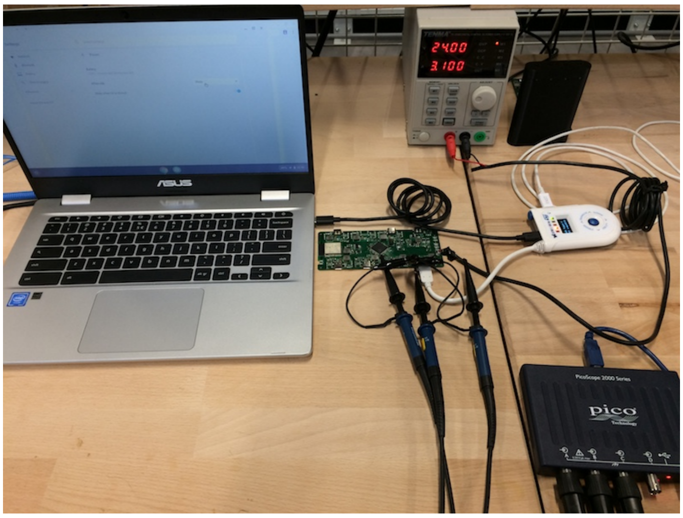


Figure 6.12: Laptop charging setup, illustrating the charging hub, Chromebook laptop, picoscope channels connected to CC Lines and bus voltage resistors. Power delivery monitoring tool is used in spy mode to capture the power delivery atomic messaging sequence.

Table 6.1: Laptop-charging hub power delivery negotiation. Source offers various fixed supply PDOs and a Variable Supply PDO. Laptop requests for 12 V-3 A. Supplying port accepts and the explicit contract is set

Timestamp(ms)	Port	Data Buffered	Header	Explanation
714	SRC	SRC Capabilities	0x51A1	Source capabilities: 5V-3A, 12V-3A, 15V-1.5A, 20V-1.5A, [3V-20V]-3A
715	SNK	GoodCRC	0x0041	Sink approves the source capabilities msg.
716	SNK	RDO:2CB10420	0x1042	Sink requests object position:2, confirming no capability mismatch
717	SRC	GoodCRC	0x01A1	Source approves the "RDO" msg
718	SRC	ACCEPT	0x0363	Source accepts the requests and adjusts its output accordingly
719	SNK	GoddCRC	0x0241	Sink approves "ACCEPT" msg.
720	SRC	PSRDY	0x05A6	Source sends "Supply Ready", confirming the contract
721	SNK	GoodCRC	0x0363	Sink approves the "PSRDY"

communication and message transmission established and the bus voltage of the corresponding source port is monitored. The voltage on CC line is observed to be around 1.1 V when the connection is established. Once the AMS has started, the data packages via bi-phase marked coding (BMC) are transmitted. This type of coding alters the voltage level on the CC line. The occurrence of the CC line voltage fluctuations show a similar pattern in terms of time series corresponding to the time stamps in the AMS in Table 6.1.

7 Conclusions and Recommendations

This chapter summarizes the key learning, project and thesis outcomes according to the stated project objective and research questions. Furthermore, the future recommendations on the next stages of the project are discussed.

7.1 Conclusions

During the realization of the project, there have been many technical challenges faced, milestones achieved and technical decisions made. When the project has started it was foreseen to implement the USB Type-C power delivery protocol without many hurdles. However, it is found out that the SMT32 USB PD project, which was the initial plan on taking as basis, did not meet the requirements of the project. The library was only able to function under deployment of maximum two ports. Furthermore, the library caused a lot of problems in terms of its initialization, management of stack and heap size. Since the core stack library was inaccessible, it caused excessive heap size on the run-time operating system. As a result, the Chromium project was chosen to be implemented after around five months of work on the STM32 project. Moreover, the adaptation of firmware into the Chromium project, creating the necessary adaptation layers and implementing it with the HAL and LL layers from the STM32 library created further challenges. These challenges eventually achieved and resulted in a firmware that is adaptable to multi port configuration (maximum eight ports, due to memory management).

The thesis and firmware deployment evolved on the way of the project. Hence, there have been many times of software debugging, in order to spot a problem and to find out the optimal solution to such a specific problem. However, in many cases, it is found out that the problem was rooted from the hardware issues on the electronic components of the designed circuits. This created vast opportunity on earning more experience on electronics side of the devices. Hardware debugging, circuit reading and problem solving has been practised on the way. It is concluded that in such a project, a hardware design should be in parallel with the software design since none of them can be implemented on its own. In the next iterations of the design and development, the collaboration about design limitations and parameters of hardware and software will be made. Nevertheless, the outcomes of the thesis created a solid base firmware implemented through the USB Type-C protocol on multi-port configuration, that can be deployed on different devices.

In order to realise the project objectives, the research questions are reflected back and discussed with the project findings, implementation and according to the case study and laboratory tests implemented.

(1) *How can power delivery over USB-C be implemented on the solar charging station for multi-port configuration?*

The power delivery firmware is implemented by firstly studying two open source projects; the STM32 Power Delivery and the Chromium Embedded Controller projects. Since the devices in the solar charging station or modular home system involve many components and different layers, an operating system that is capable of running multiple tasks and state machine has been taken as the main design objective. Hence, an architecture of device policy manager, policy engine, protocol layer and physical layer has been followed, each layer implemented individually. The state machines are deployed for these layers, that are able to cover several states and decision making processes. The transition and communication between states and different state machines are achieved by FreeRTOS operating system, dynamically allowed heap, messaging queues and event management.

Firmware is designed with each port running on its own task. On each port, there is a port controller with power electronics connected for managing the power delivery communication as well as power management. These components are mainly controlled by the Type-C state machine. The policy engine state machine implements local policy on the port, carrying out the mechanisms on negotiation and status management. The protocol layer state machine, on the one hand, is responsible of the reception message and transmission message driving, packet formatting and passing it to the specific layers.

(2) *What are the structures of data that are important to monitor, to be communicated over the ports for direct battery bank charging and energy management mechanism?*

In a system where power delivery and management is involved, the first of many parameters to be monitored are the voltage levels. First of all, in order to implement the power delivery successfully, the voltage levels on several registers of the port controllers are continuously monitored. Some of these important registers are configuration channel Lines. The alert line, that is the output of the register of the port controller and the fault line that is output of the acknowledgement pin of the power path controller, for over voltage or over current protection, are the other important parameters to be monitored for reliable performance. Additionally, the bus voltage on port is continuously monitored and communicated. Besides, the port source capabilities, with its maximum and operating current capabilities, operating and maximum voltage levels that it can offer are the other important set of data that have significant affect on the power delivery and charging patterns. In terms of the energy management, the current level being consumed is one other important parameter to be monitored. The voltage and current values result in monitoring of power consumed. The custom created unstructured vendor defined messages with load priorities is the last but not least important parameter to be communicated and monitored.

(3) *What are the user preferences tailored to rural use in Ethiopia and how to implement the demand response on the system?*

This question is left out to be answered on the next stage of the project. Although it was planned to have a field test and to implement the system in a rural village in Ethiopia, this plan has been delayed. At this stage, the project is going to continue since new revisions of the powerbank and the charging hub are developing. The field test is going to be carried out on March. There will be a month of work where which will be system implementation, data collection and surveying with the community. This field test will give deeper insight on the user preferences on rural use of electronics, appliances and how the electrification transition should actually be. These preferences and data collected on, such as the periods of usage, the load preferences, will be a basis on design criterion of demand response and reliable energy management implementation.

7.2 Outlook

After the realization of the thesis and carrying out the test results, further improvements on the firmware, hardware and project implementation have been identified. In terms of the finalization of the prototypes and preparation of the field test case, the final revisions of the prototypes designs are being made. For this purpose, some important aspects on the hardware related to data logging has been noted. The current revisions of the power banks were not able to have the port measurements regarding the bus voltage neither and current. In the next iteration, this specification will be added, with the ADC lines configured in the MCU. Furthermore, for the data transmission, extra pins will be reserved for the UART lines. Regarding the prototype of the hub, several improvements will be made. In order to remotely control, monitor and log the system and measurement data, the WiFi module on the circuit will be enabled. In order to do so, there will be two more dedicated pins on the micro controller. The Enable pins and flash booting pins are needed for remote debugging. These improvements on the prototype will make the field test as planned, feasible and will open room for remote data acquisition.

In terms of the software, the state machine and decision mechanism on load shifting has room for improvement. Currently, the total instantaneous power that can be supplied via eight ports is fixed in the memory. It is fixed to that of DC power supply connected. It is planned to configure this according to the instantaneous input power from the PV generation. A DC-DC adapter with maximum power point tracking converter can be introduced to the system via one of the ports connected to the boost converter. The data on current generation will be used as the input and be updated continuously within the decision mechanism.

The next step on the project, after finalization of the aforementioned prototypes, is pilot project or field test in Ethiopia. On March 2020, the student as well as colleagues from DC Opportunities RD, will implement the system. During this field test, it is aimed to gather broad data on user preferences, energy and power capabilities of the prototypes, projection on the payment and the actual pricing of the system. This test will also be useful for acquiring a reliable load profile in targeted rural areas. This load profile will then be used extensively to develop much broader energy management mechanism.

The field test case in Ethiopia will create an opportunity to fully test the developed USB Type-C Power delivery firmware tailored to rural electrification. The successful data acquisition will be processed in order to improve the system, validate the firmware and work towards achieving a reliable and efficient firmware that has energy management capabilities.

Bibliography

[1] The World Bank. ”More People Have Access to Electricity Than Ever Before, but World Is Falling Short of Sustainable Energy Goals”, 2019. retrieved from https://www.worldbank.org/en/news/press-release/2019/05/22/tracking-sdg7-the-energy-progress-report-2019.

[2] IEA. ”Number of people without access to electricity in sub-Saharan Africa in the STEPS”. Technical report, 2019. Retrieved from https://www.iea.org/data-and-statistics/charts/number-of-people-without-access-to-electricity-in-sub-saharan-africa-in-the-steps.

[3] Getnet Ecofys (Angelica Afanador, Michele Koper), SNV Ethiopia (Dr Mekonnen Lulie and Mengistu Teferra) Tesfaye. ’Off-grid Rural electrification in Ethiopia’ NAMA developed within the Mitigation Momentum project. Technical report, 2015. www.ecofys.com.

[4] BBoxx LTD. Our Technology - BBOXX. retrieved from https://www.bboxx.co.uk/our-technology/.

[5] ”Off-grid renewable energy solutions to expand electricity access: An opportunity not to be missed”. Technical report, 2019.

[6] Global Off-Grid Association and Altai Consulting. ”Powering Opportunity- The economic impact of off-grid solar” , year=2018.

[7] IRENA. ”Solar PV in Africa: Costs and Markets”. Technical report, 2016.

[8] Harries D. Holfort H.G. Urmee, T. ”Overview of Financing Mechanisms for Solar Home systems in Developing countries”. *Cham: Springer International Publishing*, pages 49–77, 2016.

[9] B. Rajpurohit et al. R. Chauhan, C. Phurailatpam. Demand-side management system for autonomous dc microgrid for building. *Technology and Economics of Smart Grids and Sustaianble Energy*, 2:4, 2017.

[10] S. Iqbal et al. H. Hussain, N. Javaid. An efficient demand side management system with a new optimized home energy management controller in smart grid. *Energies*, 11:1–28, 2018.

[11] Ethiopian Standards Agency. ETHIOPIAN STANDARD ES 6087 : 2017. Technical report, Ethiopian Standards Agency, 2017.

[12] Dlight. S30 Solar LED Lantern — d.light. retrieved from ://www.dlight.com/product/s30/.

[13] Dlight. X850 Solar Home System — d.light. retrieved from https://www.dlight.com/product/x850/.

[14] Azuri Technologies. Azuri — Life Changing Technology.

[15] B. Asare-Bediako. *SMART energy homes and the smart grid : a framework for intelligent energy management systems for residential customers.* PhD thesis, Technische Universitit Eindhoven, 2014.

[16] Melike Erol-Kantarci and Hussein T. Mouftah. "Wireless sensor networks for cost-efficient residential energy management in the smart grid". *IEEE Transactions on Smart Grid*, 2(2):314–325, jun 2011.

[17] Ahmad H. Sabry, W. Z. W. Hasan, MZA Ab. Kadir, M. A. M. Radzi, and S. Shafie. "DC-based smart PV-powered home energy management system based on voltage matching and RF module". *PLOS ONE*, 12(9):e0185012, sep 2017.

[18] UNDP ETH Zurich. Derisking renewable energy investment: Off-grid electrification. Technical report, United Nations Development Programme and ETH Zurich,Energy Politics Group, 2018.

[19] Global LEAP. Off-grid appliance market survey. Technical report, Global LEAP lighting and Energy Access Partnership, 2015.

[20] "Universal Serial Bus Power Delivery Specification". Technical report, 2019. Revision 3.

[21] van der Blij, Ramirez Elizondo, Laurens Mackay, Nils H van der Blij, and Pavol Bauer. "Toward the Universal DC Distribution System". *Electric Power Components and Systems*, 45(10):1032–1042, 2017.

[22] Irazusta Gorostidi L. "Power Electronics in Modular Solar Home Systems". Technical report, 2019.

[23] STMicroelectronics. STM32G0x1:Reference Manual. Technical Report November, 2008.

[24] NXP Semiconductors. PTN5110 USB PD TCPC PHY IC. *PTN5110 Product Data sheet*, 3(January):2, 2006.

[25] NXP Semiconductors. NX20P5090 High Power PD Power Switch. *Product Data sheet*, (October):1–28, 2016.

[26] STMicroelectronics. STM32F091xC Data sheet. Technical Report December, 2015.

[27] STMicroelectronics. STM32F105xx Reference manual. Technical Report December, 2018.

[28] USB 3.0 Promoter Group. Universal Serial Bus Type-C Port Controller Interface Specification. Technical Report October, 2017.

[29] FreeRTOS. "FreeRTOS Static Memory Allocation". retrieved from `https://www.freertos.org/Static{_}Vs{_}Dynamic{_}Memory{_}Allocation.html`.

[30] STMicroelectronics. User manual STM32CubeMonitor-UCPD software tool for USB Type-C ™ Power Delivery port management. (November):1–37, 2018.