



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIO ELECTRONICS

ÚSTAV RADIOELEKTRONIKY

WIRELESSHART SENSOR NODE

SENZOROVÁ SÍŤ WIRELESSHART

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Bc. Tomáš Řežucha

SUPERVISOR

VEDOUCÍ PRÁCE

Ing. Tomáš Götthans, Ph.D.

BRNO 2018

Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Bc. Tomáš Řežucha

ID: 154862

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Senzorová síť WirelessHART

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte dostupnou literaturu a seznámte se s protokolem WirelessHART. Navrhněte zařízení využívající komponenty od firmy Analog Devices. Pro demonstraci funkce snímejte alespoň dvě proměnné (například snímejte analogový Sigma-Delta ADC vstup a digitální vstup - například akcelerometr). Hardwarové řešení musí obsahovat FSK modem (dle specifikace WirelessHART), bateriové napájení a konektor pro připojení externích periférií. Využijte radiového modulu v pásmu ISM (2.4GHz) s integrovanou anténou. Zvažte též výrobu výkonového zesilovače a LNA pro rozšíření dosahu.

Navrhněte firmware pro MCU v jazyce embedded C. Dále plně implementujte HART linkovou vrstvu a Dust API která zabezpečí komunikaci s navrženým senzorem. V senzoru musí být implementovány univerzální příkazy HART, řízení výkonu, podpora služeb a aplikační doména.

DOPORUČENÁ LITERATURA:

[1] Datasheet AD5700, dostupný z:
http://www.analog.com/media/en/technical-documentation/data-sheets/AD5700_5700-1.pdf

[2] H. M. Ammari, The Art of Wireless Sensor Networks: Volume 1: Fundamentals (Signals and Communication Technology), 2014.

[3] H. M. A. Fahmy, Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis (Signals and Communication Technology), Springer; 1st ed. 2016.

Termín zadání: 5. 2. 2018

Termín odevzdání: 17. 5. 2018

Vedoucí práce: Ing. Tomáš Götthans, Ph.D.



prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení částí druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstract

The master's thesis is focused on building WirelessHART node demo and on a description of WirelessHART standard, which is used in wireless sensor networks. Out of the comprehensive WirelessHART specifications firstly, the two lowest layers of OSI networking model are described: physical and data link layer. Secondly, burst mode and device variables are described to provide necessary information for data publishing in the network. The mote is assembled according to hardware and software requirements of WirelessHART demo system. The node is complemented by its firmware and PC application demonstrating its features in a user-friendly manner.

Keywords

WirelessHART, HART, wireless sensor networks, precise sensor sampling, extended battery life

Abstrakt

Předložená diplomová práce je zaměřená na sestrojení demonstračního systému založeném na specifikaci WirelessHART a na popis průmyslového standardu WirelessHART, který je navržen pro použití v bezdrátových senzorových sítích. Z rozsáhlé specifikace WirelessHART jsou v práci nejprve popsány dvě nejnižší vrstvy síťového modelu: fyzická a linková; dále je popsán burst mód a dynamické proměnné nezbytné pro publikování dat v síti. Navržený a sestrojený demonstrační hardware je doplněn firmwarem a PC aplikacemi, které demonstrují možnosti systému v přehledném grafickém prostředí.

Klíčová slova

WirelessHART, HART, bezdrátové senzorové sítě, precizní vzorkování senzorů, prodloužená životnost baterky

Rozšířený abstrakt

Úvod

Předložená diplomová práce si klade za úkol sestavit demonstrační systém na základě protokolu WirelessHART, který by prezentoval možnosti širokého portfolia součástek firmy Analog Devices, která je zadavatelem této práce. Stěžejní část práce se soustředí na návrh obvodového a softwarového řešení bezdrátového senzorového přístupového bodu.

WirelessHART je světově první standard (IEC 62591) schválen mezinárodní komisí International Electrotechnical Commission, který definuje samo-organizující se síť typu mesh, ve které se každé připojené zařízení může také stát routerem, směrovat zprávy ostatních účastníků sítě a rozšiřovat tak geografický dosah sítě. WirelessHART je rozšířením dominantního průmyslového standardu pro senzory a aktuátory HART (Highway Addressable Remote Transducer) se kterým sdílí transportní a aplikační vrstvu síťového OSI modelu, migrace už existujícího drátového dizajnu k bezdrátovému je tím výrazně usnadněna. Zařízení splňující specifikaci WirelessHART musí obsahovat i drátový FSK modem pro zpětnou kompatibilitu se klasickým HART-em, proto je v této práci věnována pozornost obou rozhraním.

Diplomová práce je rozdělena do pěti hlavních kapitol, mimo úvodu a závěru. Druhá kapitola je zaměřená na obecný popis bezdrátových senzorových sítí, na kterou navazuje kapitola popisující už konkrétně protokol WirelessHART, jeho fyzickou a linkovou vrstvu a další vybrané podkapitoly z aplikační vrstvy, které jsou klíčové pro správné pochopení činnosti WirelessHART kompatibilního zařízení. V kapitole čtyři jsou definované požadavky na hardware a navrženy konkrétní řešení, které by umožňovali dosáhnout cílených vlastností pro extrémně nízkou spotřebou a schopnost publikovat data ze senzoru, v tomto případě z akcelerometru, v síti WirelessHART. Pátá kapitola popisuje navržené zařízení z programátorského hlediska a detailně přibližuje klíčové vlastnosti dodaného embedded firmwaru. Demonstrační systém je doplněn aplikacemi pro PC, které jsou popsány v poslední kapitole. Tyto aplikace jsou určeny pro úvodní sestavení sítě, monitorování stavu sítě po jejím ustálení a pro zobrazování dat ze senzorů v uživatelsky přívětivém grafickém prostředí.

Řešení

Jestli se zařízení může označovat za chytré, je z velké části určeno jeho softwarem, navrhované zařízení se musí držet všech doporučení pro správné publikování dat v síti, v síti typu mesh jde obzvlášť o žádost o udělení šířky pásma v síti, před periodickým vysíláním naměřených hodnot. Každé zařízení smí vysílat až po přidělení rádiových prostředků od síťového manažera. WirelessHART je síť ve které jednotliví účastníci vysílají s velmi nízkou střídou, díky které může toto zařízení strávit většinu svého času v hlubokém spánku, proto je nutno klást důraz na firmware řídicího mikrokontroleru, který může vypnout nepoužívané části zařízení (např. drátový HART modem) a sám

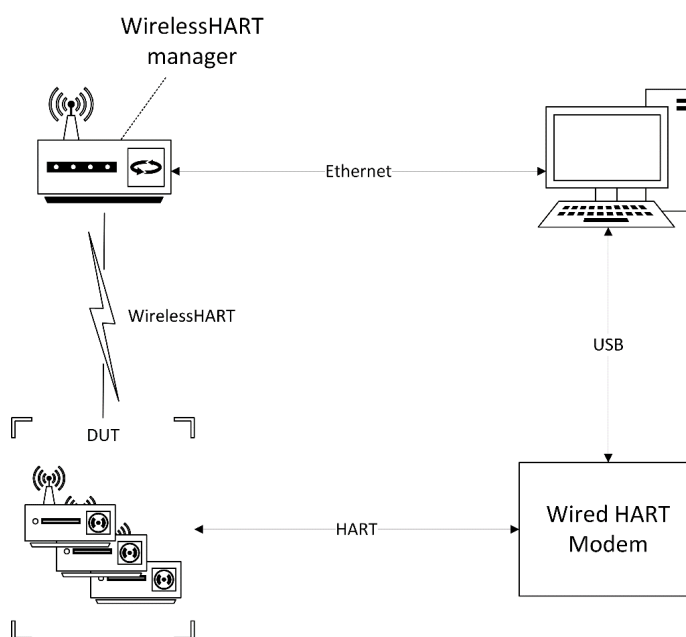
může být uložen do hibernačního módu, aby se výdrž baterky přiblížila teoretickému maximu.

Použitý Systém-on-Chip LTC5800, popsáný dále, obsahuje kromě řídicího procesoru také kompletní radiový front-end, splňující specifikace fyzické vrstvy WirelessHART. V případě že je požadován větší dosah, musí být navrhnout výkonnější koncový zesilovač a vstupní nízko šumový zesilovač. Tato možnost je diskutována v kapitole čtyři.

Na základě popsaného protokolu WirelessHART byly sestaveny požadavky na výsledné zařízení, jeho hardware i firmware s důrazem na nízkou spotřebu, napájení z baterky a sběr dat z digitálního senzoru, například akcelerometru. Dále je diskutována možnost efektivního a časově přesného vzorkování akcelerometru pro případné zpracování získaných dat pomocí rychlé Fourierovy transformace, která by doplnila měření maximální amplitudy vibrací o frekvenční analýzu, bez výrazného zvýšení spotřeby celého systému.

V dalším kroku bylo zařízení vyrobeno podle navrženého schématu a důkladně otestováno, aby byla zajištěná správná regulace napájecího napětí, správné přijímání a vysílání HART paketů po bezdrátovém rozhraní a také správná detekce nosné na drátovém rozhraní. Firmware pro aplikační mikrokontroler je neoddelitelnou součástí navržené desky plošného spoje, zabezpečuje téměř všechnu HART funkcionalitu vykonáváním, resp. ukončováním popsaných HART příkazů, vyjma výlučně bezdrátových příkazů, které jsou zajišťovány WirelessHART modulem. V současnosti zařízení podporuje všechny univerzální (povinné) příkazy a také příkazy spojené s burst módem, včetně automatických požadavků pro přidělení rádiových prostředků pro periodické publikování dat. Navíc jsou podporovány vybrané běžně používané příkazy jako Reset, nebo příkazy umožňující audio-vizuální identifikaci zařízení (blinknutí LED-kou).

Demonstrační systém podle zapojení na obrázku 1, umožňuje komplexně prověřit



Obrázek 1. Zapojení demonstračního systému

funkčnost navrženého zařízení, pomocí HART příkazů, které mohou být přijaty od síťového manažera nebo přes příruční servisní přístroj, simulovaný pomocí HART modemu. Dodaný firmware je již před-konfigurovaný, a proto lze drátové rozhraní z demonstrace vynechat.

Typické demo zahrnuje připojení alespoň pěti zařízení do sítě, po několika minutách se vytvoří síť mesh a řídicí aplikace může požadovat publikaci naměřených dat. Připojená zařízení zažádají o rádiové prostředky a posléze začínají periodicky vysílat data v požadovaném formátu, tyto se potom zobrazují v PC aplikaci, sloužící jako gateway celého systému.

Shrnutí a zhodnocení výsledků

I když byly hlavní úkoly práce splněny, v projektu zůstává velký prostor pro další vývoj. Z hardwarového hlediska by měl být přidán přepínač, který by vypnul bateriový regulátor a tím odpojil baterku od zbytku obvodu; tímto způsobem se dá dosáhnout větších úspor energie při delším nepoužívání, v porovnání s hlubokým režimem spánku, který je momentálně implementován. Po diskuzi v kapitole 4.2.3 bylo rozhodnuto vynechat externí rádiové zesilovače z návrhu.

Aplikační firmware lze také doplnit o podporu tzv. blokového přenosu, který se používá při přenosu většího množství dat, aby nezahltil síť. V tomto konkrétním případě by byl vhodný pro přenos informací o frekvenčním spektru vibrací snímaných akcelerometrem, pro získání dominantních frekvencí ve sledovaných vibracích.

Navržený systém je doplněn dvěma aplikacemi pro PC, které interagují se síťovým manažerem, a tak s celou sítí. První aplikace je určena pro uživatele s alespoň základními poznatky o standardu HART a umožňuje detailní nastavení sítě i připojených bezdrátových zařízení prostřednictvím HART příkazů. Druhá aplikace je více demonstrační, dokáže pouze zapnout nebo vypnout burst mód a zobrazit publikovaná data ze senzorů a také identifikovat jednotlivé přístupové body pomocí LED diod.

K systému je dodána další jednoduchá aplikace se svolením Analog Devices Inc., která zabezpečuje komunikaci se zařízením pomocí drátového HART rozhraní. To je typicky používáno pro inicializaci zařízení správnými přihlašovacími údaji pro vstup do sítě, nebo pro identifikaci závad, pokud se vyskytne nekorektní stav, například odmítnutý požadavek na připojení do sítě.

Můžeme tedy konstatovat, že navržený systém je schopný demonstrovat široké možnosti které předkládá poslední verze specifikace WirelessHART a nejmodernější Systém-on-Chip LTC5800 ve spojení s mikrokontrolerem s extrémně nízkou spotřebou a smart senzory. Spojení všech těchto komponent do jednoho celku vytvoří spolehlivé, robustní a chytré zařízení schopné pracovat z baterie potenciálně několik let, a navíc fungovat jako směřující prvek v síti typu mesh.

Bibliografická citace:

ŘEŽUCHA, T. *Senzorová síť WirelessHART*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 54 s. Vedoucí diplomové práce Ing. Tomáš Götthans, Ph.D..

Prohlášení

Prohlašuji, že svoji diplomovou práci na téma Senzorová síť WirelessHART jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této semestrální práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 17. května 2018

.....
podpis autora

Poděkování

Děkuji celému aplikačnímu oddělení Analog Devices Inc. v Limericku a obzvlášť mému vedoucímu Ing. Michalovi Brychtovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 17. května 2018

.....

podpis autora

Trademark information

HART® and WirelessHART® are registered trademarks of the FieldComm Group, Austin, Texas, USA. Any use of the term HART or *WirelessHART* hereafter in this document, or in any document referenced by this document, implies the registered trademark. All other trademarks used in this or referenced documents are trademarks of their respective companies.

Table of content

1	Introduction	13
2	Wireless sensor networks	14
2.1	Concept of wireless sensor networks.....	14
2.2	Mesh network	15
3	WirelessHART standard	17
3.1	Overview.....	17
3.1.1	Dust Networks LTP5903 manager	19
3.2	RF and physical layer	20
3.2.1	Coexistence	22
3.3	MAC and data link layer.....	23
3.3.1	DLPDU types.....	23
3.3.2	Media Access Control.....	24
3.3.3	Implementation on a silicon.....	24
3.4	Dynamic and device variables	25
3.5	Burst mode	26
3.5.1	Configuring a device for Burst Mode operation	27
4	Sensor node design	28
4.1	Maintenance port	28
4.1.1	AD5700-1 HART modem	29
4.1.2	HART carrier detector	29
4.1.3	HART master	30
4.2	WirelessHART interface.....	32
4.2.1	LTC5800 WirelessHART Mote-on-chip	32
4.2.2	LTP5901 – WHM module	32
4.2.3	External power and low noise amplifier	32
4.3	Power stage	33
4.4	Microcontroller selection.....	33
4.5	SensorStrobe – precise time sampling.....	34
4.5.1	Accelerometer ADXL372.....	35
4.5.2	System description.....	36

5	Node's hardware and firmware	37
5.1	HART commands	37
5.1.1	Commands aggregation	37
5.1.2	Commands termination	38
5.2	HART non-volatile settings	40
5.2.1	Write protect mode	41
5.3	Firmware description	41
5.3.1	Microcontroller initialization	41
5.3.2	Burst enabling	43
5.4	WirelessHART mote testing	44
5.5	Mote implementation	45
5.6	Operating range	47
6	Demonstrating software	48
6.1	Connecting to the network manager	48
6.2	WirelessHART manager control application	49
6.3	WirelessHART Burst viewer	51
7	Conclusion	53

Table of figures

Figure 2-1. Example use of WSN in industrial environment [2].....	14
Figure 2-2. Possible network topologies [5].....	15
Figure 3-1. Siemens SITRANS P280	18
Figure 3-2. WirelessHART network architecture [11]	19
Figure 3-3. LTP5903CEN-WHM network manager	20
Figure 3-4. WirelessHART frequency spectrum [12]	21
Figure 3-5. PHY and MAC packets in IEEE 802.15.4 [13]	21
Figure 3-6. HART and IEEE 802.15.4 in OSI model.....	22
Figure 3-7: Device and dynamic variables [17].....	25
Figure 3-8. Burst enabling sequence [18].....	26
Figure 4-1. WirelessHART sensor mote demo – block diagram.....	28
Figure 4-2. HART carrier detector circuit	29
Figure 4-3. Carrier detector function	30
Figure 4-4. DEMO-AD5700-D3Z, HART master.....	31
Figure 4-5. Simple HART master used for wired HART interface debugging.....	31
Figure 4-6. LTP5901-WHM (in the red rectangle) 42x24mm connected to Eterna programmer.....	33
Figure 4-7. ADXL372 shaker board	35
Figure 4-8. SensorStrobe mechanism timeline [24]	36
Figure 5-1: HART commands termination: a) mote terminated b) application microprocessor terminated [28].....	38
Figure 5-2. MCU initialization sequence.....	42
Figure 5-3. Burst enabling sequence.....	44
Figure 5-4. WirelessHART mote testing setup.....	45
Figure 5-5: Assembled system.....	46
Figure 6-1. Welcoming connection window.....	49
Figure 6-2. Enabling burst with HART command 109	50
Figure 6-3. HART bursts viewer	52

1 INTRODUCTION

The master's thesis aims to build a WirelessHART demo system, that could effectively represent broad possibilities that the device portfolio of Analog Devices Inc, the sponsor of this thesis, provides. Main stress is put on hardware and firmware design of wireless sensor nodes, called field devices in HART.

WirelessHART is the first global IEC-approved standard (IEC 62591) that specifies an interoperable self-organizing mesh technology in which field devices form wireless networks. WirelessHART is built atop of the broadly used HART (Highway Addressable Remote Transducer) protocol and shares transport and application layer with it, making it easier to migrate from legacy wired to state-of-the-art wireless designs. WirelessHART compliant mote must be backward compatible with wired networks, therefore both interfaces take place in the project.

Whether the field device can be deemed as 'smart' is mostly defined by its firmware; the proposed mote should utilize all HART recommendation for data publishing, especially requesting services in publish domain, when every device must request bandwidth from network manager prior periodically publishing the data. WirelessHART is a deep-low duty cycle network, which enables the mote to spend most of the time in low-power modes, hence great effort is put on correct power management of the microcontroller and both the wired and wireless HART interfaces to extend the battery life to the theoretical maximum.

The System-on-Chip LTC5800, described later, integrates complete radio front-end compliant with WirelessHART physical layer specifications. In the event that user needs a more powerful solution, to extend the operating range, external power amplifier can build. This option and its feasibility is described in chapter four.

The document is divided into five chapters apart from introduction and conclusion. The second chapter is focused on general description of wireless sensor networks and is followed by a more specific chapter describing the WirelessHART standard, its physical and data-link layer and other necessary information chosen from application layer that is important to understand in order to build a WirelessHART node. In chapter number four specific high-level design solutions are proposed, that would meet requirements for extended battery life, and sensor data publishing. The fifth chapter thoroughly introduces the designed mote from programmer's point of view and key firmware features are described in detail. The demo system is complemented by PC applications, which are described in the last chapter. These applications are intended for setting-up the network, monitoring its health, and for collecting the sensor data in a user-friendly manner.

In spite of the fact that the assignment requirements were met, there is still a lot to add in order to utilize all the functionality defined in HART specifications; suggested future improvements of the project can be found in conclusion.

2 WIRELESS SENSOR NETWORKS

In recent years an efficient design of a wireless sensor network (WSN) has become a leading area of research in both academia and industry. A WSN is a network that is made of tens to thousands of tiny, low-power, and low-cost sensing devices called sensor nodes, which are installed in the desired environment observing certain phenomenon. A sensor is a device that responds and detects some type of input from both the physical or environmental conditions, such as pressure, heat, vibration, light, etc. The output of the sensor is generally an electrical signal that is converted to digital domain and transmitted to a controller for further processing and then forwarded to the network using wireless transmitter. Modern low power manufacturing processes combined with effective primary batteries and secondary power harvesting allow designers to come up with ultra-low power sensor nodes with extended battery life, thus eliminating one of the biggest limitations of wireless sensor networks.

WSNs are designed to allow ad-hoc deployment, making them both practical and low-cost for many applications in scenarios as military target tracking, natural disaster prediction (seismic activity) or biomedical health monitoring [1]. This thesis, however, is focused on WSNs in industrial applications. An example of various use of wireless sensors in an industrial environment is shown in Figure 2-1:



Figure 2-1. Example use of WSN in industrial environment [2]

2.1 Concept of wireless sensor networks

The concept of sensor nets, small but smart devices, sprinkled on the manufacturing plant has been around for a couple of years. While there has been a plurality of proprietary wireless systems developed over the past decade, these systems have suffered from an

inability to scale well in cost and network complexity. In 2003, the IEEE 802.15.4 standard was ratified, and almost immediately silicon manufacturers began producing compliant single-chip radios [3]. At the present time in 2017, we can see next generation completed with integrated microprocessor and FLASH memory, as well as the potential for various environmental sensors to be built right into the silicon itself. The IEEE Standard 802.15.4 specifies the physical (PHY) and medium access control (MAC) layers of the networking OSI model, the upper layers are left to be defined by industry-standard protocols (such as WirelessHART).

In many WSNs the number and/or location of the sensor nodes make replacing or recharging the batteries difficult, almost infeasible. With energy-saving features designed to the fundamental IEEE standard (deep duty-cycle of transmissions), and other possibilities applied by the applications developer, IEEE 802.15.4 radios have the potential to be the cost-effective communications backbone for simple sensory mesh networks that can effectively harvest data with relatively low latency, high accuracy and ability to survive for a very long time on a primary battery (>5 years).

2.2 Mesh network

In addition to extended battery life, the modern sensor networks have moved from star topology to mesh topology (see Figure 2-2). A wireless mesh network is a fully wireless network that employs multi-hop communications to route messages to and from wired entry points, often called sink points, in case of dominantly collecting data from the network [4]. The mesh network topology is a solution for emerging market requirements for building scalable wireless networks, bringing many benefits:

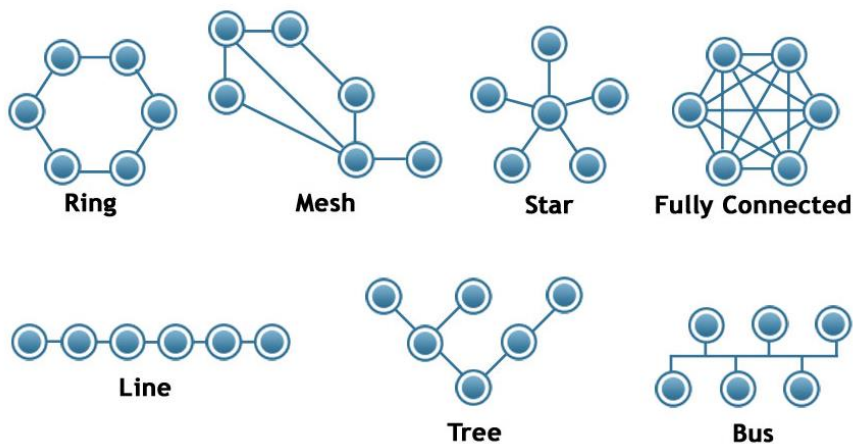


Figure 2-2. Possible network topologies [5]

When every node can also route all messages, building the network is much easier, because new nodes do not have to be connected to a limited number of access points, but to any node (with enabled routing), which consequently increases coverage area of the network. Communication over large distances is performed through multiple hops, over intermediate short links, enabling the transmitter to simultaneously transmit at low power with efficient modulation techniques. With fewer wired access-points, compared to a star

topology, the infrastructure costs are significantly reduced, yet, when deploying WSN in order of hundreds of nodes, number and location of sink points (base stations) must be carefully chosen [1].

The wireless backbone provides redundant paths between two endpoints, eliminating single points of failure or bottleneck links within the mesh. It was experimentally proven that when every node has at least three good neighbors, the network becomes stable, achieving reliability better than 99.99% (4 sigma) [6], which is crucial in industrial applications. Network fault-tolerance is increased because the network is able to re-route the message when certain path is unstable due to node crash or radio interference, this is tightly connected with so-called distributed self-management:

Data protocols make WSNs practically useful and functioning, they add ‘smart’ functions to the network so it could manage and heal itself, mainly because it is built on a peer-to-peer basis. The network services are therefore distributed and every node in the network can determine optional path to the sink point.

The diploma thesis does not aim to describe all types of WSNs that are being under research or are defined in [1], but only specific cases used in process control will be discussed. Industrial WSNs are typically terrestrial (as opposed to underground or underwater) with no mobility nor media support, resulting in periodic bursting at low data-rates.

3 WIRELESSHART STANDARD

WirelessHART is the first international IEC-approved industrial wireless standard for process control applications. By process control is typically meant a loop, that simultaneously actuates certain process/phenomenon and monitors its output via sensor feedback, completing the loop. Typical applications of process control are manufacturing plants, oil refineries or power plants. For easier deployment of process control loops in practise, standardized protocols are used for field communication between field devices i.e., sensors and actuators mounted on the plant and the host application i.e., process control system [7].

HART (Highway Addressable Remote Transducer) is one of the most widely used bi-directional industrial communication protocol for communication in process control loops between field devices and control systems. It is a hybrid protocol, it can communicate over legacy 4-20 mA analogue instrumentation current loops, also in multi-drop. It shares the pair of wires used by the analogue only host systems to send digital HART messages modulated on the 4-20 mA loop adding smart functionality to the protocol. The global installed base of HART-enabled devices is more than 20 million, mainly due to the significant installed base of 4-20 mA systems throughout the world [8].

3.1 Overview

WirelessHART was released in 2007 as a wireless extension of the already established HART protocol. The demand for wireless standard developed from the high installation and maintenance cost of wires especially in industry 4.0, where the number of sensors raised exponentially. Moreover, existing wireless protocols i.e., Zigbee, Bluetooth or Wi-Fi were not suitable for industrial process control, automation applications and ad-hoc deployment. For instance, Zigbee's biggest disadvantage is poor performance in noisy environments, thus it cannot provide desired reliability. Bluetooth only supports star type network topology with no centralized management, therefore it is not scalable for large industrial networks. Wi-Fi is originally created for high data rates and needs dense infrastructure of access points, increasing both cost and power consumption [8]. In 2017, ten years after the initial WirelessHART release, installed base exceeded 36800 networks and 11 billion operating hours, according to Emerson [9].

Leading features of WirelessHART include outstanding reliability and security comparable with wired networks. Further, real-time communication is enabled using synchronized Time Division Multiple Access (TDMA) and distribution of network services in the mesh, however, it still remains a network with centralized management using network manager. High reliability is achieved by channel hopping, channel blacklisting and using redundant paths in the mesh network [3]. Security is an inbuilt feature of WirelessHART which ensures data integrity on MAC layer and data confidentiality on the network layer using 128-bit AES encryption. Authentication and cryptographic procedures are provided solely by security manager, which must be present in every WirelessHART network.

A generic WirelessHART network is depicted in Figure 3-2. It consists of a number of network devices, which can be divided into 5 groups:

1. Field devices and adapters

Field devices consist of sensors (typical for wireless), actuators or both connected to process or plant equipment. They can be powered from line, battery or from energy harvester (typically solar, thermal or mechanical). Example of WirelessHART pressure transmitter is shown in Figure 3-1. Adapters are simple devices that add WirelessHART functionality to legacy HART devices. One adapter can be used for multiple devices.



Figure 3-1. Siemens SITRANS P280

2. Handheld devices

These portable devices are intended mostly for maintenance engineers to configure, diagnose and calibrate the field devices.

3. Network manager

The network manager is responsible for configuring the network, managing the routing tables and scheduling communication between devices. There can be only one network manager in the WirelessHART network, although redundant managers are supported.

4. Security manager

Security manager must be present, as the security cannot be turned off in WirelessHART. WirelessHART uses CCM mode 128-bit AES encryption, providing end-to-end and hop-to-hop security of both data and network.

5. Gateway

Gateway provides a single-entry point (also called sink point) to WirelessHART network, connecting the host application with field devices.

6. Access point

Grants radio resources.

Term ‘WirelessHART gateway’ is often used to refer to a device that contains gateway, security manager, network manager and access point. Interface between the gateway and host application is typically made by HART-IP, Modbus, Profinet and/or other industrial networking standards [10].

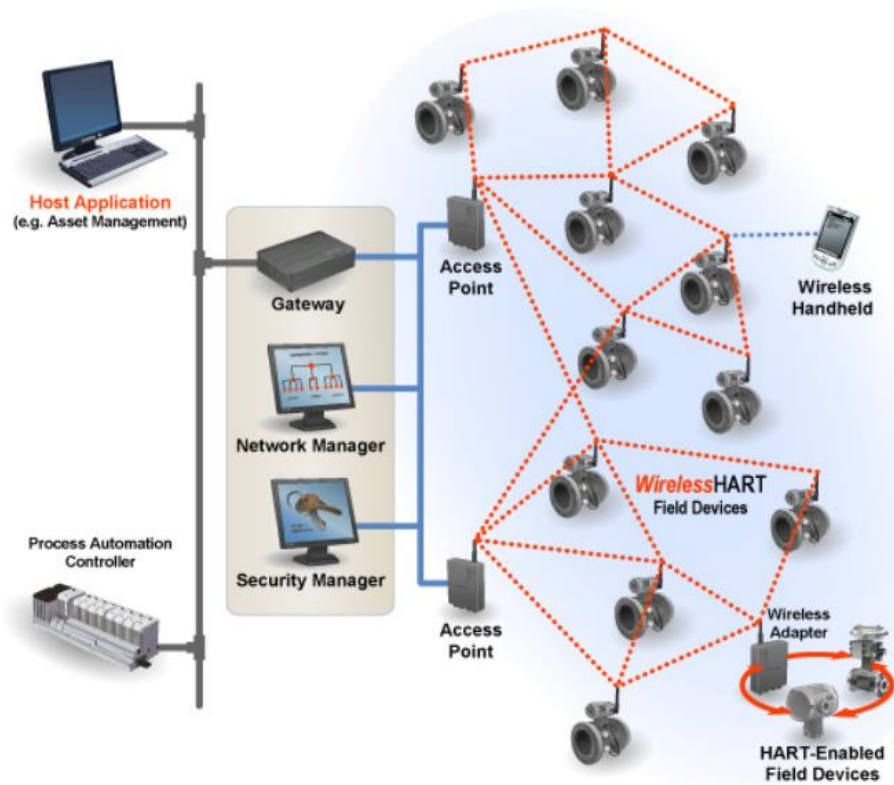


Figure 3-2. WirelessHART network architecture [11]

3.1.1 Dust Networks LTP5903 manager

In this project, all WirelessHART networks will build with network manager Dust Networks LTP5903CEN-WHM depicted in Figure 3-3. The LTP5903-WHR runs SmartMesh WirelessHART network management software to monitor and manage network performance and provide a data ingress/egress point via standard Ethernet interface. It provides the core networking and security functionality, enabling the network to achieve unsurpassed levels of resilience, reliability, and scalability with advanced network management and comprehensive security features.

The embedded network manager offers a comprehensive software application programming interface (API) to deliver rich and flexible functionality without complex coding. This API also provides full visibility and control over network configuration, security administration, network status, and performance statistics. Nevertheless, it is not intended to be used as WirelessHART gateway, as the SmartMesh software does not include the application layer of WirelessHART protocol. This layer must be built atop the LTP5903 and run on a dedicated PC.



Figure 3-3. LTP5903CEN-WHM network manager

3.2 RF and physical layer

Purpose of the physical layer is to create optical, electrical or microwave signal that represents the bits in each frame and get them on the media. It includes encoding, converting data streams into bit patterns and modulation, converting bits to signals representing log. 0 or log. 1.

Most of the industrial wireless sensor networks standards, including WirelessHART, are build atop of IEEE 802.15.4 but uses only some of the possible features introduced in this recommendation. Basic parameters of the physical and data link layers and their performance are listed in Table 3-1.

Table 3-1. WirelessHART PHY parameters

Parameter	Value
Channel frequency	$2400 + 5x \text{ ChannelNumber}$ MHz
Bandwidth	2 MHz
Max. data rate	250 kbps
Modulation	Offset-QPSK
Symbol rate	62.5 kBaud
Chip rate	2 Mchip/s
Spreading factor	32 chip/symbol
Timeslot length	10ms
Maximum radiated power	10 dBm
Range	15-20m (typical)
Latency	~15ms

WirelessHART uses only 15 channels out of 16 available (numbered 11-25 in IEEE 802.15.4) with 5 MHz span and 2 MHz bandwidth in ISM band 2400 – 2483.5 MHz; channels defined in 5GHz or sub GHz bands are not supported. Maximum data rate can be as high as 250 kbps; with offset-QPSK modulation, it results in symbol rate 62.5 kBaud. Maximum radiated power in one channel is limited to 10dBm. Direct sequence spread spectrum (DSSS) with spreading factor SF = 32 chip/symbol is used to achieve desired robustness against both narrow and broadband noise. In addition to DSSS time slotted channel hopping with channel blacklisting is implemented, giving user freedom in choosing which channels not to use, when operating in noisy ISM environment.

WirelessHART channel assignment in the frequency domain is shown in Figure 3-4:

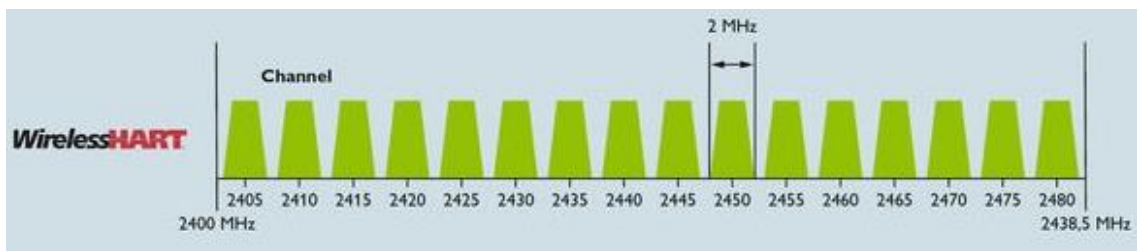


Figure 3-4. WirelessHART frequency spectrum [12]

The physical layer is also responsible for synchronization between transmitter and receiver. Preamble of physical layer packet, called synchronization header (see Figure 3-5), is 32 bits long. It is followed by PHY header, which contains start of packet delimiter and length of the payload, which can be 0-127 bytes.

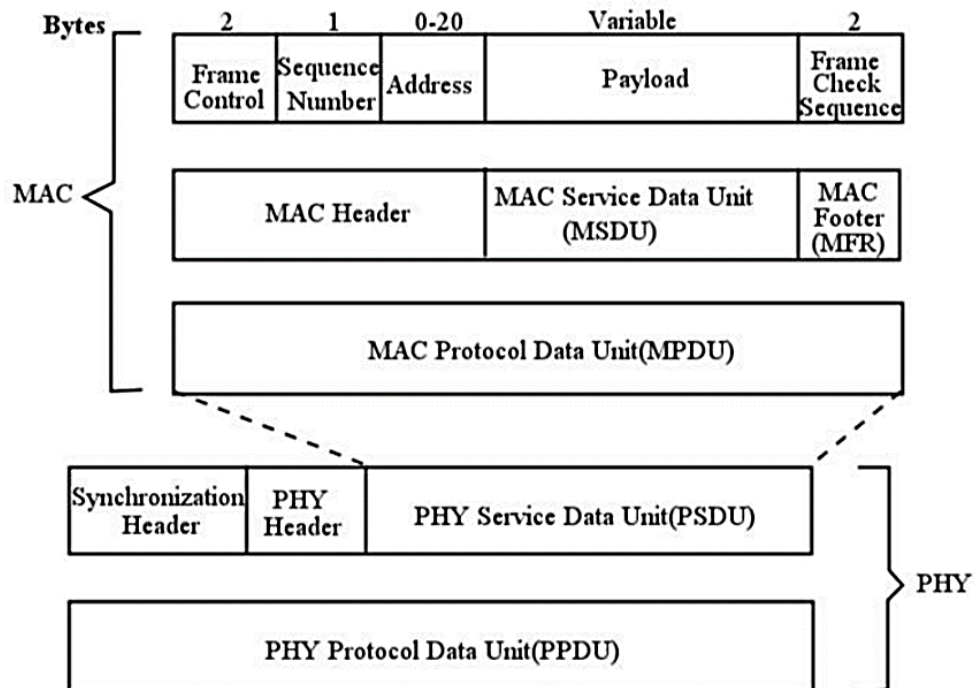


Figure 3-5. PHY and MAC packets in IEEE 802.15.4 [13]

While WirelessHART is still based on IEEE 802.15.4, this recommendation has progressed to the last version of IEEE 802.15.4e, with lots of new features that used to be exclusively in WirelessHART (i.e. channel hopping) [14].

Comparison of HART (both wired and wireless), IEEE 802.15.4 and standard OSI model is shown in Figure 3-6.

3.2.1 Coexistence

All WirelessHART networks operate in the same 2.4 GHz instrumentation, scientific, and medical (ISM) unlicensed wireless band. Here unlicensed means that anyone is free to operate in this band provided that they follow local power limits. But for the reason that anyone can use it, it is crowded with other products - 802.15.4 Wi-Fi b/g/n, Bluetooth, Cordless phones, and ZigBee all operate in the same band.

Coexistence is a key WirelessHART requirement and strength, WirelessHART has both the ability to perform successfully in the presence of interferences from other networks and protocols and the features to share the radio space.

For the IEEE 802.15.4 physical layer blacklisting generally provides limited benefits in practise. The modulation used by IEEE 802.15.4 is generally both tolerant and unobtrusive to other protocol/modulation standards in the 2.4 GHz ISM band (e.g., IEEE 802.11 WiFi).

In most cases, due to frequency hopping and very short messages used by WirelessHART, the maximum transmit power should be used to enable long distances between nodes. However, it may be beneficial for the transmit power to be reduced, in high-density networks.

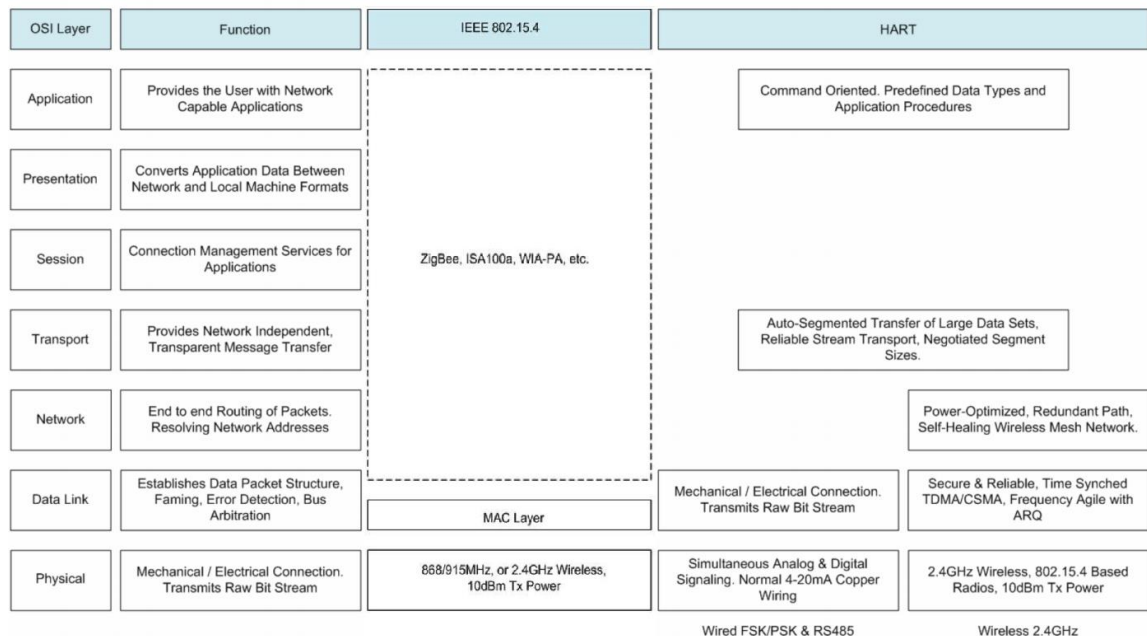


Figure 3-6. HART and IEEE 802.15.4 in OSI model

3.2.1.1 Empirical test

A dedicated test with 1000 motes split into eight WirelessHART networks was run in a small laboratory. At its peak, the total traffic was 181 packets/s in this network and the network maintained better than 99.9% reliability. However, the path stability dropped from 80% to 68% when the motes did not send data. Overall, interfering networks with high traffic increase latency and power for the network, but reliability remains above 99.9% [6].

3.3 MAC and data link layer

The data link layer is the second lowest layer in the OSI model of computer networking. Its purpose is to provide reliable transmission of data frames between two entities connected by a physical layer. It is responsible for setting up the connection, dividing data into frames and handling acknowledgments given that the reliable transfer is used. WirelessHART supports both best-effort (used mainly for data publishing) and reliable transfers (used for network management).

Establishment and structure of Data Link Layer Protocol Data Unit (DLPDU) is also included in IEEE 802.15.4 atop of the physical layer. WirelessHART employs only some of the features introduced in this recommendation, for example, all the DLPDUs are of type 'data' without the proposed encryption; WirelessHART uses its own specific frame types and security.

3.3.1 DLPDU types

The data link layer specification [15] defines five possible WirelessHART frame types:

1. Data DLPDU
2. Keep-Alive DLPDU
3. Advertise DLPDU
4. Acknowledgment DLPDU
5. Disconnect DLPDU

Data DLPDUs contain data in transit to a final destination device. This is the only packet whose payload originates from the upper layer; the network layer is responsible to forward the packet to its destination between neighbouring devices. All the others packet types are exclusively Data Link Layer packets.

The Keep-Alive DLPDUs are used, as needed, for network maintenance, i.e. network time synchronization, to confirm connectivity of a neighbour or neighbour discovery. Keep-Alive DLPDUs have an empty payload.

The Advertise DLPDU is used to invite new devices into the network. When a device wishes to join a network, it listens for these DLPDUs and then uses the information contained to synchronize with the network and initiate the join process. The payload includes basic network information: absolute slot number and security information.

Acknowledgment DLPDUs are used to manage packet flow through the network if a non-broadcast reliable packet was issued.

Disconnect DLPDUs are used to advise neighbouring devices that the device is leaving the network.

3.3.2 Media Access Control

At the Medium Access Control (MAC) layer, which is a sublayer of data link layer, WirelessHART utilizes Time Division Multiple Access (TDMA) to ensure contention free transmission. One time-slot is 10ms long; packet must be acknowledged in the same time-slot, this does not apply in case of broadcast packets. A collection of time slots forms a superframe. The size of superframe may vary; the superframes are then repeated at a fixed rate throughout the network lifetime.

The primary objectives of the MAC sublayer are to maintain slot synchronization, identify slots that must be serviced, listen for a packet being propagated from neighbours and propagate packet received from the network layer. To accomplish this, the device includes and maintains:

1. Tables of neighbours, superframes, links, and graphs
2. Link scheduler
3. Packet propagation control state machine

The tables configure the communication between the device and its neighbours; they are typically populated by the Network manager or when a neighbour is discovered. Link scheduler, along with propagation control state machine, evaluates these tables and chooses next slot to be serviced by listening for a packet or by sending a packet. Each link is associated with a set of channels used for frequency hopping.

The number one priority of the MAC sublayer is to propagate packets enqueued in the device's buffers. Its next priority is to receive packets from neighbouring devices [15]. MAC protocol data unit is a payload of PHY packet and its formatting is depicted in Figure 3-5.

Due to channel diversity provided by frequency hopping, the time slot can also be shared by different nodes. In case of collision at the destination node, the source devices use the random back-off mechanism to wait for new opportunities of transmission. For each unacknowledged transmission, the waiting window size is increased. This is similar to other contention-based channel access schemes. Broadcast messages, however, are not allowed on shared time slots.

3.3.3 Implementation on a silicon

Differences in MAC layer and mostly using the TDMA model in which all network nodes are synchronized are making it difficult to implement WirelessHART mote on IEEE 802.15.4 radios, due to the fact that most vendors only expose the MAC layer Application Programming Interfaces (APIs) to upper layer development [14]. This issue was solved with Dust Networks' (now part of Analog Devices) system on chip (SoC) LTC5800-WHM which is compliant to WirelessHART standard [16] not only in PHY layer but also in data link layer, more information about the chip can be found in chapter 4.2.1.

3.4 Dynamic and device variables

The HART protocol is designed to support smart field device technology on the legacy 4–20 mA current loop. It is a command based protocol with master-slave token passing, this means that the field devices-slaves are only responding to network manager-master commands (this does not apply when the device is in burst mode, described in the following chapter). Commands 1 and 2 return the Primary variable (PV), Loop current and Percent range. Command 3 adds to these the Secondary, Tertiary and Quaternary variables (SV, TV, QV respectively). Collectively these are called the Dynamic variables. WirelessHART device does not have to have an analogue channel, thus Loop current and Percent range variables does not have to be supported in such a device.

In addition, the protocol supports Device variables for use in more sophisticated smart field devices and multi-variable field devices. Furthermore, multi-variable field devices can configure which Device variable to connect to the current loop.

Device variables are uniquely defined data items within a field device providing process-related information or merely device information (battery life). A Device variable's value changes as its connected process varies. A code number is assigned to each Device variable and this assignment must never be changed for a given device type. Field devices must not support more than 240 Device variables. Device variables should be numbered consecutively starting from zero [17].

Each Device variable represents a direct or indirect connection between the process and the field device. An indirect connection infers the process state or level via calculations using direct process values from other Device variables (see Figure 3-7).

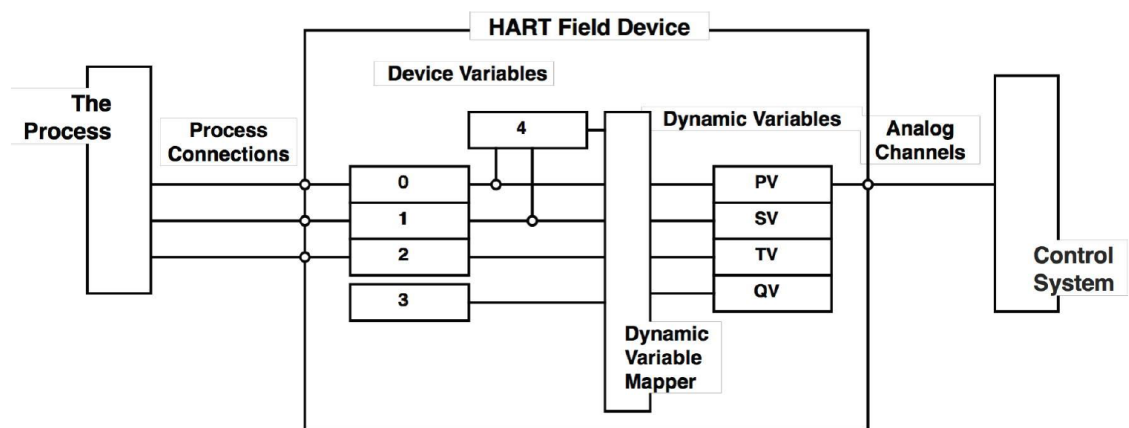


Figure 3-7: Device and dynamic variables [17]

In any case, a field device has a number of Device variables. Up to four of these Device variables are mapped to the Dynamic variables using the appropriate Device variable number by the Dynamic variable mapper, whose settings may be fixed (not exposed to the user) for simplicity.

3.5 Burst mode

The protocol supports publishing of cyclical process data (standard HART messages) using burst messaging. In this mode, a device is instructed to publish the response to a specified HART command continuously without any further master or host action. If a field device supports Burst mode then all of commands 103-105 and commands 107-109 must be implemented, for all supported Burst messages. Every Burst message can have its own trigger; when trigger conditions are met, the device enters burst mode automatically.

Each Burst message must allow a different configuration. In other words, each Burst message may have a different command, trigger, update period, set of Device variables etc. The device must retain Burst mode settings through a device reset, self-test or the power being removed and re-applied.

Burst messages are defined for wireless and wired HART interface, but they should not be supported when wired HART is only used as a maintenance port.

The command 109 is used to enter Burst mode, the field device next issues publish service request (currently called ‘timetable request’ in HART 7) to network manager. Since it may take some time to allocate requested service, the network manager starts Delayed Response (DR) sequence and after allocating the resources for the publish service, it grants the field device bandwidth in the network. The field device will then complete the DR sequence by returning a response to the application indicating that command 109 is completed and Burst mode is on, this sequence is depicted in Figure 3-8. Once initiated, the field device will publish data indefinitely until it is instructed to stop [18].

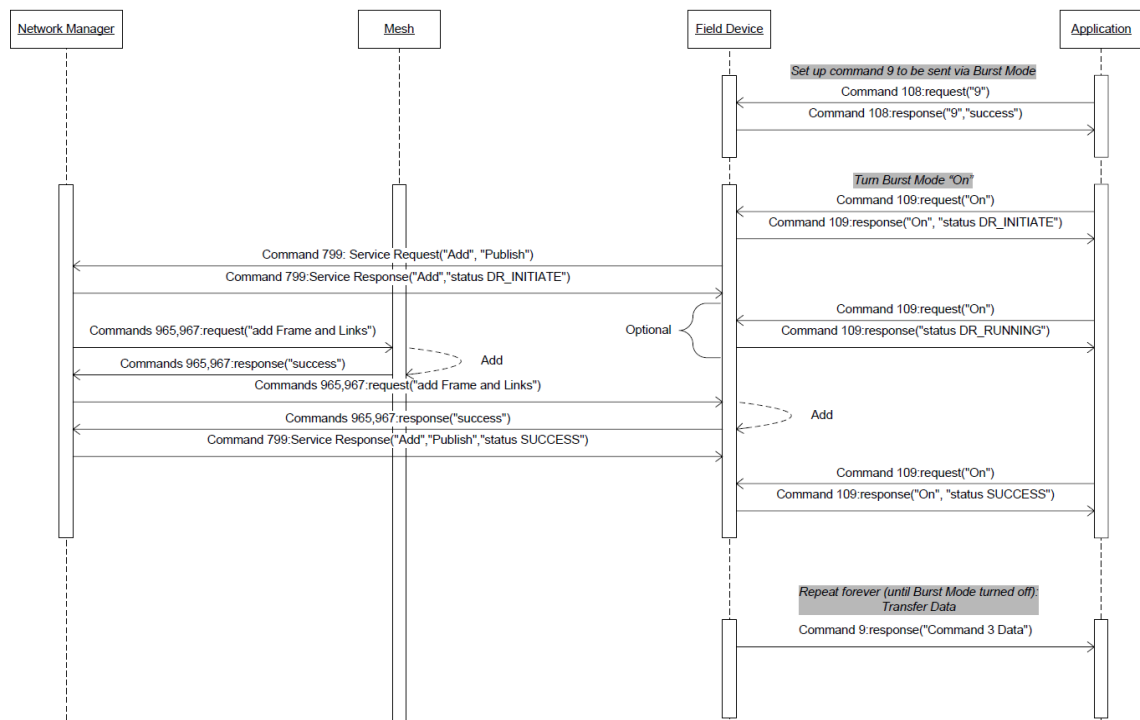


Figure 3-8. Burst enabling sequence [18]

3.5.1 Configuring a device for Burst Mode operation

The publishing period is the only limitation in configuring the burst. Allowed span is from 0.1s to 3600s (1hour) and defaults to 60s.

The procedure a master should follow to place a field device into burst mode is as follows:

1. The command response is configured using Command 108, Write Burst Mode Command Number. Commands 1, 2, 3, 9 and 48 must be available for publishing and Command 33 must be supported if it is implemented in the field device.
2. For Commands 9 and 33, Command 107, Write Burst Device Variables, is used to assign Device Variables to the response data slots. Command 9 supports up to 8 slots and Command 33 supports up to 4.
3. Use Command 54 to determine the acquisition rates for the desired Device Variables. Use Command 103 to set the Update Period and Maximum Update Period for publishing the Burst Message (see Subsection 6.9.2). The Data-Link requirements may result in the Burst Message being published more frequently than the Update Period.
4. Set the Trigger Mode for the Burst Message using Command 104. Burst Messages may be configured to publish continuously or based on a Trigger Level. The Burst Message will be at all times published at least as often as the Maximum Update Period.
5. Issuing Command 109, Burst Mode Control, will enter or exit Burst Mode. While in Burst Mode, the field device will begin transmitting the responses to the command number set by Command 108 based on the data-link requirements and the properties configured using the other Burst mode commands.

4 SENSOR NODE DESIGN

To get a closer look at how a WSN is built, an insight into a sensor node is to come first. Basic functional units are the same regardless of the protocol. Every sensor node must have a sensing unit, which converts observed phenomenon into digital domain where processing unit manages acquired data. Transceiver unit connects the node to the network. Finally, the power unit regulates supply voltages of the node and can be extended by power harvesting units such as solar cells or piezo-electric elements.

The main purpose of this sensor node design is to demonstrate possibilities that Analog Devices can offer to complete design of WirelessHART field device. In order to ensure long battery life, the strongest emphasis is put on effective power management and low-power microcontroller firmware. Block diagram of the proposed demo module is shown in Figure 4-1. All the main blocks are detailly described in the following chapter.

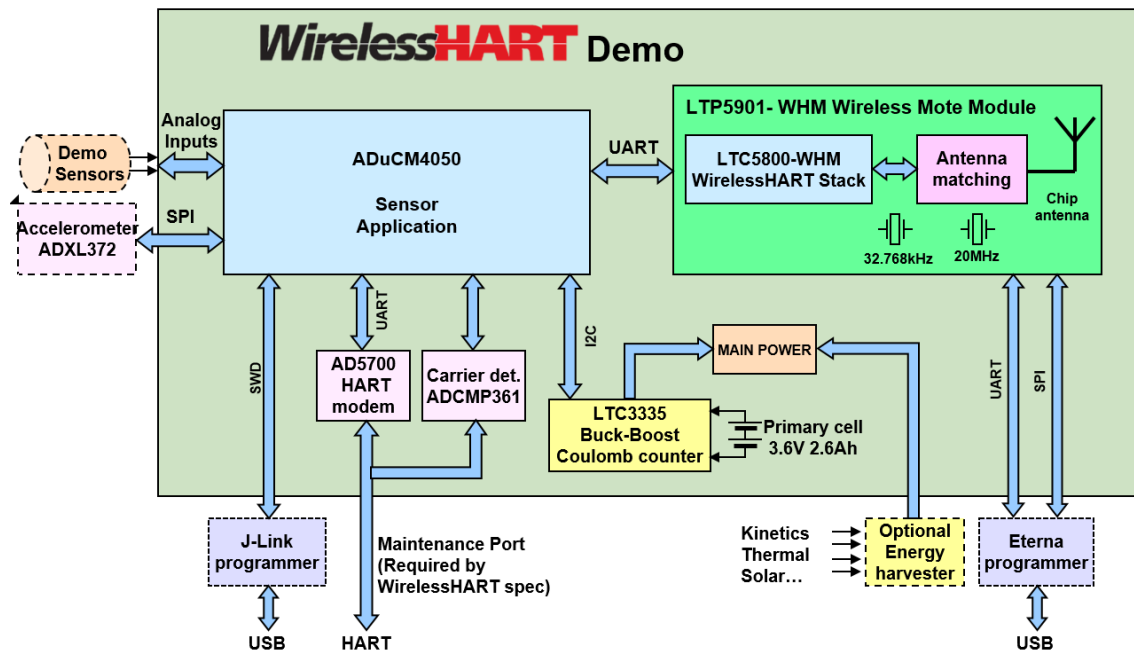


Figure 4-1. WirelessHART sensor mote demo – block diagram

4.1 Maintenance port

All WirelessHART field devices must provide maintenance port that complies with the requirements in the Token-Passing Data Link Layer Specification [19], which is used dominantly for device start-up configuration (e.g. loading the network key, network ID or monitoring the join process); after joining the network, wireless and wired interfaces become equal and the field device must accept commands received from both interfaces. When requested, the field device answers identity commands normally thus allowing the hosts to load the field device's device descriptor.

Masters connected to the maintenance port do not have access to the wireless network. The maintenance port should not support burst mode and must not be permanently wired [18].

4.1.1 AD5700-1 HART modem

The AD5700-1 is a single-chip solution, designed by Analog Devices and specified to operate as HART FSK half-duplex modem complying with the HART physical layer requirements [20]. It integrates all the necessary filtering, signal detection, biasing, modulating and demodulating and signal generation functionality, thus requiring few external components. The AD5700-1 contains accurate carrier detect circuitry, integrated RC oscillator and uses a standard UART interface. In order to reduce the board space and bill of materials (BOM), the integrated oscillator is used to clock the device.

The modem has the ability to operate in two input modes: with external or internal analogue input filter and DC bias. While the external filter provides significant robustness against electrostatic discharge (ESD) it has increased component count; 6 passive components compared with only 3 when using internal filtering. In this case, experimental input circuit was designed to combine advantages of both options, it uses an external filter to guarantee ESD robustness and internal DC bias to decrease component count.

4.1.2 HART carrier detector

Despite the low-power design of AD5700-1, the input current can be as high as 400 μ A across the temperature operating range when the internal oscillator is used, furthermore, the WirelessHART device can expect commands via maintenance port only occasionally during its lifetime (typically during initial configuration), hence it is desirable to enable the AD5700-1 only when a valid signal is present. With the aim of detecting HART signal, a simple carrier detector can be built, which would interrupt the application processor so it could power-up the FSK modem, transmit the HART response and turn the modem back off. Every HART message must have preamble time >40 ms, which gives us margin big enough to power-up the modem before the start of the message. The proposed circuit is shown in Figure 4-2.

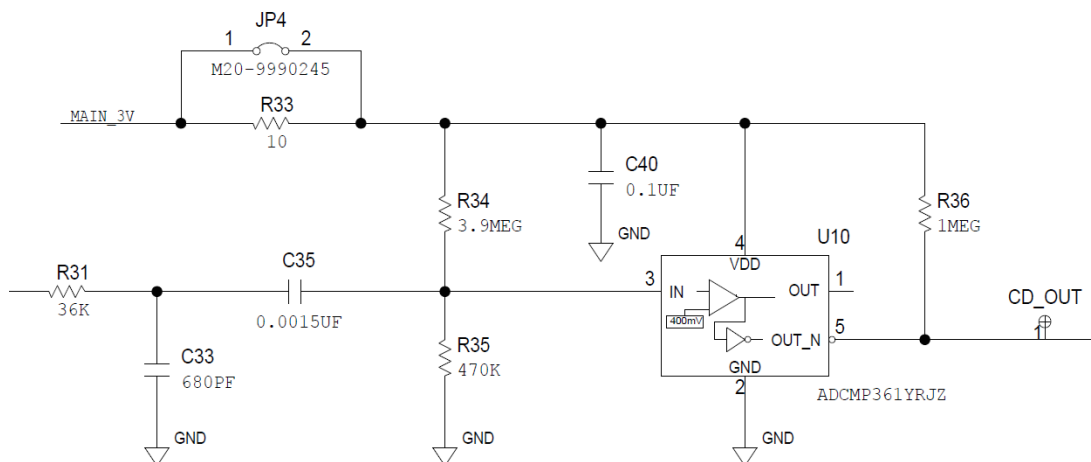


Figure 4-2. HART carrier detector circuit

HART signal is applied at R31 and then AC coupled to the input of ADCMP361 comparator. The ADCMP361 is ultra-low power comparator with an integrated 400 mV reference, dual outputs, and typical quiescent current only 6.5 μ A. Resistors R34 and R35 set DC bias a little below 400 mV. The comparator triggers its output after first half-wave of HART carrier, this signal (CD_OUT) is used to wake-up the MCU, using this mechanism we can achieve significant power savings. Power consumption of this circuit can be measured on current sensing resistor R33.

Performance evaluation of this carrier detector is depicted in Figure 4-3. ‘HART’ signal is connected to maintenance port input/output and ‘CD_OUT’ is connected to external interrupt of the MCU. HART preamble consists of multiple ‘0xFF’ bytes, which are frequency shift keyed as 1.2kHz sine signal. CD_OUT goes low immediately after its first half-wave, giving the MCU enough time to wake up and HART modem to successfully power-up.

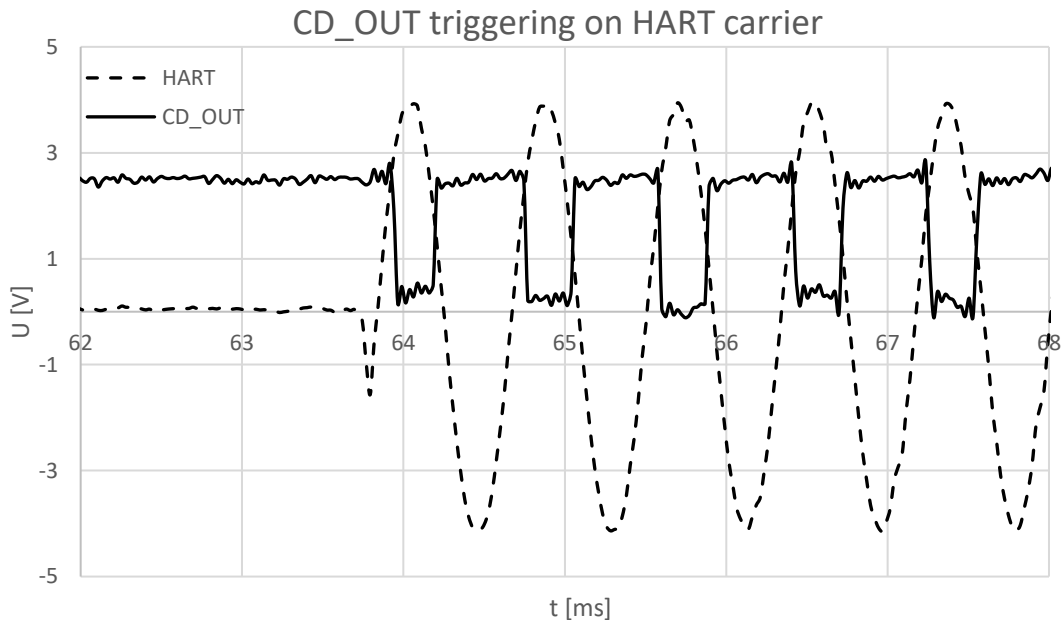


Figure 4-3. Carrier detector function

4.1.3 HART master

HART is a master-slave protocol; therefore, a HART master is needed to thoroughly test and verify proposed HART circuits in both physical and application layers. For testing and verification purposes FieldComm Group, the owner of HART specifications, provides a HART test system, which is a Linux based PC providing all the necessary stimulus for the purpose of testing HART and WirelessHART devices in all stages of development, including invalid stimulus that the field device must handle.

Undergoing all specified HART tests is not within the scope of this project, so only a very simple HART master will be used. The physical layer is built around already mentioned HART modem AD5700 (see Figure 4-4), which is complemented with galvanic insulation and USB connection to PC. The PC then serves the application layer and simulates the HART master.

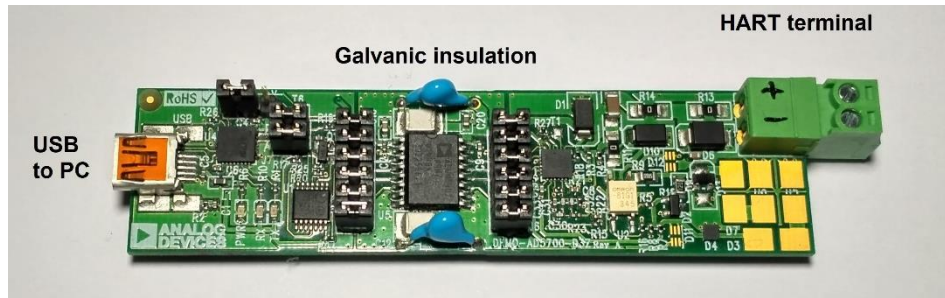


Figure 4-4. DEMO-AD5700-D3Z, HART master

HART commands and their respective responses can be issued and displayed in a user-friendly LabView application (Figure 4-5), provided by Analog Devices Inc and used with their kind approval.

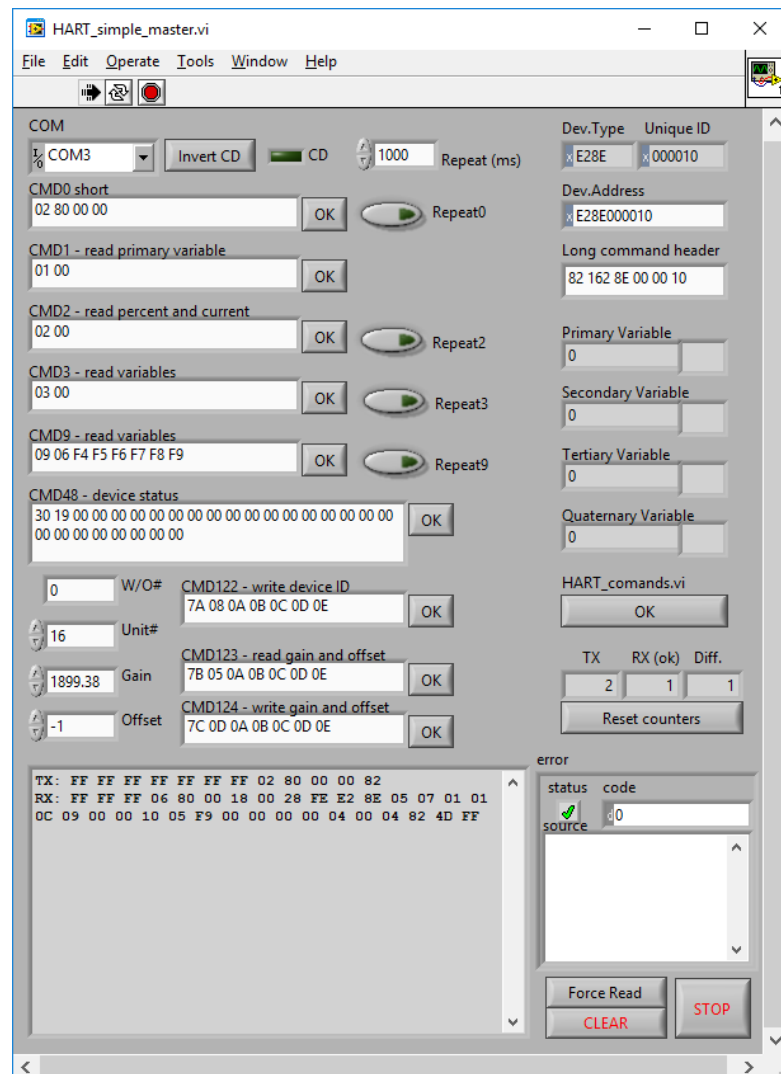


Figure 4-5. Simple HART master used for wired HART interface debugging

4.2 WirelessHART interface

During normal operation, only the wireless interface will be used to connect the sensor node to outside world. When connected to WirelessHART network, apart from publishing data, the node can be fully configured through this interface.

4.2.1 LTC5800 WirelessHART Mote-on-chip

The LTC5800-WHM is a system on a chip serving as a wireless mote in WirelessHART network [16], it incorporates low power IEEE 802.15.4 radio with an integrated power and low-noise amplifier, FLASH memory, as well as an ARM Cortex-M3 microprocessor running WirelessHART SmartMesh networking software in a compact 72-pin 10mm x 10mm QFN package. Moreover, many application commands, regarded to network control and management are implemented in networking software, which is readily configured via a software application programming interface (UART with flow control). With such a high level of integration, building a complete WirelessHART field device is significantly simplified.

4.2.2 LTP5901 – WHM module

Enhancing an existing wired HART design with wireless connectivity is not a trivial job, although both power amplifier for transmitting and low noise amplifier for receiving are integrated into the LTC5800 chip. However, the radio front-end is not complete without an antenna and its matching, for stable operation even more is needed: correct crystals, robust power-line filtering against EMI and chip shielding to suppress radiated noise. With the aim of reducing design time, radio modules, tested for compliance with IEEE 802.15.4, with all the features mentioned above were introduced to the market. In the first stage of the project, the WirelessHART mote module LTP5901-WHM with chip antenna will be used.

The module is shipped with blank FLASH memory; the user must load the SmartMesh software into the mote prior first use. Figure 4-6 depicts mounted module on a PCB being programmed via Eterna flash programmer.

4.2.3 External power and low noise amplifier

The LTC5800 includes external power and low noise amplifier support (PA, LNA) when the internal PA and LNA are bypassed and discrete amplifiers must be built in order to extend operating range. Firstly, building custom amplifiers and complete radio front-end including antenna microwave matching would require undergoing complete IEEE 802.15.4 Physical Layer Qualification Testing; this is not in the scope of this project, therefore internal PA and LNA will be used. The LTP5901-WHM module comes with PHY Layer Qualification Testing certificate, attached in appendix E.

Secondly, there already is published reference design, that illustrates correct external amplifiers connection and its performance evaluation. The reference design is available at Analog Devices webpage [21].

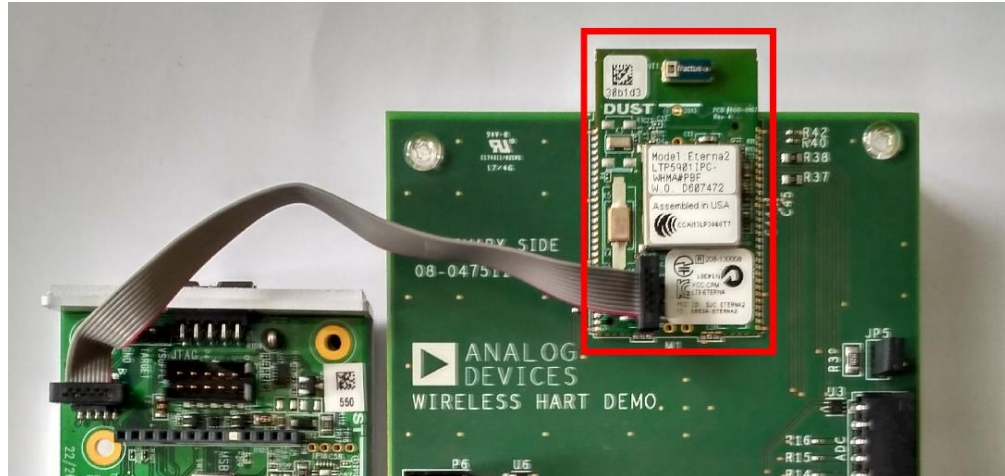


Figure 4-6. LTP5901-WHM (in the red rectangle) 42x24mm connected to Eterna programmer

4.3 Power stage

The power stage of the mote is built around Linear Technology (now part of Analog Devices) LTC3335, nano power buck-boost DC/DC converter with integrated coulomb counter. It has very low input quiescent current (<680 nA) and high (up to 90%) efficiency which make it ideal for ultra-low power applications [22]. Its integrated precision coulomb counter monitors accumulated battery discharge in long battery powered application and can inform the microprocessor if the battery is almost drained. This is particularly useful when batteries with flat discharge characteristic (such as Li-SOCl₂) are being used and the current discharge state cannot be determined from the output battery voltage. To maintain a small footprint and to attain long battery life a high energy density primary lithium thionyl chloride battery with low self-discharge Saft LS14500 was selected. It is available in standard AA-size with nominal capacity 2600 mAh and output voltage 3.6 V.

The converter can be programmed through standard I²C interface. The output voltage (1.8 – 5 V), input peak current and coulomb counter can be both set and read on the run. The goal is to run the node at a lowest possible voltage to cut down the power consumption. We are limited by means of the HART modem AD5700-1 which needs at least 2.65 V to operate correctly, therefore the main power rail is set to 2.8 V.

The mote can be also powered from the debugger (if connected) or externally; the external connection is intended for the purpose of connecting energy harvesters and evaluating their performance.

4.4 Microcontroller selection

The microcontroller (MCU) is the heart of the sensor node, controlling all the other components, therefore it must be chosen with extra care. The main focus is once again put on power management and low input current, while maintaining desired computing power. In order to control all the node's subsystem, the MCU must have peripherals listed

in Table 4-1. Floating point unit (FPU) is not compulsory, however, all device variables in HART specifications are single precision (32bit) floating point numbers; using dedicated FPU unit can considerably decrease the active time of the processor, which results in significant power savings.

Table 4-1. List of used MCU peripherals and their usage

MCU peripheral	Usage
UART 1	Interface for AD5700-1 HART modem
UART 2	Interface for LTP5901-WHM WirelessHART mote
I ² C	Interface for LTC3335 Coulomb counter
2x SPI	Interface for external sensors or ADCs
Flexi RTC	Time precision sensor sampling (SensorStrobe)
FPU	Floating point operations
Flash	HART non-volatile settings
Timers	Main Finite State Machine
Power control	Integrated Buck converter
GPIO	General purpose signals

The ADuCM4050 is a microcontroller from the ultra-low power family built around ARM Cortex-M4F core (up to 52 MHz), it comes with 512 kB of embedded flash memory and 128 kB of SRAM [23]. It integrates all the requested peripherals listed in Table 4-1. A Power Management Unit (PMU) consist of integrated switched capacitor DC-DC buck converter and advanced hibernate modes, which facilitate low power consumption even when all peripherals are running.

If analogue sensors are sampled, an external precision analog-to-digital converter (ADC) should be used, as the integrated 12-bit successive-approximation-register ADC does not reach precision $<0.1\%$, which is typically required in industrial applications.

It is available in 64-lead LFCSP (Low Flat Chip Scale Package) and in 72-ball WLCSP (Wafer Level Chip Scale Package). The LFCSP package was chosen for this design, as the GPIO count (44) is sufficient, and finally, WLCSP packages are not suitable for prototyping phase of the project and would increase complexity and cost of the following printed circuit board.

4.5 SensorStrobe – precise time sampling

A precise sampling of sensors synchronized to an accurate time base is a vital requirement if it is desirable to transfer it precisely to the frequency domain using Fast Fourier Transformation (FFT) in a signal processor. This is particularly necessary when evaluating accelerometer's data regarded to vibrations, as magnitude measurements must be complemented with evaluation of dominant vibrational frequencies.

In the traditional approach, the MCU's firmware generates a general-purpose input/output (GPIO) pulse, triggering the sensors at specific intervals to collect sensor data. The traditional approach has two issues. The approach involves considerable software overhead, which increases current consumption and takes processor time. The

triggering of a pulse depends on the firmware of the MCU and, thus, can drift as time progresses.

SensorStrobe is a mechanism developed by Analog Devices, Inc., that recognizes the low power, consistent, and synchronized data acquisition from sensors. The SensorStrobe directly addresses the issues of the traditional software approach by working in hibernate mode and using dedicated hardware to generate sampling triggers even during software execution. This results in >10x lower in current consumption, while acquiring data using SensorStrobe mechanism [24].

4.5.1 Accelerometer ADXL372

The ADXL372 is ultralow power sensor device combining a 3-axis microelectromechanical system (MEMS) accelerometer, low-pass and high pass digital filter with external trigger and 512-sample first in, first out (FIFO) buffer to store sensor data [25]. This large FIFO saves a significant amount of power because the MCU can be hibernated while the accelerometer records data in the FIFO buffer. Accelerometers registers can be accessed via a serial peripheral interface (SPI) in multibyte mode, in addition, the FIFO buffer can be drained using direct memory access (DMA) while the processor is idle to save even more power. The ADXL372 can be configured to trigger an interrupt, in selected cases, including reaching set FIFO watermark.

In this project, the accelerometer will be used for measuring vibrations with high amplitude (ADXL372 maximum range is $\pm 200g$), therefore it is vital to locate them on separate shaker boards (see Figure 4-7), so the vibrations would not damage the mote's electronics. The ADXL372 exceeds in its output data rate, which maximum value is 3200 Hz. According to Nyquist-Shannon sampling theorem, this allows us to diagnose vibrations with maximum frequency as high as 1600 Hz, which is sufficient for the demo. Motor vibrations are in most cases below 1 kHz, bandwidth ~ 1.5 kHz is deemed as low to mid-frequency vibration range [26].

The ADXL372 is a smart sensor with high integration, for instance, it autonomously stores the peak values of acceleration in all three axis since the last read of the register. This is particularly useful when frequency analysis is not required, in this case, the host MCU only polls for max peak values, which significantly reduces processor active time, thus extends battery life.



Figure 4-7. ADXL372 shaker board

4.5.2 System description

The SensorStrobe mechanism supported by ADuCM4050 is combined with external trigger feature and an integrated FIFO buffer of the ADXL372 to collect the sensor data at the lowest possible power consumption

The accelerometer is configured for measurement mode with external trigger and to interrupt the MCU when the FIFO watermark is reached. The SensorStrobe mechanism is part of ADuCM4050's real-time flexi counter (RTC1); the MCU configures it to generate triggering pulses at the desired frequency even when the MCU is in hibernate mode.

On every pulse, the ADXL372 takes a sample and stores it in the FIFO buffer. When the FIFO upper watermark is reached, the accelerometer interrupts the MCU via the external interrupt pin. The MCU can initiate DMA transfer to drain the FIFO and enter low-power Flexi mode when the core is idle and the peripherals (SPI and DMA) are running. Following the DMA completed interrupt, the microprocessor can process acquired data. This sequence is depicted in Figure 4-8:

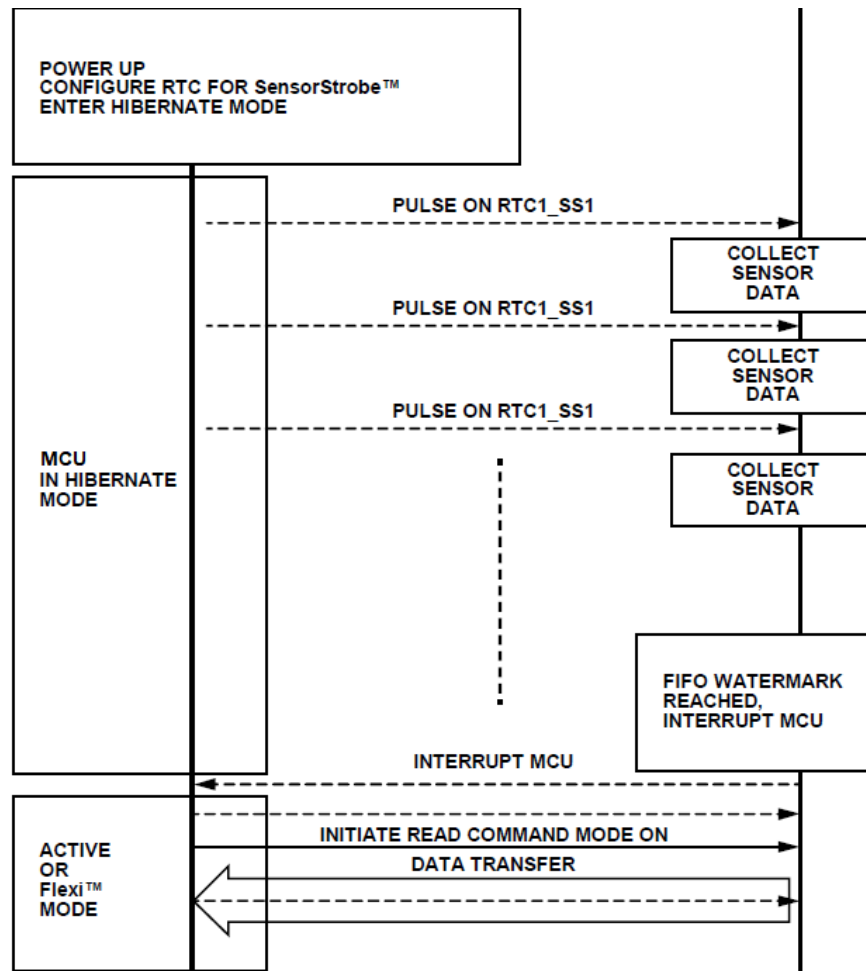


Figure 4-8. SensorStrobe mechanism timeline [24]

5 NODE'S HARDWARE AND FIRMWARE

In the following chapters, strong emphasis is put on describing specific solutions implemented in the node's custom hardware and firmware, and on practical methods used for the device testing.

The firmware does not aim to be fully compliant with HART 7 specifications, although a significant amount of compulsory functionality is supported:

1. Support all Universal Commands
2. Support Common practise commands regarding burst control (publishing)
3. Fully implement services and application domains

5.1 HART commands

The HART protocol provides specifications to facilitate two-way communication between field devices and host applications. The protocol's application layer is command based. In other words, commands from master or slave devices are the basis for HART communication and the command number, embedded in the communication, determines the content of the message. The command number indicates the specification of a unique, unambiguous packet of data with a fixed byte count [17].

The command set is divided into the following classes:

1. Universal commands. Must be implemented in every device.
2. Common practise commands. Non-compulsory, but strongly advised.
3. Wireless commands. Must be implemented in every wireless device.
4. Device and family specific. A different set of commands for every device classification (e.g. pressure, temperature, valve/actuator, flow, etc.)
5. Non-public commands. These commands are defined by device vendor and should not be used in field operation.

5.1.1 Commands aggregation

Command aggregation allows, with some limitations, multiple commands to be transported in a single transaction. This allows a host or device to aggregate/encapsulate multiple commands into a single transaction (command) for low overhead, faster communications. Using this approach, for example, multiple read commands can be aggregated in one transaction by a host application. This is especially useful when reading device configurations. Aggregated commands are a fundamental capability of WirelessHART and supported in all traditional HART communications using common practice command 78.

The device must parse each embedded command in the request sequentially and generate appropriate responses. The only limitation is, that the overall response cannot be

longer than specified HART data packet (94 bytes in wireless). This firmware does not support HART command 78, therefore commands can be aggregated only via a wireless interface.

5.1.2 Commands termination

The LTC5800 provides the wireless connectivity to the mesh network and terminates a portion of the WirelessHART command set. The HART commands required of a field device must be terminated by either the mote (LTC5800) or the microprocessor (ADuCM4050). In general, the mote terminates HART commands associated with the Wireless command specification [27], specifically commands number 768-1023 (Network health report, Neighbour report...) and Wireless device-specific commands number 64512-64765, that are appropriate to a field device. The application microprocessor is responsible for terminating all other HART commands (Read message, Configure burst...).

It is important to draw to reader's attention that command response *Command not implemented* is a valid command termination.

5.1.2.1 Mote terminated commands

The mote terminates the wireless command set (specified in HCF_SPEC-155) that are appropriate for a field device, as opposed to a network manager or gateway. Most of the HART commands terminated by the mote are related to wireless operation of the field device. The mote can receive HART commands from two different sources, depicted in Figure 5-1a:

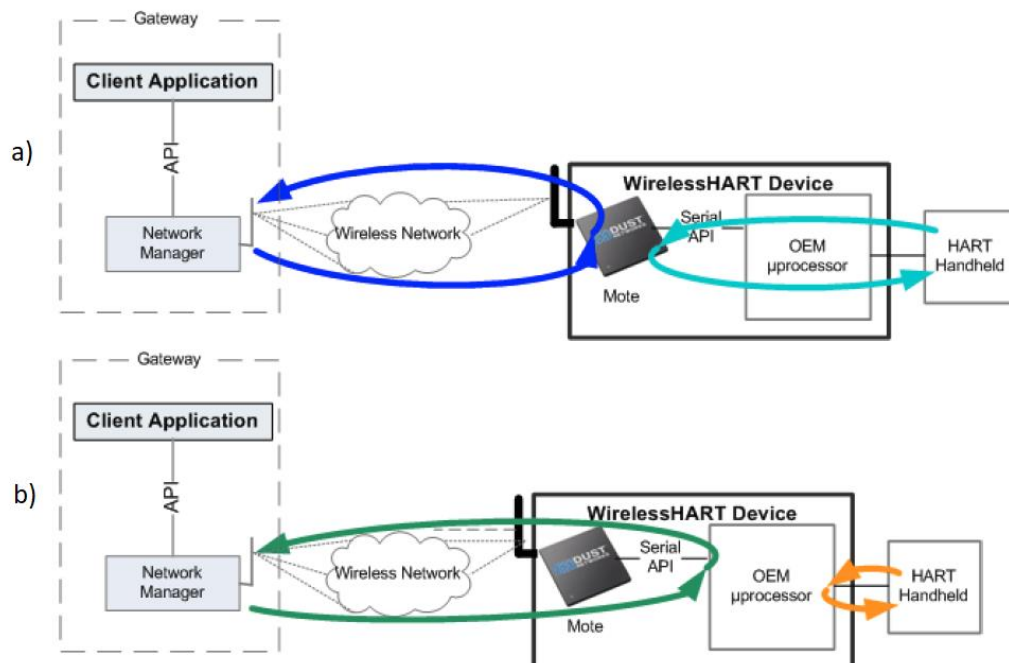


Figure 5-1: HART commands termination: a) mote terminated b) application microprocessor terminated [28]

1. Over wireless from the manager, either via gateway session or network manager session.
2. Over the mote's serial API from microprocessor: command originated from HART maintenance port (handheld device) and passed to mote for termination.

5.1.2.2 Microprocessor terminated commands

The microprocessor is responsible for terminating the non-wireless HART commands, which are not contained in HCF_SPEC-155. These commands are typically related to identification, device variables or device settings; commands implemented in this firmware are listed in Table 5-1. The microprocessor can receive the HART commands from two different sources depicted in Figure 5-1b:

1. Over serial API connection to mote, via *dataReceived* notification. This type of communication is typically reliable and the microprocessor must respond with *send* command, effectively sending the response to the manager.
2. Via HART maintenance port.

Table 5-1. Supported HART commands

Command number	Description	Details
0	Read unique identifier (UID)	
1	Read primary variable	
2	Read loop current and percent of range	Loop signalling not supported
3	Read dynamic variables and loop current	Loop signalling not supported
6, 7	Read/Write loop configuration	Loop signalling not supported
8	Read dynamic variable classifications	
11, 21	Read UID associated with Tag/Long Tag	
12, 17	Read/Write message	
13, 18	Read/Write Tag, Descriptor, Date	
15	Read device information	Only demo data
22	Write Long Tag	
33	Read device variables	
42	Perform device reset	Device disconnects then reset
54	Read device variable information	No time stamp support
72	Squawk	On, Off and Squawk once (500ms) supported
73	Find device	Hold BOOT button and issue this command
103, 104, 107, 108, 109	Write burst configuration	Burst trigger not supported
105	Read burst configuration	

5.2 HART non-volatile settings

According to HART 7 specifications, all write commands must be persistent. That means that the settings cannot be reset to its default value after switching the device on and off, after losing power or after device reset. The wireless set of these non-volatile variables (Network ID, Join Key, MAC address ...) is saved in and managed by the LTC5800. Among other non-wireless variables, all Bursts and Events settings must be persistent, which means that the device will automatically reconfigure itself to the state prior reset or power loss.

All the non-volatile settings are saved in the MCU's last page of flash memory in a structure of type *hart_flash_memImage_t*, code defining this memory space is disclosed below:

```
//===== typedef =====

//==== burst settings struct
typedef struct
{
    uint8_t mode;                // Burst on/off
    uint8_t slot[8];             // HART request, used with CMD 9 & 33
    uint16_t cmdNum;             // HART CMD number, that will be used
    uint32_t updatePeriod;       // in 1/32 s
    uint32_t maxUpdatePeriod;    // in 1/32 s
    uint8_t triggerMode;         // only continuous mode is supported
    float triggerLevel;
    uint8_t triggerClass;
    uint8_t triggerUnit;
} burstSettings_t;

//==== configuration changed counter struct
typedef struct
{
    uint16_t value;              // value of configuration changed counter
    uint8_t flags[4];           // 4 flags for 4 masters (2wired+2wireless)
} cfgCounter_t;

//==== structure of non-volatile data saved in flash
typedef struct
{
    uint8_t writeProtect;        // write protect flag
    uint8_t writeProtectPass[8]; // write protect password
    cfgCounter_t cfgCounter;      // cfg counter settings
    burstSettings_t burst[HART_BURST_NUM]; // bursts settings
    eventSettings_t event[HART_EVENT_NUM]; // events settings
    uint8_t longTag[32];         // ISO latin chars
    uint8_t shortTag[6];         // 8x packed char
    uint8_t descriptor[12];      // 16x packed char
    uint8_t date[3];             // day, month, year-1990
    uint8_t message[24];         // 32x packed char
    uint8_t pollingAddr;         // used with CMD0
    uint32_t finalAssemblyNum;    // 24bit wide
} hart_flash_memImage_t;
```

5.2.1 Write protect mode

The above-mentioned code defines non-volatile variable *writeProtect*, effectively utilizing write protect mode, when set to '1'. The write protect mode provides a method for preventing changes to the device's configuration. This mode can be enabled and disabled using hardware jumpers, but in this case, it is software lock with password defined in *writeProtectPass*. The user can enter or exit the mode via HART device-specific command number 122. The microprocessor then issues a *Lock* command to the LTC5800 mote, so it would not accept any write (configuration changing) commands, i.e. Write network ID, Write join key, etc...

When a write protected device receives a write command, it must respond with error code number 7: In write protect mode. Independent of the write protect mode, all read commands must operate normally.

5.3 Firmware description

The firmware for application microprocessor is vital for all processes taking place in the device. Integrated development environment Keil uVision 5 was used to create source code in embedded C and to generate resulting binary files, which were loaded into the MCU's flash memory using Segger J-Link debugger.

As a result of all UART peripherals being used in the application and no semihosting support in Keil, debugging messages are sent via J-Link and can be displayed on PC in J-Link RTT Viewer application.

Analog Devices, the manufacturer of used microcontroller ADuCM4050, provide low-level drivers (Device Family Package) which significantly decreases design time and are heavily used in this project, the latest version can be found on ADI website or in Keil Package Manager.

The serial API between the mote and the application processor is asynchronous (both devices may start a transmission independently at the same time) acknowledged (every request and response must be acknowledged) with flow control and error control to ensure that every packet was received and processed. Sequence number for every packet (similar to TCP/IP protocol) is provided to ensure correct receiving of every packet. Linear Technology provides open-source C-library [29] for easier implementation of this thorough communication protocol in the end-user application. This library is with minor changes and bug fixes used in this project.

5.3.1 Microcontroller initialization

The microprocessor's firmware is not a standalone application; the LTC5800 system on a chip mote must be present and pre-configured in order to guarantee proper system initialization. The pre-configuration of the mote includes loading the very fundamental non-volatile variables, necessary for joining the WirelessHART network (Join Key, Network ID, and most importantly MAC address) via the Eterna programmer (Figure 4-6). The power-up sequence depicted in Figure 5-2 starts with successful initialization of all microcontroller memory and peripherals; then the microprocessor loads the MAC address via serial API from the mote and the system is ready for operation, if the MAC

address is not a valid WirelessHART (starts with 3-byte header '00-1B-1E') address, the program exits with an error.

Secondly, the microprocessor loads state of the mote's internal finite state machine. In the event that the device was just powered-up, the mote will be in IDLE state, in this case, the microprocessor starts pre-join sequence and sets variables needed for joining the WirelessHART network (HART device info and status), issues Join command and waits (in low power mode) for join.

In case of microprocessor boot after reset, the mote's state can be various. In the event that the mote is in join process, the microprocessor will wait for join process termination (with serial API *operational* or *joinFailed* notification). In case of join fail, the processor will try to re-join, after three consequent failed joins, the device shuts down to save battery life.

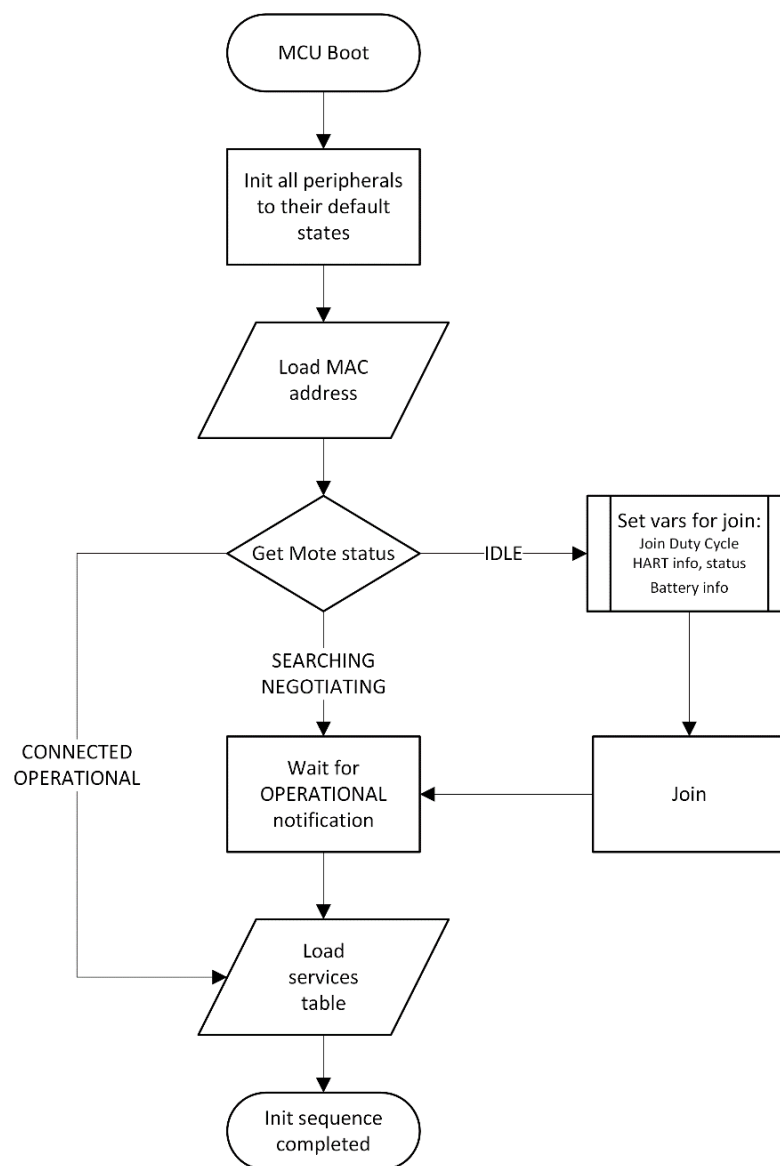


Figure 5-2. MCU initialization sequence

If the mote is already connected to a WirelessHART network after microcontroller boot, the microprocessor loads allocated radio resources (services, also called time-tables in HART ver. 7) from mote to its local memory, completing the initialization sequence.

Although the service table in mote might be blank, every WirelessHART device has automatically guaranteed ‘maintenance’ service by the network manager, which is used for device setup utilizing HART commands. This service, however, cannot be used for data publishing (HART bursts). The device must request dedicated radio resources, in particular, the ‘publish’ service, in order to periodically publish data. In addition to ‘maintenance’ and ‘publish’ services, services of type ‘events’ (used for irregular data notifications) and ‘block transfer’ (used for large data packets) are defined, but not implemented in this firmware.

5.3.2 Burst enabling

The burst enabling sequence, depicted in Figure 5-3, is executed when one of the following conditions is met:

1. Microprocessor successfully completed initialization
2. HART command 109 was received

The burst settings are non-volatile, as a result, the microprocessor must check if the burst was not enabled prior the microcontroller boot, in case of power loss or soft reset. Similarly in the latter case, the HART command 109 is cable of enabling or disabling the burst and the microprocessor must act accordingly.

If the burst is enabled, the microprocessor must check if there is an active publish service for this burst in the service table. Normally, the service is inactive, because the mote saves service table in SRAM region, so it is blank on power-up, it can be active however, if only the microprocessor was reset and the mote retained its state. If the service is inactive the device must request it with parameters defining the publishing period and wait for the service approval before entering the burst mode. Naturally, the service might have been already active and the device can enter burst mode directly. This sequence is repeated for every supported burst; thus, every enabled burst must have its dedicated publish service with possibly different publishing period. In the burst mode the microprocessor configures internal real-time clock, that will periodically wake-up the device for burst publishing. Lastly, it must configure the sensing device to correctly acquire its measured data.

In the case of disabling the burst, the device must delete the publishing service to allow the network manager to effectively manage radio resources, then it must disable the publishing timer, consequently exiting the burst mode.

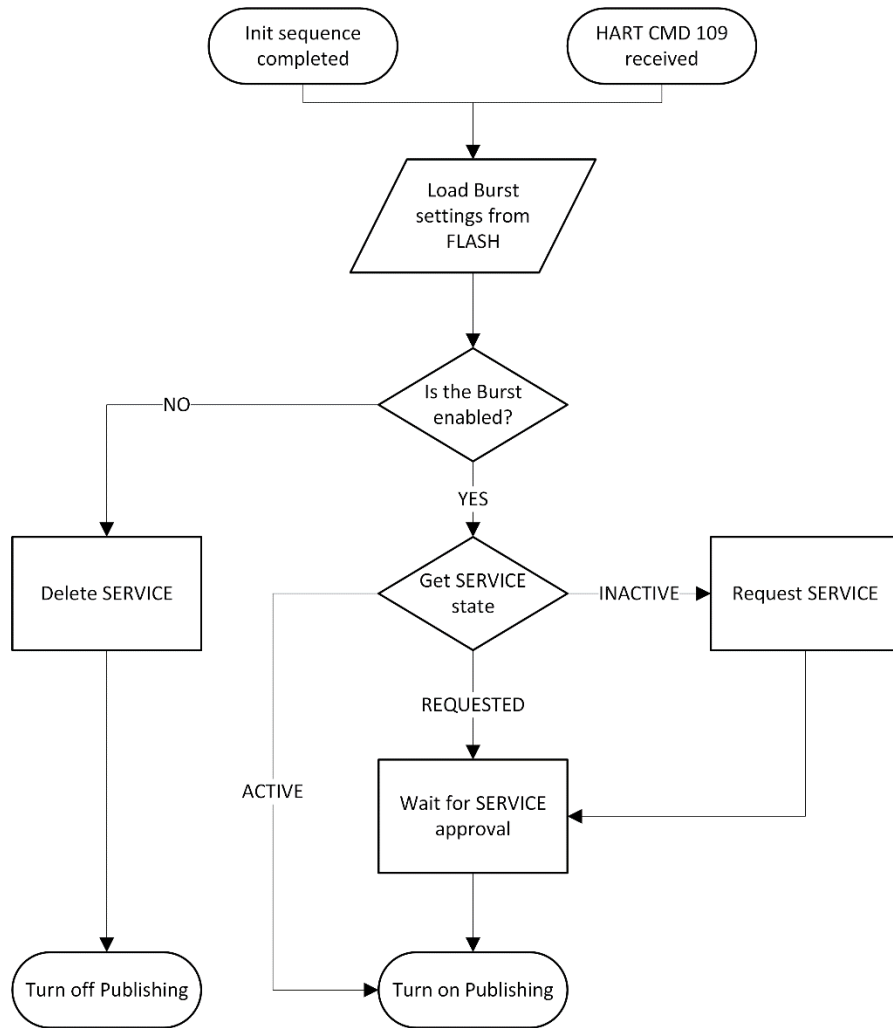


Figure 5-3. Burst enabling sequence

5.4 WirelessHART mote testing

The FieldComm Group (FCG), sells two kits – HCF-Kit 192 (a Linux based system for testing wired HART protocol) and HCF-Kit 193 (the wireless test hardware and test scripts), and a multi-channel packet sniffer called WiAnalys. Using this setup, full HART compliance can be tested and verified.

The FCG test kits were not available for full duration of this project, therefore similar but simpler setup (see Figure 5-4) was created to thoroughly test wireless and wired communication ports. The setup consists of the device under test (DUT), WirelessHART network manager, HART modem, and PC. The application running on PC, described in chapter 6, is capable of setting-up the device, issuing supported HART commands and displaying their responses and burst data (i.e. device status or sensor data).

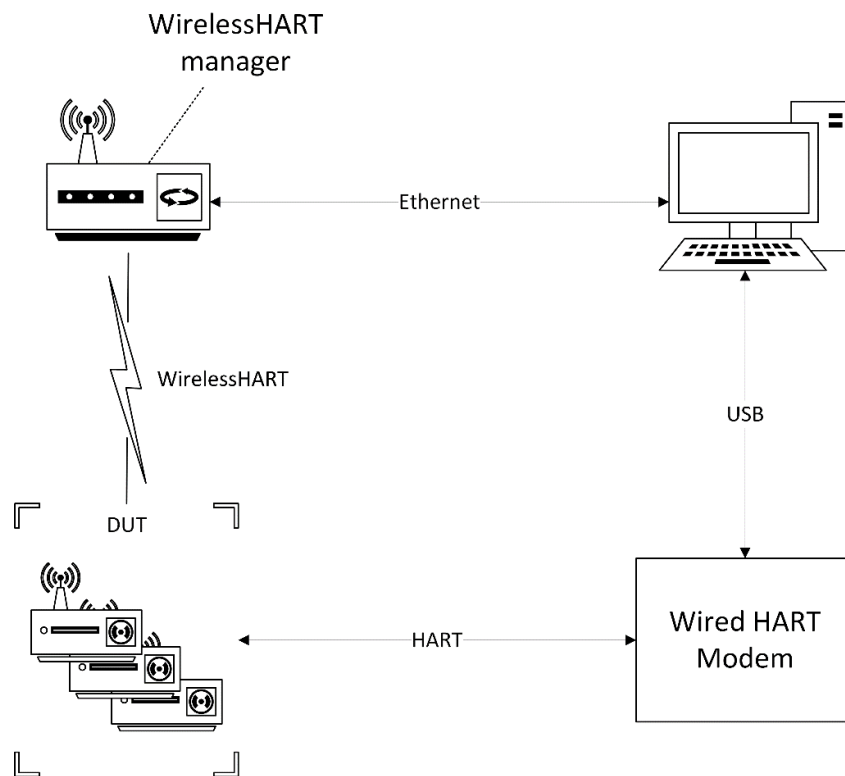


Figure 5-4. WirelessHART mote testing setup

5.5 Mote implementation

In the first stage of the project the main functional blocks, in particular, the microcontroller with accelerometer and the two HART interfaces, were connected into one functional block to verify the proposed circuit. All the key components have their evaluation boards which make it easier to prototype and debug the system.

Finally, the prototype circuit was coupled with HART carrier detector and battery power stage, described in chapter 4.1.2. and 4.3 respectively, to form the final WirelessHART node demo schematic. Routing of the printed circuit board (PCB) was performed by layout department of Analog Devices Inc.

A photo of the resulting PCB connected to the rest of the test and debugging system is shown in Figure 5-5 (battery holder is on the bottom side of the PCB). Complete schematic and PCB masks files can be found in appendix B and C, the Bill of Materials (BOM) in appendix D.

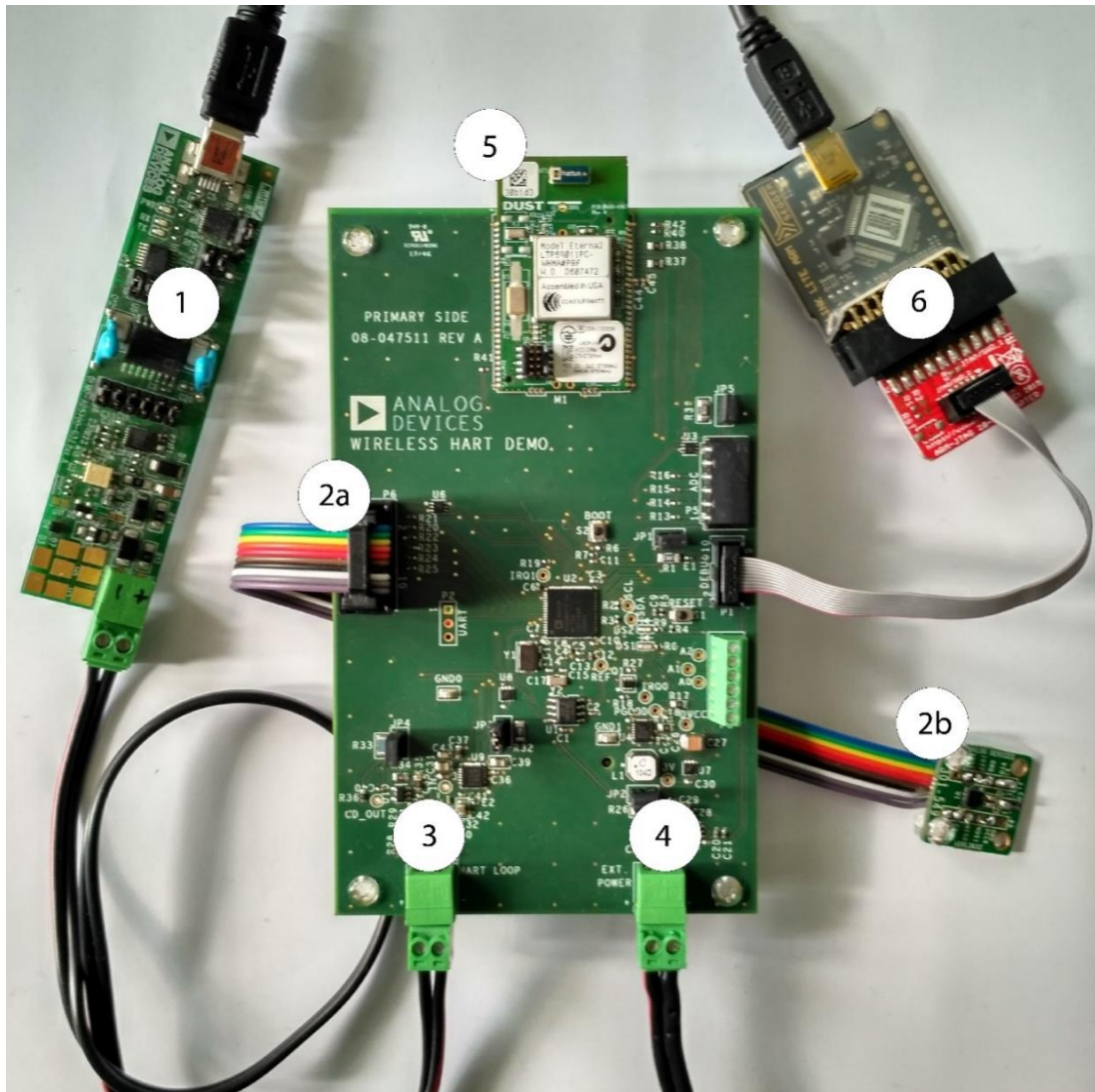


Figure 5-5: Assembled system

Legend to Figure 5-5:

1. DEMO-AD5700-D3Z, simulates HART handheld device
2. MEMS accelerometer ADXL363 shaker board (b) with its flat cable connector (b).
3. Mote's maintenance port
4. External power supply connector
5. WirelessHART radio module mounted on custom PCB
6. Segger J-Link programmer

5.6 Operating range

Operating range is one of the most important parameters of every radio system. Despite not focusing the main attention on the radio front-end in this thesis, a brief evaluation of radio performance is necessary.

The maximum possible operating range that can be theoretically achieved only in a laboratory line of sight conditions, with a low noise radio environment, no obstacles in the first Fresnel zone and perfect antenna alignment can be calculated from logarithmic transmission equation 5.1:

$$P_r = P_t + D_t + D_r + 20 \log \left(\frac{\lambda}{4\pi d} \right) \quad (5.1)$$

Where P_t is the power delivered to the transmit antenna, D_t and D_r are directivities (sometimes called gains) of the transmit and receive antennas in dBi. λ is a wavelength that is used for communication over distance d . Finally, P_r is available power at the receive antenna terminal. In the following calculations, also the antennas effectivity will be taken into account, as utilizing small chip antennas has a significant impact on the radio performance. List of relevant radio parameters with enabled power amplifier and packet error rate 1% is in Table 5-2. It can be stated, that the power balance is very similar for down-link (manager to mote) and up-link (mote to manager), although the link budget is tighter in up-link due to worse managers receive sensitivity, so it will be used for maximum theoretical range calculations.

Table 5-2: System's radio parameters

Parameter	Manager	Mote
Transmit power	8 dBm	8 dBm
Receive sensitivity	-90 dBm	-93 dBm
Antenna effectivity	-2dB (80%)	-2.5dB (75%)
Antenna directivity	1 dBi	1 dBi

In equation 5.1 the received power will be substituted with receive sensitivity as that is the desired output with only one variable input parameter – distance d . For up-link parameters in Table 5-2, the maximum operating range becomes $d_{uplink} = 590 \text{ m}$. The maximum range that can be expected in noisy indoors environment is typically 13-15x lower [16] than the theoretical in free space, resulting in range 40-45 m.

The proposed system was tested to prove its radio performance. The maximum distance between the mote and the manager, while attaining 1% packet error rate was ~25m. Every reliable packet that is issued, while the link is down, is not acknowledged, so the application is aware of its failed reception. It will autonomously try to re-send the packet, when the link is back on, in other words, when the mote is placed back in the operating range.

The used WirelessHART module is also available with MMCX antenna connector, that can be used for connecting to a more effective dipole antenna. With this enhancement, the operating range can be brought closer to the theoretical limits, specifically ~100m indoors and ~300m outdoors [16].

6 DEMONSTRATING SOFTWARE

To fully demonstrate all supported HART functionality, a software running on PC (see Figure 5-4) must be provided. Firstly, this software is responsible for setting-up the WirelessHART nodes in the network and the network itself, secondly, it must collect and display the publish data formatted as HART messages.

The proposed Graphical User Interface (GUI) software does not need to manage the network and transport layer in WirelessHART (e.g. redundant path stability, neighbours list, packet segmentation...), because of the fact that these layers are managed by the network manager LTP5903CEN-WHM depicted in Figure 3-3. The following chapter is focused on connecting to the manager via API and implementing demo WirelessHART function in Python programming language.

6.1 Connecting to the network manager

The SmartMesh WirelessHART manager functions as the server that provides an interface between a client application and a WirelessHART network. The manager has two interfaces, the already mentioned API is intended for machine-to-machine communications (e.g. a host program communicating with the manager) and is contacted via an Ethernet network whereas, the command line interface uses serial RS232 connection and is intended for human interaction with a manager during development stage or for interactive troubleshooting. This interface is useful for the manager initialization most importantly, adjusting the manager's IP address. Only the API interface will be discussed in the following text.

In the XML API, the manager uses two channels for communication with a client: a two-way control channel that permits requests and responses between the manager and a client, and a one-way notification channel that streams data from the manager to a client [30].

1. The Control channel provides a client with a Remote Procedure Call interface for querying manager status, updating configuration information, and interacting with the mote network. The Control channel uses the XML-RPC protocol. By default, the control channel port is set to 4445.
2. The Notification channel is a persistent TCP/IP stream over which the manager sends notifications to the client in an XML format. The host system must subscribe to the notifications in order to receive them.

The first command from the host to the manager is login, therefore every application that uses the connection to the manager must successfully establish a connection with valid credentials (user: *admin* pass: *admin*), IP address and control channel port number. Welcoming window of proposed GUI is depicted in Figure 6-1 (credentials are set to the default values and are not exposed to the user).

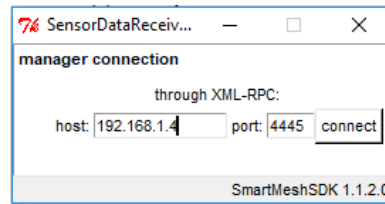


Figure 6-1. Welcoming connection window

6.2 WirelessHART manager control application

The essential purpose of the WirelessHART manager control application is to send HART compliant packets to field devices in the WirelessHART network through the network manager, effectively controlling all settings in the field device. The application has also its secondary function that can issue commands for the network manager via API to set-up the network and the manager itself.

Dust Networks provide Software Development Kit (SDK) for the network manager [29] written in Python that serves as a low-level driver for accessing the manager via ethernet connection and issuing API commands, moreover it contains a number of example applications. The SDK was used as a guideline in the manager control application development process.

After successfully establishing serial connection to the network manager and subscribing to *DataReceived* and *Events* notifications, a main application window is opened (see Figure 6-2). The first three frames (labelled ‘command’, ‘response’ and ‘tooltip’) are used to generate API request to the manager, the user can use this method to get information about network settings, motes connected, network health etc. The other four frames are used to generate a HART command and send it to the selected mote. The destination mote can be selected from a list in ‘HART motes’ frame. This list is updated every time a mote connects or disconnects. Please note that a mote can be listed as connected even though it is not responding since disconnection timeout is rather long (in order of minutes), this state can be reached whenever a mote does not send ‘disconnect’ notification prior resetting or shutting down. A broadcast message can be sent with MAC address set to ‘FF-FF-FF-FF-FF-FF-FF-FF’.

In the ‘HART command’ frame, the user can select one of the supported commands listed in Table 5-1, choose its parameters and send it to the selected destination mote by pressing the ‘send’ button. In Figure 6-2, command 109: Burst Mode Control is used to enable burst message number ‘0’ over a wireless interface. The other parameter’s values (a different burst message or bursting via wired FSK port) are not supported in this project. After receiving the response, it is decoded and the response fields are displayed in ‘HART response’ frame. The ‘HART command description’ provides additional information about the selected command.

The first three rows in Figure 6-2 are used to create requests that do not address the motes, but the manager itself. These frames are taken without any changes from already mentioned Dust Networks SDK and are used for the network initialization and health monitoring. All commands in the manager API are described in the manager’s API guide [30].

The screenshot shows the 'WirelessHART Manager Control' application. The 'command' section has a 'getTime' button and a 'send' button. The 'response' section shows the results of the 'getTime' command in a table:

getTime	
utc_time	asn_time
1522440222.66	176863863
float	8B (int)

The 'tooltip' section contains the text: 'Get the current time.' and 'Note: to enter a value in a field of type HEXDATA, please enter two digits for each byte (zero padded).'

The 'HART notes' section has a text input field containing '00-1B-1E-E2-8E-00-00-10'.

The 'HART command' section shows '109: Burst Mode Control' selected in a dropdown. Below it are two input fields: 'Burst Control Code' (set to 'Wireless') and 'Burst Message' (set to '0'). Both have a data type of 'INT8U'. There is a 'send' button at the bottom.

The 'HART response' section shows the response for '109: Burst Mode Control' in a table:

RC	Burst Control Code	Burst Message
0 (Success)	2 (Wireless)	0
INT8U	INT8U	INT8U

The 'HART command description' section contains the text: 'This command is used to enter and exit the Burst Mode' and 'Note: to enter a value in a field of type HEXDATA, please enter two digits for each byte (zero padded).'

The bottom right corner of the window shows 'SmartMeshSDK 1.1.2.0'.

Figure 6-2. Enabling burst with HART command 109

Every HART command supported by the manager control application must have, apart from its name and description, defined its request and response fields according to HART universal/common practise commands specifications. An example of code in Python defining HART command 109 (Burst mode control) is disclosed below. The code results in GUI in Figure 6-2, digestedly representing HART command request and its response.

```
### CMD 109 BURST MODE CONTROL ###
{
    'id'          : 109,
    'name'        : '109: Burst Mode Control',
    'description' : 'This command is used to enter/exit the Burst Mode',
    'request'     : [
        ['Burst Control Code', INT, 1, 'tab9'], #Common table no.9
        ['Burst Message',      INT, 1, None ],, #only burst '0'
    ],
    'response'    : {
        FIELDS    : [
            ['RC',
             INT, 1, RC ], #Response Code
            ['Burst Control Code', INT, 1, 'tab9'],
            ['Burst Message',      INT, 1, None ],,,
        ],
    },
},
```

The 'tab9' parameter indicates, the value cannot be arbitrary, but only values specified in Common Table number 9 are valid. In such a case the GUI permits only valid values and shows their specific meaning as depicted in Figure 6-2.

Furthermore, the field device can generate error response, when the command, its format or its parameters are not supported. The error response contains only one byte, the response code (RC) is substituted by error code and the command is not executed by the field device. The GUI checks the response and informs the user of such a case.

The response codes definition is an integral part of HART specifications [31]. Similarly, all standardized codes (i.e. unit codes, classification code, burst control codes...) can be found in HART Common Tables Specification [32].

6.3 WirelessHART Burst viewer

In order to provide user-friendly GUI that could display published data (HART bursts) to the user, who has only a little knowledge about the HART protocol or about the mote's functionality, a simple demo application called WirelessHART Burst viewer was created.

After successfully establishing the serial connection to the network manager and subscribing to *DataReceived* and *Events* notifications, the Burst viewer firstly creates a frame for each of the connected motes and displays its MAC address and HART Long Tag. Every frame includes four sub-frames, one for every dynamic variable. These frames are also opened or closed when *MoteConnect* or *MoteDisconnect* notification is received.

Secondly, the application issues broadcast of HART command number 123. Command 123 is a non-public command that was defined especially for Burst viewer application purposes. The response of this command contains all necessary information about the device variable: minimum and maximum values, unit and classification codes. Using HART command number 105: Read burst configuration the application loads and displays burst settings (the burst mode and burst command number) of every field device, so the user would know whether to expect burst messages or not. HART command number 72: Squawk is also issued, causing the device to blink its yellow LED to ensure correct receiving of the HART request. Lastly, command 3: Read dynamic variables is issued to load current dynamic variables values. All mentioned commands are aggregated into one broadcasted, unreliable WirelessHART packet.

Using 'Enable burst' and 'Disable burst' buttons, HART command 109: Burst mode control, the burst '0' can be enabled or disabled, but no further settings are available. Any detailed burst settings can be configured with above-mentioned application WirelessHART manager control. The 'Blink' button can be used to easily distinguish the motes from each other. The device blinks its yellow LED for 500ms upon receiving command 72: Squawk with parameter *Squawk once*.

The Burst viewer decodes HART bursts and displays received values, upon *DataReceived* notification from the manager. The data is displayed in corresponding sub-frame numerically and with a slider along with its units and min/max values, the GUI with three active motes is shown in Figure 6-3. The application does not include any higher logic, it merely shows published data in a readable format and sends HART requests upon user's action.

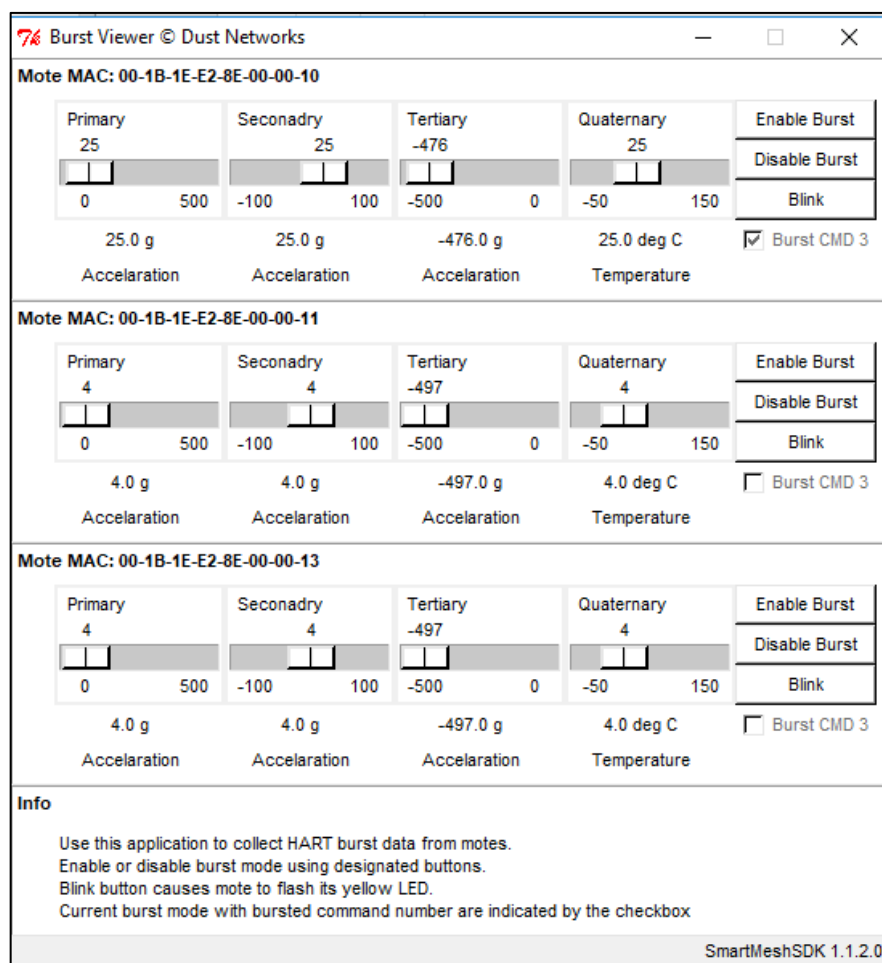


Figure 6-3. HART bursts viewer

Out of HART commands regarded to device variables only a small subset can be used for publishing; burst enabled commands with their response description are listed in Table 6-1.

Table 6-1: Burst enabled HART commands

Command number	Description	Returned values
1	Read primary variable	Primary dynamic variable
2	Read loop current and percent of range	Primary variable, Loop current, Percent range
3	Read dynamic variables and loop current	Loop current and four dynamic variables
9	Read dynamic variables with status	Up to 8 selected device variables with status, unit and classification
33	Read device variables	Up to 4 device variables with unit

7 CONCLUSION

With the intention of building WirelessHART network demo, WirelessHART protocol was briefly described and requirements for both hardware and software of the demo field device (sensor node) were defined with a strong focus on extreme low-power design and acquiring data from digital accelerometer.

Following a careful review of the proposed schematics, the node was assembled and its hardware was thoroughly tested to ensure correct voltage regulation, transmitting and receiving of wireless packets and detecting carrier at maintenance port. The microprocessor's firmware is an inseparable part of the mote. It facilitates all of the described HART functionality by means of executing or terminating HART commands. At this point in time, the node supports all universal (mandatory) commands and all commands regarded to bursting, including automatic bandwidth request in the network prior bursting. In addition, a few selected common-practice commands are supported, for instance, Perform device reset, Find device or Squawk, which are described in chapter 5.1.

Although the main goals of the project were met, there is considerable space for further improvements. From a hardware point of view, a toggle switch should be added for disabling the battery regulator, when the device is not in use for longer period of time. This way the battery is cut off and can serve longer compared to deep-sleep modes that are entered at the present time. After a brief discussion in chapter 4.2.3 it was decided to omit external RF amplifiers from the project.

The microprocessor's firmware can also be upgraded with added support of *BlockTransfer* services. The block transfer is especially useful when publishing a large set of data, without congesting the network in *Publish* domain; in this particular case, it could be used for transferring FFT processed data from the accelerometer to determine dominant frequencies in vibration measurement.

The demo system is complemented with two PC applications that interact with the network manager and thus with the network. The first is intended for users with basic knowledge of HART protocol and is capable of setting-up the mote via HART commands send wirelessly to the mote. The second one is used as demonstrating application, merely setting-up the burst mode and displaying motes in the network with their post-processed data of measured values in a user-friendly manner.

Another simple application is provided to facilitate communication with the device via wired HART interface, so-called maintenance port, this is typically used for loading the initial values (e.g. network credentials) into the device or for troubleshooting when the device fails to join the network or behaves unexpectedly. This application is provided with kind approval of Analog Devices Inc.

To summarize, the designed system is able to demonstrate broad possibilities that are provided by the newest WirelessHART specifications and the state-of-the-art System-on-Chip LTC5800 in conjunction with Analog Devices' extreme low-power microcontroller and digital smart sensors. Combining these components into one sensor node results in reliable, robust and smart field device capable of operating from battery potentially for years, additionally serving as routing node in the mesh network.

Bibliography

- [1] H. M. A. Fahmy, Wireless Sensor Networks, Vols. Concepts, Applications, Experimentation and Analysis, Springer, 2016.
- [2] B. Deshmukh, "Industrial Wireless Sensor Network Market Analysis and Forecast Report | 2022," 23 6 2017. [Online]. Available: <http://www.satprnews.com/2017/06/23/industrial-wireless-sensor-network-market/>. [Accessed 6 12 2017].
- [3] J. T. Adams, "An Introduction to IEEE STD 802.15.4," in *Aerospace Conference* , Big Sky, MT, USA, 2006.
- [4] H. M. Ammari, The Art of Wireless Sensor Networks, vol. 1: Fundamentals, Springer, 2014.
- [5] Certiology.com, "Network Topology," [Online]. Available: <http://www.certiology.com/computing/computer-networking/network-topology.html>. [Accessed 6 12 2014].
- [6] Linear Technology, "SmartMesh WirelessHART Application Notes," 2017.
- [7] D. Chen, M. Nixon and A. Mok, Why WirelessHART, Springer, 2010.
- [8] Emerson Process Management, "System Engineering Guidelines IEC 62593 WirelessHART," 2016.
- [9] Emerson, "Industrial Wireless Technology," [Online]. Available: <http://www.emerson.com/en-us/expertise/automation/industrial-internet-things/pervasive-sensing-solutions/wireless-technology>. [Accessed 21 December 2017].
- [10] FieldComm Group, "HART Technology - Leading the digital transformation," 2017. [Online]. Available: <https://www.fieldcommgroup.org/sites/default/files/technologies/hart/HART%20brochure%20web%20view.pdf>.
- [11] HART Communication Foundation, "Network Management Specification," 2012.
- [12] Phoenix Contact, "WirelessHART," Phoenix Contact, [Online]. Available: https://www.phoenixcontact.com/online/portal/web/main/products/technology_pages/subcategory_pages/Industrial_wireless_wirelessHART. [Accessed 18 April 2018].
- [13] S. Mohanty, "Energy Efficient Routing Algorithms for Wireless Sensor Networks and Performance Evaluation of Quality of Service for IEEE 802.15.4 Networks," 2010. [Online]. Available: https://www.researchgate.net/publication/47737848_Energy_Efficient_Routing_Algorithms_for_Wireless_Sensor_Networks_and_Performance_Evaluation_of_Quality

- of_Service_for_IEEE_802154_Networks. [Accessed 21 11 2017].
- [14] D. Chen, M. Nixon, S. Han, A. K. Mok and X. Zhu, *WirelessHART and IEEE 802.15.4e*, Busan, Korea: IEEE, 2017.
 - [15] FieldComm Group, *TDMA Data Link Layer Specification*, 2016.
 - [16] Linear Technology, *LTC5800-WHM Datasheet: SmartMesh WirelessHART Node*, 2013.
 - [17] FieldComm Group, "Command summary specification," 2016.
 - [18] HART Communication Foundation, "WirelessHART Device Specification," 2008.
 - [19] HART Communication Foundation, "Token-passing data Link Layer Specification," 2012.
 - [20] Analog Devices Inc., *AD5700/AD5700-1 Datasheet: Low Power HART Modem*, 2016.
 - [21] Analog Devices, "SmartMesh Extended Range," [Online]. Available: http://www.analog.com/media/en/technical-documentation/product-information/SMARTMESH_EXTENDED_RANGE_LTC5800_WITH_SE2438T.zip. [Accessed 30 March 2018].
 - [22] Linear Technology: Nanopower Buck-Boost DC/DC with Integrated Coulomb Counter, *LTC3335 Datasheet*, 2015.
 - [23] Analog Devices Inc., *ADuCM4050 Datasheet: Ultra Low Power ARM Cortex-M4F MCU with Integrated Power Management*, 2017.
 - [24] Analog Device Inc., *AN-1427 Application Note: SensorStrobe, Ultralow Power, Time Synchronized, Sensor Data Sampling*, 2017.
 - [25] Analog Devices Inc., *ADXL372 Datasheet: Micropower, 3-Axis +/-200 g Digital Output, MEMS*, 2017.
 - [26] FES Systems Inc., "Principles and practises of vibrational analysis," [Online]. Available: <https://www.irc.wisc.edu/export.php?ID=158>. [Accessed 25 March 2018].
 - [27] HART Communication Foundation, "Wireless Command Specification," 2012.
 - [28] Linear Technology, "SmarMesh WirelessHART User's Guide," 2015.
 - [29] Dust Networks, "DustCoud," Dust Networks, [Online]. Available: <https://dustcloud.atlassian.net/>. [Accessed 16 March 2018].
 - [30] Linear Technology, "SmartMesh WirelessHART Manager API guide," 2015.
 - [31] HART Communication Foundation, "Command Response Code Specification," 2007.
 - [32] FieldComm Group, "Common Tables Specification," 2016.

Appendix: Table of content

A	CD folder structure	57
B	PCB schematic	57
C	PCB masks	61
C.1	Copper masks.....	61
C.2	Silkscreen.....	62
D	PCB Bill of Materials	63
E	IEEE 802.15.4 PHY certificate	66

A CD folder structure

The enclosed CD is inseparable part of the diploma thesis, it contains design files of the proposed PCB and source codes in C and Python of described firmware and software.

The folders on the CD and their descriptions are in the following table:

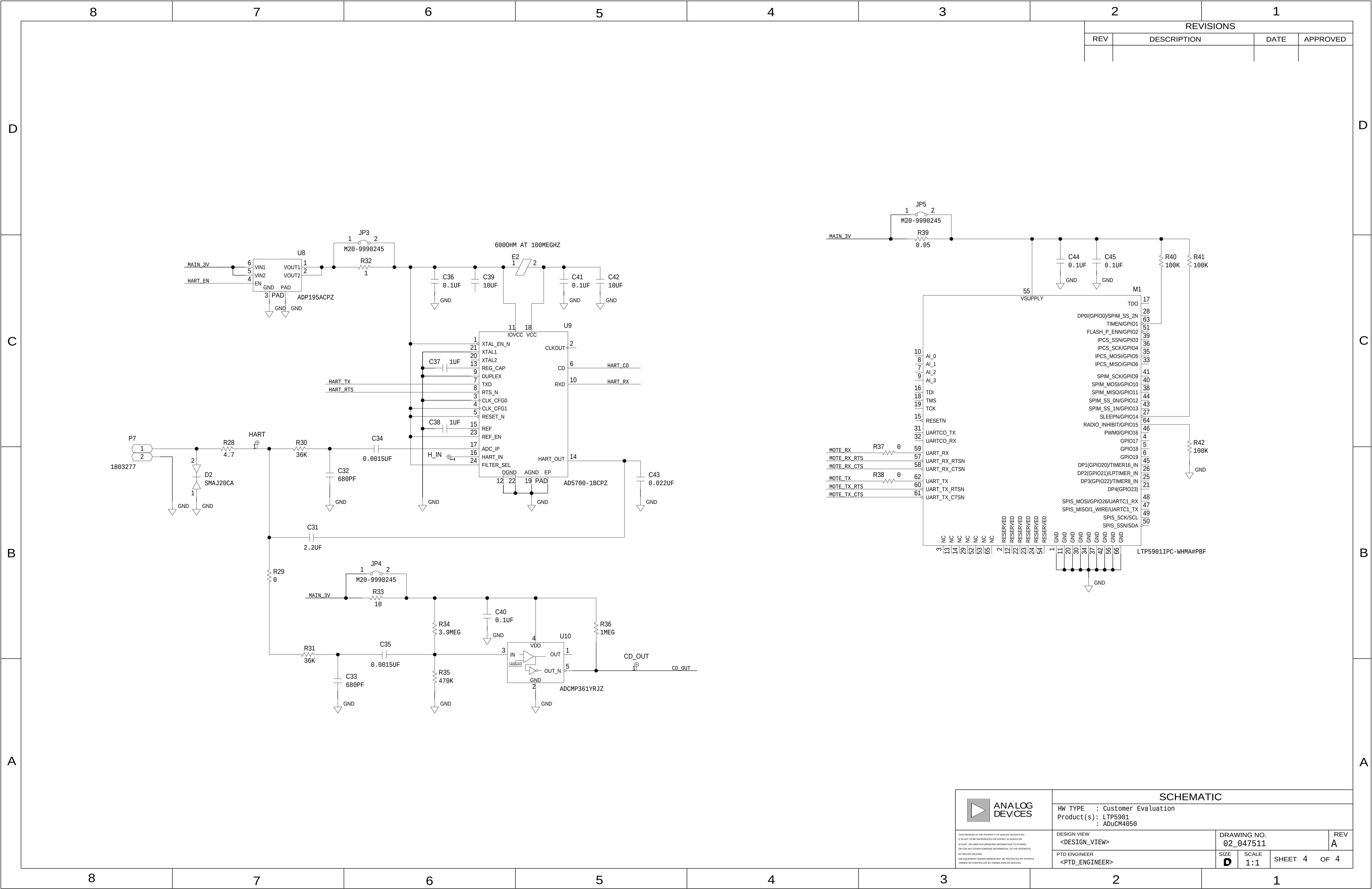
Path	Description
<i>..\WirelessHART_ManagerGUI</i>	Source files for Python applications: WirelessHART Manager and Burst viewer
<i>..\WirelessHART_NodeFirmware</i>	Source files for application microprocessor firmware in C.
<i>..\WiredHART</i>	LabView application for communicating via maintenance port
<i>..\PCB</i>	Schematic, PCB masks, and BOM
<i>..\Datasheets</i>	Datasheets of used components

B PCB schematic

Page 1: Microcontroller, voltage reference, buttons, LEDs

Page 2: Power voltage regulator, connectors to external accelerometer and ADC

Page 3: HART modem with carrier detector, WirelessHART radio module

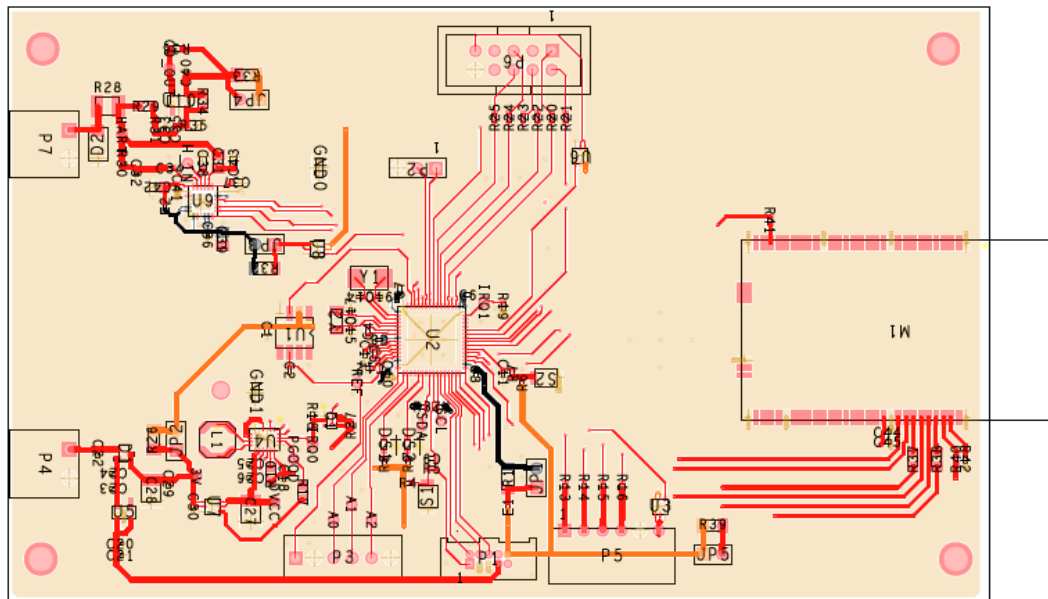


C PCB masks

C.1 Copper masks

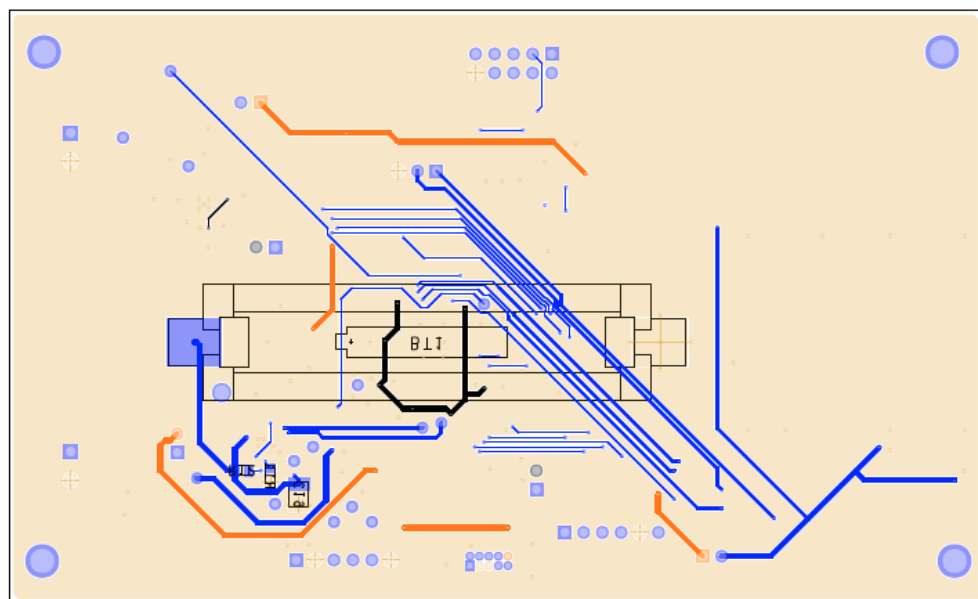
Top layer:

Scale 1:1

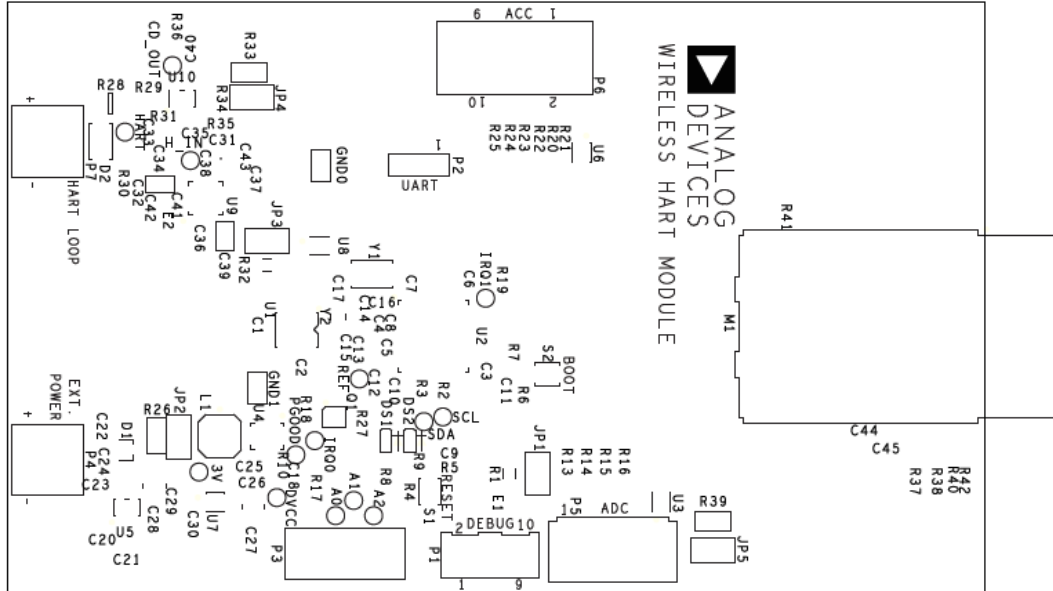


Bottom layer:

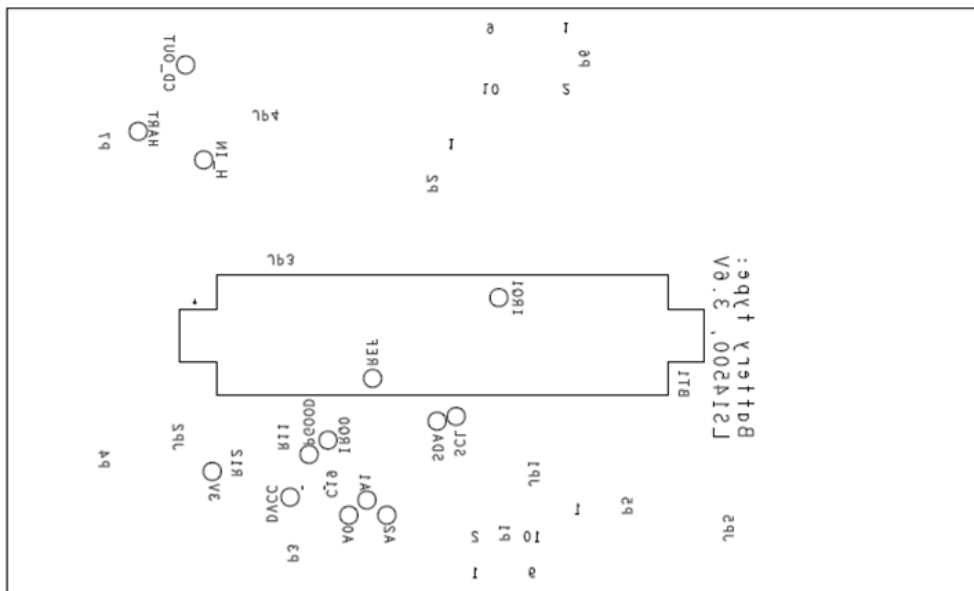
Scale 1:1



Scale 1:1



Scale 1:1



D PCB Bill of Materials

Idx	Qty	Description	Value	Package	Location	Manufacturer	Part number
1	1	PCB	?	?	?	Analog Devices	08_047511a
2	1	BATTERY HOLDER AA	1024	SKTKEY1024	BT1	KEYSTONE	1024
3	21	CAP CER X7R	0.1uF	C0402	C1-11,C20,C22-23,C25, C30,C36,C40-41,C44-45	KEMET	C0402C104K4RACTU
4	1	CAP CER JB	0.47UF	C0402	C12	TDK	C1005JB1C474K050BC
5	6	CAP CER X5R,	1uF	C0402	C13,C18,C21,C24,C26,C29	MURATA	GRM155R60J105KE19D
6	2	CAP CER NP0(C0G),	16pF	C0402	C14,C16	MURATA	GJM1555C1H160GB01D
7	2	CAP CER C0G(NP0),	8pF	C0402	C15,C17	MURATA	GRM1555C1H8R0DA01D
8	3	CAP CER X7R,	47uF	C1210-2	C19,C27,C28	MURATA	GRM32ER71A476KE15L
9	1	CAP CER X7R	2.2uF	C0805H60	C31	TAIYO YUDEN	UMK212BB7225KG-T
10	2	CAP CER C0G(NP0),	680pF	C0402	C32,C33	MURATA	GRM1555C1H681JA01D
11	2	CAP CER U2J,	0.0015uF	C0402	C34,C35	MURATA	GRM1557U1A152JA01D
12	2	CAP CER X7R	1uF	C0603	C37,C38	TDK	CGA3E1X7R1C105K080AC
13	2	CAP CER X7R,	10UF	C0805H53	C39,C42	MURATA	GRM21BR70J106KE76L
14	1	CAP CER X7R	0.022uF	C0603	C43	AVX	06035C223KAZ2A
15	1	DIODE SCHOTTKY	-	SOT323	D1	DIODES INC	SDMG0340LC-7-F
16	1	DIODE TVS	SMAJ20CA	DO214AC-4	D2	ST MICRO	SMAJ20CA
17	1	LED YELLOW	yellow	LED0603-2	DS1	OSRAM OPTO	LY L29K-H1J2-26-Z
18	1	DIODE GREEN	green	LED0603-2	DS2	OSRAM OPTO	LG L29K-G2J1-24-Z
19	2	IND CHIP FERRITE	600Ohm	L0402	E1,E2	MURATA	BLM15PX601SN1D
20	2	SMT TEST POINTS	5015	CNKEY5015TP	GND0,GND1	KEYSTONE	5015
21	5	CONN-PCB JMPR ST MALE 2P,1X M020779	M20- 9990245	M20-9990246	JP1-5	HARWIN	M20-9990245

Idx	Qty	Description	Value	Package	Location	Manufacturer	Part number
22	1	IND SHIELDED POWER	100uH	LSMSQ189H123	L1	COILCRAFT	LPS5030-104MRB
23	1	MOD SMARTMESH WIRELESSHART NODE	LTP5901IPC	MLTP5901	M1	LINEAR TECHNOLOGY	LTP5901IPC-WHMA#PBF
24	1	CONN-PCB SHROUDED HDR VERT 1.27MM	-	STD	P1	SAMTEC INC.	SHF-105-01-L-D-TH-TR
25	1	CONN-PCB TERM BLOCK 2.54MM P	1725698	CN1X6H335	P3	PHOENIX CONTACT	1725698
26	2	CONN-PCB HEADER RT 3.81MM	1803277	CNHDRRA1X2H285	P4,P7	PHOENIX CONTACT	1803277
27	1	CONN-PCB 6POS MALE HDR, 2.5MM PITCH	1778667	STD	P5	PHOENIX CONTACT	1778667
28	1	CONN-PCB 10POS SOLDER TAIL, 2.54MM	5103309-1	STD	P6	TE CONNECTIVITY	5103309-1
29	1	TRANS DUAL N & P CHANNEL FET	FDG6321C	SC70-6	Q1	FAIRCHILD SEMI	FDG6321C
30	2	RES PRECISION	1	R1206	R1,R32	PANASONIC	ERJ-8RQF1R0V
31	7	RES THIN FILM CHIP	0	R0805	R10-12,R17, R29,R37-38	MULTICOMP	MC 0.1W 0805 0R
32	10	RES GEN PURPOSE	1K	R0402	R13-16, R20-25	YAGEO	RC0402JR-071KL
33	3	RES GEN PURPOSE	10MEG	R0402	R18,R19,R27	YAGEO	RC0402FR-0710ML
34	4	RES STD THICK FILM	10K	R0402	R2,R3,R5,R7	VISHAY	CRCW040210K0JNED
35	2	RES THICK FILM CURRENT SENSING	0.05	R1206-2	R26,R39	STACKPOLE ELECTRONICS	CSR1206FK50L0
36	1	RES THICK FILM CHIP	4.7	R1210	R28	PANASONIC	ERJ-14YJ4R7U
37	2	RES GEN PURPOSE	36K	R0402	R30,R31	YAGEO	RC0402FR-0736KL
38	1	RES THIN FILM PREC	10	R1206-2	R33	TT ELEC	WINT1206LF0310R0B
39	1	RES GEN PURPOSE	3.9MEG	R0805	R34	YAGEO	2.32E+11
40	1	RES GEN PURPOSE	470K	R0805	R35	YAGEO	RC0805FR-07470KL
41	1	RES THICK FILM CHIP	1MEG	R0603	R36	VISHAY	CRCW06031M00JNEA

Idx	Qty	Description	Value	Package	Location	Manufacturer	Part number
42	5	RES THICK FILM CHIP,	100K	R0402	R4,R6,R40-42	VISHAY	CRCW0402100KJNED
43	2	RES GEN PURPOSE	470	R0402	R8,R9	YAGEO	RC0402FR-07470RL
44	2	SWITCH	B3U-1000P	SWSML118W98H69	S1,S2	OMRON	B3U-1000P
45	1	PREC MICROPOWER VOLTAGE REF,	2.048V	SO8NB-5	U1	ANALOG DEVICES	REF191ESZ
46	1	SINGLE COMPARATOR AND REFERENCE	400mV	SOT23-5L-2	U10	ANALOG DEVICES	ADCMP361YRJZ
47	1	ULTRA LOW POWER ARM M4F MCU	-	QFN64_9X9	U2	ANALOG DEVICES	ADUCM4050BCPZ-U1
48	4	LOGIC, HI-SIDE P SW	-	DFN6_2X2	U3,U6,U7,U8	ANALOG DEVICES	ADP195ACPZ-R7
49	1	NANOPOWER BUCK- BOOST DC/DC	-	QFN20_4X3	U4	LINEAR TECHNOLOGY	LTC3335EUDC#PBF
50	1	CMOS LIN REG	3.3V	TSOT-5	U5	ANALOG DEVICES	ADP160AUJZ-3.3-R7
51	1	HART MODEM	-	QFN24_4X4	U9	ANALOG DEVICES	AD5700-1BCPZ-RL7
52	1	CRYSTAL SMD	16MHz	YSML197W126H59	Y1	NDK	NX5032GA 16MHZ AT-W
53	1	IC XTAL LOW PROFILE	32.768kHz	YSML126W59H35	Y2	ABRACON	ABS07-32.768KHZ

E IEEE 802.15.4 PHY certificate

IEEE STD 802.15.4 PHY LAYER QUALIFICATION PROGRAM

This is to certify that

DUST NETWORKS / LINEAR TECHNOLOGY

has successfully completed IEEE Std 802.15.4 PHY Layer Qualification Testing as performed by
NATIONAL TECHNICAL SYSTEMS based on the ZigBee Alliance test plan for compliance testing
(available online at the ZigBee website - <http://www.zigbee.org/>)

Product Information	
Product Name:	LTC5800/LTM5850
Firmware+Hardware Version:	1.4.0 + A1
Vendor Name:	Dust Networks / Linear Technology
Vendor Street Address:	1630 McCarthy Blvd.
City & Postal Code:	Milpitas, CA 95035
Country:	USA



NTS

ENGINEERING SUCCESS

SIGNATURE

OCTOBER 7, 2015
DATE

NATIONAL TECHNICAL SYSTEMS