

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ**

KATEDRA TECHNICKÝCH ZAŘÍZENÍ BUDOV



**VYUŽITÍ IoT PRO MONITORING STAVEB A
KONSTRUKCÍ**

DIPLOMOVÁ PRÁCE

Bc. Jakub Matolín

Vedoucí diplomové práce:

doc. Ing. Michal Kabrhel, Ph.D.

Konzultant:

Ing. Aleš Vodička

**Univerzitní centrum energeticky
efektivních budov ČVUT**

Akademický rok 2016/2017



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Matolín Jméno: Jakub Osobní číslo: 395984
Zadávací katedra: K125
Studijní program: Inteligentní budovy
Studijní obor: _____

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce: Využití IoT pro monitoring staveb a konstrukcí

Název diplomové práce anglicky: Use of IoT for monitoring of buildings and structures

Pokyny pro vypracování:

Do průmyslového PLC implementujte komunikaci s již existující senzorovou platformou UCEEB pro monitoring stavebních konstrukcí a kvality vnitřního prostředí po sběrnici RS-485 s Modbus RTU. Vyčtené hodnoty zobrazujte na webserveru PLC s možností konfigurace senzorové sítě.

Navrhněte funkce pro obousměrnou komunikaci pomocí rozhraní ethernet a LoRa. Zprovozněte přenos dat do databázového systému UCEEB pomocí obou těchto rozhraní.

Seznam doporučené literatury:

Návody k PLC SDS, dostupné z: http://wiki.merenienergie.cz/index.php/Hlavn%C3%AD_strana

Internetové stránky LoRa Alliance, dostupné z: <https://www.lora-alliance.org/>

Jméno vedoucího diplomové práce: doc. Ing. Michal Kabrhel, Ph.D.

Datum zadání diplomové práce: 20.2.2017 Termín odevzdání diplomové práce: 21.5.2017
Údaj uveďte v souladu s datem v časovém plánu příslušného ak. roku

Podpis vedoucího práce

Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v diplomové práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

28.2.2017

Datum převzetí zadání

Podpis studenta(ky)

Prohlašuji, že jsem svoji diplomovou práci vypracoval samostatně s použitím uvedené literatury a podkladů.

V Praze dne 21.5.2017

.....
Jakub Matolín

Poděkování

Chtěl bych poděkovat především Ing. Aleši Vodičkovi za cenné rady a čas, který mi věnoval v průběhu zpracovávání diplomové práce. Dále bych chtěl poděkovat Ing. Marku Maškovi za poskytnutí základních informací k programování PLC modulu SDS-BIG a pomoc při sestavení funkcí pro přenos dat do databáze. V neposlední řadě bych chtěl poděkovat také doc. Ing. Michalovi Kabrhelovi, Ph.D. za věcné připomínky a revizi obsahu práce.

Obsah

Úvod	7
I. Teoretická část.....	8
1 Internet věcí.....	8
1.1 Low-Power Wide Area Network.....	10
1.2 Sigfox.....	12
1.3 LoRa.....	14
1.4 IoT v ČR.....	18
2 Monitoring budov a konstrukcí s využitím IoT.....	20
2.1 Měřené veličiny při monitoringu konstrukcí.....	21
2.2 Monitoring vlhkosti dřevěných konstrukcí.....	22
II. Praktická část.....	26
3 Systém Moisture Guard.....	26
4 Implementace komunikace po sběrnici RS485.....	29
4.1 Vlastnosti sběrnice RS485.....	30
4.2 Protokol Modbus.....	30
4.3 Specifikace PLC modulu SDS-BIG.....	32
4.4 FULL-C kód pro komunikaci po sběrnici.....	35
5 Přenos dat do databázového systému.....	38
5.1 Přenos přes rozhraní Ethernet.....	38
5.2 Přenos po LPWAN síti LoRa.....	42
6 Sestavení webové stránky.....	47
Závěr.....	52
Použitá literatura a zdroje.....	53
Seznam použitých zkratk.....	57
Seznam obrázků a grafů.....	59
Seznam příloh.....	61

Anotace

Teoretická část této práce je věnována bezdrátovým sítím pro internet věcí Sigfox a LoRa a prezentuje možnosti jejich využití v systémech pro monitoring budov a stavebních konstrukcí.

V následující praktické části je popsáno naprogramování řídicí jednotky systému Moisture Guard pro kontinuální monitoring vlhkosti v dřevostavbách. Řídicí jednotkou je programovatelný automat (PLC) SDS-BIG. Pro uložení naměřených hodnot ze senzorů je zajištěn přenos dat přes rozhraní Ethernet a po síti LoRa do databáze Univerzitního centra energeticky efektivních budov ČVUT (UCEEB) v Buštěhradě. Součástí této práce je také webová stránka, která zobrazuje aktuální údaje ze senzorů a umožňuje měnit nastavení programu běžícího v řídicí jednotce.

Klíčová slova: Monitoring konstrukcí, vlhkost, internet věcí, Sigfox, LoRa, senzory, přenos dat do databáze.

Abstract

The theoretical part of this paper summarizes features of wireless networks for the Internet of Things Sigfox and LoRa and presents possibilities of their use with systems for structural health monitoring.

The following practical part describes programming of a control unit of a system for monitoring of moisture levels in wooden structures. The monitoring system is called Moisture Guard. The control unit consists of programmable logic controller (PLC) SDS-BIG. In order to save measured values from sensors, transfer of data via Ethernet interface and LoRa network into the database of University Centre for Energy Efficient Buildings (UCEEB) is implemented. Part of this work is also a web page that displays current measured values from sensors and allows the user to change settings of the program running in the control unit.

Keywords: Structural health monitoring, moisture, Internet of Things, Sigfox, LoRa, sensors, transfer of data to a database.

Úvod

Náklady na údržbu a opravy představují významnou část nákladů životního cyklu každé stavby. Jednou z možností jak výdaje na opravy snížit, je provádět pravidelnou údržbu stavby a tím předejít vzniku rozsáhlých poruch, jejichž oprava by si vyžádala nemalé finanční prostředky. Existuje ale i sofistikovanější přístup, kterým je provádění údržby na základě analýzy hodnot fyzikálních veličin, které indikují začínající degradaci konstrukce.

Měření těchto veličin a jejich analýzu obstarávají systémy pro monitoring staveb a konstrukcí. V případě dosažení kritických hodnot mají tyto systémy za úkol varovat správce stavby před vznikající poruchou. Pro přenos informací ze senzorů do vyhodnocovacího zařízení nebo databáze, kde se data ukládají, se tradičně používají kabelové sítě. Alternativou je použití radiokomunikačních sítí pro bezdrátový přenos dat, se kterými se můžeme setkat pod označením „sítě pro internet věcí“.

Teoretická část práce prezentuje vlastnosti těchto sítí a nastiňuje možnosti jejich využití v systémech pro monitoring stavebních konstrukcí. Nedílnou součástí systémů pro monitoring staveb tvoří senzory, v práci jsou proto uvedeny principy měření některých fyzikálních veličin, které indikují riziko vzniku poruchy konstrukce. Jako úvod k praktické části této práce je zařazena kapitola věnovaná problematice zvýšené vlhkosti v dřevostavbách a sensorům vlhkosti dřeva.

V praktické části je popsáno fungování konkrétního systému pro monitoring staveb, kterým je systém Moisture Guard pro kontinuální měření vlhkosti v dřevostavbách. Hlavní část práce se věnuje naprogramování řídicí jednotky, jejímž úkolem je získávání naměřených hodnot ze senzorů systému Moisture Guard a jejich odesílání do databáze, kde jsou data uložena a následně vyhodnocena. Odesílání do databáze je uskutečněno přes rozhraní Ethernet a po bezdrátové síti LoRa.

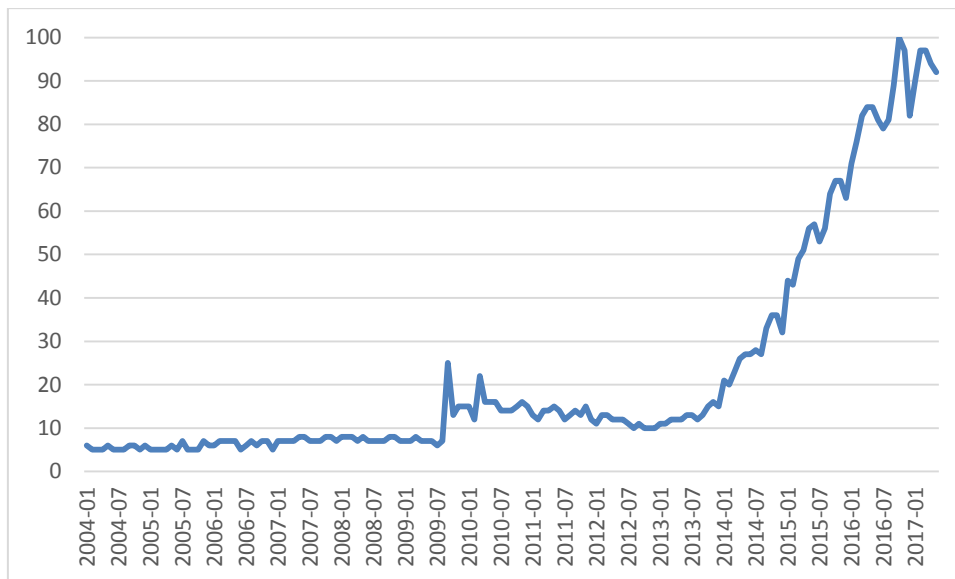
I. Teoretická část

1 Internet věcí

Termín „internet věcí“ jako první na světě použil v roce 1999 Kevin Ashton, který se zabýval tím, jak efektivně a bez lidské pomoci přenášet na internet informace, které se týkají nejrůznějších objektů. Za klíčovou součást internetu věcí považoval senzory, umožňují totiž počítačům porozumět reálnému světu a sledovat stav objektů a identifikovat je, aniž by k tomu potřebovaly lidskou pomoc. Kevin Ashton věřil, že díky propojení *internetu* a *věcí* by bylo možné získat obrovské množství dat o různých objektech a díky tomu bychom mohli přesně analyzovat, kde se pohybují, kolik jich je a v jakém jsou stavu [1].

Dnes obecně používaná definice pojmu „internet věcí“ (anglicky *Internet of Things*, běžně se můžeme také setkat se zkratkou *IoT*) vychází z této myšlenky a definuje ho jako *síť* navzájem propojených *věcí*. *Síť* je myšlena datová síť používající protokol IP a *věcmi* jsou chápány jakékoliv objekty vybavené senzory nebo akčními členy (případně obojím) a komunikačním rozhraním. Internet věcí dává nejrůznějším předmětům možnost komunikace přes internet a tím umožňuje nové formy interakce mezi uživateli a *věcmi* nebo mezi *věcmi* samotnými [2]. Společnost Sigfox, která buduje celosvětovou telekomunikační síť pro IoT, prohlašuje na svých webových stránkách, že jejich misí je „dát věcem hlas“, a tím trefně vystihuje ve třech slovech celou podstatu internetu věcí [3].

Přestože pojem IoT byl poprvé použit téměř před dvaceti lety, na popularitě získává až v posledních letech. Tento trend je možné dokumentovat na nárůstu četnosti vyhledávání tohoto sousloví ve vyhledávači Google mezi roky 2013 a 2017.



Obrázek 1: Relativní četnost vyhledávání sousloví „internet věcí“ ve vyhledávači Google od roku 2004 do současnosti [4].

Při pohledu na graf se nabízí otázka, proč od prvotní myšlenky po uvedení pojmu IoT v širší povědomí uběhla tak dlouhá doba. Tento fakt je dán technickými problémy, které omezují rozvoj internetu věcí a které byly ještě před několika lety obtížně překonatelné. Pro ilustraci těchto problémů jsou uvedeny základní předpoklady, které musí systémy využívající internet věcí splňovat:

1. Nízká energetická náročnost – základní předpoklad pro použití bateriově napájených senzorů, které umožňují monitorování objektů i v místech, kde by zajištění jiné formy napájení bylo nákladné nebo obtížně proveditelné.
2. Dlouhý komunikační dosah – abychom mohli data ze senzorů umístěných v mnoha od sebe vzdálených lokalitách spolehlivě přenášet a okamžitě vyhodnocovat, je třeba zajistit bezdrátovou komunikaci na velké vzdálenosti.
3. Nízké provozní náklady – nejen ceny koncových zařízení, ale i náklady na vybudování síťové infrastruktury a přenos dat musí být nízké.
4. Zpracovatelnost dat – součástí systému musí být výkonný databázový a cloudový systém, který je schopný zpracovávat obrovské množství dat přicházející ze senzorů.

Ještě před několika lety by byly systémy, které by splňovaly všechny výše zmíněné vlastnosti, technicky nerealizovatelné nebo příliš nákladné. To se v současnosti mění s vývojem nových výkonných cloudových systémů a s nástupem Low-Power Wide Area

Network (LPWAN) sítí, které by v budoucnu měli hrát zásadní roli v rozšíření internetu věcí do nejrůznějších oblastí lidské činnosti.

1.1 Low-Power Wide Area Network

V minulosti bylo vyvinuto již mnoho komunikačních technologií, které umožňují připojení věcí k internetu, žádná však zatím nezpůsobila masové rozšíření IoT produktů na trhu. Nejsou totiž schopny zajistit energeticky úsporný provoz, dlouhý komunikační dosah a nízké provozní náklady zároveň. Často splňují dva z těchto požadavků, ve třetím však selhávají.

Existují komunikační standardy jako na příklad ZigBee nebo Bluetooth, které jsou energeticky úsporné a lze je tedy implementovat do bateriově napájených přístrojů. Nevýhodou těchto technologií je však jejich omezený komunikační dosah v jednotkách desítek metrů, a proto plošné pokrytí rozsáhlého území nepřipadá v úvahu [2].

Mobilní telekomunikační sítě se z tohoto hlediska jeví jako slibnější technologie pro IoT díky komunikačnímu dosahu v jednotkách kilometrů a existující infrastruktuře. Tradičně se jedná o přenos informací pomocí standardu GSM, nověji se pak můžeme setkat s technologiemi jako na příklad LTE nebo UMTS. Nevýhodou těchto technologií je vysoká cena GSM/LTE modemů a vysoká energetická náročnost přenosu dat, což komplikuje jejich použití v bateriově napájených zařízeních, u kterých požadujeme schopnost pracovat autonomně po dobu několika let.

Novou alternativou jsou LPWAN sítě, které operují na podobném principu jako sítě pro mobilní telefony. Nabízí dlouhý komunikační dosah, díky čemuž lze signálem snadno pokrýt rozsáhlá území. Spojení mezi koncovými zařízeními a IP světem zajišťují základnové stanice (také nazývané *koncentrátory* nebo *gatewaye*), stejně tak jako je tomu v případě mobilních sítí. Sítě LPWAN jsou navrženy v topologickém uspořádání hvězda, zařízení tedy navazují komunikaci přímo se základnovou stanicí. Díky tomu, že LPWAN a mobilní telekomunikační sítě si jsou do značné míry podobné, lze pro rozšiřování LPWAN sítí s výhodou využít již existující infrastrukturu základnových stanic mobilních operátorů. Je proto logické, že se do budování LPWAN sítí u nás i v zahraničí pustili i nadnárodní telekomunikační operátoři, kteří v internetu věcí vidí zajímavou obchodní příležitost [2].

Na rozdíl od mobilních sítí, přenos dat po sítích LPWAN probíhá při velmi nízkých přenosových rychlostech v řádech stovek bitů za sekundu. Takové rychlosti se jeví z dnešního hlediska jako nedostatečné a pro aplikace jako prohlížení webových stránek nebo přenos hlasu naprosto nepoužitelné. Pro IoT zařízení jsou ale tyto rychlosti dostačující, protože objemy přenášených dat jsou malé a není vyžadována okamžitá odezva, typicky totiž uživateli nezáleží na tom, jestli údaj z čidla dojde okamžitě nebo až o několik vteřin později. V naprosté většině případů jsou zprávy posílané po LPWAN sítích velmi krátké a obsahují pouze několik jednotek, maximálně desítek bajtů. Díky nízké rychlosti a malému objemu dat je přenos informací po sítích LPWAN výrazně energeticky úspornější v porovnání s GSM nebo LTE. Je tak možné dosáhnout bateriově napájeného provozu zařízení po dobu až několika let bez nutnosti výměny baterie [5].

Ve prospěch LPWAN technologií hovoří také cenová dostupnost komunikačních modulů, jejichž ceny se pohybují v řádech jednotek dolarů. To je v porovnání s LTE modemy zhruba dvacetkrát méně [6]. IoT řešení se tak díky LPWAN technologiím výrazně zlevňují a nacházejí uplatnění i ve velmi jednoduchých aplikacích, kde se připojení pomocí GSM nebo LTE ekonomicky nevyplatí.

Díky cenové dostupnosti LPWAN modemů, dlouhému komunikačním dosahu a nízké energetické náročnosti lze očekávat v nejbližších letech nástup IoT zařízení ve formě nejrůznějších senzorů v mnoha odvětvích. Mezi tyto odvětví a některé příklady použití patří:

- energetika – vzdálené odečítání spotřeby vody, plynu a elektřiny, vzdálené monitorování produkce energie z obnovitelných zdrojů
- průmysl – prediktivní údržba, monitoring průmyslových zařízení a strojů, měření průtoků, úrovně hladin apod., monitoring úniku plynu a dalších látek
- logistika – sledování pohybu zboží, vzdálená diagnostika vozového parku, lokalizace odcizených vozidel a objektů
- zemědělství – monitoring vlhkosti půdy, sledování stavu naplnění vodních nádrží a sýpek, autonomní meteostanice, inteligentní řízení zavlažování, systémy upozornění na vznik lesních požárů, sledování pohybu dobytka
- města – monitoring úrovně naplnění odpadních kontejnerů, sledování kvality ovzduší, řízení pouličního osvětlení, parkovací senzory, dálkové ovládání dopravního značení

- budovy – kouřové detektory, záplavové detektory, zabezpečovací systémy, monitoring stavu stavebních konstrukcí

LPWAN sítě lze postavit na různých technologiích, mezi ty nejrozšířenější momentálně patří Sigfox, LoRa, Nwave a Ingenu. V následujících kapitolách budou krátce popsány technologie Sigfox a LoRa, kterým se v současnosti dostává v souvislosti s internetem věcí nejvíce pozornosti.

1.2 Sigfox

Technologie Sigfox byla vyvinuta stejnojmennou společností, která byla založena v roce 2009 ve Francii. Pokrytí sítí Sigfox je aktuálně dostupné ve 32 zemích světa [7]. První verze Sigfoxu umožňovala pouze jednosměrnou komunikaci, zprávy tedy mohly být odesílány pouze z koncového zařízení směrem k uživateli (tzv. *uplink*). To už v současnosti neplatí a je podporována obousměrná komunikace s omezením na 4 zprávy denně směrem od uživatele ke koncovému zařízení (tzv. *downlink*). Jedna Sigfox základnová stanice je schopna vyřídit požadavky až jednoho milionu připojených zařízení a pokrýt oblasti do vzdálenosti 30 – 50 km od koncentrátoru v nezastavěném prostoru, ve městech je pak dosah snížen na 3 – 10 km [2].

Sigfox je vhodný zejména pro bateriově napájené senzory, které potřebují posílat velmi malé objemy dat na velké vzdálenosti. Senzory si vystačí s velmi jednoduchým hardwarem, protože většina výpočetních operací je prováděna externě v cloudovém úložišti. Díky tomu je snížena jejich pořizovací cena a energie potřebná na jejich provoz.

Komunikační Sigfox protokol je uzpůsoben pro posílání krátkých zpráv. Každá zpráva může obsahovat maximálně 12 bajtů užitečných dat (tzv. *payload*) a maximální počet uplinkových zpráv je stanoven na 140 denně. Hlavička zprávy je složena ze 14 bajtů, které obsahují identifikační číslo zařízení a další informace potřebné pro autentizaci zprávy na straně serveru. Touto vlastností se Sigfox protokol liší od standardních síťových protokolů, které nejsou optimalizované pro internet věcí a které s každou zprávou (i když krátkou) odesílají velké množství dat ukryté hlavičce. Pro představu, v jedné 12bajtové zprávě lze poslat tyto informace:

- 2 GPS souřadnice s přesností 3 metry

- 6 hodnot z teploměru v rozmezí -100 až 200 °C s přesností na dvě desetinná místa
- 12 záznamů o rychlosti v rozsahu 0 – 255 km/h
- 96 binárních hodnot signalizujících stav on nebo off

Zařízení mohou odesílat také prázdné zprávy, které dávají uživateli informaci o tom, že zařízení je „na živu“.

Zprávu lze poslat i opačným směrem, tedy od uživatele k Sigfox jednotce. Tento typ zprávy může obsahovat maximálně 8 bajtů užitečných dat. Pomocí downlinkových zpráv je možné zasahovat do nastavení zařízení, např. upravit měřící rozsah senzoru, nastavit frekvenci posílání zpráv, požádat o dodatečná data nebo zahájit update firmware. V případě update firmware se jedná pouze o pokyn pro zahájení, protože vzhledem k objemu dat potřebnému pro update je nutno navázat jiný typ spojení (např. přes GSM modem). Při posílání downlinkových zpráv je třeba dát pozor na skutečnost, že citlivost přijímače levného koncového zařízení je horší, než citlivost přijímače drahého koncentrátoru. Z tohoto důvodu se může ve vzdálenějších lokalitách od koncentrátoru stát, že základnová stanice bude úspěšně přijímat uplinkové zprávy od zařízení, ale to nebude schopné přijímat downlinkové zprávy [8].



Obrázek 2: Komunikační prvky v síti Sigfox [9].

Jednotlivá zařízení v síti Sigfox nejsou spárována s konkrétní základnovou stanicí, to znamená, že jednu zprávu může zachytit více stanic (v průměru je jedna zpráva zachycena třemi různými stanicemi). Tím je zajištěna vyšší stabilita přenosu dat pro případ, že by došlo k porušení zprávy při přenosu do jednoho z koncentrátorů.

Poté co základnová stanice zprávu přijme, data jsou přenesena do Sigfox cloudu, který je provozován společností Sigfox. Z něj pak může svá data získat koncový zákazník. Jiná možnost jak se k datům odeslaným přes Sigfox dostat neexistuje, cesta vždy vede přes oficiální Sigfox cloud, což někteří uživatelé mohou vnímat jako omezující. Centralizované řešení má ale i svou výhodou a tou je fakt, že neexistují poplatky za roaming. To může být velmi zajímavé především pro logistické společnosti, které potřebují sledovat pohyb zboží, palet apod. napříč různými státy.

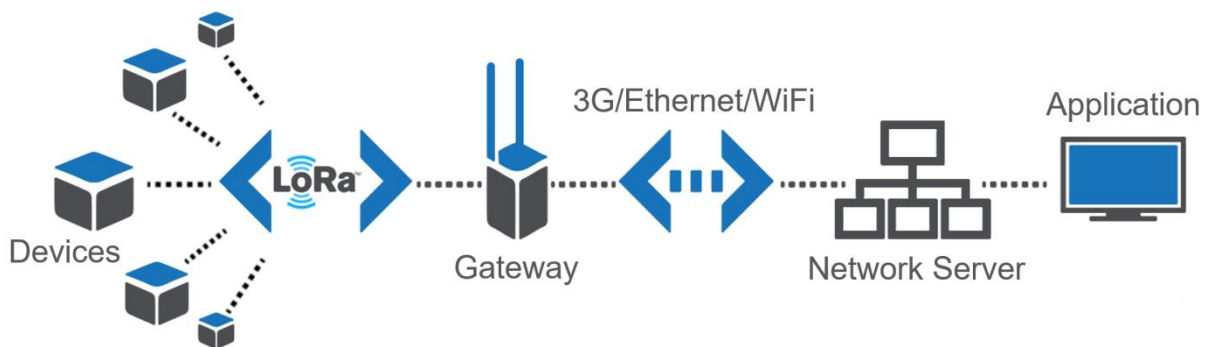
Bezpečnost přenosu dat je zajištěna pomocí klíče, který je uložen v nepřístupné nevolatilní paměti každého Sigfox zařízení, a identifikačního čísla, které je na rozdíl od klíče veřejné. Každá zpráva obsahuje unikátní autorizační podpis vytvořený pomocí klíče a identifikačního čísla, který slouží k autentifikaci odesílatele [10].

Společnost Sigfox poskytuje výrobcům hardwaru všechnu dokumentaci zdarma, výrobci mohou tedy (pokud splní certifikační podmínky) bez omezení vyrábět vlastní Sigfox radiokomunikační komponenty. Data odeslaná přes síť Sigfox jsou přístupná pouze přes oficiální Sigfox server, jedná se tedy o uzavřenou technologii, jejíž chod závisí pouze na jednom poskytovateli služeb. Společnost Sigfox působí v několika zemích přímo jako operátor své sítě, např. ve Francii nebo Spojených státech, v jiných zemích pak zajišťují pokrytí exkluzivní operátoři. V ČR a na Slovensku to je operátor SimpleCell. Sigfox operátoři si účtují za objem přenesených dat a podobně jako mobilní operátoři poskytují různé tarify pro méně náročné zákazníky (několik uplinkových zpráv denně, žádné downlinkové zprávy) až po ty náročnější (maximálně 140 uplinkových zpráv a 4 downlinkové zprávy denně). Obchodní strategie Sigfoxu je založena na zpoplatnění svých telekomunikačních služeb a výnosu z každé přenesené zprávy [8].

1.3 LoRa

LoRa je způsob modulace radiofrekvenčního signálu vyvinutý kalifornskou firmou Semtech pro přenos zpráv mezi zařízeními a koncentrátory na velké vzdálenosti s nízkou energetickou náročností. Stejně jako Sigfox, operují LoRa sítě v bezlicenčních frekvenčních pásmech ISM (Industrial, Scientific and Medical) kolem frekvence 868 MHz (v Evropě, na jiných kontinentech se frekvence liší) a jsou postaveny na síťové architektuře typu hvězda. Základnové stanice LoRa jsou připojeny do IP sítě a plní funkci zprostředkovatele přenosu dat mezi koncovými jednotkami a zákazníkem [11].

Rozmístění koncentrátorů se doporučuje tak, aby každé čidlo ve vnitřním prostoru bylo umístěno ve vzdálenosti maximálně do 1 km. V případě venkovního umístění lze počítat s dosahem 7 km a při přímé viditelnosti až 20 km [12].



Obrázek 3: Schéma přenosu zpráv v síti LoRa [13].

Downlinkové zprávy jsou odesílány vždy přes koncentrátor s nejlepšími radiovými podmínkami, které jsou vyhodnoceny na základě předchozí komunikace. Uplinkové zprávy jsou odeslány stejně jako v případě Sigfoxu na všechny základnové stanice v dosahu.

Důležitým prvkem v komunikaci je řídicí server (Network Server), který obstarává komunikaci se základnovými stanicemi, autorizuje senzory a směřuje komunikaci na jednotlivé aplikační servery. Mezi základní činnosti řídicího serveru patří [12]:

- odstraňování duplicitních zpráv, které byly doručeny více cestami na základě sekvenčního čísla zprávy
- směřování dat z jednotlivých zařízení k odpovídajícím aplikačním serverům (zodpovídá za to, aby data byla přenesena ke správnému uživateli)
- vysílání potvrzovacích zpráv o přijetí zprávy (tzv. *ACK*), jsou-li vyžadovány
- nastavování provozních podmínek (spreading faktor, přenosová rychlost, vysílací výkon, použité kanály apod.) v závislosti na vzdálenosti od koncentrátoru pro optimalizaci provozu zařízení (více níže)

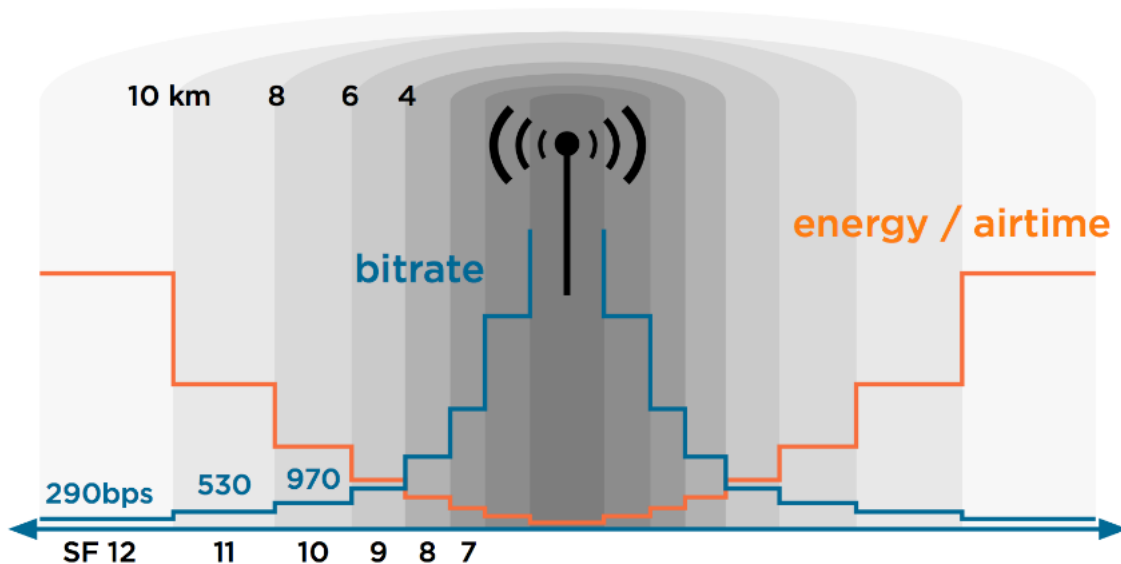
Úkolem aplikačního serveru je potom komunikace s řídicím serverem, dešifrování zpráv, ukládání dat v podobě, která je vhodná pro jejich další zpracování a analýzu, a vytvoření následných akcí (např. spuštění alarmu v případě, že je v zabezpečeném objektu zaznamenán pohyb) [12].

Při návrhu specifikace pro technologie LoRa byly brány v úvahu různé aplikace a byly proto zavedeny tři třídy, do kterých jsou zařízení podle nároků na spotřebu energie a způsob komunikace rozdělené [2], [12]:

- Třída A – Hodí se v situacích, kdy není k dispozici napájení a je požadavek zejména na sběr dat. Zařízení třídy A vždy zahajují komunikaci a to na základě nějaké události (např. detekce kouře u kouřového čidla) nebo v závislosti na intervalu časovače uvnitř čidla. Po odeslání zprávy je zařízení schopno ve dvou intervalech o celkové délce 2 sekundy přijmout zprávu ze serveru, na základě které pak může vyvolat nějakou akci (např. sepnout relé, přenastavit interval odesílání zpráv apod.). Čidla třídy A jsou tedy po většinu času v režimu spánku a díky tomu spotřebovávají jen malé množství energie.
- Třída B – V mnoha uživatelských aplikacích je vyžadována obousměrná komunikace a v takových případech je vhodné použít zařízení třídy B. Oproti třídě A tyto zařízení aktivují v pravidelném časovém intervalu svoji přijímací radiovou část, aniž by došlo k odeslání nějaké zprávy (což je energeticky nejnáročnější operace). Pomocí synchronizačních impulsů (tzv. *beaconů*), které se šíří sítí, se interní časovače zařízení synchronizují s časem v síti. Interní časovač lze pak nastavit tak, aby se zařízení aktivovalo pro příjem zprávy v intervalu, který vyžaduje daná aplikace. Interval můžeme nastavit například na 1 minutu, pokud chceme, aby zařízení zareagovalo na downlinkovou zprávu nejdéle do jedné minuty. Podobného efektu bychom mohli dosáhnout i nastavením intervalu časovače zařízení třídy A na 1 minutu, tím bychom ale popřeli jeho energetickou nenáročnost (každou minutu by byla odeslána uplinková zpráva).
- Třída C – Zařízení třídy C mohou být použita v případech, kdy není vyžadována energetická nenáročnost a je požadavek na okamžitou odezvu na downlinkové zprávy. Zařízení třídy C neustále naslouchají příchozím zprávám a jejich přijímací radiová část je tedy stále aktivní, což způsobuje jejich vyšší energetickou náročnost. Typicky se jedná o akční členy, které nejsou bateriově napájené, a slouží například k ovládání strojů, spínání relé apod.

Díky zavedení různých tříd zařízení má uživatel možnost zvolit rychlou komunikační odezvu nebo optimalizovat chod zařízení pro co nejdelší výdrž baterie.

Komunikační protokol sítí LoRa nese označení LoRaWAN a stejně jako protokol sítě Sigfox je navržen pro nízkou energetickou náročnost přenosu dat. Maximální délka užitečných dat (payloadu) je 240 bajtů při hodnotách spreading factoru (SF) 0 až 7. Pro vyšší hodnoty SF se délka zprávy zkracuje až na 51 bajtů pro SF 12, což souvisí s delší dobou vysílání zprávy [12]. Vztah mezi spreading factorem, přenosovou rychlostí (*bitrate*), dobou vysílání zprávy (*airtime*) a komunikačním dosahem je zobrazen na následujícím obrázku.



Obrázek 4: Vliv spreading factoru na přenosovou rychlost, komunikační dosah a energii potřebnou k odeslání zprávy [13].

Bezpečnost komunikace je podobně jako u Sigfoxu zajištěna pomocí bezpečnostních klíčů. Šifrování probíhá ve dvou úrovních, v první úrovni je šifrování provedeno pomocí síťového klíče, ve druhé pak za použití aplikačního klíče. Síťový klíč šifruje celou zprávu s výjimkou identifikačního čísla zařízení EUI, šifruje tedy sekvenční číslo zprávy, payload a veškeré radiové informace, které slouží pro nastavení optimálního radiového přenosu (SF, vysílaný výkon, použitý kanál a další informace). Síťový klíč šifruje zprávu mezi zařízením a řídicím serverem a zajišťuje, aby do sítě nebyly vpuštěny podvrhy čidel. Aplikační klíč má za úkol zajistit privátnost dat během průchodu celým komunikačním řetězcem. Správa síťových klíčů je v odpovědnosti

provozovatele aplikace, což může být buď koncový uživatel (např. v případě dálkového měření spotřeby distributor energií), nebo provozovatel služby (např. poskytovatel služeb zabezpečení domácností) [12].

Výrobci modulů a základnových stanic pro komunikaci s protokolem LoRaWAN jsou sdruženi v organizaci LoRa Alliance, která zajišťuje certifikaci a kompatibilitu mezi produkty jednotlivých výrobců. Budování LoRa sítí se podobně jako v případě Sigfoxu chopili mobilní operátoři, ve svých zdrojích LoRa aliance uvádí aktuálně 37 veřejných operátorů, zejména v Evropě a Severní Americe [14].

Na rozdíl od Sigfoxu, LoRa nabízí možnost vytvořit si privátní síť pomocí vlastních základnových stanic a je tedy v tomto směru flexibilnější a otevřenější než Sigfox, kde všechny zprávy musí směřovat přes exkluzivní operátory a oficiální Sigfox cloud. LoRa má ale také určitý omezující element a tím je fakt, že radiové komponenty pro LoRa moduly vyrábí pouze jediná firma a tou je společnost Semtech. Strategie obou LPWAN konkurentů je tedy značně odlišná – tržby Sigfoxu plynou z poskytování telekomunikačních služeb a jeho cloudového backendu, zatímco Semtech, který stojí za vývojem technologie LoRa, těží z prodeje hardwaru [8].

1.4 IoT v ČR

Mezi nejvýznamnější subjekty v České Republice, které se pohybují v oblasti internetu věcí, patří společnosti SimpleCell a České Radiokomunikace. Obě tyto společnosti budují v ČR vlastní LPWAN síť, každá ale vsází na jinou technologii.

SimpleCell je prvním a z podstaty technologie Sigfox také jediným operátorem sítě Sigfox na českém území. Infrastrukturu buduje ve spolupráci se společností T-Mobile a aktuálně pokrývá svojí sítí 86% území [15]. Prvním komerčním projektem v ČR postaveným na technologii Sigfox se stal jeho spuštěním koncem roku 2016 systém pro monitorování parkovacích míst ve městě Liberec. V rámci tohoto projektu je sledována obsazenost celkem 240 parkovacích míst pomocí senzorů magnetického pole zalitých ve vozovce. Jako další projekt využívající síť Sigfox lze zmínit pilotní projekt odečtů stavu vodoměrů realizovaný společností Solidus Tech [16].

České Radiokomunikace (ČRa) zvolily pro vybudování své IoT sítě technologii LoRa. V současnosti sítí pokrývají pouze krajská města v ČR a vybraná okresní města, SimpleCell je tedy co se pokrytí týče napřed. Husté plošné pokrytí plánují České

Radiokomunikace zprovoznit v rozmezí 2 – 3 let [12]. Přenos dat z koncových zařízení k uživateli probíhá pomocí řídicího serveru ve vlastnictví ČRa, je tedy realizován odlišně než tomu je v případě SimpleCellu, kde data k zákazníkovi proudí přes oficiální Sigfox cloud, který provozuje společnost Sigfox.



Obrázek 5: Schéma přenosu zpráv v síti LoRa provozované Českými Radiokomunikacemi [17].

České Radiokomunikace nabízí několik různých balíčků v závislosti na tom, kolik je do sítě připojených zařízení. Poskytují také služby spojené se správou dat a jejich ukládáním do databáze [17]. Jako jeden z projektů, který využívá platformu ČRa, lze uvést např. projekt GEO Tracker, který umožňuje přibližně lokalizovat zařízení nezávisle na technologiích GPS a GSM pomocí známých poloh základnových stanic a dostupných parametrů sítě (útlum signálu, odstup signálu od šumu apod.). Společnost RWE potom ve spolupráci s ČRa testuje nasazení čidel pro vzdálený odečet stavu plynůměrů [18]. U obou zmíněných projektů se v případě úspěšných testů plánuje komerční využití.

Na českém poli se pohybuje poměrně velké množství firem vyrábějících různé senzory a moduly s LPWAN konektivitou, na nějaké masovější rozšíření IoT projektů se ale v ČR stále čeká. Co se týče monitoringu budov a konstrukcí s využitím LPWAN sítí, zatím u nás nebyl podle dostupných informací na internetu žádný takový projekt realizován.

2 Monitoring budov a konstrukcí s využitím IoT

Rozmach IoT v posledních letech přináší změny do mnoha odvětví. Nejinak je tomu ve stavebnictví, kde se nové telekomunikační technologie ve spojení se senzory dají využít pro získávání cenných dat o stavu budov a konstrukcí. Systém monitoringu stavebních konstrukcí umožňuje kontinuálně sledovat a diagnostikovat stav stavebních konstrukcí a na základě analýzy dat také předpovídat vývoj poškození a plánovat jejich údržbu. Systém pro monitoring se typicky skládá ze senzorů, výpočetní jednotky, která přijatá data ze senzorů vyhodnocuje, a rozhraní pro přenos dat [19].

Mezi hlavní výhody monitoringu stavebních konstrukcí patří:

- včasné varování před selháním konstrukce
- snížení výdajů na opravy včetně snížení jejich časové náročnosti díky včasnému odhalení závad
- možnost lokalizace poruch
- získání cenných dat pro zhotovitele stavby a výrobce stavebních materiálů, kteří mohou tyto data využít pro zlepšení stavebních postupů a vylepšení svých produktů
- zefektivnění práce správců objektů a údržbářů, kteří díky monitoringu nemusí provádět údržbu na základě daných časových intervalů, ale mohou ji naplánovat podle reálné potřeby konstrukce

Konstrukce jsou tak ve výsledku bezpečnější (včasná detekce závad předchází selhání konstrukce) a jejich údržba je méně nákladná [20]. LPWAN sítě výrazně zlevňují a zrychlují realizaci monitorovacích systémů, jelikož odpadá nutnost nákladné instalace kabelů a drahé údržby kabelové sítě. V budoucnu se proto očekává příchod nových systémů pro sledování stavu konstrukcí, které budou využívat některou z LPWAN technologií.

Při rozhodování jakou ze dvou výše zmíněných LPWAN technologií je vhodnější pro monitoring staveb použít, je třeba vzít v úvahu počáteční náklady spojené s pořízením potřebného hardwaru a provozní náklady na přenos a správu dat. Pokud by se jednalo o menší systém, který monitoruje pouze jednu stavbu a nepoužívá velké množství senzorů, bylo by výhodnější postavit systém na technologii Sigfox nebo využít služeb velkého operátora sítě LoRa. Poplatky operátorovi za přenos dat by totiž díky malému počtu senzorů nebyly velké.

Pro rozsáhlejší systémy, které monitorují větší stavby nebo případně i několik staveb blízko sebe (např. skupinu budov univerzitního kampusu apod.), by bylo vhodné uvažovat o instalaci vlastní základnové stanice a vybudování privátní sítě na technologii LoRa. LoRa díky své otevřenosti toto umožňuje a byly by tím eliminovány poplatky operátorovi sítě za přenos dat. Nevýhodou tohoto řešení jsou vyšší počáteční náklady spojené s investicí do vlastní gatewaye.

Přestože obě technologie používají pro přenos dat frekvenční pásma v oblasti pod 1 GHz a díky tomu dobře prostupují skrze fyzické překážky, v případě umístění senzorů uvnitř budov je potřeba vzhledem k velkému množství překážek počítat s horší dostupností signálu. Při umístění senzorů uvnitř budov a zejména v suterénech a podzemních patrech je třeba s touto skutečností počítat a tomu přizpůsobit výběr technologie. Z hlediska penetrace do vnitřního prostředí má lepší výsledky Sigfox, který pro přenos dat používá frekvenční modulaci na velmi úzkém pásmu (jedná se o tzv. *ultra-narrow band* technologii).

Samozřejmostí je vyvarování se umístění senzorů do kovových skříní, např. rozvaděčů, které se chovají jako Faradayova klec a zabraňují průchodu radiových vln. Tento problém se dá obejít použitím externí antény, která je k senzoru připojena kabelem a je umístěna mimo skříň.

2.1 Měřené veličiny při monitoringu konstrukcí

Stav stavebních konstrukcí a jejich životnost je ovlivňována velkým množstvím vnějších faktorů. Jako některé z nejčastěji sledovaných veličin lze uvést [21]:

2.1.1 Vibrace

Vliv vibrací se sleduje zejména u mostních konstrukcí pro vyhodnocení vlivu dynamických zatížení od projíždějící dopravy nebo namáhání při extrémních povětrnostních podmínkách. S monitoringem vibrací se můžeme setkat i u staveb nacházejících se v blízkosti vozovek s frekventovanou těžkou nákladní automobilovou dopravou nebo u staveb v seismických oblastech.

Pro detekci vibrací se využívají akcelerometry. Často jde o senzory velmi malých rozměrů spadajících do kategorie mikro-elektro-mechanických součástek (tzv. MEMS).

2.1.2 Deformace

Měření deformací je důležité na příklad pro monitorování stavu historických budov nebo sledování rozvoje trhlin v betonových konstrukcích. Pro tyto účely se využívají nejčastěji tenzometry. Jedná se o odporové senzory, u nichž je elektrický odpor závislý na stavu deformace. Další možností sledování deformací jsou piezoelektrické senzory, které využívají piezoelektrických vlastností některých krystalů.

2.1.3 Posuny a průhyby

Posuny stavebních konstrukcí často souvisí se změnami v podloží nebo s nekvalitním provedením základových konstrukcí. Pro detekci posunů lze použít na příklad Linear Variable Differential Transformer (LVDT) snímače, které převádějí lineární posun na elektrický signál. Zajímavým řešením pro monitorování posunů a průhybu konstrukcí je použití technik pro zpracování obrazu, kdy je vyhodnocován obraz snímáný fotoaparátem s teleobjektivem. Ten je umístěn na pevném bodě a je zaměřen na sledovaný bod konstrukce. Posun pixelů obrazu je pak vyhodnocen jako posun konstrukce. Toto řešení je použitelné spíše u větších staveb, např. mostů [21].

2.1.4 Klimatické vlivy

Teplota a vlhkost mají zásadní vliv na stav stavebních konstrukcí a často jsou příčinou mnoha problémů. Degradace může být způsobena cyklickým zatěžováním konstrukce vlivem tepelné roztažnosti materiálů nebo přítomností vody, se kterou se do konstrukcí dostávají rozpuštěné soli a další látky, které způsobují narušování zdiva a omítek.

Vlhkost stavebního materiálu je důležitým faktorem zejména pro údržbu dřevostaveb, jelikož je hlavním iniciátorem degradačních procesů dřeva. Včasná detekce rostoucího obsahu vody ve dřevě umožňuje tyto degradační procesy včas identifikovat a zamezit jejich rozvoji. Je třeba sledovat také vzdušnou vlhkost a teplotu rosného bodu a předejít tak případné kondenzaci vodních par uvnitř stavby.

2.2 Monitoring vlhkosti dřevěných konstrukcí

Mezi hlavní faktory způsobující poruchy dřevěných konstrukcí patří atmosférická koroze a napadení dřeva dřevokazným hmyzem a houbami [22]. Tyto degradační

procesy se často vyskytují současně a vlhkost dřeva je jednou z příčin těchto procesů (dalšími faktory jsou teplota a pH hodnota dřeva). Jako minimální vlhkost pro napadení dřeva houbami a plísněmi se uvádí absolutní vlhkost dřeva kolem 20%, při vlhkosti 25% - 55% mají pak dřevokazné houby ideální podmínky pro jejich vývoj [23].

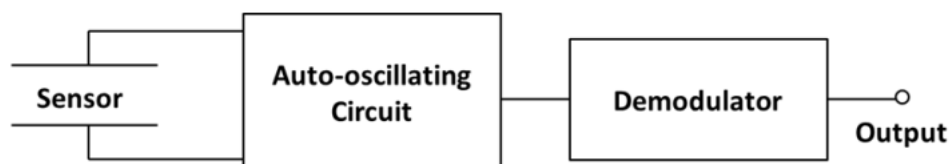
Nárůst vlhkosti v dřevěných konstrukcích staveb je většinou spjat s netěsnostmi v rozvodech vody nebo s kondenzací vodních par uvnitř stavby. Ve většině případů dojde k odhalení závady až ve chvíli, kdy už je dřevo zasaženo degradačními procesy. Nezbyvá pak nic jiného, než přistoupit k nákladné a rozsáhlé opravě napadených konstrukcí. Použití systému pro monitoring dřevěných konstrukcí si klade za cíl tomuto scénáři předejít a včas informovat uživatele o vzniklé závadě [24]. Pro sběr dat a sledování vlhkosti se nejčastěji používají kapacitní nebo rezistivní senzory.

2.2.1 Kapacitní senzory

Pro měření vlhkosti dřeva senzory je nutné měřit vlhkost pomocí nepřímých metod, které využívají známých závislostí mezi měřenou veličinou a vlhkostí. Jednou z těchto závislostí je vztah mezi kapacitou a vlhkostí materiálu, kdy vlivem změny vlhkosti materiálu dochází také ke změně permitivity dielektrika, kterým je v tomto případě dřevo. Kapacita kondenzátoru C (F) je dána rovnicí:

$$C = \frac{\epsilon S}{d} \quad (2.1)$$

kde ϵ ($F \cdot m^{-1}$) je permitivita dielektrika, S (m^2) je plocha elektrod kondenzátoru a d (m) je vzdálenost mezi elektrodami. Blokové schéma senzoru fungujícího na tomto principu je znázorněno na následujícím obrázku.



Obrázek 6: Blokové schéma kapacitního senzoru vlhkosti [23].

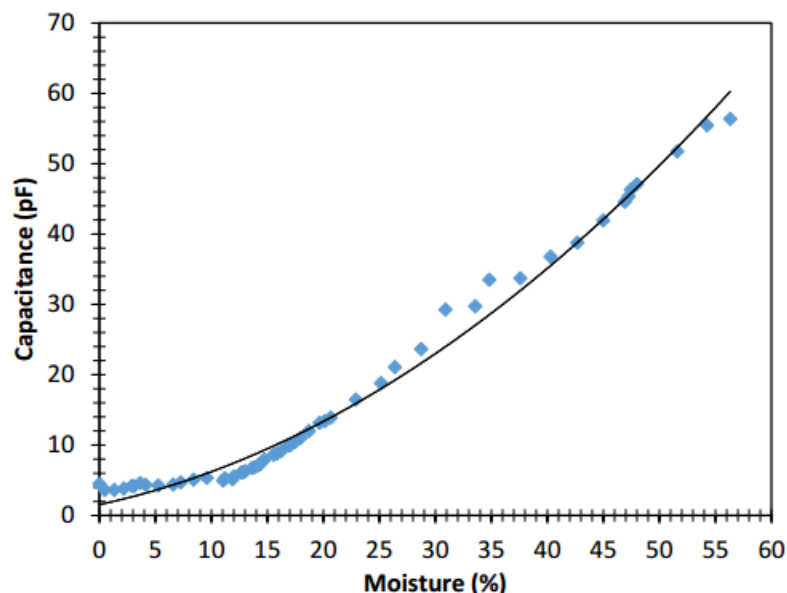
Přístroj samotný je složen z elektrod, rezonančního obvodu a demodulátoru. Rezonanční frekvence ω (Hz) je závislá na kapacitě kondenzátoru a indukčnosti cívky L (H) uvnitř obvodu, tento vztah je popsán Thomsonovou rovnicí (2.2) [23]:

$$\omega = \frac{1}{\sqrt{LC}} \quad (2.2)$$

Kapacitu kondenzátoru získáme úpravou rovnice (2.2):

$$C = \frac{1}{L\omega^2} \quad (2.3)$$

Z tohoto vztahu lze následně vyjádřit kapacitu a přepočtem získat vlhkost dřeva. Pro získání správné hodnoty je třeba použít křivku odpovídající typu dřeva, změna permitivity je totiž závislá na hustotě, stáří dřeva a také na druhu dřeviny. Závislost kapacity dřeva na vlhkosti je parabolická [23]. Relativní permitivita vody má hodnotu 80, zatímco relativní permitivita dřeva se pohybuje v rozmezí 2 – 7, proto se kapacita se zvyšujícím se podílem obsažené vody ve dřevě zvyšuje [25].

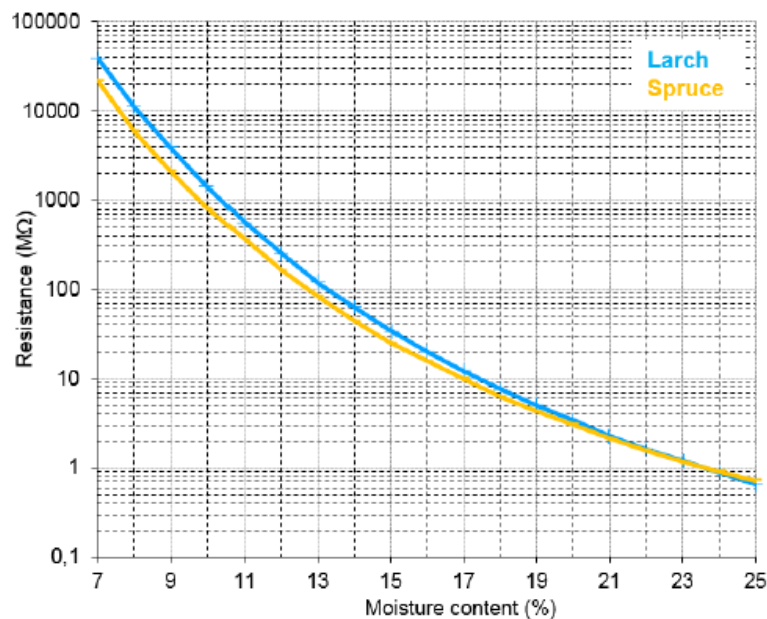


Obrázek 7: Závislost kapacity borového dřeva na absolutní vlhkosti dřeva [23].

Jak je vidět v grafu, rozdíl hodnot kapacity senzoru se pohybuje v řádech pikofaradů na rozsahu absolutní vlhkosti dřeva 0 – 60%.

2.2.2 Rezistivní senzory

Na rozdíl od kapacitních senzorů, kde se měřená veličina pohybuje na malém rozmezí hodnot, elektrický odpor dřeva se v závislosti na vlhkosti dřeva mění v rozsahu od desítek k Ω do stovek G Ω .



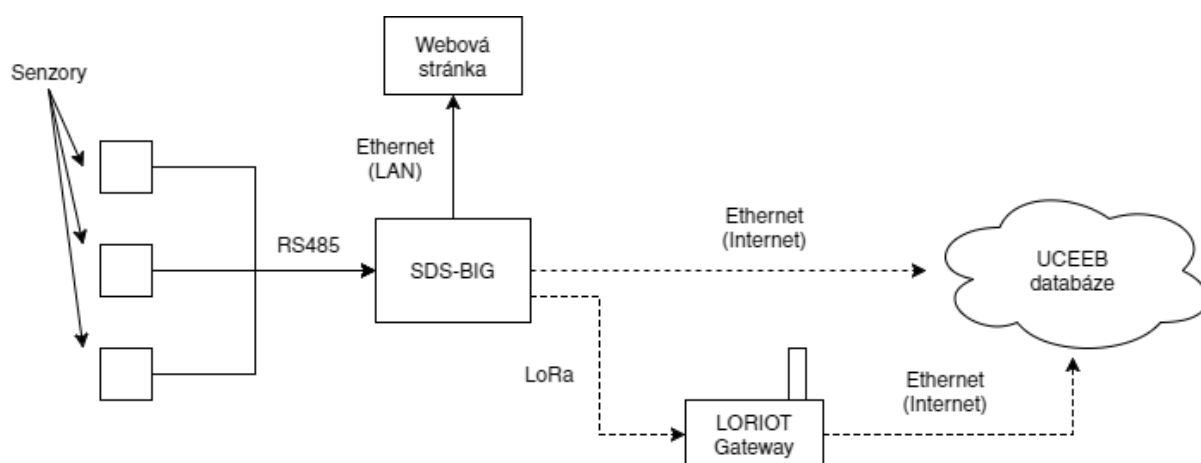
Obrázek 8: Závislost elektrického odporu dřeva na absolutní vlhkosti dřeva [26].

Z grafu je patrné, že s rostoucí vlhkostí odpor klesá v důsledku toho, jak se zvyšuje podíl vody v materiálu a stoupá jeho vodivost. Závislost odporu na vlhkosti dřeva je hyperbolická.

Výsledný naměřený odpor je třeba korigovat na vliv teploty, který není nezanedbatelný, a je třeba také zohlednit konfiguraci měřících elektrod a typ dřeviny, ze které je konstrukce postavena [26]. Různé druhy dřevin mají totiž za stejných podmínek jiné hodnoty elektrického odporu, což je způsobeno především jejich rozdílnou hustotou.

II. Praktická část

Cílem praktické části této práce je představit naprogramování řídicí jednotky pro sběr dat ze senzorů systému Moisture Guard a jejich následný přenos do databázového systému. Řídicí jednotkou je programovatelný automat (PLC) SDS-BIG (Síťový Dohledový Systém) od společnosti Onlinetechnology.cz. Do jednotky bylo implementováno zobrazování naměřených hodnot a ovládání nastavení programu pomocí webové stránky, která běží na webovém serveru zařízení SDS-BIG. Přenos dat do databázového systému Univerzitního centra pro energeticky efektivní budovy (UCEEB) je proveden pomocí rozhraní Ethernet a LoRa. Komunikaci mezi SDS-BIG a databází po síti LoRa zajišťuje gateway LORIOT.



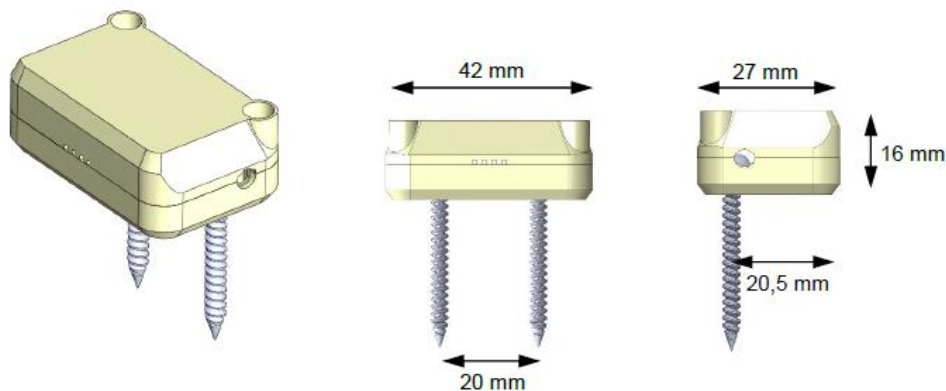
Obrázek 9: Schéma přenosu dat ze senzorů do databáze UCEEB.

3 Systém Moisture Guard

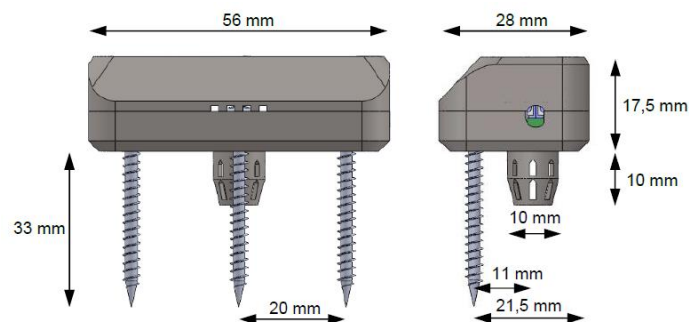
Moisture Guard je systém pro kontinuální monitoring vlhkosti uvnitř dřevěných konstrukcí vyvinutý výzkumnou skupinou na ČVUT-UCEEB. Skládá se z několika jednotlivých kombinovaných senzorů spojených napájecí a komunikační sběrnici a z centrální jednotky, která zajišťuje vyhodnocování dat ze senzorů a jejich odesílání do databáze UCEEB [24].

Obrázek 10: Logo Moisture Guard [27].

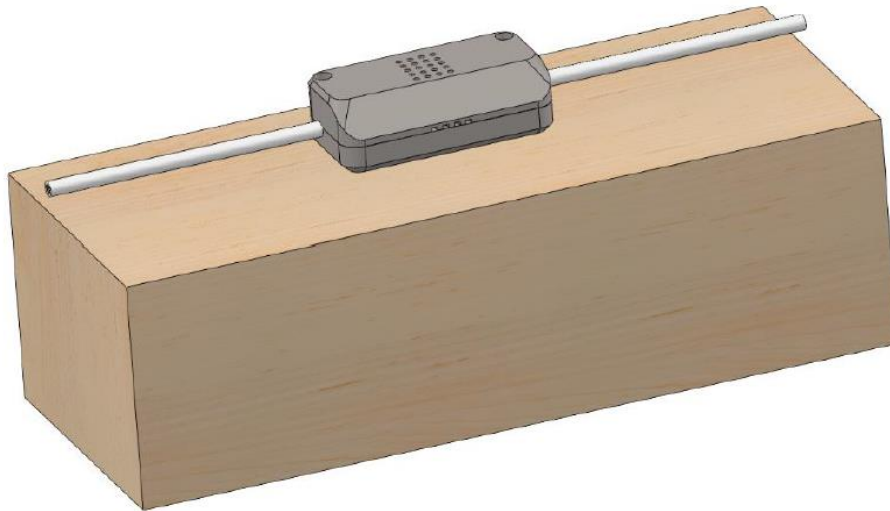
Hlavní senzor systému Moisture Guard nese označení MHT02485 a jedná se o kombinovaný senzor vlhkosti stavebních materiálů, vzdušné vlhkosti a teploty. Měřený materiál je penetrován měřicími elektrodami (vruty), které slouží zároveň pro připevnění senzoru ke sledované části konstrukce. Mezi elektrodami je měřen elektrický odpor, který je následně přepočítán na vlhkost materiálu. Výsledná hodnota je korigována na vliv teploty a druh dřeviny [28].



Obrázek 11: Konstrukční provedení senzoru MHT02485, hardware verze 2.2 [28].



Obrázek 12: Konstrukční provedení senzoru MHT02485, hardware verze 2.3 [29].



Obrázek 13: Připevnění senzoru na konstrukci pomocí vrutů [29].

Senzor MHT02485 v hardware verzi 2.2 je schopný měřit tyto veličiny:

- teplotu
- rosný bod
- relativní vzdušnou vlhkost
- absolutní vlhkost materiálu
- elektrický odpor mezi elektrodami
- absolutní vzdušnou vlhkost

Novější verze senzoru 2.3 měří kromě výše zmíněných veličin také:

- napájecí napětí
- teplotu v kavitě materiálu
- relativní vzdušnou vlhkost v kavitě
- rosný bod v kavitě
- absolutní vzdušnou vlhkost v kavitě

System Moisture Guard lze rozšířit o doplňkový senzor HT01485 určený pro monitorování teploty a vzdušné vlhkosti v místech, kde není třeba monitorovat vlhkost masivního materiálu.



Obrázek 14: Doplnkový senzor vzdušné vlhkosti a teploty [30].

Všechny senzory jsou navzájem propojeny sběrnici RS485, která umožňuje komunikaci na dostatečnou vzdálenost a je odolná vůči vnějšímu rušení. Na sběrnici je možné připojit až 32 senzorů, v případě připojení většího počtu zařízení je nutné použít opakovač. Pro komunikaci je použit protokol Modbus RTU, který je velmi rozšířený a umožňuje použití sensorového systému Moisture Guard s většinou průmyslových řídicích automatů [24]. Připojení senzoru na sběrnici se provádí pomocí krouceného kabelu s dvěma páry vodičů (jeden pár slouží pro napájení senzoru, druhý pro přenos dat).



Obrázek 15: Schéma připojení senzoru MHT02485 na sběrnici RS485 a připojení napájení [28].

4 Implementace komunikace po sběrnici RS485

Pro vyčítání hodnot ze senzorů Moisture Guard byla řídicí jednotka SDS-BIG naprogramována pro komunikaci po sběrnici RS485 s komunikačním protokolem Modbus RTU.

4.1 Vlastnosti sběrnice RS485

Sběrnice RS485 je sériová sběrnice hojně využívaná pro přenos dat a řízení v průmyslu. Na jedno signální vedení může být připojeno více zařízení, často v konfiguraci master/slave, kde každá jednotka slave má svoji unikátní adresu a odpovídá pouze na zprávy jí určené. Tyto zprávy generuje master a odesílá je na všechny připojené slave jednotky [31].

Fyzicky jsou zařízení na sběrnici propojeny pomocí dvou do sebe zkroucených vodičů (kroucený pár). Signál se přenáší po obou vodičích stejným směrem, přičemž úroveň napětí na vodičích se liší podle toho, jestli je přenášen logický stav 0 nebo 1. Logický stav je přijímačem vyhodnocován jako rozdíl napětí mezi vodiči A a B. Pokud je rozdíl V_{AB} větší než 200 mV, přijímač detekuje logický stav. Pokud je rozdíl V_{AB} menší než -200 mV, přijímač detekuje logický stav opačný.

Díky fyzickému uspořádání a vyhodnocování signálu pomocí rozdílu napětí mezi vodiči je sběrnice odolná vůči vnějšímu rušení. Šum je totiž indukován jak ve vodiči A, tak i ve vodiči B, a ve výsledku je souhlasná rušivá složka v přijímači od výsledného rozdílu napětí eliminována. Tato vlastnost sběrnice umožňuje přenášet signály pomocí sběrnice RS485 na relativně velké vzdálenost (až 1200 m) [31].

Standard RS485 popisuje fyzickou podobu sběrnice v rámci ISO/OSI modelu, nestanovuje však, jakou podobu mají mít telegramy na vyšších vrstvách, které se přes sběrnici posílají. Struktura přenášených telegramů je definována pomocí komunikačních protokolů. Pro komunikaci mezi řídicí jednotkou Moisture Guard a senzory je použit protokol Modbus RTU.

4.2 Protokol Modbus

Protokol Modbus je komunikační protokol, který slouží pro přenos informací po sériových linkách mezi elektronickými zařízeními. Byl vyvinut společností Modicon v roce 1979 pro použití s programovatelnými automaty této firmy [32]. Modbus je otevřený protokol a výrobci ho tedy mohou zdarma použít pro vlastní účely. V průmyslové automatizaci se Modbus stal standardem a je jedním z nejpoužívanějších běžně dostupných způsobů komunikace v tomto odvětví. Modbus existuje v několika verzích: Modbus RTU a Modbus ASCII pro komunikaci po sériových rozhraních a

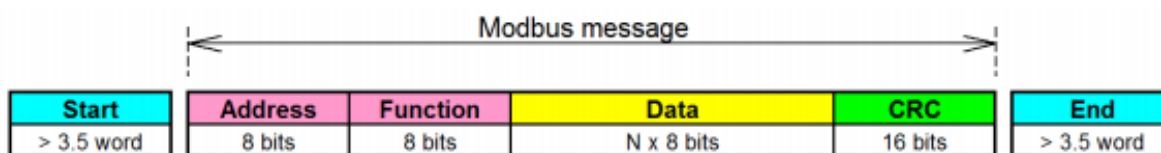
Modbus TCP pro komunikaci po rozhraní Ethernet. Každá verze se od sebe odlišuje strukturou zpráv, kterou je potřeba dodržet pro úspěšný přenos dat.

Protokol pracuje v konfiguraci master – slave, ve standardní Modbus síti se nachází vždy jeden master a až 247 jednotek slave, každá s vlastní unikátní adresou v rozsahu 1 až 247. Po sériové lince jsou data přenášena definovanou modulační rychlostí, typicky se jedná o rychlost 9600 nebo 19200 baud [32]. Data se přenášejí po znacích o 8 bitech a každý znak je reprezentován jedním číslem v hexadecimální soustavě v rozsahu 0x00 až 0xFF.

Pro navázání komunikace musí master nejprve odeslat dotaz, který v prvním bajtu uvádí adresu jednotky slave a v druhém bajtu kód funkce. Tento kód předá jednotce slave informaci o tom, zdali je požadován zápis do registrů nebo je požadováno vyčtení hodnot z registrů. Kód funkce obsahuje také informaci o tom, s jakými registry chceme pracovat.

4.2.1 Modbus RTU

Modbus RTU (Remote Terminal Unit) využívá pro přenesení jednoho znaku 11 bitů: 1 start bit, 8 bitů obsahujících znak, 1 paritní bit a 1 stop bit. Zřetězením několika znaků vznikne Modbus zpráva dlouhá maximálně 256 znaků. Jednotlivé zprávy jsou od sebe odděleny rámcovou mezerou dlouhou alespoň 3,5 znaku [28].



Obrázek 16: Struktura Modbus RTU zprávy [28].

Poslední dva bajty slouží pro přenos výsledku výpočtu CRC (Cyclic Redundancy Check). CRC výpočet slouží k detekci chyb vzniklých při přenosu zprávy a je vypočítán ze všech bajtů dané zprávy. Výpočet CRC provede nejprve odesílatel a připojí výslednou hodnotu na konec zprávy. Příjemce zprávy provede CRC výpočet z přijatých dat a porovná výsledek s hodnotou přijatou od odesílatele. V případě chyby při přenosu zprávy nejsou bity v odeslané a v přijaté zprávě identické a výsledná hodnota CRC výpočtu na straně odesílatele a příjemce je různá. Příjemce si tak může ověřit, zdali opravdu dostal stejnou zprávu, kterou mu odesílatel poslal [32].

4.3 Specifikace PLC modulu SDS-BIG

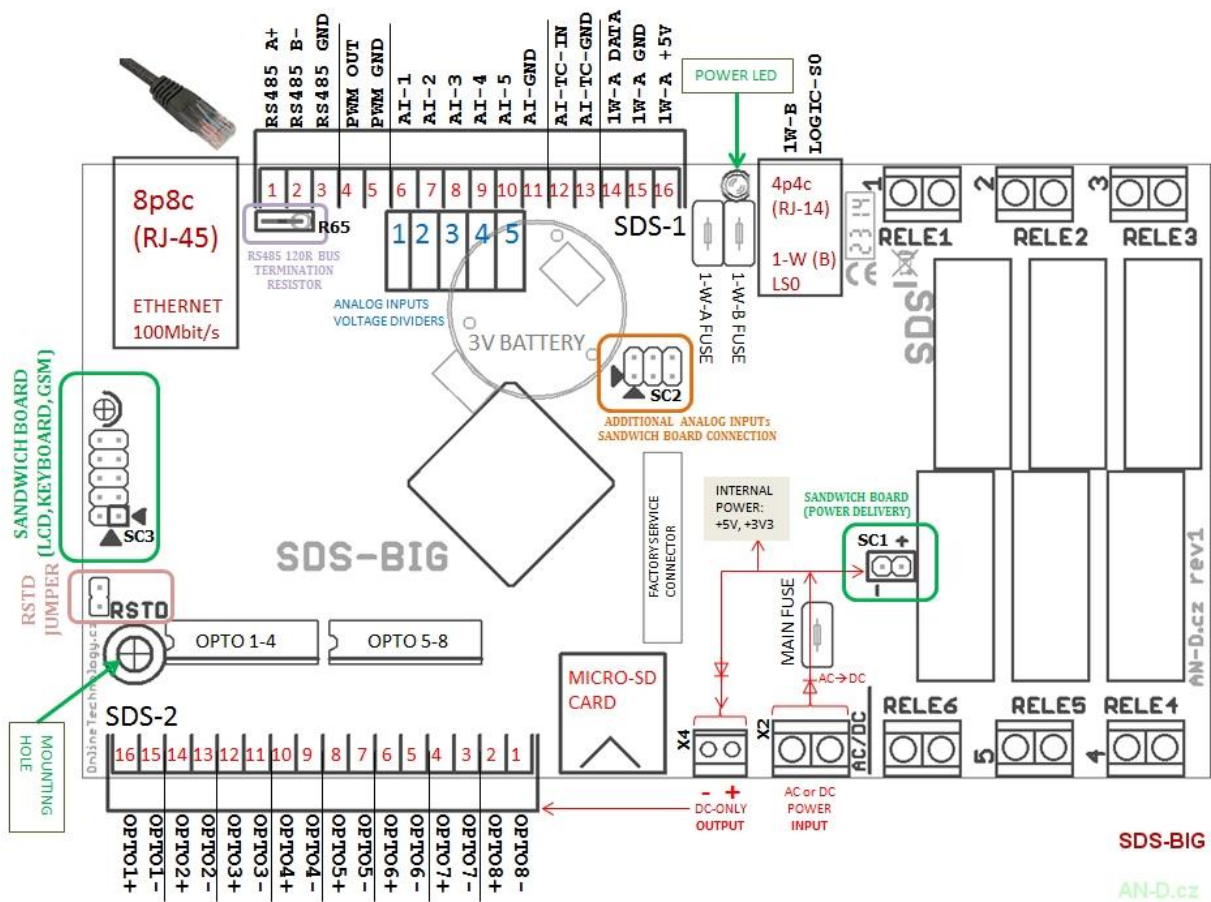
Pro vyčítání dat ze senzorů a jejich následné zpracování byl použit monitorovací a řídicí programovatelný automat SDS-BIG. SDS-BIG využívá řadu komunikačních protokolů pro získávání informací a měřených hodnot a umožňuje také nahrání vlastní HTML stránky do interního webserveru pro zobrazování informací a ovládání nastavení programu běžícího v SDS.

SDS-BIG je možné napájet napětím 12 – 24 V AC/DC, pro nahrávání programů do zařízení a připojení do sítě slouží rozhraní Ethernet s přenosovou rychlostí 100 Mbit/s. Pro získávání dat z připojených zařízení a jejich ovládání je k dispozici [33]:

- 8 optických vstupů pro měření odběru elektrické energie, vody a plynu, případně pro kontrolu stavu ON/OFF jiných zařízení
- 5 analogových vstupů s rozsahem 0 – 33 V
- PWM výstup
- odporový vstup
- 2 komunikační rozhraní sběrnice 1-wire pro připojení teploměrů
- převodník RS485
- 6 spínacích relé



Obrázek 17: Fotografie modulu SDS-BIG bez plastové DIN krabičky, ve které je modul umístěn [34].



Obrázek 18: Základní prvky modulu SDS-BIG a označení vstupů a výstupů [34].

Nastavení základních parametrů jako na příklad heslo nebo IP adresa probíhá skrze webový konfigurační systém, který je ve výchozím nastavení přístupný na adrese 192.168.1.250.

4.3.1 Programovací jazyk FULL-C

Pro zápis programu pro zařízení SDS-BIG je určen programovací jazyk FULL-C, který odpovídá standardu C90 [35]. Kromě omezeného počtu standardních funkcí definovaných v ANSI-C, FULL-C nabízí také řadu vlastních funkcí pro interakci se zařízením SDS [36]. Jedná se o funkce pro:

- zápis do konzole
- změnu hodnoty a čtení systémových proměnných (např. spínání relé, čtení teplot ze senzorů)
- práci s pamětí flash
- obsluhu sériových rozhraní
- řízení PWM výstupu
- šifrování odesílaných zpráv (funkce pro symetrickou kryptografii)
- komunikaci se serverem (pomocí HTTP *get* dotazů)
- odesílání e-mailů
- odesílání UDP paketů
- a mnoho dalších funkcí

Výpis z konzole je možné sledovat ve webovém prohlížeči po zadání IP adresy zařízení s příponou */echo.html* do adresového řádku. Ve výchozím nastavení lze tedy výpis z konzole sledovat na adrese *http://192.168.1.250/echo.html*.

Po sepsání programu se tento program vloží do aplikace FULLC.exe, která je dostupná na stránkách výrobce. Tato aplikace nejprve program přeloží do bajtkódu a ten následně přes síťové rozhraní nahraje do zařízení SDS. Program se automaticky spouští po každém jeho nahrání do SDS nebo po každém novém zapnutí zařízení [35].

Pro přístup k proměnným „z venku“ lze ve FULL-C programu deklarovat speciální proměnné (číselné hodnoty a textová pole), do kterých lze přes síťové připojení zapisovat hodnoty nebo je lze z modulu vyčítat. Díky tomu je možné sdílet hodnoty běžícího programu s uživatelskou webovou stránkou nahranou v zařízení SDS nebo s

jiným systémem komunikujícím po síti a také zasahovat do chodu programu bez nutnosti nahrávat upravený program znovu [37].

Pro změnu hodnoty proměnné je třeba zaslat zařízení HTTP *get* požadavek s potřebnými údaji obsaženými v URL adrese požadavku. Pokud bychom chtěli např. změnit hodnotu proměnné S00 na -123, proměnné U02 na 1000 a proměnné na T00 na „hello“, HTTP požadavek pro tento příkaz by vypadal následovně [37]:

```
192.168.1.250/sv?S00=-123&U02=1000&T00=hello
```

Ze struktury požadavku je vidět, že nejprve se uvádí IP adresa zařízení, poté název příkazu (*sv*) a následně názvy proměnných a požadované hodnoty oddělené znakem &.

4.4 FULL-C kód pro komunikaci po sběrnici

Aby mohla být zahájena komunikace po sběrnici RS485, je nejprve třeba v programu nastavit parametry sériové linky. To se provádí pomocí funkce `SDS_serial_config` [38]. Parametry pro komunikaci s využitím protokolu Modbus RTU jsou následující:

- `busID` – pro komunikaci po sběrnici RS485 se volí hodnota 1
- `BaudRateCfg` – 19200 Bd
- `DataSizeCfg` – 9 bitů
- `StopBitsCfg` – 1 bit
- `ParityCfg` – 2 (sudá parita)

Program podporuje komunikaci až s 64 senzory současně. Počet byl zvolen s ohledem na maximální předpokládaný počet připojených senzorů v rámci jednoho systému Moisture Guard a na velikost paměti zařízení SDS-BIG.

4.4.1 Odesílání a přijímání zpráv Modbus RTU

První část zdrojového kódu má na starosti komunikaci se senzory pomocí protokolu Modbus RTU. Pro tento účel byly v programovacím jazyce FULL-C sestaveny tyto funkce:

`int countCRC(int length, int bufferType)` – Provede CRC výpočet ze vstupních dat o délce `length` uložených v poli `txBuffer` (pole pro odeslání dotazu jednotce slave) nebo v poli `res` (pole pro uložení odpovědi). Výběr pole je proveden zadáním argumentu `TX_BUFFER_TYPE` nebo `RX_BUFFER_TYPE` na místo parametru `bufferType` při volání funkce.

`void TxCRC()` – Přidá na konec odesílané zprávy dva bajty s výsledkem CRC výpočtu.

`void RxCRC()` – Porovná výsledek CRC výpočtu s hodnotou CRC v přijaté zprávě. Pokud se oba výsledky shodují, do proměnné `CRCError` je uložena hodnota `FALSE`. V případě, že se neshodují, je do proměnné `CRCError` uložena hodnota `TRUE`.

`void sendFunction(int sensorAddr, int functionCode, int startAddr, int registerCount)` – Připraví hodnoty v poli `txBuffer` pro odeslání jednotce slave. Obsah indexů v poli odpovídá pravidlům protokolu Modbus RTU: první bajt zprávy obsahuje adresu jednotky slave, druhý bajt obsahuje kód funkce, následují dva bajty s počáteční relativní adresou registru, ze kterého chceme data číst, a dva bajty označující počet registrů, které chceme číst.

`int readData(int maxRepeat)` – Naplní pole `res` daty přijatými ze sériové linky. Argument `maxRepeat` stanovuje maximální počet opakování funkce pro případ, že data dorazí se zpožděním nebo dojde k porušení zprávy během přenosu. Ve výchozím nastavení je počet pokusů pro čtení hodnot nastaven na 10, pro enumeraci (více o funkci `enumeration` níže) je pak počet pokusů stanoven na 5, aby se celý proces enumerace zrychlil. Interval opakování je nastaven ve výchozím nastavení konstantou `READ_TIMEOUT` na dobu 100 ms. Funkce vrací hodnotu 0, pokud nebyla přijata žádná data nebo celé číslo odpovídající délce přijaté zprávy.

`void readValues(int registerCount)` – Funkce, která nejprve odešle dotaz pro vyčtení hodnot, počká dobu stanovenou konstantou `WRITE_TIMEOUT` (ve

výchozím nastavení 100 ms) a poté zavolá funkci `readData`. Vráti hodnotu 1, pokud data byla úspěšně přijata, nebo 0, pokud došlo při přenosu k chybě.

`void readSensors()` - Naplní příslušná pole naměřenými hodnotami ze senzorů, např. pole `tempArray` naplní údaji o teplotě. Funkce zavolá funkci `readValues` pro každý připojený senzor pomocí cyklu `for`.

4.4.2 Enumerace

Další sekce zdrojového kódu obsahuje funkce pro detekci připojených senzorů na sběrnici a rozpoznání jejich typů.

`void enumeration()` - Funkce odešle dotaz pomocí funkce `readSensorType()` na každou adresu v rozsahu stanoveného sdílenými proměnnými `U03` a `U04`. Vy výchozím nastavení funkce `enumeration()` provádí hledání připojených senzorů v rozsahu adres 1 až 21. Pokud je na dotaz přijata odpověď, adresa a typ senzoru se zapíše do pole `addrArray`, resp. `typeArray`.

4.4.3 DataFlash

V této části jsou deklarovány funkce pro práci s pamětí flash. Paměť flash uchovává data i po odpojení napájení a je proto vhodná pro uložení nastavení programu, které díky tomu není nutné nastavovat znovu po každém zapnutí zařízení. Nedoporučuje se do paměti flash ukládat proměnné, jejichž hodnoty se často mění, protože životnost tohoto typu paměti je omezená počtem zápisů (mezi 10 tisíci a 100 tisíci zápisů) a předcházíme tím jejímu poškození [39].

`void saveToFlash()` - Uloží do paměti flash informace o adresách a typech připojených senzorů. Díky tomu není nutné spouštět enumeraci, která je poměrně zdlouhavá, po každém zapnutí zařízení. Do paměti se uloží také nastavení programu, konkrétně jde o rozsah adres pro enumeraci (proměnné `U03` a `U04`), nastavení odesílání po Ethernetu (`U08`), šifrování dat (`U09`), nastavení odesílání dat po síti LoRa (`U10`) a interval mezi odesláním jednotlivých měření po Ethernetu (`U12`) a síti LoRa (`U13`).

`void loadFromFlash()` – Nahraje do programu po jeho spuštění informace uložené v paměti flash. Pokud je paměť prázdná, (uživatel si žádné vlastní nastavení neuložil), zahájí enumeraci a nastaví výchozí nastavení programu.

`void deleteFlash()` – Vymaže všechny informace uložené v DataFlash.

5 Přenos dat do databázového systému

Naměřené hodnoty ze senzorů jsou ukládány v databázovém systému UCEEB a následně jsou tyto data pomocí algoritmů vyhodnocena. V případě dosažení kritického stavu je uživatel upozorněn na vznikající poruchu. Odesílání dat z SDS-BUG na server je uskutečněno přes rozhraní Ethernet pomocí zpráv protokolu UDP a po LPWAN síti LoRa. Díky implementaci komunikace po síti LoRa je možné přenášet data do databáze také z míst, kde není přístup k internetu. V současné době nejsou sítě postavené na technologii LoRa v ČR příliš dostupné, to by se ale v budoucnosti mělo změnit.

Řídící jednotka je také schopna získat nastavení ze serveru pomocí dotazů HTTP *get*, komunikace přes rozhraní Ethernet je tedy oboustranná. Na straně serveru tyto funkce vzdáleného nastavení zatím nejsou připravené, nicméně funkčnost komunikace byla otestována pomocí zprávy, kterou SDS-BIG ze serveru přijalo, jak je vidět na výpisu z konzole níže.

```
IP address loaded.  
Resolving data.moistureguard.cz  
Received IP address: 147.32.99.171  
Calling http get...  
Http get called.  
OK. Response value: 200 Received data size: 5  
Server response: 00300
```

Obrázek 19: Výpis z konzole. Řídící jednotka přijala ze serveru zprávu „00300“.

5.1 Přenos přes rozhraní Ethernet

UDP slouží pro komunikaci mezi klientem a serverem, přenos dat je zprostředkován pomocí krátkých zpráv (tzv. *datagramů* nebo *paketů*). Payload

datagramu je složen ze znaků, jejichž pořadí a význam musí odpovídat struktuře požadované serverem. Server, na který jsou data odesílána (nachází se v UCEEBu v Buštěhradě), vyžaduje datagramy složené z nezašifrované části (hlavička datagramu) a zašifrované části obsahující přenášená data. První část datagramu je dlouhá 23 bajtů a obsahuje tyto informace:

23B			
4B	1B	16B	2B
ID	Typ PLC	IV	Délka

Obrázek 20: Nezašifrovaná část datagramu.

- ID – Identifikace konkrétního PLC. Musí být unikátní v databázovém systému.
- Typ PLC – Rozlišení mezi typem odesílacího PLC. Důležité kvůli rozdílnostem v šifrování.
- IV – Inicializační vektor pro šifrování.
- Délka – Počet následujících bajtů (délka zašifrované části).

Následuje zašifrovaná část obsahující libovolný počet bajtů, vždy však musí být celkový počet bajtů dělitelný čtyřmi.

14B						N x 4B	2B
4B	2B	2B	2B	2B	2B	N x 4B	2B
Čas	Start	Počet	Packet ID	Číslo packetu	Počet paketů	Data	ChckSum

Obrázek 21: Struktura zašifrované části datagramu.

- Čas – Časová značka UNIX timestamp.
- Start – Počáteční adresa (index) dat v konkrétním paketu.
- Počet – Počet přenášených hodnot (počet čtyřbajtových slov) v konkrétním datagramu.
- Packet ID – Pořadové číslo datagramu. Datagramy, které patří k sobě, mají stejný identifikátor.

- Číslo paketu – Obsahuje pořadové číslo datagramu z množiny datagramů, které přenášejí k sobě patřící data.
- Počet paketů – Celkový počet datagramů, které přenášejí k sobě patřící data.
- Data – Pole dat po čtyřech bajtech. Datový typ může být integer 32-bit nebo float, typ musí být předem známý jak v PLC, tak v databázi.
- Checksum – Kontrolní součet rozšifrovaných dat (slouží jako detekce správného klíče). Jedná se o jednoduchý součet všech bajtů zašifrované části.

Zabezpečení přenášených dat je zajištěno šifrováním AES s datovým blokem délky 128 bitů a mechanismem Counter-Mode (CTR). Aby protistrana mohla data dekódovat, je potřeba odeslat v první části datagramu nezašifrovaný inicializační vektor (IV) o délce 128 bitů, který je složen ze čtyř náhodných 32bitových čísel. IV musí být pro každý datagram unikátní, v opačném případě by útočník mohl šifru snadno rozluštit. Kromě IV musí protistrana znát také klíč, který je neměnný a nastavuje se ve webové administraci SDS-BIG. Klíč je dlouhý 128 bitů a zapisuje se do flash (nevolatilní) paměti zařízení. Samotné zašifrování zprávy provádí funkce `SDS_crypto`, která je jednou z vlastních funkcí SDS-BIG [40].

CHANGE CRYPTO KEY

Password	<input type="text"/>
CRYPT-16B KEY (32-chars)	<input type="text" value="770A8A65DA156D24EE2A0932775"/>
Set the new key	<input type="button" value="store"/>

Obrázek 22: Pole pro zapsání klíče ve webovém konfigurátoru SDS-BIG.

5.1.1 FULL-C kód pro komunikaci po rozhraní Ethernet

`void createPacket(int start)` – Vyplní pole `udpBuffer` daty, které tvoří nezašifrovanou hlavičku paketu. Parametr `start` určuje počáteční adresu (index) dat v konkrétním paketu.

`void checksum()` – Provede kontrolní součet všech bajtů zašifrované části zprávy.

`void sendSensorAddrToServer()` – Naplní příslušná pole datagramu údaji o Modbus adresách připojených senzorů. Obdobně fungují i následující funkce, které posílají do databáze údaje o typech senzorů, teplotách apod. Je zajištěno, aby část paketu obsahující data byla dlouhá vždy alespoň 16 bajtů kvůli šifrování (funkce AES šifrování vyžaduje délku paketu dělitelnou 16).

`void encrypt()` – Zašifruje datagram metodou AES-128-CTR s využitím vlastní funkce zařízení SDS `SDS_crypto`.

`void sendUDPPacket()` – Odešle UDP paket pomocí funkce `udp_send`.

`void sendToServer(int start)` – Odešle postupně všechny pakety na server. Nejprve jsou odeslány pakety obsahující informaci o adresách a typech senzorů, následují pakety s naměřenými údaji (teplota, vlhkost atd.). Celkem je odesláno 8 paketů.

`void waitForIP(int start)` – Vrátí hodnotu 1, pokud je řídicí jednotce přidělena IP adresa z DHCP serveru. V případě, že do 5 sekund není přidělena IP, funkce vrátí hodnotu 0.

`void getServerIP(int start)` – Odešle dotaz na DNS server s doménovým jménem, které je uloženo v konstantě `IP_DOMAIN_STRING`. Pokud byla přijata IP adresa od DNS serveru, je uložena do proměnných `IP0` až `IP3`. V případě selhání vypíše funkce do konzole hlášku o chybě a číslo chyby. Tato funkce byla v kódu zakomentována, jelikož potřebuje k jejímu chodu příliš velké množství paměti a blokuje běh programu. V budoucnu bude potřeba tento problém vyřešit snížením počtu globálních proměnných nebo přechodem na řídicí jednotku s větší pamětí.

`void getResponseFromServer(int start)` – Odešle na server požadavek HTTP *get* pomocí vlastní funkce SDS `http_get`. Přijatá data ze serveru se uloží do dynamicky alokovaného pole `receiveDataBuffer`. Tato funkce byla v kódu zakomentována ze stejného důvodu jako funkce `getServerIP`.

5.2 Přenos po LPWAN síti LoRa

Přenos dat po síti LoRa umožňuje zařízení SDS-BIG speciální modul, který je možné zakoupit společně se zařízením nebo časem zařízení SDS o tento modul rozšířit. Připojení k libovolné LoRa síti a sestavení zpráv obstarává FULL-C program. Pro úspěšnou komunikaci je třeba po instalaci LoRa modulu ve FULL-C programu nastavit unikátní identifikátory – aplikační klíč, pomocí kterého dešifruje zprávu koncový uživatel (pro provozovatele sítě je tedy zpráva nečitelná – nezná aplikační klíč) a síťový klíč, který zabezpečuje komunikaci mezi zařízením SDS a provozovatelem sítě.

Obecně lze zprávy posílat s očekáváním potvrzení přijetí (ACK) od provozovatele nebo bez potvrzení. V případě SDS-BIG, program očekává ACK. Pokud ho neobdrží, je odeslání zprávy opakováno [41].

Před připojením do sítě je potřeba do FULL-C programu zadat tyto údaje: [13]:

- DevEUI – Device EUI, globální unikátní 64-bitový identifikátor zařízení (analogický k MAC adrese)
- DevAddr – Device Address, 32-bitový identifikátor, který musí být v síti jedinečný (analogický k IP adrese)
- NwkSKey – Network Session Key, 128-bitový síťový klíč
- AppSKey – Application Session Key, 128-bitový aplikační klíč

Všechny údaje pro připojení poskytuje provozovatel sítě, kterým je v tomto případě UCEEB.

5.2.1 Komunikační modul RN2903

V zařízení SDS-BIG je použit LoRa modul RN2903 od společnosti Microchip, popis všech příkazů určených k ovládání modulu je k nalezení v příslušném manuálu. Modul s SDS komunikuje pomocí sériového rozhraní UART. Pro zahájení komunikace je proto nejprve potřeba nastavit parametry sériové linky (podobně jako v případě nastavení komunikace po sériové lince RS485). Nastavení probíhá pomocí SDS funkce

SDS_serial_config. Tato funkce vyžaduje 5 parametrů, kromě busID jsou zbylé hodnoty k nalezení v manuálu k RN2903 [42]:

- busID – pro komunikaci s LoRa modulem se používá sběrnice COM6, proto busID = 6
- BaudRateCfg – 57600 Bd
- DataSizeCfg – 8 bitů
- StopBitsCfg – 1 bit
- ParityCfg – 0 (bez parity)

Příkazy jsou odesílány v kódování ASCII a každý příkaz musí být zakončen znaky <CR><LF>. V jazyce C jsou těmto znakům ekvivalentní znaky \r\n.

5.2.2 Odesílání příkazů a příjem odpovědí

`void senddataLora(char * write)` – Odešle po sériové lince znaky, které jsou funkci předány v poli `write`.

`int readdataLora()` – Naplní pole `_answer` znaky, které SDS přijalo od LoRa modulu. Funkce vrátí nulu v případě, že nebyla přijata žádná data, nebo celé číslo odpovídající délce přijaté zprávy.

5.2.3 Tovární nastavení a připojení do sítě

Po zapnutí zařízení SDS a povolení komunikace po síti LoRa je provedeno nastavení parametrů a klíčů modulu zavoláním funkce `factoryreset` a následné připojení do LoRa sítě.

`factoryreset()` – SDS-BIG odešle modulu RN2903 sérii příkazů pomocí rozhraní UART:

- `sys factoryRESET` – obnoví výchozího nastavení modulu a provede restart modulu
- `mac set deveui` – nastaví unikátní identifikátor modulu DevEUI
- `mac set devaddr` – nastaví unikátní síťový identifikátor DevAddr
- `mac set nwkskey` – nastaví síťový klíč
- `mac set appskey` – nastaví aplikační klíč

- `mac set retx 7` – nastaví počet opakovaných odeslání zprávy pro případ, že nebylo od serveru přijato potvrzení o doručení ACK
- `mac set adr on` – zapne režim ADR (Adaptive Data Rate), který umožňuje upravit přenosovou rychlost tak, aby byla minimalizována spotřeba energie zařízení, byla zajištěna spolehlivost přenosu zpráv a bylo optimalizováno využití sítě. V praxi to znamená, že zařízení blízko ke koncentrátoru zvolí vyšší rychlost přenosu, díky čemuž vysílají zprávy kratší dobu a spotřebují méně energie. Naopak zařízení nacházející se dále od základnové stanice vysílají při nižší rychlosti a s vyšší spotřebou energie, aby bylo možné zprávy přenést na větší vzdálenost [43].
- `mac save` – uloží všechna nastavení, která byla nastavena pomocí příkazů `mac set` (kromě `set retx`) do EEPROM paměti modulu

Následně je odeslán příkaz `mac join abp`, který připojí zařízení do sítě LoRa v módu `abp` (Activation by Personalization). Poslední funkcí, která je zavolána v rámci připojení zařízení do sítě je funkce `getOutputPower`.

`int getOutputPower()` – Odešle příkaz `get pwridx` a vrátí aktuální hodnotu indexu vysílacího výkonu.

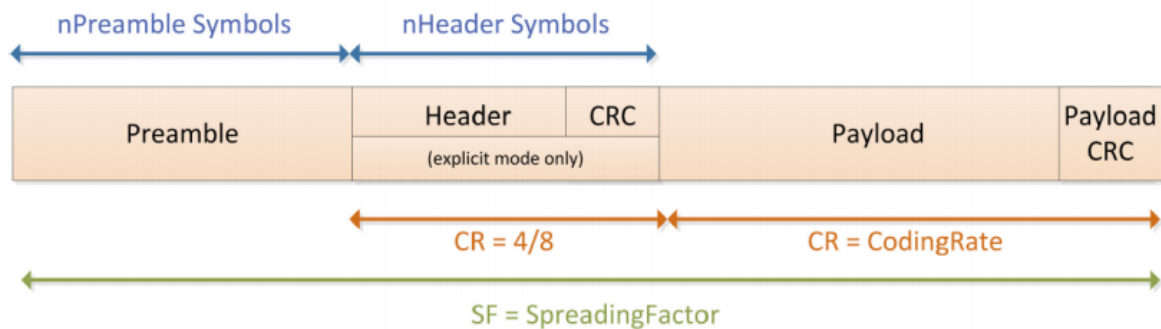
5.2.4 Odesílání dat ze senzorů

Interval odesílání zpráv, které obsahují data (payload) je ve výchozím nastavení stanoven na 60 min. Sestavení payloadu je provedeno pomocí funkcí `sendAddrLora` (pro adresy a typ senzorů) a `sendValuesLora` (pro měřené veličiny). Postupně je odesláno 8 paketů, které obsahují adresu senzoru, jeho typ, teplotu, vzdušnou relativní vlhkost, absolutní vlhkost materiálu, elektrický odpor, teplotu a vzdušnou relativní vlhkost v kavitě.

Jednotlivé zprávy od sebe musí dělit pauzy tak, aby přenos respektoval podmínku ISM pásma na klíčovací poměr 1% (*duty cycle*). V praxi to znamená, že pokud vysílací doba zprávy je 1 s, zařízení musí následně počkat alespoň 99 s než odešle další zprávu. Vysílací doba zprávy závisí na [44]:

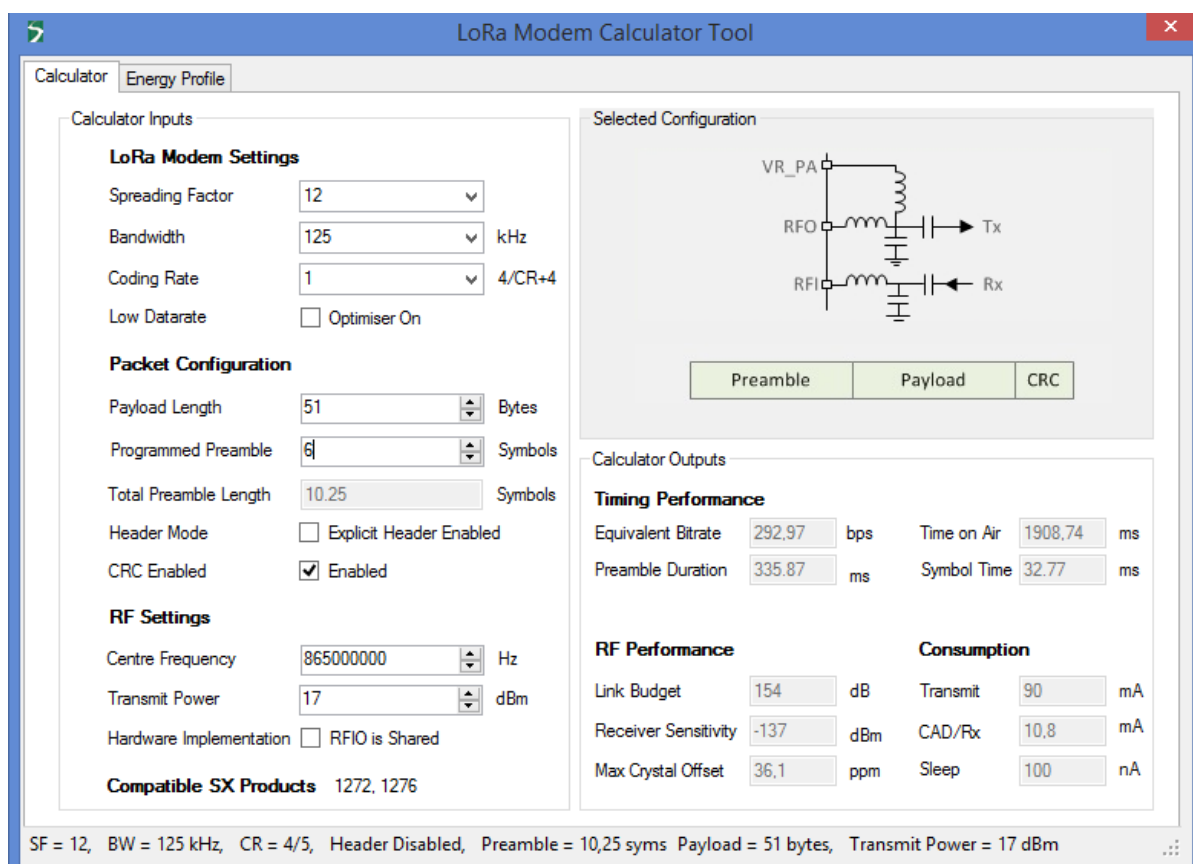
- délce payloadu zprávy
- šířce frekvenčního pásma (*bandwidth*)

- spreading factoru
- poměru kódování (*coding rate*) - poměr mezi počtem přenášených „užitečných“ bitů (které obsahují např. data z čidla) a „přebytečných“ bitů (např. v hlavičce)
- přítomnosti hlavičky zprávy (*header*) – může obsahovat informace o délce payloadu, výpočet CRC apod.
- aktivaci módu pro optimalizaci pro nízké přenosové rychlosti



Obrázek 23: Struktura LoRa paketu. Každý paket musí obsahovat preamble a payload, ostatní části jsou volitelné [44].


Semtech uvádí na svých stránkách rovnici pro orientační výpočet vysílací doby zprávy, pro zjednodušení výpočtu je k dispozici také aplikace pro OS Windows. Po zadání nejnejpříznivějších hodnot, tedy spreading factoru 12, šířce pásma 125 kHz a maximální délce payloadu 51 bajtů pro SF 12 (odpovídá odesílání dat z 25 senzorů) je aktivní vysílací doba zprávy (*time on air*) odhadována na 1908 ms.



Obrázek 24: Aplikace pro výpočet vysílací doby LoRa zprávy a dalších komunikačních parametrů [45].

V reálné instalaci bude vysílací doba zprávy velmi pravděpodobně nižší, protože zpráva bude kratší (bude připojeno méně senzorů) a SF bude nastaven na nižší hodnotu (zařízení se nebude nacházet na samém okraji dosahu koncentrátoru). Interval mezi jednotlivými zprávami lze ve FULL-C programu nastavit změnou konstanty MESSAGE_INTERVAL. Ve výchozím nastavení je tento interval stanoven na 60 vteřin, aby nedocházelo k porušení podmínky klíčovacího poměru 1%. V případě, že tento interval není dostatečně dlouhý, modul RN2903 (stejně jako všechny LoRa certifikované moduly) sám hlídá dodržení této podmínky. Pokud dojde k jejímu porušení, modul vrátí odpověď `no_free_ch` a po jedné minutě se pokusí o odeslání zprávy znovu.

Přijaté pakety je možné zobrazit v aplikaci LORIoT, která slouží také ke správě koncových zařízení připojených ke koncentrátoru.

Dashboard > Applications > SampleApp Proof of Concept  pavel.mlejnek@fel.cvut.cz

an AppEUI

Downlink (TX)
enables downlink API Enabled

Gateway information feed
per-gateway data feed verbosity

Output verbosity
feed radio information per packet Extended

Device EUI	SeqNo	Time	Port	Data
BE-7A-00-00-00-00-00-2E	8	3 minutes	1	00 00 00 00 0d 60 d 7b
BE-7A-00-00-00-00-00-2E	7	4 minutes	1	00 00 00 00 09 69 09 55
BE-7A-00-00-00-00-00-2E	6	5 minutes	1	03 fa 04 06 0c 5f c2 00
BE-7A-00-00-00-00-00-2E	5	6 minutes	1	00 00 00 00 00 00 0b eb
BE-7A-00-00-00-00-00-2E	4	8 minutes	1	09 7c 09 76 08 68 02 9e
BE-7A-00-00-00-00-00-2E	3	9 minutes	1	0e a5 0e 40 0d 3a 0d 80
BE-7A-00-00-00-00-00-2E	2	10 minutes	1	09 3a 09 4d 09 43 09 2e
BE-7A-00-00-00-00-00-2E	1	11 minutes	1	04 04 06 06
BE-7A-00-00-00-00-00-2E	0	12 minutes	1	01 02 0c 0d

Obrázek 25: Zobrazení přijatých paketů v aplikaci Lorient.

6 Sestavení webové stránky

SDS-BIG obsahuje webový server, který umožňuje uživateli do zařízení nahrát vlastní dokument v jazyce HTML a ten pak zobrazit v libovolném webovém prohlížeči po zadání výchozí adresy `http://192.168.1.250/user.htm`. Je tedy možné vytvořit webovou stránku, která zobrazuje stav programu (např. hodnotu sdílených proměnných) nebo zasahovat do jeho chodu a tím ho řídit (např. sepnout relé, upravit interval odesílání zpráv na server apod.).

Webová stránka, která byla pro účely tohoto projektu vytvořena, používá programovací jazyk HTML pro sestavení základní struktury webové stránky, CSS (kaskádové styly) pro úpravu vzhledu HTML prvků a JavaScript pro odesílání dotazů na server a dynamickou změnu obsahu webové stránky. Webová stránka je rozdělena do několika bloků.

Moisture Guard System Overview

Status: **RUNNING** SDS Uptime: 4h 9m 20s
 Sensor count: 6

Enumeration Status: DONE Enumeration range: from 1 to 56 Set a new range: first address <input type="text"/> last address <input type="text"/> <input type="button" value="Submit"/> <input type="button" value="Start enumeration"/>	Sending to server - Ethernet Status: Enabled Packet interval: 10 min Encryption: Enabled Set a new interval (in minutes): <input type="text"/> <input type="button" value="Disable sending to server"/> <input type="button" value="Disable encryption"/> <input type="button" value="Submit"/>
DataFlash Status: Settings loaded from flash <input type="button" value="Save settings"/> <input type="button" value="Delete settings"/>	Sending to server - LoRa Status: Disabled Packet interval: 60 min <input type="button" value="Enable sending with LoRa"/> Set a new interval (in minutes): <input type="text"/> <input type="button" value="Submit"/>

Sensor No.	Address	Type	Temperature	Humidity	Moisture	Resistance	Temperature 2	Humidity 2
1	1	4	24.96 °C	30.54 %	24.07 %	1.0 MOhm	-	-
2	2	4	24.62 °C	31.38 %	24.12 %	1.0 MOhm	-	-
3	12	6	24.69 °C	28.57 %	21.36 %	3.2 MOhm	25.12 °C	28.38 %
4	13	6	24.74 °C	27.98 %	6.63 %	200.0 GOhm	25.25 °C	27.48 %
5	55	1	24.80 °C	30.38 %	-	-	-	-
6	56	1	27.47 °C	30.93 %	-	-	-	-

Obrázek 26: Uživatelská webová stránka pro zobrazení aktuálních naměřených hodnot a změnu nastavení programu.

Vrchní panel zobrazuje stav programu, který může nabývat stavů *RUNNING*, pokud program běží, nebo *Starting*, pokud program načítá nastavení z paměti flash a nebo probíhá počáteční enumerace (v případě, že v paměti flash žádné předchozí nastavení uloženo není).

Status: **RUNNING** SDS Uptime: 1h 2m 52s
 Sensor count: 4

Obrázek 27: Zobrazení stavu programu a počtu připojených senzorů.

Pod stavem se zobrazuje počet senzorů načtených pomocí poslední enumerace a tlačítko pro restart programu. Restart opětovně načte nastavení z paměti flash a vynuluje interval pro odesílání zpráv po síti Lora a přes rozhraní Ethernet. Kromě toho také nastaví proměnnou `packetID` na hodnotu nula (UDP pakety budou znovu číslovány od nuly). Na pravé straně se zobrazuje *SDS uptime*, tedy čas uplynulý od připojení zařízení k napájení.

V bloku *Enumeration* lze nastavit rozsah Modbus adres, ve kterých je prováděno hledání připojených senzorů, a zahájit enumeraci kliknutím na tlačítko *Start enumeration*.

Enumeration
 Status: **DONE**
 Enumeration range: from **1** to **14**
 Set a new range: first address last address

Obrázek 28: Nastavení parametrů enumerace.

Stav enumerace je zobrazen ve statusu a je buď *DONE* nebo *Enumeration in progress*, pokud enumerace zrovna probíhá.

Panel DataFlash umožňuje uložit stávající nastavení do paměti flash nebo nastavení z paměti vymazat. V takovém případě při příštím restartu programu nebo připojení SDS k napájení proběhne enumerace ve výchozím rozsahu adres 1 – 21, interval odesílání UDP datagramů se nastaví na 5 min a interval odesílání LoRa zpráv se nastaví na 60 min.

DataFlash
 Status: **Settings loaded from flash**

Obrázek 29: Blok pro uložení nebo vymazání nastavení z Data Flash.

Parametry přenosu dat do databázového systému UCEEB je možné upravit v bloku *Sending to server – Ethernet*. Uživatel může povolit odesílání UDP paketů na server, zapnout jejich šifrování pomocí metody AES-128 nebo nastavit interval odesílání jednotlivých zpráv. Ve výchozím nastavení je šifrování a odesílání na server pomocí rozhraní Ethernet povoleno. Minimální interval je 1 min, maximální 255 minut. Do budoucna se počítá se zabezpečením této sekce pomocí hesla.

Sending to server - Ethernet
 Status: **Enabled** Packet interval: **5** min
 Encryption: **Enabled** Set a new interval (in minutes):

Obrázek 30: Nastavení odesílání UDP paketů na server.

Obdobně lze také povolit odesílání pomocí sítě LoRa a nastavit interval mezi zprávami v sekci *Sending to server – LoRa*. Intervalem se rozumí doba mezi odesláním zpráv se stejným typem informace (např. mezi dvěma zprávami s údaji o adresách senzorů). Minimální povolený interval je 8 minut z důvodu dodržení podmínky klíčovacího poměru 1% (jednotlivé pakety od sebe dělí 60 vteřin a celkem je paketů 8), maximální interval je pak 255 minut.

Obrázek 31: Nastavení odesílání zpráv po síti LoRa.

Ve spodní části webové stránky se nachází tabulka zobrazující Modbus adresy senzorů nalezených pomocí enumerace, jejich typ a aktuální hodnoty měřených veličin. Program podporuje tři typy Moisture Guard senzorů, konkrétně se jedná o HT01485 (typ 1), MHT02485 verze 2.2 (typ 4) a MHT02485 verze 2.3 (typ 6). Počet řádků tabulky se mění s počtem připojených senzorů. Pokud dojde při vyčítání dat z některého ze senzorů k chybě, místo naměřené hodnoty se zobrazí text *Error*.

Sensor No.	Address	Type	Temperature	Humidity	Moisture	Resistance	Temperature 2	Humidity 2
1	1	4	25.00 °C	29.94 %	24.06 %	1.0 MOhm	-	-
2	2	4	24.61 °C	30.98 %	24.09 %	1.0 MOhm	-	-
3	12	6	24.68 °C	28.21 %	21.36 %	3.2 MOhm	25.11 °C	27.67 %
4	13	6	24.75 °C	28.18 %	6.63 %	200.0 GOhm	25.28 °C	27.67 %
5	55	1	24.84 °C	29.90 %	-	-	-	-
6	56	1	27.57 °C	30.01 %	-	-	-	-

Obrázek 32: Zobrazení aktuálně naměřených údajů.

Díky použití technik AJAX (Asynchronous JavaScript and XML) jsou zobrazovány aktuální hodnoty měřených veličin dynamicky, aniž by bylo třeba pro získání nových údajů nutně webovou stránku obnovit. AJAX není programovacím jazykem, jedná se o souhrn programovacích metod, které lze použít v rámci skriptu napsaného v jazyce

JavaScript. Poslední písmeno ve zkratce odkazuje na datový formát XML (Extensible Markup Language), ve kterém je možné informaci ze serveru pomocí metod AJAX obdržet [46].

Kroky potřebné pro odeslání dotazu na server a změnu údajů, které probíhají na pozadí webové stránky, jsou následující: [47]:

1. Na stránce nastane nějaká událost: uživatel klikne na jedno z tlačítek nebo je zavolána funkce `readSensors`. Funkce `readSensors` je volána periodicky každou vteřinu pro načtení aktuálních hodnot ze senzorů.
2. Je vytvořen nový objekt `XMLHttpRequest` pomocí JavaScriptu a uložen do proměnné `xmlHttp`.
3. Je navázáno spojení se serverem pomocí metody `xmlHttp.open`, která vyžaduje tři parametry: typ metody, kterou je v tomto případě `get`, URL adresu a booleovskou hodnotu, která je nastavena na `true`, tedy pro asynchronní provádění požadavku. Adresa je zadávána ve tvaru:
 - a. pro změnu hodnoty proměnné, např: `/sv?U08=1` (změní hodnotu proměnné `U08` na 1
 - b. `shared.txt` pro vyčtení hodnot všech sdílených proměnných
4. Je odeslán požadavek na server pomocí metody `xmlHttp.send`.
5. V případě vyčítání hodnot je do proměnné `data` uložena odpověď serveru.
6. Pomocí DOM (Document Object Model) metod `document.getElementById` a `document.innerHTML` jsou přepsány staré hodnoty zobrazené na stránce aktuálními hodnotami ze senzorů.

Kompletní zdrojový kód webové stránky je k nalezení v příloze této práce. V systému Moisture Guard bude webová stránka sloužit jako přehled aktuálně měřených hodnot a konfigurátor, pomocí kterého lze rychle nastavit požadované parametry programu jako na příklad rozsah enumerace nebo interval odesílání zpráv na server. Díky tomu není nutné zasahovat do zdrojového kódu programu pro řídicí jednotku SDS a je možné ho použít univerzálně pro různé instalace systému Moisture Guard.

Závěr

Úvodní teoretická část této práce byla věnována popisu technologií Sigfox a LoRa pro bezdrátový přenos dat pomocí radiových vln, které umožňují přenášet hodnoty ze senzorů na velké vzdálenosti při zachování nízké energetické náročnosti. Následně byly uvedeny výhody použití systémů pro monitoring stavebních konstrukcí pro údržbu staveb a možnosti kombinace technologií Sigfox a LoRa s těmito systémy.

V praktické části práce bylo prezentováno naprogramování řídicí jednotky pro komunikaci se senzory systému Moisture Guard pro monitoring vlhkosti v dřevostavbách. Kromě získávání dat ze senzorů program obstarává také jejich přenos do databáze přes rozhraní Ethernet a po síti LoRa. V rámci této práce byla vytvořena webová stránka, která zobrazuje aktuální údaje ze senzorů a umožňuje změnit nastavení programu běžícího v řídicí jednotce.

Výsledkem práce je funkční program pro programovatelný automat (PLC) SDS-BIG, který přenáší informace mezi senzory a databázovým systémem. Důraz byl při vytváření programu kladen na univerzálnost kódu, díky čemuž mohou být jeho části v budoucnu využity v dalších aplikacích, které se na UCEEBu připravují (např. monitorování kvality vnitřního ovzduší budov, sledování ozáření pro vyhodnocování účinnosti fotovoltaických elektráren a další).

Použitá literatura a zdroje

- [1] **Ashton K.** That 'Internet of Things' Thing. RFID Journal, 2009. [online] <http://www.rfidjournal.com/articles/view?4986>
- [2] **Centenaro M., Vangelista L., Zanella A., Zorzi M.** Long-Range Communications in Unlicensed Bands: the Risings Stars in the IoT and Smart City Scenarios. IEEE Wireless Communications, 2016. [online] <https://arxiv.org/pdf/1510.00620.pdf>
- [3] **Sigfox.** Webové stránky společnosti Sigfox. [online] <https://brand.sigfox.com/presentation>
- [4] **Google.** Google Trends. [online] <https://trends.google.com/trends/explore?date=all&q=internet%20of%20things>
- [5] **LinkLabs.** A Comprehensive Look at Low Power, Wide Area Networks. [online] 2016. <http://cdn2.hubspot.net/hubfs/427771/LPWAN-Brochure-Interactive.pdf>
- [6] **Gallis R.** World's First Ultra Low Cost Modules for Sigfox IoT Network. Thinxtra blog, 2016. [online] <http://www.thinxtra.com/2016/11/sigfox-ultra-low-cost-modules/>
- [7] **Sigfox.** Webové stránky společnosti Sigfox. [online] <http://www.sigfox.com/en/coverage>
- [8] **Schatz G.** Sigfox Vs. LoRa: A comparison Between Technologies & Business Models. LinkLabs blog, 2016. [online] <https://www.link-labs.com/blog/sigfox-vs-lora>
- [9] **YouTube video.** Sigfox Network Architecture. YouTube, 2016. [online] <https://youtu.be/7zc6bH-9qZk>
- [10] **Sigfox.** Sigfox Technology Overview. [online] <http://sigfox.com/en/sigfox-iot-technology-overview>
- [11] **LoRa Alliance.** LoRa Alliance Technology [online] <https://www.lora-alliance.org/What-Is-LoRa/Technology>
- [12] **České Radiokomunikace.** Technické aspekty technologie LoRa. [online] <http://pripoj.me/hub/technicke-aspekty-technologie-lora/>

- [13] **Depuydt J.** LoRaWAN Simply Explained. Jensd's I/O Buffer, 2016. [online] <http://jensd.be/755/network/lorawan-simply-explained>
- [14] **LoRa Alliance.** Webové stránky LoRa Alliance. [online] <https://www.lora-alliance.org/>
- [15] **SimpleCell.** Webové stránky společnosti SimpleCell. [online] <https://www.simplecell.eu/>
- [16] **Solidus Tech.** LPWAN v ČR a první reálné zkušenosti v pilotním projektu. Webové stránky společnosti Solidus Tech, 2017. [online] <http://www.solidustech.cz/clanek/lpwan-v-cr-a-prvni-realne-zkusenosti-v-pilotnim-projektu/>
- [17] **České Radiokomunikace.** Webové stránky společnosti České Radiokomunikace. [online] <https://www.cra.cz/sluzby-iot>
- [18] **Sedlák J.** České Radiokomunikace a RWE testují síť pro internet věcí. Lupa.cz, 2015. [online] <https://www.lupa.cz/clanky/ceske-radiokomunikace-a-rwe-testuji-sit-pro-internet-veci/>
- [19] **Balageas D., Fritzen C., Güemes A.** Structural Health Monitoring. ISTE, 2016. ISBN 13: 978-1-905209-01-9.
- [20] **Spencer B.F.** Prezentace: Structural Health Monitoring of Civil Infrastructure: from Research to Engineering Practice. University of Illinois. [online] <http://www-smartinfrastucture.eng.cam.ac.uk/files/20130118CSICSeminarProfBillSpencerpresentation.pdf>
- [21] **Arun Sundaram B., Ravisankar K., Senthil R., Parivallal S.** Wireless Sensors for Structural Health Monitoring and Damage Detection Techniques. Current Science, vol. 104, no. 11, 2013. [online] <http://www.currentscience.ac.in/Volumes/104/11/1496.pdf>
- [22] **Moravskoslezský dřevařský klastr.** Atmosférická a biologická koroze dřeva. [online] <http://drevostavebniportal-popularizace.msdk.cz/files/atmosfericka-a-biologicka-koroze-dreva.pdf>
- [23] **Moron C., Garcia-Fuentevilla L., Garcia A., Moron A.** Measurement of Moisture in Wood for Application in the Restoration of Old Buildings. MDPI, 2016. [online] <http://www.mdpi.com/1424-8220/16/5/697/pdf>

- [24] **Vodička A., Mlejnek P., Maška M., Včelák J.** Moisture Guard – Systém pro kontinuální monitoring vlhkosti. Sympozium Společnosti pro techniku prostředí, 2015. [online]
http://www.uceeb.cz/system/files/souboryredakce/vodicka_moisture_guard.pdf
- [25] **Bureš J.** Relativní permitivita. Converter.cz. [online]
<http://www.converter.cz/tabulky/relativni-permitivita.htm>
- [26] **Vodička A., Maška M., Včelák J., Mlejnek P.** Sensor System for Continuous Moisture Monitoring in Wooden Buildings and Structures. World Conference on Timber Engineering, Vienna University of Technology, 2016. ISBN 978-3-903039-00-1.
- [27] **Moisture Guard.** Webové stránky systému Moisture Guard. [online]
<http://www.moistureguard.cz/>
- [28] **Vodička A.** Datasheet MHT02485: Kombinovaný senzor vlhkosti stavebních materiálů, vzdušné vlhkosti a teploty. Verze: 2.0.1. Univerzitní centrum energeticky efektivních budov ČVUT, Buštěhrad, 2017.
- [29] **Vodička A.** Datasheet MHT02485: Kombinovaný senzor vlhkosti stavebních materiálů, vzdušné vlhkosti a teploty. Verze: 2.1. Univerzitní centrum energeticky efektivních budov ČVUT, Buštěhrad, 2017.
- [30] **Vodička A.** Datasheet HT01485: Kombinovaný senzor vzdušné vlhkosti a teploty. Verze 2.3.1. Univerzitní centrum energeticky efektivních budov ČVUT, Buštěhrad, 2017.
- [31] **Staněk J.** RS 485 & 422. Vyroj.hw.cz, 1998. [online]
<http://vyvoj.hw.cz/teorie-a-praxe/dokumentace/rs-485-422.html>
- [32] **Simply Modbus.** Frequently Asked Questions. [online]
<http://www.simplymodbus.ca/FAQ.htm#Modbus>
- [33] **Online technology.** Uživatelský návod SDS-BIG. [online]
http://www.onlinetechnology.cz/data/soubory/dokumentace_2016/SDS_BIG_uziv_2015CZ_rev2.pdf
- [34] **Online technology.** SDS-BIG. Síťový dohledový systém – WiKi, 2016. [online]
<http://wiki.merenienergie.cz/index.php/SDS-BIG>
- [35] **Online technology.** FULL-C. Síťový dohledový systém – WiKi, 2017. [online]
<http://wiki.merenienergie.cz/index.php/FULL-C>

- [36] **Online technology.** FULL-C functions. Síťový dohledový systém – WiKi, 2015.
[online] http://wiki.merenienergie.cz/index.php/FULL-C_functions
- [37] **Online technology.** FULL-C sharing. Síťový dohledový systém – WiKi, 2015.
[online] http://wiki.merenienergie.cz/index.php/FULL-C_sharing
- [38] **Online technology.** FULL-C serialbus functions. Síťový dohledový systém – WiKi, 2016. [online] http://wiki.merenienergie.cz/index.php/FULL-C_serialbus_functions
- [39] **Online technology.** FULL-C dataflash functions. Síťový dohledový systém – WiKi, 2015. [online] http://wiki.merenienergie.cz/index.php/FULL-C_dataflash_functions
- [40] **Online technology.** FULL-C crypto functions. Síťový dohledový systém – WiKi, 2015. [online] http://wiki.merenienergie.cz/index.php/FULL-C_crypto_functions
- [41] **Online technology.** LoRaWAN. Síťový dohledový systém – WiKi, 2016.
[online] <http://wiki.merenienergie.cz/index.php/LoRaWAN>
- [42] **Microchip.** RN2903 LoRa™ Technology Module Command Reference User's Guide. [online] 2015.
<http://ww1.microchip.com/downloads/en/DeviceDoc/40001811A.pdf>
- [43] **Semtech.** LoRa FAQ. [online] <http://www.semtech.com/wireless-rf/lora/LoRa-FAQs.pdf>
- [44] **Semtech.** SX1272/3/6/7/8: LoRa Modem Designer's Guide. [online] 2013.
https://www.semtech.com/images/datasheet/LoraDesignGuide_STD.pdf
- [45] **Semtech.** LoRa Modem Calculator Tool. Instalační soubor ke stažení na:
<http://sx1272-lora-calculator.software.informer.com/download/>
- [46] **ITnetwork.cz.** 1. díl – Úvod do AJAXu. [online]
<https://www.itnetwork.cz/javascript/ajax/uvod-do-ajaxu>
- [47] **ITnetwork.cz.** 2. díl – Metody GET a POST v AJAXu. [online]
<https://www.itnetwork.cz/javascript/ajax/metody-get-a-post-v-ajaxu>

Seznam použitých zkratek

ABP	Activation by Personalisation
ACK	Acknowledgement
ADR	Adaptive Data Rate
AES	Advanced Encryption Standard
AJAX	Asynchronous JavaScript and XML
ANSI-C	American National Standards Institute – C Programming Language
CSS	Cascading Style Sheets
CRC	Cyclic Redundancy Check
CTR	Counter Mode
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DOM	Document Object Model
GSM	Global System for Mobile Communications
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
ISO/OSI	International Standards Organization / Open Systems Interconnection
IV	Inicializační vektor
LPWAN	Low-Power Wide Area Network
LTE	Long-Term Evolution
OS	Operační systém
PLC	Programmable Logic Controller
RTU	Remote Terminal Unit
SDS	Síťový dohledový systém
SF	Spreading Factor
UART	Universal Asynchronous Receiver/Transmitter
UCEEB	Univerzitní centrum energeticky efektivních budov ČVUT

UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
XML	Extensible Markup Language

Seznam obrázků a grafů

Obr. 1: Relativní četnost vyhledávání sousloví „internet věcí“ ve vyhledávači Google od roku 2004 do současnosti [4].....	9
Obr. 2: Komunikační prvky v síti Sigfox [9].	13
Obr. 3: Schéma přenosu zpráv v síti LoRa [13].....	15
Obr. 4: Vliv spreading factoru na přenosovou rychlost, komunikační dosah a energii potřebnou k odeslání zprávy [13].....	17
Obr. 5: Schéma přenosu zpráv v síti LoRa provozované Českými Radiokomunikacemi [17].....	19
Obr. 6: Blokové schéma kapacitního senzoru vlhkosti [23].	23
Obr. 7: Závislost kapacity borového dřeva na absolutní vlhkosti dřeva [23].	24
Obr. 8: Závislost elektrického odporu dřeva na absolutní vlhkosti dřeva [26].	25
Obr. 9: Schéma přenosu dat ze senzorů do databáze UCEEB.....	26
Obr. 10: Logo Moisture Guard [27].	27
Obr. 11: Konstrukční provedení senzoru MHT02485, hardware verze 2.2 [28].	27
Obr. 12: Konstrukční provedení senzoru MHT02485, hardware verze 2.3 [29].	27
Obr. 13: Připevnění senzoru na konstrukci pomocí vrutů [29].	28
Obr. 14: Doplnkový senzor vzdušné vlhkosti a teploty [30].	29
Obr. 15: Schéma připojení senzoru MHT02485 na sběrnici RS485 a připojení napájení [28].	29
Obr. 16: Struktura Modbus RTU zprávy [28].	31
Obr. 17: Fotografie modulu SDS-BIG bez plastové DIN krabičky, ve které je modul umístěn [34].	33
Obr. 18: Základní prvky modulu SDS-BIG a označení vstupů a výstupů [34].	33
Obr. 19: Výpis z konzole. Řídící jednotka přijala ze serveru zprávu „00300“.	38
Obr. 20: Nezašifrovaná část datagramu.	39
Obr. 21: Struktura zašifrované části datagramu.	39
Obr. 22: Pole pro zapsání klíče ve webovém konfigurátoru SDS-BIG.....	40
Obr. 23: Struktura LoRa paketu. Každý paket musí obsahovat preamble a payload, ostatní části jsou volitelné [44].	45

Obr. 24: Aplikace pro výpočet vysílací doby LoRa zprávy a dalších komunikačních parametrů [45].	46
Obr. 25: Zobrazení přijatých paketů v aplikaci Lorient.	47
Obr. 26: Uživatelská webová stránka pro zobrazení aktuálních naměřených hodnot a změnu nastavení programu.	48
Obr. 27: Zobrazení stavu programu a počtu připojených senzorů.	48
Obr. 28: Nastavení parametrů enumerace.	49
Obr. 29: Blok pro uložení nebo vymazání nastavení z Data Flash.	49
Obr. 30: Nastavení odesílání UDP paketů na server.	49
Obr. 31: Nastavení odesílání zpráv po síti LoRa.	50
Obr. 32: Zobrazení aktuálně naměřených údajů.	50

Seznam příloh

Příloha 1 – CD, které obsahuje:

- diplomovou práci ve formátu .pdf
- zdrojový kód FULL-C programu pro řídicí jednotku SDS-BIG
- zdrojový kód webové stránky