



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

BEZDRÁTOVÝ SBĚR DAT Z BOSCH XDK V LABVIEW

WIRELESS DATA ACQUISITION FROM BOSCH XDK IN LABVIEW

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Martin Hlaváč

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Stanislav Pikula, Ph.D.

BRNO 2021

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Martin Hlaváč

ID: 203561

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Bezdrátový sběr dat z Bosch XDK v LabVIEW

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvoření aplikace v LabVIEW pro bezdrátový záznam dat ze zařízení Bosch XDK. Aplikace bude umožňovat záznam do různých datových formátů. Bude zvoleno rozhraní Bluetooth nebo Wifi. Bude připraveno několik variant nastavení Bosch XDK pro posílání dat z různých snímačů přes bezdrátové rozhraní. Zadání bakalářské práce lze shrnout do následujících bodů:

1. Popište zařízení Bosch XDK včetně parametrů jeho sensorů, dostupných komunikačních rozhraní a prostředí XDK-Workbench pro programování a konfiguraci zařízení.
2. Proveďte návrh aplikace v LabVIEW pro záznam dat přijímaných prostřednictvím bezdrátových rozhraní.
3. Navrhněte několik scénářů pro posílání dat přes bezdrátová rozhraní a zdokumentujte potřebná nastavení Bosch XDK.
4. Realizujte navrženou aplikaci v LabVIEW, zdokumentujte ji a otestujte funkčnost pro navržené scénáře.

DOPORUČENÁ LITERATURA:

[1] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. Začínáme s LabVIEW. Ilustroval Viktorie VLACHOVÁ. Praha: BEN - technická literatura, 2008. ISBN 978-8073002459.

[2] BLUME, Peter A. The LabVIEW style book. Upper Saddle River: Prentice Hall, 2007. ISBN 978-0131458352.

Termín zadání: 8.2.2021

Termín odevzdání: 4.8.2021

Vedoucí práce: Ing. Stanislav Pikula, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

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

Abstrakt

Práce popisuje zařízení Bosch XDK 110 a možnosti bezdrátové komunikace s ním v programovacím prostředí LabVIEW. Podrobně je rozebráno rozhraní Bluetooth Low Energy, kdy je využit USB dongle z důvodu nekompatibility tohoto standardu s LabVIEW. Dále jsou navrženy scénáře pro vyčítání dat z XDK 110 a je popsána vytvořená aplikace pro vyčítání a záznam dat měřených pomocí XDK v LabVIEW. V práci je také popsán software pro XDK schopný spolupracovat s navrženou aplikací a jsou popsány balíky dat určených ke komunikaci s aplikací.

Klíčová slova

LabVIEW, Bosch XDK 110, Bluetooth Low Energy (BLE), USB dongle BLED112-V1

Abstract

The work summarizes what the XDK 110 is, what sensors it has, their properties and what communication interfaces can be used to communicate with the XDK device. Since the XDK 110 enables communication using Bluetooth Low Energy (BLE), this work presents the possibilities of solving communication using this interface, which is not yet directly supported in LabVIEW. Furthermore, scenarios for reading data from XDK 110 are proposed in the work, and then an application for reading and writing measured data from XDK in LabVIEW is developed and described. The work also describes the software for XDK able to cooperate with the proposed application and describes the data packages intended for communication with the application.

Keywords

LabVIEW, Bosch XDK 110, Bluetooth Low Energy (BLE), USB dongle BLED112-V1

Bibliografická citace

HLAVÁČ, Martin. Bezdrátový sběr dat z Bosch XDK v LabVIEW. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/136693>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Stanislav Pikula.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: *Martin Hlaváč*

VUT ID studenta: *203 561*

Typ práce: *Bakalářská práce*

Akademický rok: *2020/21*

Téma závěrečné práce: *Bezdrátový sběr dat z Bosch XDK v LabVIEW*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucího závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 25. července 2021

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Stanislavu Pikulovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: 25. července 2021

podpis autora

Obsah

| | |
|---|-----------|
| SEZNAM OBRÁZKŮ | 9 |
| ÚVOD | 10 |
| 1. BOSCH XDK 110..... | 11 |
| 1.1 VESTAVĚNÉ SNÍMAČE..... | 11 |
| 1.2 KOMUNIKAČNÍ ROZHRANÍ..... | 12 |
| 1.2.1 Rozdíly mezi komunikačními rozhraními Wi-Fi a BLE:..... | 12 |
| 1.3 CO JE TO BLE..... | 13 |
| 1.4 XDK-WORKBENCH..... | 16 |
| 1.5 PŘÍKLADY V XDK-WORKBENCH..... | 17 |
| 1.5.1 Examply v XDK Workbench..... | 18 |
| 2. KOMUNIKACE LABVIEW SE ZAŘÍZENÍM STANDARDU BLE..... | 20 |
| 2.1 LABVIEW..... | 20 |
| 2.2 KOMUNIKACE LABVIEW PŘES BLUETOOTH..... | 21 |
| 2.3 KOMUNIKACE LABVIEW PŘES BLE..... | 22 |
| 2.3.1 Řešení pomocí nástroje od komunity..... | 22 |
| 2.3.2 Řešení pomocí knihoven OS Windows | 23 |
| 3. NÁVRH APLIKACE V LABVIEW | 25 |
| 3.1 ČTENÍ Z XDK..... | 25 |
| 3.1.1 Aktivní čtení | 26 |
| 3.1.2 Blokové čtení..... | 26 |
| 3.1.3 Jednorázové čtení..... | 26 |
| 3.2 SCÉNÁŘE PRO POSÍLÁNÍ DAT | 27 |
| 3.3 POKUS ŘEŠENÍ PŘES WINDOWS KNIHOVNY..... | 27 |
| 3.4 POZNATKY K ŘEŠENÍ POMOCÍ KOMUNITNÍHO TOOLKITU | 27 |
| 3.4.1 Reset..... | 27 |
| 3.4.2 Přepnutí XDK mezi bootingem a aplikací..... | 27 |
| 3.4.3 Úprava UUID z XDK-Workbench do LabVIEW..... | 28 |
| 3.4.4 Handle tabulka..... | 28 |
| 3.4.5 Globální proměnné | 29 |
| 4. POPIS APLIKACE V LABVIEW | 30 |
| 4.1 APLIKACE..... | 30 |
| 4.1.1 Volba zjišťování připojitelných zařízení..... | 30 |
| 4.1.2 Volba měření..... | 31 |
| 4.1.3 Komunikace mezi XDK a aplikací..... | 34 |
| 4.1.4 Program pro XDK..... | 36 |
| 4.1.5 Použitelné znaky pro data ze snímačů | 36 |
| 4.2 MOŽNÉ CHYBY A JEJICH ŘEŠENÍ | 36 |
| 4.2.1 Připojení donglu a error -1073807343..... | 36 |
| 4.2.2 BLE v XDK není zapnuto a error -1073807339..... | 37 |
| 4.2.3 Špatné servisní nebo charakteristické UUID a error 0..... | 37 |
| 4.2.4 Neplatná složka pro ukládání hodnot a error 7 | 38 |

| | | |
|-----------|--|-----------|
| 4.2.5 | <i>Chyba při resetu USB donglu a error -1073807338</i> | 38 |
| 4.3 | SHRNUTÍ FUNKČNOSTI APLIKACE | 39 |
| 5. | ZÁVĚR | 40 |
| | LITERATURA | 41 |
| | SEZNAM SYMBOLŮ A ZKRATEK | 43 |
| | SEZNAM PŘÍLOH | 44 |

SEZNAM OBRÁZKŮ

| | |
|--|----|
| Obr. 1-1 Zařízení XDK 110..... | 11 |
| Obr. 1-2 Rozšiřující rozhraní XDK Gateway | 13 |
| Obr. 1-3 Datová hierarchie GATT [17] | 15 |
| Obr. 1-4 Atributy vložené do tabulky [17] | 16 |
| Obr. 1-5 Úvodní obrazovka XDK-Workbench..... | 16 |
| Obr. 1-6 Obrazovka při prvním otevření projektu | 17 |
| Obr. 1-7 Examply v XDK..... | 17 |
| Obr. 1-8 Ukázka Call Hierarchy | 18 |
| Obr. 2-1 Úvodní obrazovka LabVIEW | 20 |
| Obr. 2-2 Knihovna funkcí pro komunikaci s Bluetooth..... | 21 |
| Obr. 2-3 Složka pro BLE toolkit..... | 22 |
| Obr. 2-4 Funkce BLE toolkitu | 23 |
| Obr. 2-5 Call Library Function Node..... | 23 |
| Obr. 2-6 Obrazovka nastavení Call Library Function Node | 24 |
| Obr. 2-7 Obrazovka nastavení vstupů a výstupů | 24 |
| Obr. 3-1 Schéma návrhu programu pro LabVIEW | 25 |
| Obr. 3-2 Handle tabulka..... | 28 |
| Obr. 3-3 VI z toolkitu se čtením a zápisem pomocí handlu | 29 |
| Obr. 3-4 VI z toolkitu se čtením a zápisem pomocí UUID..... | 29 |
| Obr. 4-1 Aktuální struktura programu | 30 |
| Obr. 4-2 Vzhled čelního panelu programu při volbě ‚Zjišťování připojitelných zařízení‘ | 31 |
| Obr. 4-3 Čelní panel – Volba měření..... | 31 |
| Obr. 4-4 Příklad postupu při měření | 34 |
| Obr. 4-5 Data odesílaná programem při zapnutí měření..... | 34 |
| Obr. 4-6 Struktura dat přijímaných programem..... | 35 |
| Obr. 4-7 Data o vyžádaných přístrojích | 35 |
| Obr. 4-8 Chyba kód -1073807343 | 37 |
| Obr. 4-9 Chyba kód -1073807339 | 37 |
| Obr. 4-10 Chyba kód 0 – servisní nebo charakteristické UUID | 38 |
| Obr. 4-11 Chyba kód 7 | 38 |
| Obr. 4-12 Chyba kód -1073807338 | 39 |

ÚVOD

Tato bakalářská práce se zabývá komunikací XDK 110 a programu LabVIEW. Při práci se všemi snímači je většinou potřebné následné zpracování dat, to můžeme provádět například pomocí LabVIEW. Avšak před zpracováním je prvně potřeba data do LabVIEW dostat a jedním ze způsobů, kterým to jde, je Bluetooth.

Práce se skládá ze čtyř kapitol. V první kapitole práce je popsáno, co je XDK 110, jak ho naprogramovat a jaké má toto zařízení integrované senzory. Dále jsou uvedeny možné prostředky, umožňující komunikaci s LabVIEW. Z těchto prostředků jsou v práci více rozvinuty rozdíly mezi Wi-Fi a Bluetooth.

V další kapitole se práce zabývá LabVIEW a komunikací přes Bluetooth. Jelikož XDK obsahuje vyšší standart Bluetooth nazvaný BLE, který však LabVIEW nepodporuje, jsou popsány i možnosti, jak tento problém řešit.

V další kapitole práce je uveden návrh aplikace pomocí blokového diagramu. Následně jsou navrženy varianty vyčítání dat, které odpovídají návrhu. Tyto varianty jsou dále zapracovány do navržených scénářů. V této kapitole je také ukázán postup pokusu řešení pomocí Windows knihoven a poznámky k řešení pomocí USB donglu s BLE toolkitem.

Celá práce je ukončena kapitolou s popisem vytvořeného programu a sepsány některé chyby, které se mohou objevit během chodu programu a navrženy možnosti jejich řešení.

1. BOSCH XDK 110

Bosch XDK 110 je programovatelné zařízení umožňující měření pomocí osmi vestavěných snímačů. Zařízení má vnitřní paměť. Je možné s ním komunikovat (a tedy i např. odesílat naměřená data pomocí USB, Bluetooth nebo Wi-Fi. Mimo vestavěných senzorů obsahuje také dvě programovatelná tlačítka označená jednou a dvěma tečkami a čtyři LED diody (zelená LED dioda pro signalizaci stavu nabíjení a programovatelné LED diody: červenou, oranžovou a žlutou). Pracovní podmínky jsou: teplota od $-20\text{ }^{\circ}\text{C}$ do $60\text{ }^{\circ}\text{C}$, vlhkost vzduchu od 10 % do 90 %, napájecí napětí 5 V DC.

XDK 110 je zařízení vhodné pro testování nových zařízení a technologií pro IoT a usnadňuje začátky produkce. Zařízení je vytvořeno tak, aby umožnilo rychle se připojit k jakémukoli elektronickému zařízení a spotřebiči, a to i k již vytvořenému zařízení. [11] a [12]



Obr. 1-1 Zařízení XDK 110

1.1 Vestavěné snímače

Bosch XDK 110 obsahuje osm snímačů, jmenovitě: akcelerometr, gyroskop, magnetometr, senzor intenzity osvětlení, teploměr, senzor tlaku, senzor vlhkosti a hlukový senzor.

Tab. 1-1 Základní údaje snímačů [1], [2] a [16]

| Snímač | Rozsah | Vzorkovací frekvence |
|----------------------------|--|---|
| Akcelerometr | Programovatelný ±2 g až ±16 g | 2 000 Hz (BMA280) 12,5 Hz - 1600 Hz (BMI160) |
| Gyroskop | Programovatelný ±125 °/s až ±2000 °/s | 2 000 Hz (BMG160) 3200 HZ (BMI160) |
| Magnetometr | ±1300 μT (osa X a Y) ±2500 μT (osa Z) | 300 Hz |
| Senzor intenzity osvětlení | 0,045 luxu až 188 000 luxu (22 bitů) | Není uveden |
| Teploměr | -20 °C až 60 °C | 182 Hz |
| Tlakoměr | 300 hPa až 1100 hPa | 182 Hz |
| Měřič vlhkosti | 10 % až 90 % | 182 Hz |
| Zvukový snímač | 60 Hz – 12,5 kHz | --- |

Rozdíl mezi akcelerometrem a gyroskopem BMI160 a akcelerometrem BMA280 a gyroskopem BMG160 je v tom, že BMI je inerciální snímač.

1.2 Komunikační rozhraní

Bosch XDK 110 umožňuje komunikovat přes rozhraní USB, pomocí kterého se dá i programovat. Dále umožňuje komunikovat na dálku pomocí Wi-Fi a již zmíněného BLE. Dále obsahuje konektor pro připojení rozšiřující desky “XDK Gateway“, která rozšiřuje možnosti základní desky. XDK Gateway je ukázán na Obr. 1-2.

1.2.1 Rozdíly mezi komunikačními rozhraními Wi-Fi a BLE:

Rychlost – BLE je vhodné pro posílání menších bloků dat, které dokáže posílat rychlostí od 125 kb/s do 2 Mb/s [20], n rozdíl od Wi-Fi, která je určena k posílání velkoobjemových dat a umožňuje je posílat rychlostí 54 Mb/s [19]

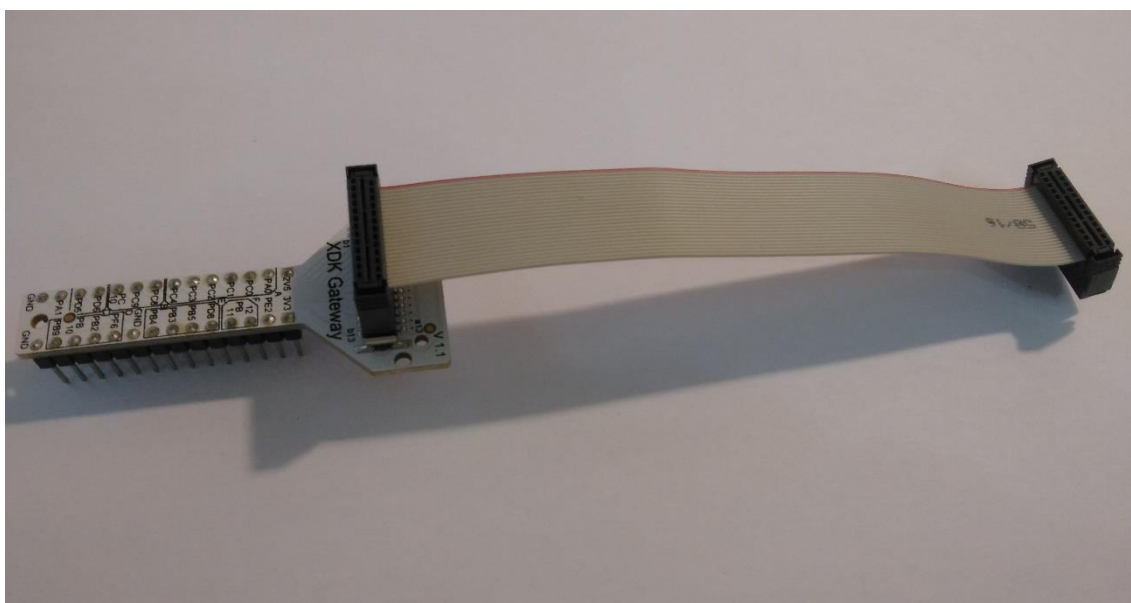
Bezpečnost – obě komunikační prostředí jsou v běžných případech dostatečně bezpečná, avšak pokud by se mělo jedna o choulostivá data je lepší využít rozhraní Wi-Fi,

kteřé má již delší dobu zaimplementované bezpečnostní protokoly jako WEP, WPA a další novější protokoly.

Vzdálenost – dosah BLE je obvykle mezi 15 m až 140 m (záleží však na mnoha faktorech) [21], dosah Wi-Fi je až 92 metrů. [16]

Kompatibilita – většina novějších zařízení dokáže komunikovat s nejnovějšími standardy jak Wi-Fi, tak Bluetooth, proto zde by problém být neměl. Toto však nemusí platit pro aplikace, se kterými chceme pracovat (například LabVIEW nepodporuje BLE, více k problematice v kapitole 2.3)

Spotřeba – ve většině případů mají zařízení BLE menší spotřebu jak zařízení Wi-Fi, a to i s relativně významným rozdílem. [16]



Obr. 1-2 Rozšiřující rozhraní XDK Gateway

1.3 Co je to BLE

Následující informace jsou čerpány ze zdrojů [8] a [9]. Bluetooth Low Energy je bezdrátová komunikační technologie s nízkou spotřebou energie. Je to poměrně nový standard, který vznikl kolem roku 2009 pro použití v zařízeních s nízkou spotřebou. Pro Bluetooth Low Energy se používá mnoho označení jako například Bluetooth LE, Bluetooth Smart, či zkratka BLE. Standard byl původně navržen společností Nokia pod názvem Wibree.

Zařízení BLE jedná buď v “centrální” nebo v “periferní” roli, kdy centrální rolí je myšlen klient, zařízení, které iniciuje příkazy a požadavky, např. počítač, a periferní rolí je myšlen server, zařízení, které přijímá příkazy a požadavky, např. XDK. Tato

komunikace je označována jako online režim a používá se pro ni protokol GATT. Tento protokol definuje, jak dvě zařízení vybavená BLE mezi sebou komunikují.

GATT (**G**eneric **A**tttribute **P**rofile) definuje způsob, jakým dvě zařízení BLE mezi sebou komunikují pomocí konceptů zvaných ‘services’ a ‘characteristics’. Používá se protokol s názvem Attribute protocol (ATT), který využívá k ukládání dat jednoduchou tabulku obsahující služby, charakteristiky a související data [14].

Následující odstavce do konce podkapitoly jsou ze zdroje [17]. Nejmenší datová jednotka definovaná pro ATT je atribut. To je adresovatelná informace, která může obsahovat různá uživatelská data. ATT dokáže pracovat pouze s atributy, a proto všechny informace musí být organizovány v této podobě. Atributy jsou vždy uloženy na serveru, takže práce s ATT nevyžaduje žádnou interní paměť. Každý atribut obsahuje informace o sobě samém a poté skutečná data v polích. K popsání jednotlivých atributů se používá takzvaný Attribute Handle, zkráceně Handle. To je hexadecimální číslo mezi 0001 až FFFF.

Mimo Handle se při komunikaci přes GATT používají ještě dvě další specifikace, a to typ a oprávnění. Typ je UUID, což je jedinečný identifikátor tak velký, aby nebyl opakovatelný, používají se 16, 32 nebo 128bitové UUID a určují druh dat, který je uložen v hodnotě atributu. Ačkoli typ je vždy UUID, tak lze použít mnoho druhů UUID, jako například servisní, charakteristické a profilové. Druhou specifikací je oprávnění, to jsou metadata, která určují, co lze s atributem dělat a s jakými bezpečnostními požadavky.

Přístupová oprávnění:

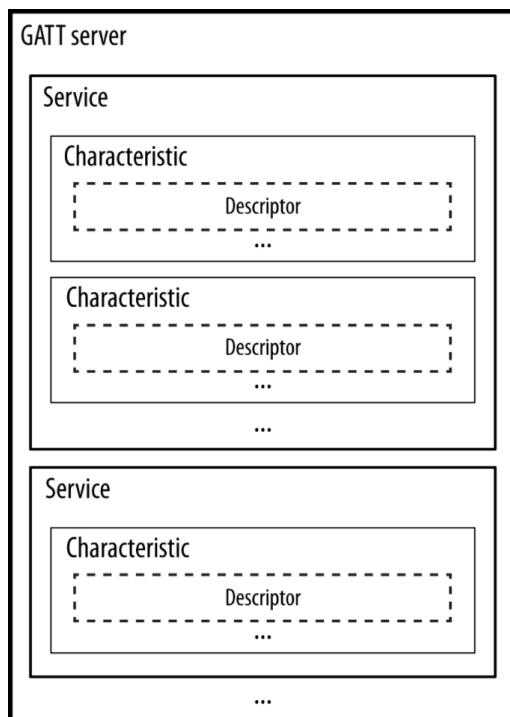
- Žádné (None) – nelze číst ani zapisovat
- Readable – lze číst
- Writable – lze zapisovat
- Readable and writable – Lze číst a zapisovat

Šifrování:

- Není vyžadováno
- Neověřené šifrování – musí být šifrováno, ale šifrovací klíče není třeba ověřovat
- Ověřené šifrování – pro připojení musí být šifrováno autorizovaným klíčem

Autorizace:

- Vyžadována autorizace
- Není vyžadována autorizace



Obr. 1-3 Datová hierarchie GATT [17]

Pokud tyto atributy vložíme do tabulky, vznikne nám tabulka podobná tabulce na Obr. 1-3 **Chyba! Nenalezen zdroj odkazů.** Pokud bychom tuto tabulku lehce upravili (,Type‘ nahradíme za dvě UUID – servisní a charakteristické a ,Permissions‘, ,Value‘ a ,Value length‘ je definované přímo v samotných datech na začátku celého pole) získáme tabulku, kterou vyčteme z XDK pomocí BLE (viz. Obr. 3-2).

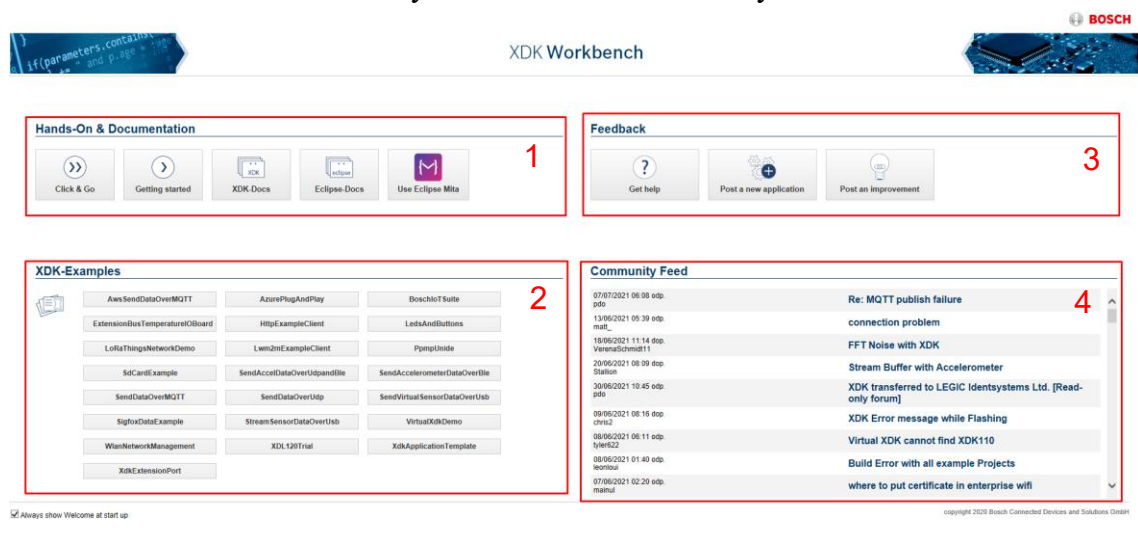
| Handle | Type | Permissions | Value | Value length |
|--------|-----------------------------|---|---------------------------|--------------|
| 0x0201 | UUID ₁ (16-bit) | Read only, no security | 0x180A | 2 |
| 0x0202 | UUID ₂ (16-bit) | Read only, no security | 0x2A29 | 2 |
| 0x0215 | UUID ₃ (16-bit) | Read/write, authorization required | "a readable UTF-8 string" | 23 |
| 0x030C | UUID ₄ (128-bit) | Write only, no security | {0xFF, 0xFF, 0x00, 0x00} | 4 |
| 0x030D | UUID ₅ (128-bit) | Read/write, authenticated encryption required | 36.43 | 8 |
| 0x031A | UUID ₁ (16-bit) | Read only, no security | 0x1801 | 2 |

Obr. 1-4 Atributy vložené do tabulky [17]

1.4 XDK-Workbench

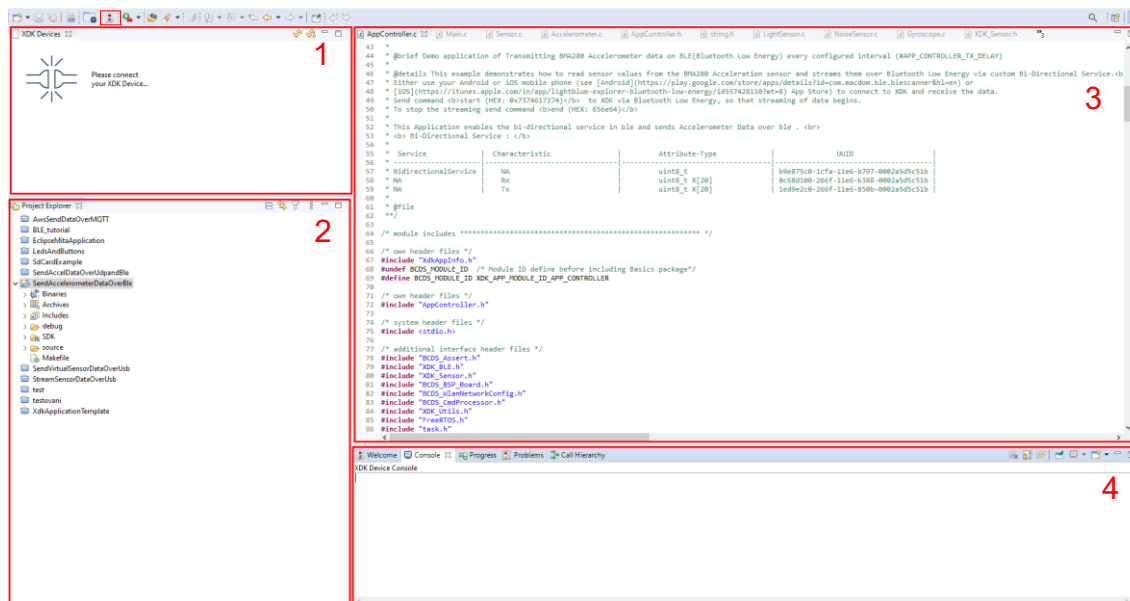
Aplikace XDK-Workbench slouží k programování zařízení Bosch XDK 110. XDK-Workbench umožňuje programovat v jazycích C, C++ a Mita. (Mita je jazyk zaměřující se na vytváření IoT a je určen zvláště pro vývojáře bez hlubších znalostí vývojového prostředí [10])

Úvodní obrazovka programu XDK-Workbench je na Obr. 1-5. Rámeček č. 1 označuje skupinu příkazů pro práci s projektem a s dokumenty. V rámečku č. 2 jsou vzorové projekty pro XDK, v rámečku č. 3 jsou ikony pro komunikaci se společností Bosch a v rámečku č. 4 lze nalézt odkazy na čerstvá témata a otázky na fóru XDK.



Obr. 1-5 Úvodní obrazovka XDK-Workbench

Na Obr. 1-6 je obrazovka po vytvoření projektu. Ikonka XDK na horní liště (označená na obrázku malým červeným rámečkem) umožní návrat na úvodní obrazovku programu. V rámečku č. 1 se nachází seznam připojených zařízení, zde je vidět, zda je zařízení aktuálně připojeno. Zařízení je zde možno přejmenovat, případně do něj nahrát napsaný kód. V rámečku č. 2 jsou vidět projekty, které jsou nebo byly používány. Dále je zde vidět, zda je projekt uzavřen nebo otevřen a u otevřených projektů je zobrazena jejich struktura. V rámečku č. 3 se objeví po otevření zdrojového kódu programovací list, do kterého lze vepsat program pro XDK. Rámeček č. 4 zobrazuje konzoli, ve které se nacházejí informace o překladu a nahrávání programu, dále pak také změřené hodnoty při komunikaci přes USB. V sekci Window → Perspective → Customize Perspective lze obrazovku po vytvoření projektu různě upravovat podle vlastních preferencí. Proto vzhled, který je na Obr. 1-6 nemusí být pro každého uživatele stejný.

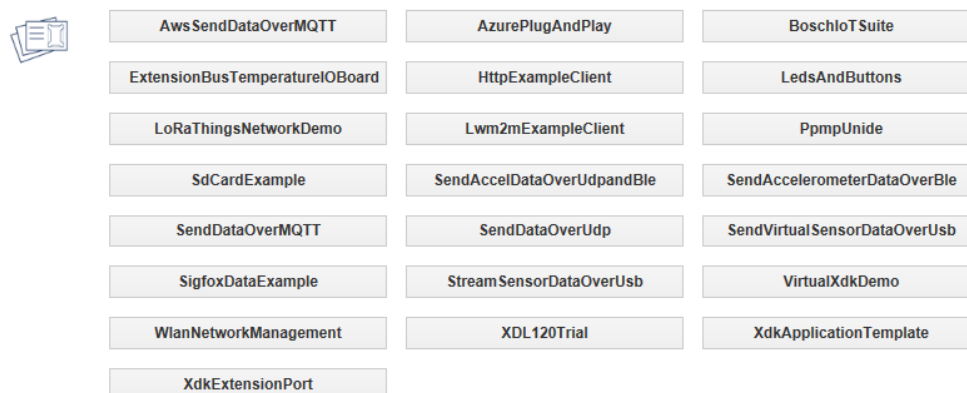


Obr. 1-6 Obrazovka při prvním otevření projektu

1.5 Příklady v XDK-Workbench

XDK-Workbench obsahuje mnoho příkladů (examples), toto anglické označení se v odborné literatuře obvykle užívá, a proto se ho budu držet i v této práci. Všechny tyto ,exampley‘ jsou vidět na Obr. 1-7. Během této bakalářské práce byl použit example ,SendAccelerometerDataOverBle‘. Mimo tento example byl použit také example ,SendVirtualSensorDataOverUSB‘ pro otestování nahrávání do XDK a práce s XDK-Workbench. Všechny exampley jsou složeny z ,main‘ souboru, souboru aplikace a mnoha dalších hlavičkových souborů. Pro samotnou práci s exampley je nejdůležitější hlavně soubor AppControler, který obsahuje popis a kód jednotlivých funkcí.

XDK-Examples

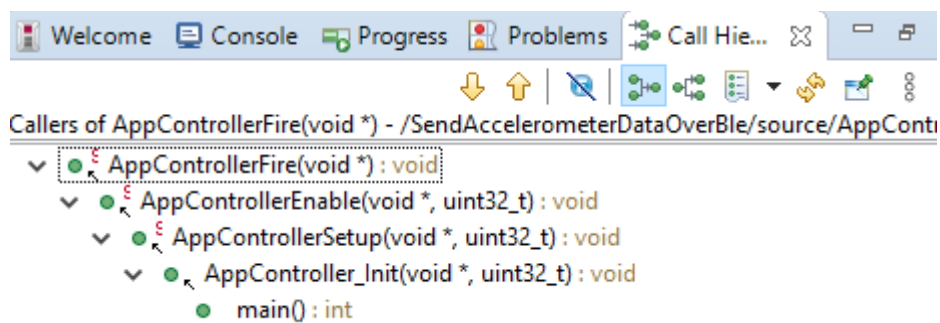


Obr. 1-7 Examply v XDK

Pro práci s examplem je důležitější si přečíst úvodní komentář v souboru AppController obsahující důležité informace o fungování exemplu, případně i parametry, které je potřeba nastavit pro začátek posílání dat a například i informace o tom, kde se nachází měřená data a kam naopak zapisovat data.

Mimo tento popis obsahuje soubor AppController definice proměnných a následně jednotlivé funkce obsahující samotný kód programu. Práce se samotnými snímači je poté v samostatných souborech jako například ,senzor.c‘ a ,accelerometr.c‘.

Tyto funkce jsou navzájem hodně provázané proto je dobré pro zjištění přesných volání použít příkaz Call Hierarchy. Ta se dá otevřít po označení funkce kliknutím na pravé tlačítko myši a poté levým tlačítkem na Open Call Hierarchy. Ukázka vzhledu je na Obr. 1-8.



Obr. 1-8 Ukázka Call Hierarchy

1.5.1 Example v XDK Workbench

AwsSendDataOverMQTT – Program pro použití protokolu MQTT pro ukládání do cloudového úložiště AWS, v případě, že je XDK připojeno k počítači, bude se na konzoli v XDK-Workbench zobrazovat stav komunikace.

AzurePlugAndPlay – Program umožňující pracovat jako zařízení plug and play pro připojení a komunikaci se službami Azure IoT. Zařízení po registraci u služby Azure začne odesílat data podle nastavení uživatele. Uživatel může rozhodovat o používaných senzorech i o intervalech odesílání dat. Zařízení bude také možno za pomoci cloudových služeb restartovat.

BoschIoTSuite – Program umožňuje komunikovat s uživatelem za pomoci zprostředkovatele, IoT, od Bosch.

ExtensionBusTemperatureIOBoard – Program umožňující připojit externí snímač teploty a dva digitální vstupy a dva digitální výstupy.

HttpExampleClient – Program ukazuje, jak použít síťový zásobník pro provedení požadavku klienta http.

LedsAndButtons – Program ukazuje práci s tlačítky a LED diodami zabudovanými v rozhraní XDK.

LoRaThingsNetworkDemo – Vzorový program pro přenos dat tlaku, teploty, vlhkosti a osvětlení pomocí přenosu LoRa.

Lwm2mExampleClient – Tento program bude fungovat jako klient protokolu LWM2M a je zde ukázáno, jak se na server LWM2M zaregistrovat.

PpmpUnide – Tento program bude řídit tok dat na server za pomoci PPMP

SdCardExample – Ukázkový program pro využívání SD karty. V případě že je přítomna karta rozsvítí LED diodu a poté do ní vytvoří a zapíše textový soubor.

SendAccelDataOverUdpandBle – Program ukazuje, jak číst hodnoty ze senzoru zrychlení BMA280 a posílat je BLE a UDP.

SendAccelerometerDataOverBle – Program ukazuje, jak číst hodnoty ze senzoru zrychlení BMA280 a posílat je BLE.

SendDataOverMQTT – Example ukazuje přenos dat z enviromentálního snímače pomocí protokolu MQTT.

SendDataOverUdp – Example je určen pro odesílání dat pomocí UDP na server. Zařízení se připojí k přístupovému bodu, získá IP adresu a spustí posílání dat v předdefinovaném intervalu na předdefinovaný server na předem definovaném portu

SendVirtualSensorDataOverUsb – Example vypisuje hodnoty virtuálních snímačů na sériový port.

SigfoxDataExample – Tento example ukazuje možnost připojení desky innocomm sigfox pomocí rozšiřující XDK sběrnice.

StreamSensorDataOverUsb – Example vytiskne jednou za čas všechna naměřená data z definovaných senzorů na sériový port.

VirtualXdkDemo – Example ukazuje hravě práci s XDK pomocí mobilního telefonu. Interakce s fyzickým XDK ovládá virtuální XDK.

WlanNetworkManagement – Example ukazuje, jak pracovat s XDK WLAN Abstraction pro práci se sítí a IP adresami.

XdkApplicationTemplate – Šablona na práci s XDK bez jakékoli funkce, vhodná pro vývoj vlastní aplikace.

XdkExtensionPort – Example obsahuje ukázkou práce s rozhraními jako jsou SPI, I2C, UART a GPIO. Uživatel má možnost pomocí marker odebrat část kódu, kterou nebude potřebovat.

Na základě zvoleného BLE jsem si pro práci vybral example SendAccelerometerDataOverBle, tento example je také vhodný pro snadné rozšíření pro další snímače.

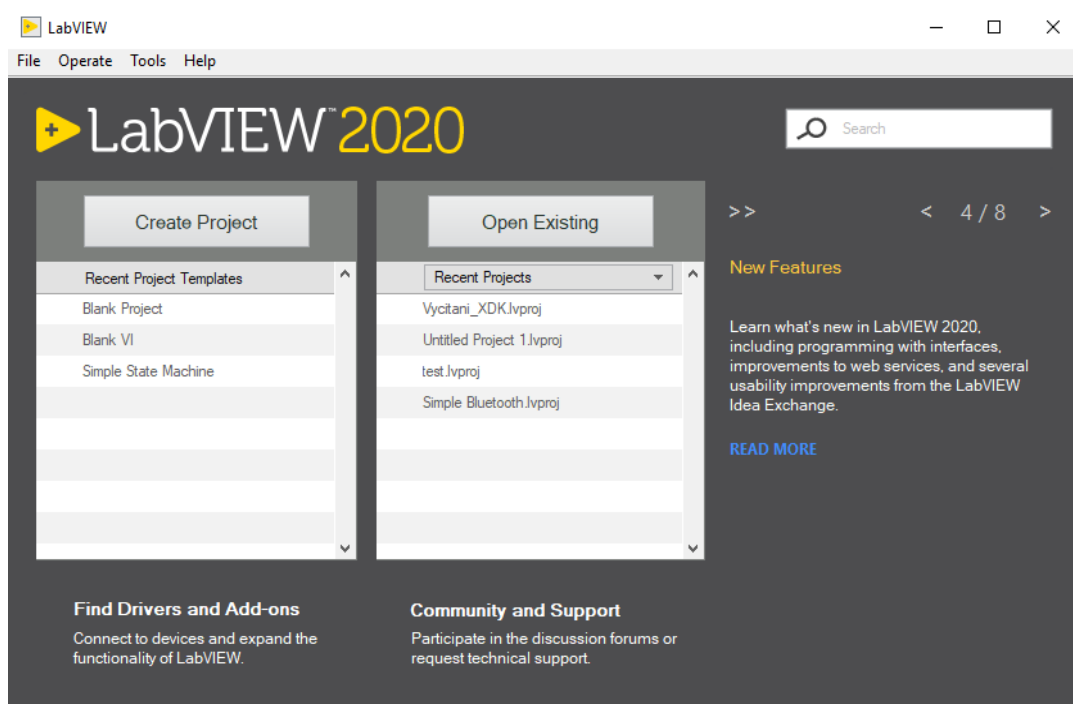
2. KOMUNIKACE LABVIEW SE ZAŘÍZENÍM STANDARDU BLE

2.1 LabVIEW

LabVIEW je programovací a vývojářské prostředí od americké firmy National Instruments. Při vytváření aplikací se v LabVIEW využívá grafického programování, které je vhodné nejen pro měření a analýzu signálů, ale lze ho také použít při programování složitých systémů jako je třeba řízení robota. Hlavním cílem virtuální instrumentace je nahradit hardware za softwarové řešení, které je ve značné míře flexibilnější a levnější [13].

Vývoj LabVIEW začal v roce 1983 po uvedení desky pro rozhraní GPIB od firmy National Instrument. Za ‘otce LabVIEW’ je považován Jeffrey Kodosky, který zahájil vývoj grafického vývojového nástroje, na kterém LabVIEW staví. Hlavním předpokladem bylo, aby technik, který umí zapsat své poznatky a požadavky do blokového diagramu, dokázal samostatně a stejným způsobem napsat i program [13].

Při spuštění aplikace se zobrazí úvodní obrazovka LabVIEW. Po kliknutí na ‚Create Project‘ můžeme vytvořit nový projekt, a to nejen prázdný, ale je zde i mnoho šablon a vzorových projektů, ze kterých můžeme vycházet. Po kliknutí na ‚Open Existing‘ můžeme otevřít již vytvořený projekt.



Obr. 2-1 Úvodní obrazovka LabVIEW

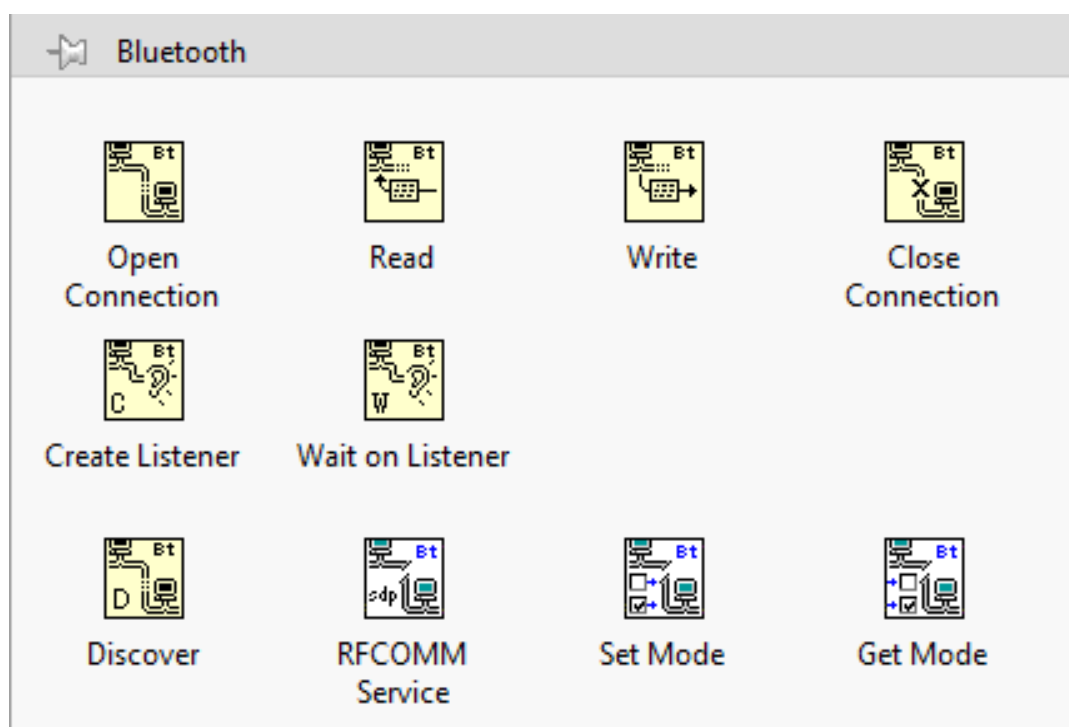
Každá aplikace VI (Virtual Instrument, prvek programu LabVIEW) se skládá ze dvou oken, a to Čelního panelu a Blokového diagramu. Čelní panel slouží jako uživatelské rozhraní k vypracované aplikaci. Blokový diagram je místo pro programátora aplikace, kde lze pomocí funkčních bloků poskládat kód programu. Mimo bloků, které již obsahuje samotné LabVIEW, lze využít také uživatelské knihovny, případně si vytvořit vlastní subVI.

2.2 Komunikace LabVIEW přes Bluetooth

LabVIEW má v sobě vloženu knihovnu pro komunikaci přes Bluetooth. Tuto knihovnu lze najít po otevření výběrového menu na blokovém diagramu, kliknutím pravým tlačítkem myši v blokovém diagramu, následně v záložce Data Communication → Protocols → Bluetooth.

Základem komunikace s LabVIEW přes Bluetooth jsou čtyři VI ukázané na Obr. 2-2 v prvním řádku, a to prvky Open Connection, Read, Write a Close Connection. Prvky Open Connection / Close Connection slouží ke spuštění / ukončení komunikace se zařízením. Prvek Read slouží k vyčítání změřených hodnot. Prvek Write slouží k nastavení potřebných vlastností zařízení.

Tato volba však neumožňuje komunikovat se zařízeními, které používají technologii BLE, jelikož v aktuální verzi (2020) není standard BLE podporován [6]. Jak lze tento problém vyřešit je uvedeno v kapitole 2.3 Komunikace LabVIEW přes BLE.



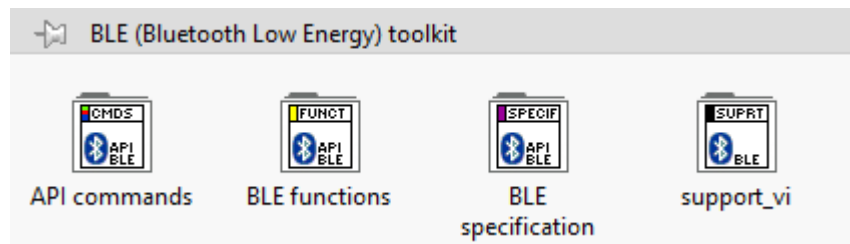
Obr. 2-2 Knihovna funkcí pro komunikaci s Bluetooth

2.3 Komunikace LabVIEW přes BLE

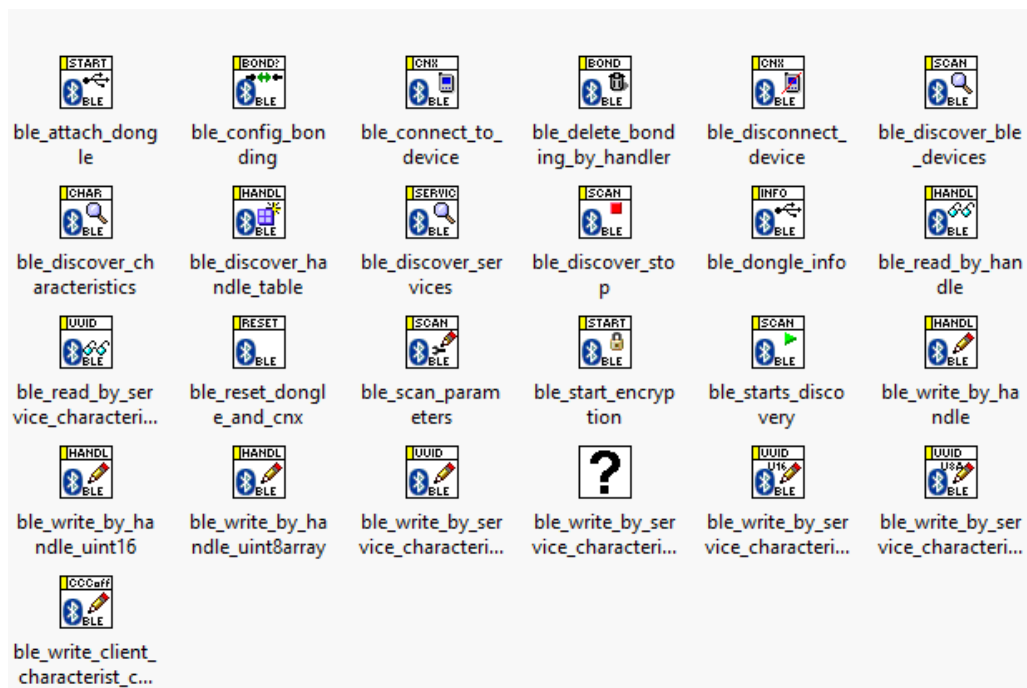
LabVIEW v sobě neobsahuje žádné knihovny, které by umožnily komunikaci přes standard BLE [6], na rozdíl od klasického Bluetooth, a proto nelze bez úpravy využít komunikaci přes BLE. Problém lze řešit dvěma způsoby, a to pomocí nástroje vytvořeného komunitou programátorů LabVIEW nebo použitím knihovných funkcí Windows. Na žádnou z těchto variant však nelze použít základní vestavěné primitivní funkce LabVIEW, založené na starších ovladačích Windows, které BLE nepodporují.

2.3.1 Řešení pomocí nástroje od komunity

Možnost komunikace pomocí komunitou vytvořené knihovny pro práci s BLE zařízeními v LabVIEW [7], přesněji toolkit vytvořila společnost Silicon Lab. Při použití tohoto programového nástroje je potřeba převodník z USB na Bluetooth (USB dongle), který změní tuto komunikaci tak, že nebude pro LabVIEW nutné komunikovat přes Bluetooth. Bude totiž komunikovat pouze s USB, kdy toolkit bude upravovat požadavky do podoby takové, aby USB dongle dokázal zprávy převést na komunikaci přes Bluetooth. Využitelné funkce jsou na Obr. 2-3 a podsložce Obr. 2-4.



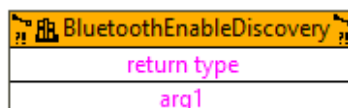
Obr. 2-3 Složka pro BLE toolkit



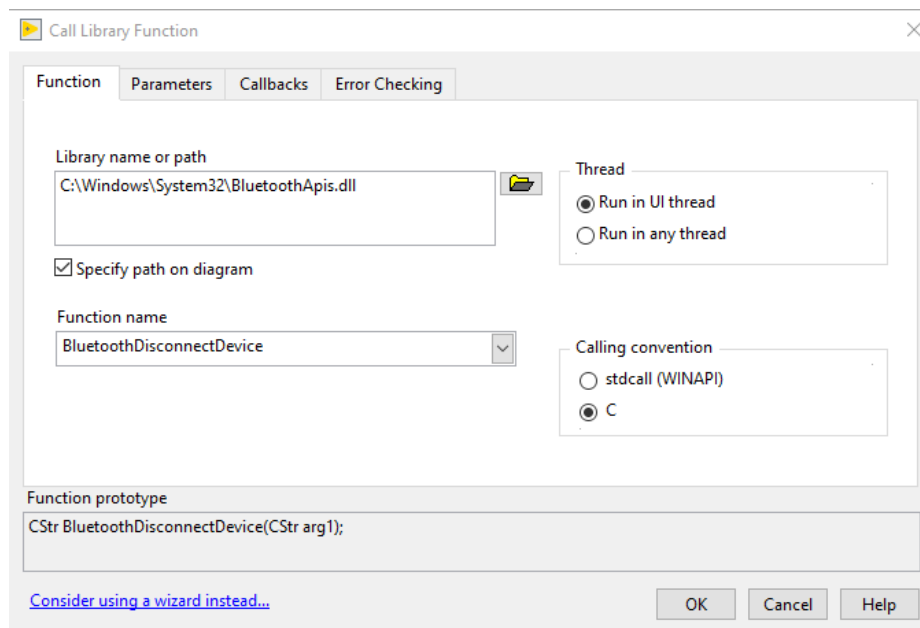
Obr. 2-4 Funkce BLE toolkitu

2.3.2 Řešení pomocí knihoven OS Windows

Další možností je využít knihovnu v rámci OS Windows [6]. Toto řešení je možné pouze pro systémy Windows 8 a vyšší, které obsahují knihovnu se standardem Bluetooth 4.0 a vyšším. K tomuto musíme využít funkci “Call Library Function Node”, která slouží pro vložení kódu z knihovny do LabVIEW. (Obr. 2-5)



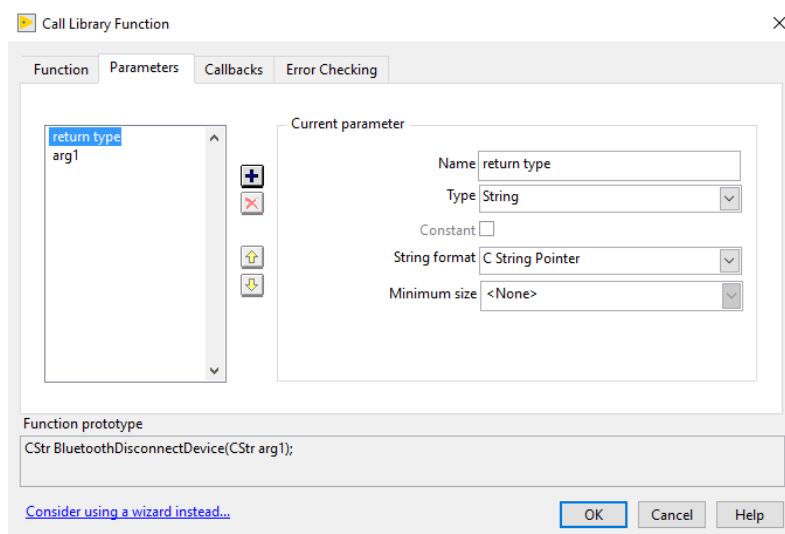
Obr. 2-5 Call Library Function Node



Obr. 2-6 Obrazovka nastavení Call Library Function Node

- Na Obr. 2-5 je ukázána funkce „Call Library Function Node“, tak jak vypadá na Blokovém diagramu. Na Obr. 2-6 je ukázána po rozkliknutí jejího nastavení. Do kolonky Library name or path se vloží cesta ke knihovně, nebo pouze název knihovny, pokud je cesta definována pomocí vstupu ‚path‘, který se zobrazí při zakliknutí checkboxu pod oknem s názvem souboru. Do této kolonky si nastavíme adresu k souboru knihovny BluetoothApis.dll, která obsahuje funkce pro komunikaci s BLE. Kolonka function name určuje, kterou funkci z knihovny chceme využít a kolonka Function prototype ukazuje funkci i se vstupy a výstupy, které jsou aktuálně nastaveny do funkce Call Library Function. Na Obr. 2-7 je vidět obrazovka vstupů a výstupů funkce Call Library Function, tyto vstupy je potřeba nastavit podle požadované funkce.

-
-



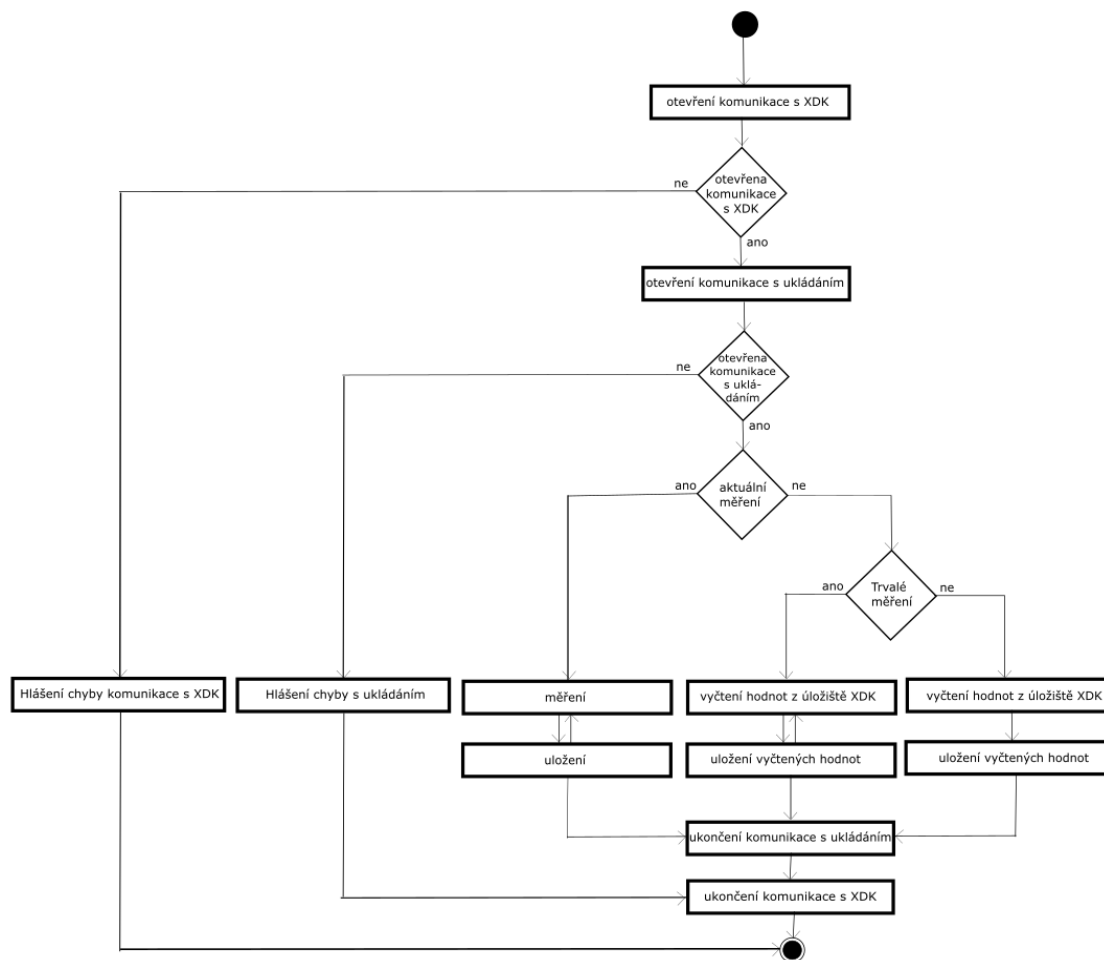
Obr. 2-7 Obrazovka nastavení vstupů a výstupů

3. NÁVRH APLIKACE V LABVIEW

Na základě dohody s vedoucím jsem se rozhodl pro práci využít komunikaci přes BLE. Po zkušenostech s vytvářením bych si BLE vybral také, vzdálenost mají BLE i Wi-Fi podobnou, obě varianty jsou relativně bezpečné. Nevýhodou může být jeho rychlost a kompatibilita, kdy kompatibilitu lze celkem snadno vyřešit pomocí Bluetooth toolkitu a USB donglu. Naopak výhodou je podstatně menší spotřeba energie.

3.1 Čtení z XDK

Návrh aplikace počítá s tím, že aplikace bude obsahovat tři varianty vyčítání dat z XDK „Aktivní čtení“, „Blokové čtení“ a „Jednorázové čtení“. Schéma je uvedeno na Obr. 3-1.



Obr. 3-1 Schéma návrhu programu pro LabVIEW

3.1.1 Aktivní čtení

Aktivní čtení vyčítá z XDK vždy jednu poslední změřenou hodnotu ze sledované veličiny. Na Obr. 3-1 se aktivní čtení nachází v levé větvi schématu. Při spuštění se nejdříve otevře komunikace s XDK, potom se nastaví komunikace se souborem, do kterého se budou data ukládat. V případě, že soubor ještě nebyl vytvořen, vytvoří se, v případě, že již soubor bude existovat, soubor se buď přepíše, nebo se záznam přidá za stávající data. Poté se zapne nekonečná měřící smyčka, která vždy vyčte údaje z XDK a následně je uloží. V případě, že si uživatel vyžádá vypnutí měření, program přejde na uzavření souboru, do kterého jsou vkládány změřené hodnoty, a následně ukončí komunikaci s XDK.

Popis jednoho vyčtení: Poslední hodnota od každé měřené veličiny.

3.1.2 Blokové čtení

Blokové čtení vyčítá z XDK vždy blok hodnot sledovaných veličin uložený v paměti XDK. Na Obr. 3-1 se blokové čtení nachází v prostřední větvi schématu. Při spuštění se nejdříve otevře komunikace s XDK, potom se nastaví komunikace se souborem, do kterého se budou data ukládat. V případě, že soubor ještě nebyl vytvořen, vytvoří se, v případě, že již soubor bude existovat, soubor se buď přepíše, nebo se záznam přidá za stávající data. Poté se zapne nekonečná vyčítací smyčka, která v každém cyklu vyčte blok dat z paměti XDK a následně je uloží. V případě, že si uživatel vyžádá vypnutí vyčítání, program přejde na uzavření souboru, do kterého jsou vkládány změřené hodnoty, a následně ukončí komunikaci s XDK.

Popis jednoho vyčtení: Blok hodnot uložených v paměti s daty všech měřených veličin.

3.1.3 Jednorázové čtení

Jednorázové čtení vyčítá z XDK právě jeden blok hodnot sledovaných veličin uložených v paměti XDK. Na Obr. 3-1 se jednorázové čtení nachází v pravé větvi schématu. Při spuštění se nejdříve otevře komunikace s XDK, potom se nastaví komunikace se souborem, do kterého se budou data ukládat. V případě, že soubor ještě nebyl vytvořen, vytvoří se, v případě, že již soubor bude existovat, soubor se buď přepíše, nebo se záznam přidá za stávající data. Poté se vyčte blok dat z paměti XDK a uloží se. Následně program uzavře soubor a ukončí komunikaci s XDK.

Popis jednoho vyčtení: Blok hodnot uložených v paměti s daty všech měřených veličin.

3.2 Scénáře pro posílání dat

Pro aplikaci navrhuji následujících devět scénářů pro posílání dat mezi XDK a LabVIEW, které vycházejí z výše zmíněných variant vyčítání dat z XDK:

- 1) Aktivní čtení, aktivované sensory teploty, tlaku a vlhkosti
- 2) Blokované čtení, aktivované sensory teploty, tlaku a vlhkosti
- 3) Jednorázové čtení, aktivované sensory teploty, tlaku a vlhkosti
- 4) Aktivní čtení, aktivovaný gyroskop a akcelerometr
- 5) Blokované čtení, aktivovaný gyroskop a akcelerometr
- 6) Jednorázové čtení, aktivovaný gyroskop a akcelerometr
- 7) Aktivní čtení, aktivovaný senzor intenzity osvětlení
- 8) Blokované čtení, aktivovaný senzor intenzity osvětlení
- 9) Jednorázové čtení, aktivovaný senzor intenzity osvětlení

3.3 Pokus řešení přes Windows knihovny

První pokus vytvořit aplikaci byl pomocí Windows knihoven. Stručný popis tohoto řešení je v kapitole 2.3.2 Řešení pomocí knihoven OS Windows. Toto řešení se však nevedlo k výsledku, protože na internetu jsem nenašel ideální popis všech funkcí. V literatuře [15] je mnoho popisů funkcí, ale chybí popisy některých funkcí jako je například popis funkce ‚Disconnect‘. Pro zprovoznění aplikace je však nutná minimálně znalost vstupů a výstupů. Proto jsem se rozhodnul tímto způsobem bakalářskou práci neřešit a zvolil raději variantu pomocí komunitního toolkitu.

3.4 Poznatky k řešení pomocí komunitního toolkitu

Pro řešení problému jsem proto nakonec zvolil variantu toolkitu umožňující komunikaci s BLE zařízeními pomocí USB donglu.

3.4.1 Reset

Protože při vytváření programu se objevovala chyba, která zablokovala komunikaci USB a následně bylo nutné restartovat celý počítač, byl na začátek programu přidán příkaz ‚Reset USB‘. Tento reset odstraní případné zablokování USB konektoru z důvodu jeho předchozího obsazení. Toto následně zajistí bezchybný chod programu.

3.4.2 Přepnutí XDK mezi bootingem a aplikací

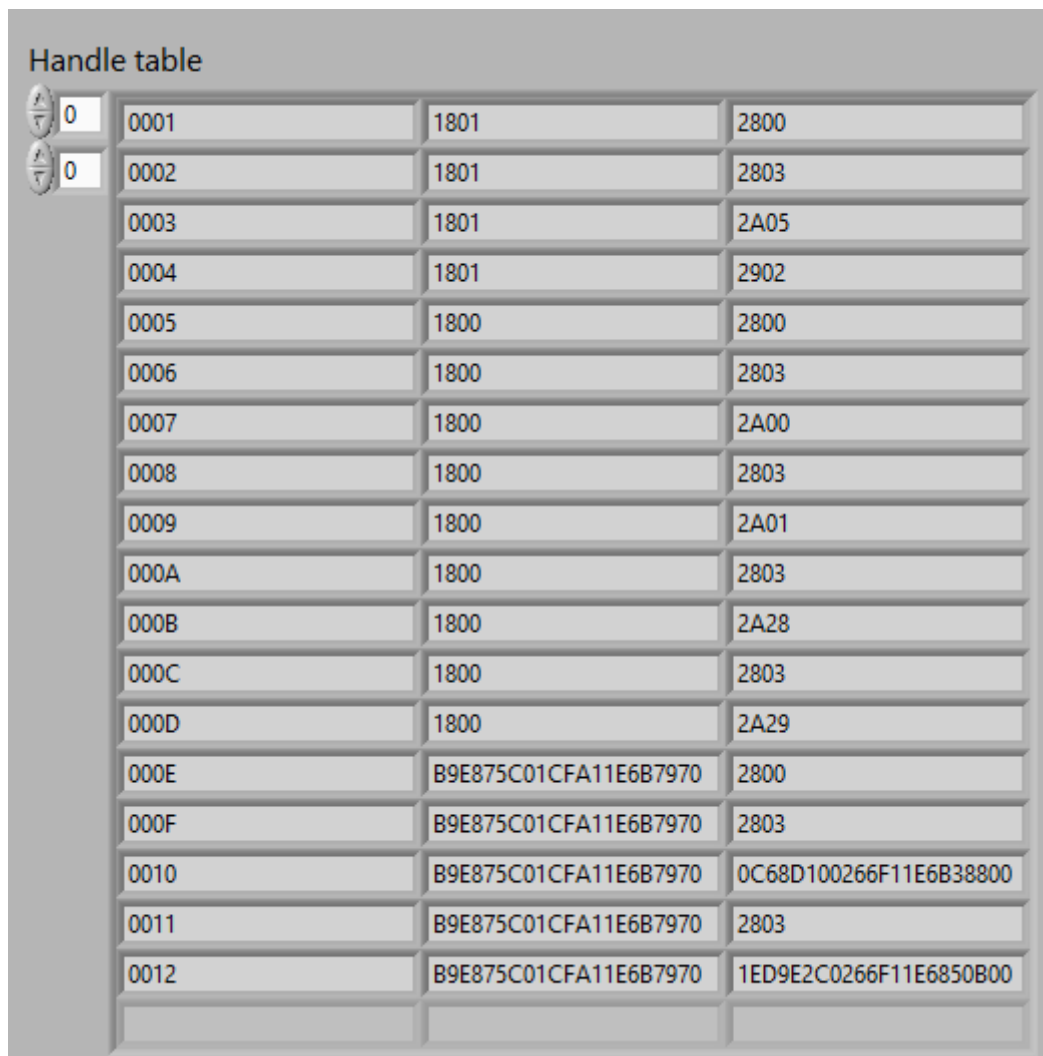
V případě, že komunikujete s XDK (případně před spuštěním komunikace), např pomocí LabVIEW, není dobré spouštět aplikaci XDK-Workbench. Při jejím spuštění se XDK přepne do režimu bootování, čímž se vypne aktuálně běžící program. To způsobí, že se

vypne komunikace a LabVIEW ohlásí chybu, že komunikační zařízení BLE není v XDK zapnuto. Řešení je uvedeno v kapitole 4.2.2.

3.4.3 Úprava UUID z XDK-Workbench do LabVIEW

Pro komunikaci přes Bluetooth je nutné znát UUID, to se dá pro example, který byl použit, zjistit v komentářích v programu, tyto UUID však nelze přímo použít v LabVIEW, je potřeba je prvně upravit. UUID z XDK-Workbench obsahuje pomlčky, ty je nutno odebrat a také obsahuje malá písmena, ta je nutno přepsat na velká. Například z exemplu přečteme, že Service UUID je b9e875c0-1cfa-11e6-b797-0002a5 ale do LabVIEW musíme zapsat B9E875C01CFA11E6B7970002A5. Tyto UUID lze najít také v Handle tabulce.

3.4.4 Handle tabulka

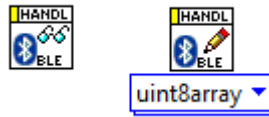


| Handle | Service UUID | Device UUID |
|--------|--------------|------------------------|
| 0 | 0001 | 1801 |
| 0 | 0002 | 1801 |
| | 0003 | 1801 |
| | 0004 | 1801 |
| | 0005 | 1800 |
| | 0006 | 1800 |
| | 0007 | 1800 |
| | 0008 | 1800 |
| | 0009 | 1800 |
| | 000A | 1800 |
| | 000B | 1800 |
| | 000C | 1800 |
| | 000D | 1800 |
| | 000E | B9E875C01CFA11E6B7970 |
| | 000F | B9E875C01CFA11E6B7970 |
| | 0010 | B9E875C01CFA11E6B7970 |
| | 0011 | B9E875C01CFA11E6B7970 |
| | 0012 | B9E875C01CFA11E6B7970 |
| | | 2800 |
| | | 2803 |
| | | 2A05 |
| | | 2902 |
| | | 2A00 |
| | | 2A01 |
| | | 2A28 |
| | | 2A29 |
| | | 0C68D100266F11E6B38800 |
| | | 1ED9E2C0266F11E6850B00 |

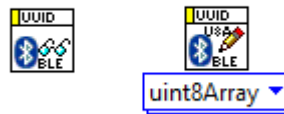
Obr. 3-2 Handle tabulka

Handle tabulka je základem pro komunikaci pomocí Bluetooth. Vzhled Handle tabulky je ukázán na Obr. 3-2. Každý řádek obsahuje informace poskytované XDK,

do jednotlivých řádků můžeme zapisovat a vyčítat pomocí toolkitu dvěma způsoby, pomocí handlu, nebo za pomoci dvou UUID.



Obr. 3-3 VI z toolkitu se čtením a zápisem pomocí handlu



Obr. 3-4 VI z toolkitu se čtením a zápisem pomocí UUID

Pro vyčítání pomocí handlu jsou na Obr. 3-3 ukázány funkce, které umožňují vyčítat, nebo zapisovat informace pomocí handlu. Tento handle najdeme v Handle tabulce v prvním sloupečku. Druhým způsobem čtení je pomocí dvou UUID, kterými jsou Charakteristické UUID a Servisní UUID. Servisní UUID se dá v Handle tabulce najít ve druhém sloupečku. Charakteristické UUID se v Handle tabulce nachází v třetím sloupečku.

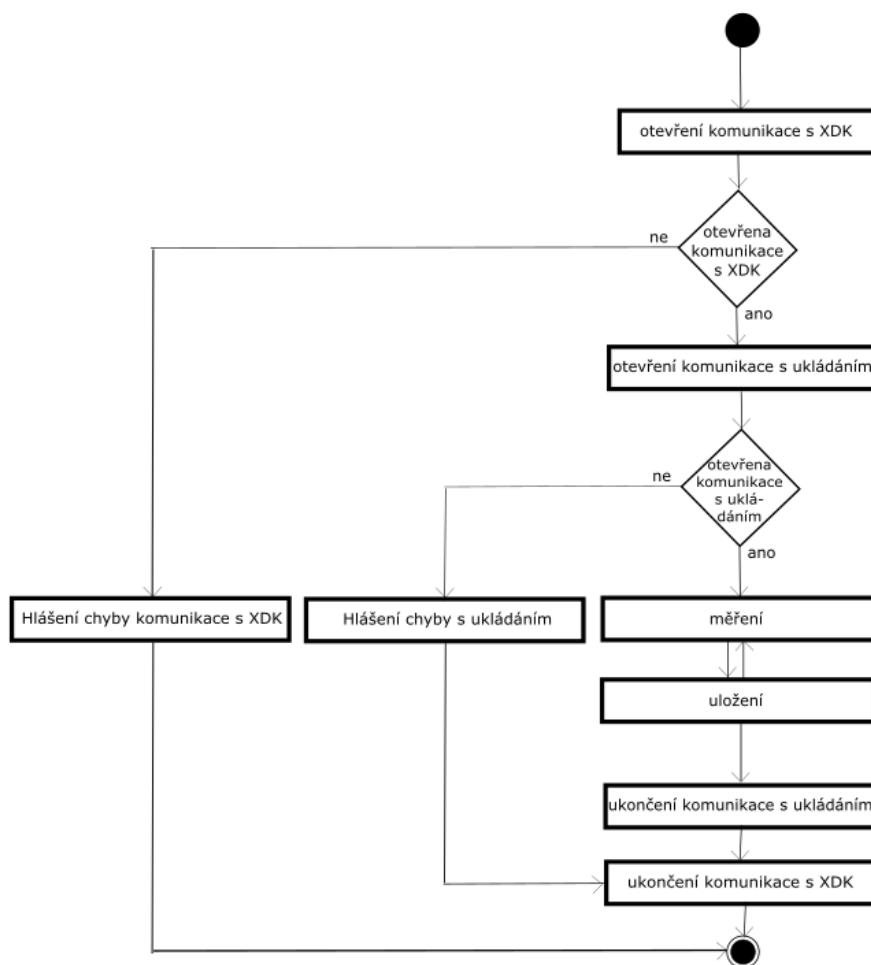
3.4.5 Globální proměnné

V kódu byly využity globální proměnné pro zpřehlednění kódu, avšak mělo by být pomocí ‚case‘ zajištěno přesné pořadí toho, kdy se co vyčte. Takže by neměla hrozit nějaká chyba nebo pád programu spojený s vyčtením či zapsáním do proměnné ve špatném okamžiku.

4. POPIS APLIKACE V LABVIEW

4.1 Aplikace

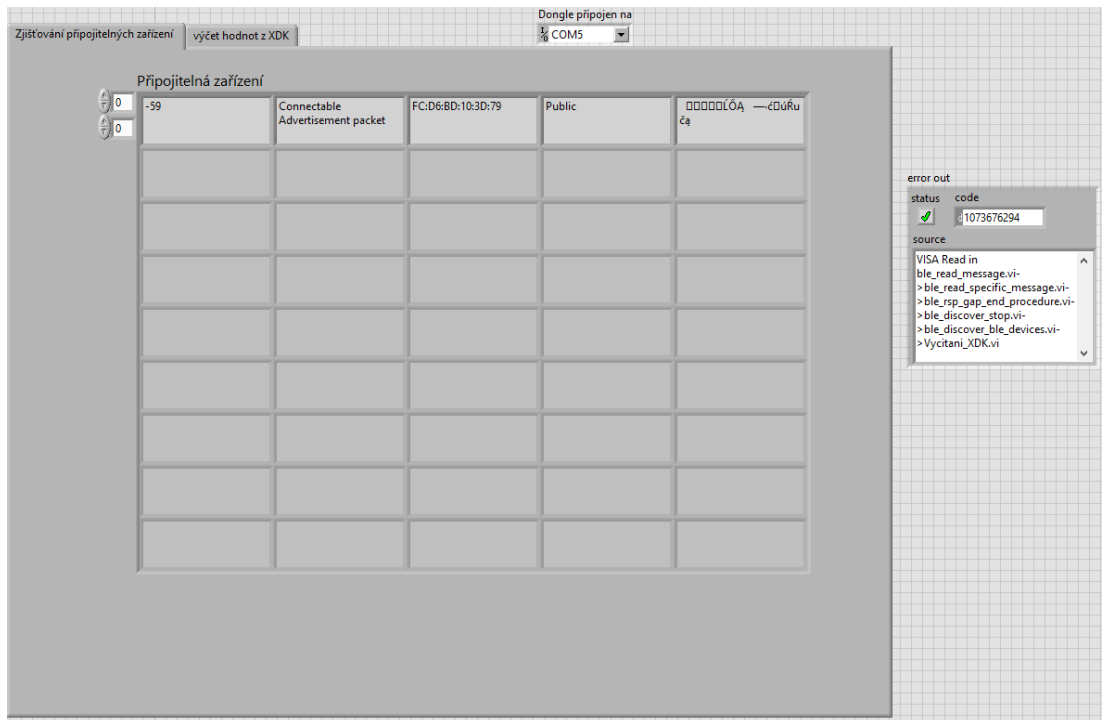
Původním plánem bylo vytvořit devět scénářů pro vyčítání, nakonec jsem se však rozhodl z výše uvedených scénářů realizovat pouze aktivní čtení. Tuto volbu jsem však v aplikaci upravil tak, aby bylo možné si ze seznamu vybrat požadovaný snímač. Aplikace proto obsahuje jediný scénář, který je však přizpůsobitelný práci s jakýmkoli snímačem.



Obr. 4-1 Aktuální struktura programu

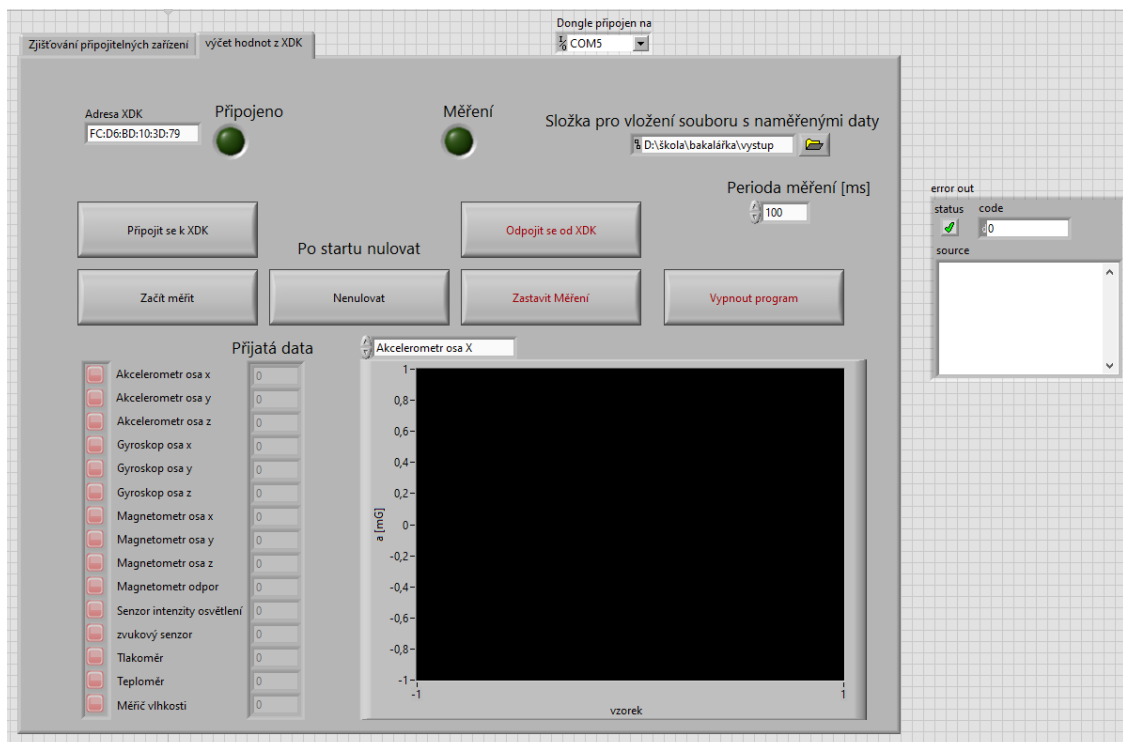
4.1.1 Volba zjišťování připojitelných zařízení

Aplikace umožňuje dvě volby spuštění. Jednou z nich je ‚Zjišťování připojitelných zařízení‘. V této volbě je možno zjistit, zda požadované zařízení je aktivní a zda se na něj dá připojit, či nikoli. Také se zde dá nalézt adresa, kterou má zařízení XDK a kterou je potřeba nastavit při měření. Na Obr. 4-2 je ukázán vzhled čelního panelu v programu LabVIEW při volbě zjišťování připojitelných zařízení.



Obr. 4-2 Vzhled čelního panelu programu při volbě „Zjišťování připojitelných zařízení“

4.1.2 Volba měření



Obr. 4-3 Čelní panel – Volba měření

Pro samotné měření se využívá záložka ‚Výčet hodnot z XDK‘. Tato volba je uvedena na Obr. 4-3. Kde kolonka Adresa XDK je určena pro adresu XDK, která je zjištěna pomocí volby zjišťování připojitelných zařízení. Indikátory ‚Připojeno‘ a ‚Měření‘ informují, zda došlo k připojení XDK nebo byl odeslán příkaz pro začátek vysílání dat. Samotné připojení není přímo kontrolováno, například pomocí přijetí potvrzovacích dat přes BLE, ale v aplikaci je vloženo místo do stavového automatu pro možnost této kontroly. Dále je potřeba do aplikace zadat adresu pro ukládací soubor. Zde stačí udat jen složku a aplikace si sama vytvoří soubor s názvem ‚mereni‘ + připojí do názvu datum a čas měření. Dále je na panelu možno zadat periodu měření. Jedná se o periodu jednoho balíku dat z měření. Čím více balíků si v aplikaci zvolím, tím bude perioda delší, maximálně však při posílání všech dat může dosáhnout pětinasobku periody zadané. Ideální proto pro měření je, aby byl nastaven jen jeden balík, kdy perioda měření všech nastavených dat může dosáhnout minimální hodnoty a to zhruba 30 ms. Tato perioda měření však nemusí být přesná, aplikace si volbu zaokrouhlí na sudé číslo pro jednodušší práci s automatem a toto zaokrouhlené číslo se následně pošle do XDK.

Důvodem pro zaokrouhlení periody měření na sudé číslo je, že samotné měření probíhá ve dvou stavech měřícího automatu, které musejí trvat stejnou dobu a doba čekání může být pouze v celých milisekundách (int32), je potřeba, aby bylo zadané číslo převedeno na sudé.

Pro zapínání a vypínání obsahuje aplikace pět tlačítek. Tato tlačítka je potřeba spouštět postupně podle Obr. 4-4. V případě, že je například zmáčknuto tlačítko ‚Začátek měření‘, ale zatím není aplikace připojena k XDK, tak se měření nezapne, dokud nebude stisknuto tlačítko ‚Připojit se k XDK‘. Pod tlačítkem ‚Začít měřit‘ se nachází volba snímačů, při zeleně rozsvíceném tlačítku se bude údaj ukládat, a naopak při červeném nebude. Toto však neplatí pro vyčítání, aplikace vždy bude vyčítat všechny hodnoty, které budou odeslány z XDK. Tato volba snímačů je spolu s volbou periody odesílána do XDK. Napravo od výběru snímačů jsou poté zobrazeny zpracované hodnoty snímačů. Vzhledem k možnosti odesílání jen tří veličin v jeden čas, se všechny nepřijaté hodnoty ukládají v nezměněné hodnotě a jsou přepsány jen ty, co jsou nově přijaty. Pro případ, že by bylo třeba při začátku měření smazat uložené hodnoty, je napravo od tlačítka ‚Začít měřit‘ tlačítko pro volbu mazání uložených hodnot při stlačení tlačítka ‚Začít měřit‘. Poslední část na panelu měření je graf posledních naměřených hodnot s přepínačem veličin.

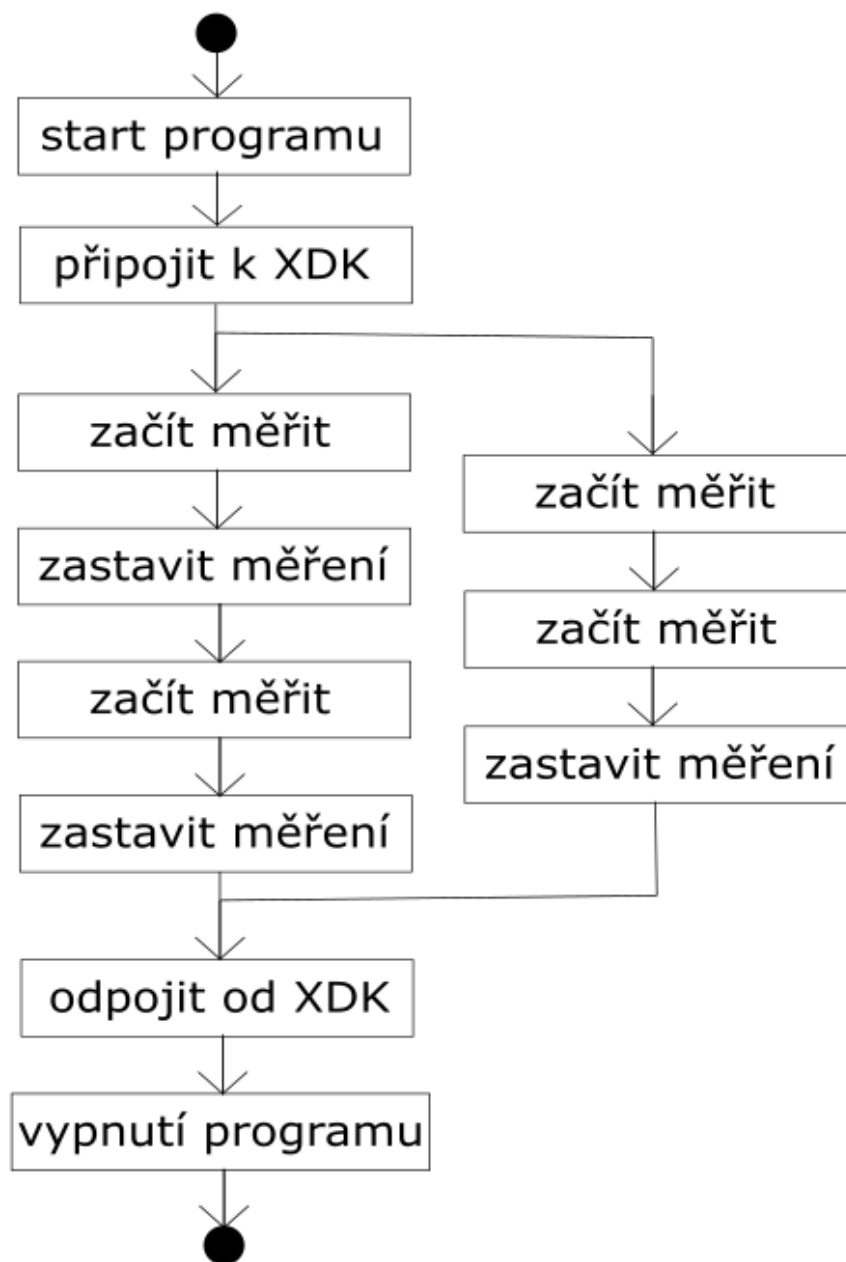
Na Obr. 4-4 jsou ukázány dvě možnosti použití tlačítek na panelu aplikace, a to jedna se střídáním začátku a konce měření. V tomto případě se vytvoří dva samostatné soubory každý s jedním popiskem na začátku souboru. U každého souboru může být jiný popisek podle zvolených snímačů (dvě samostatná měření). Druhá možnost ukazuje použití dvou tlačítek ‚Začít měřit‘. V tomto případě se vytvoří pouze jeden soubor, který obsahuje dva popisky, jeden na začátku souboru a druhý uprostřed souboru na začátku druhé části

měření. Tyto popisky mohou být od sebe odlišné, záleží na změněných snímačích (jedno měření rozdělené na dvě části).

Příklad na Obr. 4-4 je potřeba brát pouze orientačně jednotlivé větve se mohou vzájemně překrývat, doplňovat či natahovat. Tento obrázek jen ukazuje jen možnost, jak řešit více měření, tak aby se vytvořily pro každé měření samostatné soubory, na obrázku levá větve a jak řešit jedno měření se změnou snímačů během měření, tak aby se vše dalo pouze do jednoho souboru.

Například chci měřit teplotu a od okamžiku, kdy dojdou do určité teploty, chci zapnout měření tlaku a zároveň chci, aby obě měření zapsaly do stejného souboru. Po dokončení chci změřit v místě měření ještě osvětlení, které ale chci uložit do samostatného souboru. Proto využiji obrázek níže a upravím si ho tak, že jako základ využiji levou větve, ve které však nahradím první měření pravou stranou a vznikne mi následující postup:

Zapnu program a nechám se připojit k XDK, následně si v nastavení snímačů nastavím měření teploty a zmáčknou ,Začít měřit'. V okamžiku, kdy teplota vyleze na požadovanou teplotu, nastavím snímač tlaku na zapnuto a zmáčknou ,Začít měřit' znovu. Jakmile mám měření hotové zmáčknou tlačítko ,Zastavit měření', kterým se ukončí zápis do souboru. Následně chci ještě měření osvětlení do samostatného souboru tak nastavím snímač osvětlení na zapnuto a ostatní snímače vypnu. Jakmile mám snímače nastaveny zmáčknou ,Začít měřit', po ukončení měření zmáčknou ,Zastavit měření'. Tímto postupem mi vzniknou dva soubory, jeden obsahující měření teploty a později k teplotě přidán i tlak a druhý obsahující jen snímač osvětlení.



Obr. 4-4 Příklad postupu při měření

4.1.3 Komunikace mezi XDK a aplikací

| | | | |
|----------------------|------------------------------|------------------|--|
| 1 byte | 2 byte | 4 byte | |
| Charakterizační znak | Data o vyžádaných přístojích | Perioda vyčítání | |

Obr. 4-5 Data odesílaná programem při zapnutí měření



Obr. 4-6 Struktura dat přijímaných programem

Komunikace mezi XDK a LabVIEW probíhá pomocí tří datových celků, a to start a stop signál, balík dat pro nastavení XDK a data naměřených veličin. Startovací a stop signály jsou nastaveny tak, že posílají do XDK string s textem ‚start‘ a ‚stop‘.

Balík dat pro nastavení XDK se skládá ze tří částí, a to z charakterizačního znaku, dat o snímačích a z délky periody vyčítání. Charakterizační znak slouží pro rozeznání, o jaká data se jedná, pro nastavení snímačů je definován znak ‚c‘, kód z ASCII tabulky 65. Následují dva byty obsahující požadavky na snímače, kdy každý bit značí jeden snímač. První bit z těchto dat značí akcelerometr, druhý bit gyroskop, třetí bit magnetometr bez měření odporu, čtvrtý bit odpor měřený magnetometrem, pátý bit je snímač osvětlení, šestý bit snímač zvuku, sedmý bit snímač tlaku, osmý bit teploměr a devátý bit značí snímač vlhkosti, zbylé byty jsou bez využití a je možnost je využít pro případné rozšíření aplikace. Poslední část tohoto balíku je perioda měření v ms, která zabírá čtyři byty. Strukturu balíku dat pro nastavení XDK je možno najít na Obr. 4-5.

Poslední balík, který je posílán, je balík dat ze snímačů. Ten se skládá ze tří částí (viz. Obr. 4-6). Bytů BLE komunikace je pět a obsahují informace o samotném balíku dat, například pátý byte obsahuje data o délce balíku bez bytů BLE komunikace. Další částí balíku je znak skupiny snímačů, jeho délka je jeden byte a obsahuje informaci o tom, ze kterého snímače či snímačů přijatá data pocházejí. Znak může obsahovat až pět variant, a to ASCII kódy písmen od B do F.

| | | | | | | | | | |
|--------------|----------|--------------------|----------------------|---------------------|-------------------|----------|----------|--------------------|---------------|
| Akcelerometr | Gyroskop | Magnetometr osy | Magnetometr odpor | Snímač osvětlení | Zvukový snímač | Tlakoměr | Teploměr | Snímač vlhkosti | Volné bity |
|--------------|----------|--------------------|----------------------|---------------------|-------------------|----------|----------|--------------------|---------------|

Obr. 4-7 Data o vyžádaných přístrojích

Písmeno A původně mělo být využito pro potvrzování připojení komunikace BLE, ale do programu jsem již tuto vlastnost nedělal a potvrzování je kontrolováno jen tím, že se neobjeví chyba při připojení. Znak skupiny snímačů B je určen pro vyčítání dat z akcelerometru. Znak C je určen pro gyroskop. Znak D je určen pro magnetometr bez magnetického odporu. Znak E obsahuje změřená data odporu z magnetometru, data ze snímače osvětlení a zvukového senzoru. Poslední používaný znak F pak obsahuje data z environmentálních snímačů – tlakoměru, teploměru a snímače vlhkosti.

4.1.4 Program pro XDK

Vytvořená aplikace dokáže komunikovat s různě upraveným programem vytvořeným podle exemplu „SendAccelerometerDataOverBle“. Je potřeba jen přidat na začátek posílaných dat znak skupiny snímačů. V okamžiku, kdy se tento znak přidá, dokáže aplikace již s programem pracovat. Je možno také upravit používané snímače jako například přepsat akcelerometr na gyroskop. Mimo takto jednoduché aplikace, které je v případě změny potřeba přepsat, dokáže aplikace pracovat i se složitějšími programy, které obsahují například i změnu snímačů, či periody.

Při vytváření programu je však třeba si hlídat, aby nedocházelo k přetečení dat mimo maximální délku odesílaného řetězce (20 byte, z toho jeden byte je určen pro zvolení snímače). V kódu XDK se dá nastavit, jak velká bude maximální délka odesílaných dat. V případě nastavení více než dvaceti bytů se mi však objevovala chyba komunikace. V případě, že data překročí maximální nastavenou délku řetězce (využíval jsem 20 byte), přebývající znaky nebudou odeslány a aplikace bude ukazovat špatnou hodnotu. U některých snímačů jsem se setkal s tím, že vytvořená zpráva měla délku (bez znaku skupiny snímačů) i dvacet tři bytů. Pro variantu příliš dlouhého řetězce je do aplikace přidána možnost použití znaku E, který změní tvar odesílaného čísla na „x*10^E“. Ideální délka hodnoty veličiny je maximálně pět znaků včetně případného znaku – a znaku E.

4.1.5 Použitelné znaky pro data ze snímačů

Aplikace dokáže pracovat pro data ze snímačů se všemi čísly (0 až 9) se znakem -, znakem E, mezerou a prázdným znakem.

Znak E v komunikaci značí, kolik nul za předchozím číslem se ještě nachází.

Znak mezera je určen pro označení další veličiny a slouží k jejich rozdělení.

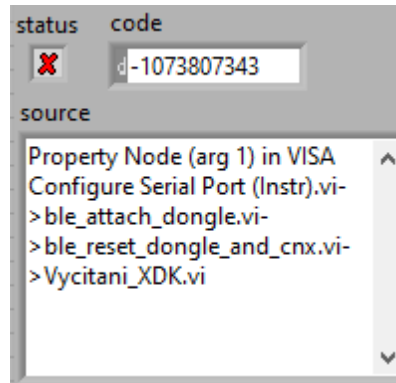
Prázdný znak (odeslána 0) se nachází na konci datového balíku v okamžiku, kdy všechna data jsou již odeslána.

4.2 Možné chyby a jejich řešení

V této podkapitole si představíme možné chyby, které mohou nastat nebo nastaly při testování aplikace.

4.2.1 Připojení donglu a error -1073807343

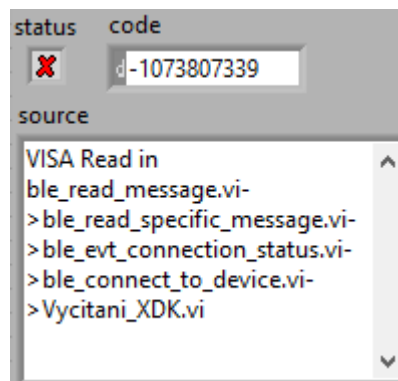
V případě, že není připojen dongle do správného USB, nebo není připojen vůbec, objeví se na výstupu kód chyby -1073807343 (na Obr. 4-8), tuto chybu lze odstranit připojením USB donglu.



Obr. 4-8 Chyba kód -1073807343

4.2.2 BLE v XDK není zapnuto a error -1073807339

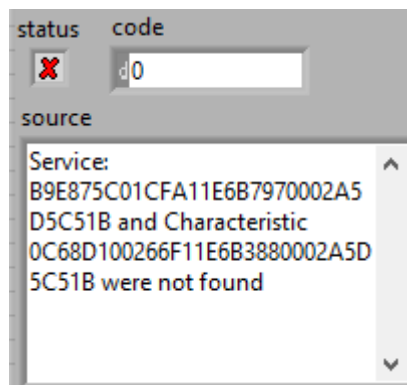
Tato chyba se objevuje v případě, že v XDK není zapnuto Bluetooth zařízení, toto může být způsobeno špatným programem v XDK, nebo přepnutím XDK do bootovacího režimu (k tématu přepnutí do bootovacího režimu je více uvedeno v kapitole 3.4.2). Tuto chybu lze vyřešit nahráním správného programu do XDK. Vzhled chyby je na Obr. 4-9.



Obr. 4-9 Chyba kód -1073807339

4.2.3 Špatné servisní nebo charakteristické UUID a error 0

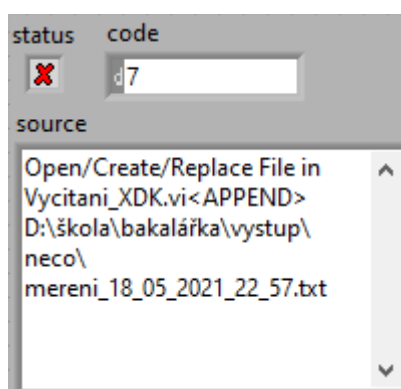
Tato chyba se objevuje v případě, že je v kódu špatně zadaný servisní nebo charakteristické UUID, případně obojí. Výstup chyby je uveden na Obr. 4-10. Postup úpravy UUID je v kapitole 3.4.3.



Obr. 4-10 Chyba kód 0 – servisní nebo charakteristické UUID

4.2.4 Neplatná složka pro ukládání hodnot a error 7

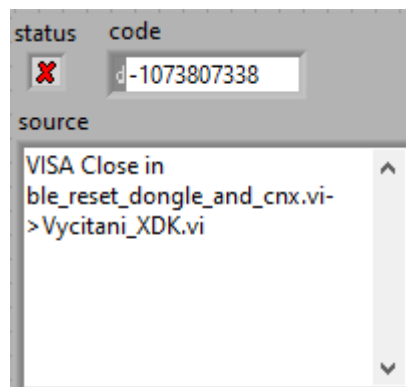
Tato chyba se objevuje v případě, že je v kódu na ukládání změřených hodnot uvedena složka, která neexistuje a program proto nemá kam vložit soubor s naměřenými hodnotami. Výstup chyby je uveden na Obr. 4-11.



Obr. 4-11 Chyba kód 7

4.2.5 Chyba při resetu USB donglu a error -1073807338

Tato chyba se objevuje v případě, že se mezi jednotlivými měřeními vysune USB dongle, tuto chybu není nutno nijak opravovat, stačí aplikaci spustit znovu a chyba se již neobjeví. Výstup chyby je uveden na Obr. 4-12.



Obr. 4-12 Chyba kód -1073807338

Mimo výše uvedené chyby se mohou v kódu objevit i další, které se během vytváření kódu neprojevily.

4.3 Shrnutí funkčnosti aplikace

Aplikace umožňuje komunikaci mezi XDK a LabVIEW pomocí BLE. Aplikace umí volbu snímačů podle požadavků uživatele, zároveň umožňuje i volbu periody vyčítání a veškeré nastavení přeposílá pomocí BLE i do XDK. Přijaté hodnoty z XDK umožňuje uložit do textového souboru a zároveň umožňuje zobrazit jednu veličinu do grafu.

5. ZÁVĚR

Hlavním cílem bakalářské práce bylo vytvořit návrh programu v LabVIEW pro stažení změřených dat z programovatelného zařízení XDK 110 při využití bezdrátového přenosu dat. Pro tuto bakalářskou práci bylo jako komunikační rozhraní zvoleno Bluetooth.

Vedlejším úkolem bylo vytvořit stručný popis zařízení Bosch XDK a návod na práci s ním. Tento návod se nachází v kapitole 1. V první části této kapitoly jsou popisovány jednotlivé snímače a druhá část popisuje program XDK-Workbench, který se používá k programování zařízení XDK. V této kapitole je také ukázán example, který byl v práci použit a probrána jeho struktura.

Kapitola 2 se zabývá komunikačním rozhraním Bluetooth a prací s Bluetooth v LabVIEW. Je zde popsána komunikace jak s klasickým Bluetooth, tak úprava pro komunikaci s Bluetooth Low Energy a také je zde popsáno, co Bluetooth Low Energy je. V práci jsou popsány dvě varianty řešení komunikace mezi BLE a LabVIEW, které použití BLE zatím přímo nepodporuje.

V kapitole 3 je uvedena původní představa aplikace, která se však z důvodu problémů s BLE změnila a aktuální verze aplikace je uvedena v kapitole 4. Mimo původní představu je v kapitole 3 uveden popis postupu řešení a také poznatky z řešení aplikace.

V kapitole 4 je popsána vypracovaná aplikace a jsou rozebrány chyby, které nastaly během řešení nebo se domnívám, že by se mohly během práce s aplikací vyskytnout.

Při pokračování práce je možno přidat do programu funkce pro práci s pamětí XDK, jak je uvedeno v kapitole 3.1 Čtení z XDK.

LITERATURA

- [1] Cross Domain Development Kit. Bosch [online]. Gerlingen-Schillerhöhe: Bosch Connected Devices and Solutions, 2017 [cit. 2020-11-18]. Dostupné z: https://developer.bosch.com/documents/422133/482930/XDK_Node_110_combined_Datasheet.pdf/ace7ca7c-fd0e-9741-acf6-e9d5dc10957d
- [2] Technical information. Bosch [online]. Gerlingen-Schillerhöhe: Bosch Connected Devices and Solutions, 2018 [cit. 2020-11-18]. Dostupné z: <https://developer.bosch.com/web/xdk/technical-information>
- [3] BMG160. Alldatasheet [online]. [cit. 2020-12-30]. Dostupné z: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1131979/BOSCH/BMG160.html>
- [4] BMI160. Bosch [online]. Reutlinger: Bosch Sensortec, c2020 [cit. 2020-12-30]. Dostupné z: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi160-ds000.pdf>
- [5] BMA160. Bosch [online]. Reutlinger: Bosch Sensortec, c2020 [cit. 2020-12-30]. Dostupné z: <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bma280-ds000.pdf>
- [6] Communicate with Bluetooth Low Energy (BLE) Devices in LabVIEW. NI Community [online]. Austin: NATIONAL INSTRUMENTS, c2020 [cit. 2020-12-30]. Dostupné z: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019LuHSAU&l=cs-CZ>
- [7] BLE Toolkit. NI Community [online]. Austin: NATIONAL INSTRUMENTS, c2020, 2016 [cit. 2020-12-30]. Dostupné z: <https://forums.ni.com/t5/Community-Documents/LabVIEW-BLE-Bluetooth-Low-Energy-toolkit/thread/3538612?profile.language=en>
- [8] WHAT IS BLUETOOTH LOW ENERGY? ElaInnovation [online]. Montpellier: ELA Innovation, 2019 [cit. 2020-12-28]. Dostupné z: <https://elainnovation.com/what-is-ble.html>
- [9] What is Bluetooth Low Energy (BLE) and how does it work? Centare [online]. Brookfield: Centare, c2019, 2019-03-07 [cit. 2020-12-30]. Dostupné z: https://www.centare.com/blog/what_is_bluetooth_low_energy/
- [10] Anatomie programu Mita. Eclipse foundation [online]. Ottawa: Eclipse Foundation [cit. 2021-02-03]. Dostupné z: <https://www.eclipse.org/mita/language/introduction/>
- [11] Bosch Connected Devices and Solutions prototypová deska XDK 110 Cross-Domain Development Kit XDK 110. Conrad Electronic [online]. Vinohradská 2828/151, 130 00, Praha 3 - Žižkov: Conrad Electronic Česká republika, c2020 [cit. 2021-5-4]. Dostupné z: <https://www.conrad.cz/p/bosch-connected-devices-and-solutions-prototypova-deska-xdk-110-cross-domain-development-kit-xdk-110-1421124>

- [12] Bosch Connected Devices and Solutions presents XDK. Youtube [online]. Reutlingen: Bosch Connected Devices and Solutions, 2016 [cit. 2021-5-4]. Dostupné z: <https://www.youtube.com/watch?v=FAlyjBO0-7g>
- [13] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. Začínáme s LabVIEW. Praha: BEN – technická literatura, 2008. ISBN 978-80-7300-245-9.
- [14] GATT. Adafruit [online]. adafruit, 2014 [cit. 2021-5-5]. Dostupné z: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt>
- [15] Bluetoothapis.h header. Microsoft [online]. Redmond: Microsoft, c2021 [cit. 2021-5-9]. Dostupné z: <https://docs.microsoft.com/en-us/windows/win32/api/bluetoothapis/>
- [16] Bluetooth vs WiFi Comparison For the IoT Solutions. Netguru [online]. Poznań: Netguru S.A., c2021 [cit. 2021-5-23]. Dostupné z: <https://www.netguru.com/codestories/bluetooth-vs-wifi-comparison-for-the-iot-solutions>
- [17] AKU340. Mouser Electronic [online]. Brno: Mouser Electronics, c2021 [cit. 2021-5-23]. Dostupné z: <https://www.mouser.com/datasheet/2/720/DS26-1.04%20AKU340%20Datasheet-770074.pdf>
- [18] GATT (Services and Characteristics). O'Reilly [online]. Sebastopol: O'Reilly Media, c2021 [cit. 2021-5-23]. Dostupné z: https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch04.html#gatt_svc_disc
- [19] Přehled doplňků standardu IEEE 802.11. Access server [online]. Praha: České vysoké učení technické v Praze, FEL, c2007 [cit. 2021-7-20]. Dostupné z: <http://access.fel.cvut.cz/view.php?cisloclanku=2005113002>
- [20] Bluetooth Technology Overview. Bluetooth [online]. Kirkland: Bluetooth SIG, c2021 [cit. 2021-7-20]. Dostupné z: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>
- [21] Understanding Bluetooth Range. Bluetooth [online]. Kirkland: Bluetooth SIG, c2021 [cit. 2021-7-20]. Dostupné z: <https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/>

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

| | |
|-------|--|
| ATT | Attribute protocol |
| AWS | Amazon Web Services, poskytovatel cloudových služeb |
| BLE | Bluetooth Low Energy |
| DC | Direct Current, stejnosměrný proud |
| DEC | Dekadické číslo |
| GATT | Generic Attribute |
| GET | Metoda pro vyžádání dat z určitého zdroje |
| GPIO | General Purpose Interface Bus |
| HEX | Hexadecimální číslo |
| IoT | Internet of Things |
| LED | Light-Emitting Diode |
| LoRa | Řešení pro bezdrátový přenos dat |
| LWM2M | Protokol pro správu a služby zařízení mezi stroji nebo IoT |
| MQTT | Message Queue Telemetry Transport (otevřený protokol pro výměnu dat na vzdálená místa) |
| OS | Operační systém |
| txt | Koncovka textového souboru |
| USB | Universal Serial Bus (Univerzální sériová sběrnice) |
| UUID | Universally unique identifier |
| VI | Virtual Instrument, přípona souborů LabVIEW |
| WEP | Wired Equivalent Privacy, zastaralý standard zabezpečení bezdrátové sítě |
| WPA | Wi-Fi Protected Access, zabezpečení bezdrátových sítí |

SEZNAM PŘÍLOH

PŘÍLOHA A - PROGRAM NA MĚŘENÍ POMOCÍ BLE V LABVIEW..... PŘILOŽENO NA CD