



IÐN- OG TÆKNIFRÆÐIDEILD

B.Sc. í Rafmagnstæknifræði

Hönnun sírita fyrir rennsli hlutfylltra lagna

Desember, 2022

Nafn nemanda: Bjarki Guðjónsson

Kennitala: 080398-2209

Umsjónarmaður: Guðmundur Kristjánsson

Leiðbeinandi: Sverrir Óskar Elefsen

24 ECTS ritgerð til B.Sc. í Rafmagnstæknifræði

Iðn- og tæknifræðideild**Heiti verkefnis:**

Hönnun á sírita fyrir hlutfylltar lagnir

Námsbraut:

Rafmagnstæknifræði

Tegund verkefnis:

Lokaverkefni í tæknifræði BSc

Önn:

Haust 2022

Námskeið:

RTLOK1012

Ágrip:

Í þessu verkefni var hönnuð frumgerð að sírita fyrir hlutfylltar lagnir, með fyrirfram ákveðum skynjurum. Frumgerðin þarf að lesa af skynjurum á fimm sekúnda fresti, skrá niður mínútu meðaltal mældra gilda, ásamt því að skila gögnum á gagnaver.

Var það leyst með notkun Arduino tölvu og Sim7600.

Höfundur:

Bjarki Guðjónsson

Umsjónarkennari:

Guðmundur Kristjánsson

Leiðbeinandi:

Sverrir Óskar Elefsen

Fyrirtæki/stofnun:

Mannvit ehf.

Dagsetning:

16.12.2022

Lykilorð íslensk:Síriti, Arduino,
Sim7600**Lykilorð ensk:**Data logger, Arduino,
Sim7600**Dreifing:**opin lokuð

til:

Formáli

Verkefnið var unnið sem B.Sc.lokaverkefni í rafmagnstæknifræði við Háskólann í Reykjavík á haustönn 2022 og var það unnið í samvinnu við verkfræðistofuna Mannvit.

Sérstaklega vil ég þakka Mannvit fyrir að veita mér aðgang að aðstöðunni í Ögurhvarfi til skrifta og vinnu að lokaverkefni.

Þá vil ég þakka leiðbeinandanum mínum Sverri Óskari Elefsen fyrir góða leiðslu í verkefninu ásamt því að hleypa mér út að vinna í mælingum þegar hausinn var alveg að springa við skriftina. Án þess hefði verkefnið ekki orðið eins þar sem það leyfði mér að endurstilla hugann.

Jafnframt vil ég þakka Eylvallíu Kolbrúnu Ágústsdóttur og Sif Guðjónsdóttur fyrir vel unna yfirlit og lesningu. Án leiðréttingu þeirra á stafsetningu hefði líklega ekki verið hægt að skilja meira en tvær setningar í ritgerðinni.

Að lokum færi ég sérstakar þakkir til Borgþórs, kokksins í Mannvit fyrir einstaklega góðan mat í hádeginu. Án hans hefði lítið komist á blað þar sem erfitt getur reynst að skrifa þegar svengdin tekur völd.

Efnisyfirlit

Lykilsíða	i
Formáli	ii
Myndaskrá	v
Töfluskrá	vi
1 Inngangur	1
1.1 Uppsetning verkefnis	2
1.2 Markmið	2
2 Síritar	3
2.1 Síritun flæðis í hlutfylltri lögn	4
2.2 Skynjari í síritanum	5
2.3 Núverandi lausn Mannvits	6
3 Heili sírita	7
3.1 Uppbygging örtölvu	7
3.1.1 Örgjörvi (CPU)	7
3.1.2 Minnið	8
3.1.3 Inn- og útgangar	9
3.2 Valin örtölva	11
4 Fjarskipti	12
4.1 Mótun og kóðun	12
4.2 Flutningsrás	14
4.3 Truflanir	15
4.4 Fjarskiptalausnir sírita	16
4.4.1 Símkertið	16
4.4.2 LoRaWAN	17
4.4.3 SigFox	19
4.5 Valin fjarskiptalausn	19
5 Samskiptamátar raftækja	22
5.1 Serial samskipti	24
5.1.1 Universal Asynchronous Receiver/Transmitter (UART)	25

5.1.2	RS485 og RS232	26
5.1.3	Inter-Integrated Circuit (I2C)	26
5.1.4	Serial Peripheral Interface (SPI)	27
5.2	Modbus	28
5.3	AT skipanir	30
6	Samsetning síritans	31
6.1	Tengjast við skynjara	32
6.2	Koma gögnum á gagnaver Mannvits	35
6.3	Skrifa á SD kort	35
6.4	Forritun	36
7	Samantekt	38
7.1	Verðsamanburður	39
7.2	Vandamál sem komu upp við smíðina	39
7.2.1	SD kort	40
7.2.2	Sim7600	40
7.2.3	Modbus	41
7.3	Straumnotkun búnaðarins	41
7.4	Framtíðar viðbætur	42
8	Lokaorð	43
	Heimildir	44
	Viðaukar	50
A	Rásateikning	50
B	Forrit	51

Myndaskrá

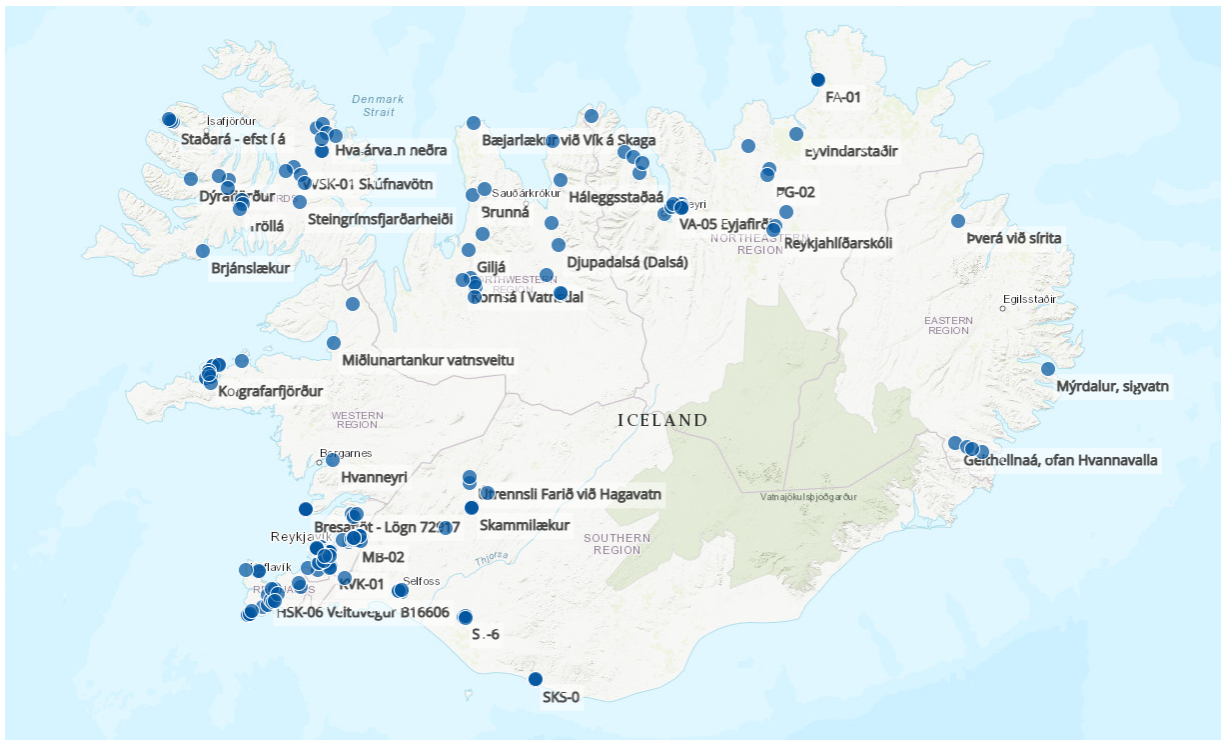
1	Staðsetning vatnamælinga á vegum Mannvits [1].	1
2	Mismunur á sekúndu gildum og hlaupandi meðaltali.	3
3	Útskýringarmynd á flæði í lögnum [2].	4
4	Hugmynd af uppsetningu á síritum í hlutfylltum lögnum [8].	5
5	Samband tenginga innan örtölvu.	7
6	Uppbygging CPU [12].	8
7	Munurinn á mismunandi safntíðni á merkjagæðum [15].	10
8	Mismunandi upplausn merkis sem örtölva nær að lesa. Innmerki er alltaf sínus bylgja [14].	10
9	Uppbygging fjarskiptakerfa [20].	12
10	Munur á milli AM og FM mótnar, burðarbylgjan er það sem er eins í FM og AM [24].	13
11	Munur á milli ASK og FSK [25].	14
12	Séð hvernig suð getur truflað, vinstri myndin sýnir mikið suð en nánast ekkert suð er á hægri mynd [28].	15
13	Hvernig bjögun hefur áhrif á merki, efri mynd sýnir upprunalega merkið og neðri mynd það bjagaða [29].	15
14	Uppgefinn 4g dreifing Símans [31].	17
15	LoRaWAN dreifni frá Lýsi [38].	18
16	Munur á serial og parallel samskiptum [51].	23
17	Tenging á serial samskiptum [50].	25
18	Myndræn útfærsla á UART [54].	25
19	I^2C sending [56].	27
20	Tengimynd af SPI [54].	28
21	Blökkmynd af síritanum.	31
22	Hvernig er hægt að lesa spennumerki út frá straummerki.	33
23	Flæðirit forrits í síritanum.	37
24	Uppsetning prófun á frumgerðar.	38

Töfluskrá

1	Verð á núverandi síritum.	6
2	Eiginleikar fjarskiptalausna [35] [39] [41] [42] [43].	19
3	Samanburður á búnaði sem fjarskipta lausn [45] [46] [47].	20
4	Munur á serial og parallel [52].	23
5	Sýnir þær töflur sem geyma upplýsingarnar sem tækin geyma [58].	29
6	Fuction codes í Modbus samskiptum [58].	29
7	Hvaða merki skynjarar gefa frá sér og hvaða spennu þeir nota [3] [5].	32
8	Mismunur á nákvæmni á milli 10 bit og 16 ADC [3] [5].	34
9	Verð í frumgerð síritans [44] [71] [72] [62].	39
10	Straumnotkun búnaðar [17] [47] [63] [73] [74].	41

1. Inngangur

Verkfræðistofan Mannvit sérhæfir sig í verkefnum tengdum orku, iðnaði, mannvirkjagerð og vatnamælingum. Við vatnamælingar er notað ýmsar leiðir mælinga, hvort sem það er handmælingar eða sírítar að verki. Síriti er tæki sem mælir og skráir mælingar sjálfvirk (sjá kafla 2). Mannvit framkvæmir mælingar víðs vegar um land allt og má sjá staðsetningu mælistöðvanna á mynd 1.



Mynd 1: Staðsetning vatnamælinga á vegum Mannvits [1].

Verkefnum Mannvits í vatnamælingum má skipta í tvo flokka:

- Mælingar fyrir gagnasöfnum fyrir stærri verka, til dæmis virkjana kosti, stækkun hafnar-garðar eða þá vatnstaka.
- Almennar eftirlitsmælingar með vatni, til dæmis stöðu grunnvatns, vatnsgæði og fylgjast með rennsli í skolplögnum sem dæmi mætti nefna.

1.1. Uppsetning verkefnis

Uppsetning verkefnis var hugsuð þannig að smíða átta frumgerð sírita sem gæti mælt úti á mörkinni og skilað því til gagnavers Mannvits, svipað eins og núverandi síritar gera. Var þá hugsunin að geta smíðað lausn sem væri ódýrari en núverandi lausn Mannvits. Til að einfalda verkefnið var ákveðið að hanna sírita sem ætti að mæla í hlutfylltum lögnum (skolpi), frekar en sírita sem gæti mælt við allar aðstæður sem núverandi síritar geta. Einnig var verkefnið sett upp þannig að nota ætti sömu skynjara og Mannvit notar nú þegar. En til þess að mæla flæðið í skolplögnum þarf að geta mælt bæði hraða á vökvanum ásamt þverskurðarflatarmáli hans til að geta reiknað út flæðið.

1.2. Markmið

Markmiðin í lokaverkefninu eru:

- Finna tölvu sem getur lesið af skynjaranum og skráð það í minni.
- Geta geymt skráð gildi ef tæki verður rafmagnslaust.
- Geta skilað mældum gögnum til gagnavers Mannvits.

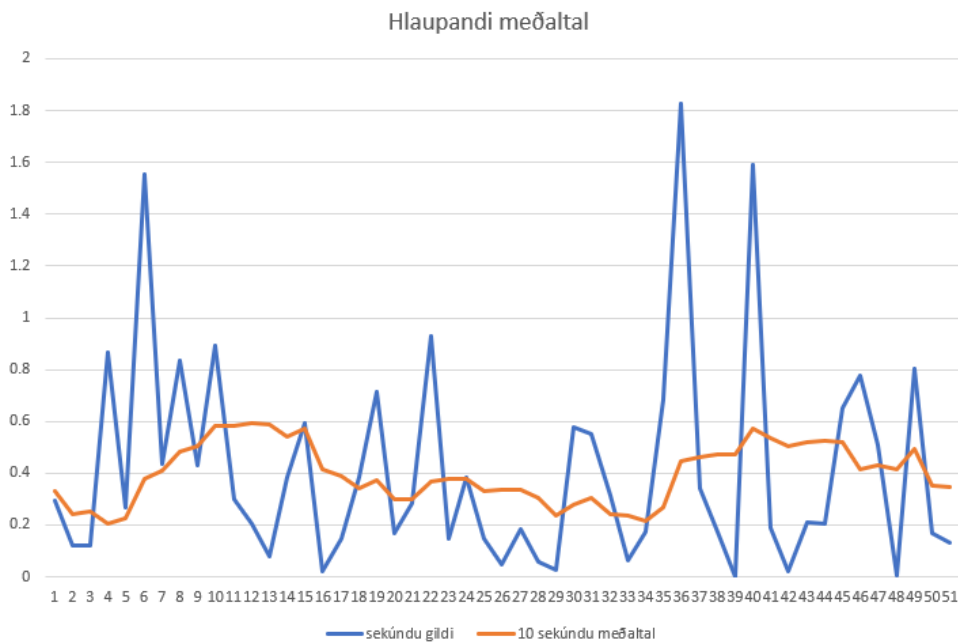
Virknilysing sírita skal vera þannig á 5 sekúnda fresti les hann gildi af mælum en skráir síðan einungis mínútu meðaltal í minnið, það er að segja meðaltal af seinustu 12 gildum. Síritinn skal síðan senda afrit af seinustu klukkustund til gagnavers Mannvits.

Einnig var þælt í straumnotkun tækis til að vita hversu lengi það getur keyrt á batterí.

2. Síritar

En hvað er síriti? Síriti er tæki sem les gildi af skynjara sem er innbyggður í tækið eða tengdur að utanverðu við hann. Skráir hann og geymir lesna gildið við tímastimpil, sem geymir hvaða ár, mánuður, dagur og klukkan er við lesningu. Hversu oft síritir mælir fer alfarið eftir hönnunarforsendum en þær geta verið stilltar á að mæla mörgum sinnum á sekúndu í að mæla einu sinni á dag eða sjaldnar.

Síritinn sem verður hannaður á að geta mælt á fimm sekúnda fresti en skráir síðan einungis niður mínútu meðaltal. Er það gert til að minnka þau suð sem geta verið í mælingum, sem skapast vegna öra breytinga. Má til dæmis nefna þegar það er verið að mæla vatnshæð sjávar, en þar hefur öldugangur áhrif á mælingar og myndar mikið suð. Suðið myndast vegna þess að ekki er hægt að tryggja að mælirinn hittir á sama stað á öldu, þar af leiðandi verða örar breytingar þegar hæð sjávarfalla er mæld. Ef mynd 2 er skoðuð má sjá áhrif þess að taka mínútumeðaltal.



Mynd 2: Mismunur á sekúndu gildum og hlaupandi meðaltali.

Svona suð getur einnig gerst í skolplögnum þar sem rennsli þar getur verið mjög flöktandi, því þarf að nota mínútu meðaltal til að hreinsa burt allt suð.

2.1. Síritun flæðis í hlutfylltri lögn

Hægt er að setja flæði hvers vökva fyrir sig upp með einfaldri formúlu.

$$Q = A \cdot \vec{v} \quad (1)$$

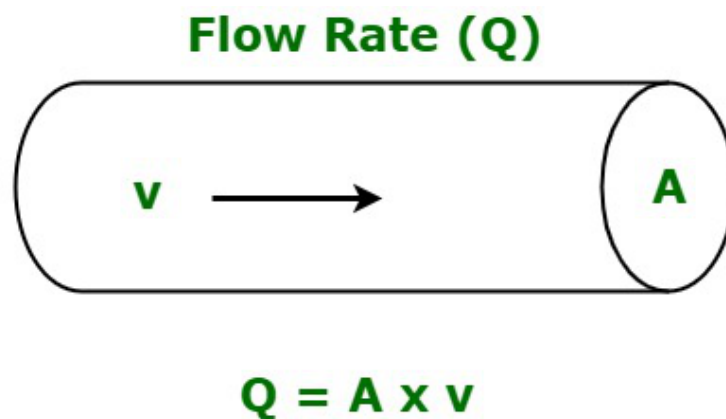
Þar sem:

$A [m^2]$ = Þverskurðarflatarmál af vökvanum, þ.e. vota flatarmálið.

$\vec{v} [m/s]$ = Meðalhraði vökva í sniðinu.

$Q [m^3/s]$ = Flæðið.

Til þess að geta síritað flæði á vökva þarf að mæla hraðann á vökvanum ásamt því að vita þverskurðarflatarmálið. Þetta er fremur einfalt ef um er að ræða fylltar lagnir þar sem þverskurðarflatarmálið er fasti og þarf því einungis að meta meðalstraumhraðann á vökvanum. Mynd 3 sýnir flæði í lögnum.



Mynd 3: Útskýringarmynd á flæði í lögnum [2].

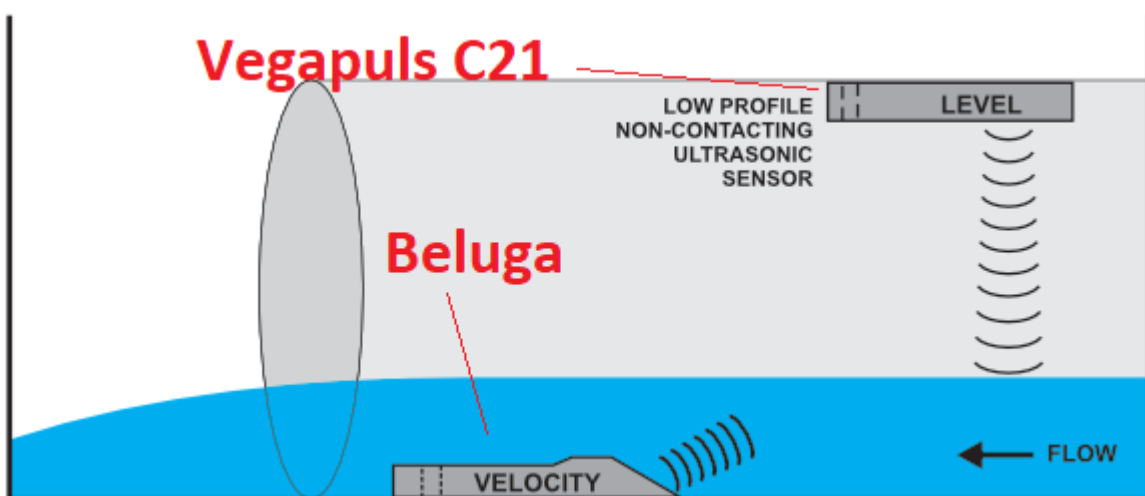
En ef um hlutfylltar lagnir er að ræða þarf einnig að mæla þverskurðarflatarmálið vökvens, þar sem vatnsmagn getur breyst. Í öllum tilfellum er vota þverskurðarflatarmálið fundið út frá mælingum á vatnsdýpi í lögnum sem oftast er gert með þrýstinema.

2.2. Skynjari í síritanum

Til að mæla vatnshæðina í lögninni er skynjari frá Vegapuls að nafni C21, en það er mælir sem Mannvit hefur verið að nota til hæðarmælinga. Sá mælir notar radar til að mæla fjarlægð niður á vatnsborð [3]. Virkar hann þannig að skynjarinn sendir frá sér radar bylgju sem endurkastast til baka af yfirborðinu. Skynjarinn sjálfur bíður eftir því að endurkastið skili sér til baka inn í skynjarann. Mælir hann þá tímann sem radar bylgjan ferðast, og reiknar þá út fjarlægðina út frá tímanum. Vitað er hversu hratt radar bylgjan ferðast í lofti sem er á við ljóshraða, en ljóshraði er $2,998 \cdot 10^8$ m/s [4].

Til að mæla straumhraðann er skynjari frá Flow-tronics notaður sem heitir Beluga, en það er mælir sem Mannvit notar til vatnshraðamælinga. Er honum komið fyrir í botni lagnar og notar síðan úthljóðs Dopplersáhrifa (e. Ultrasonic Doppler effect) til að mæla straumhraðann á vatninu [5]. Beluga notar úthljóðsbylgjur til að mæla hraðann á vatninu svipað eins og Vegapuls C21. Beluga sendir út úthljóðsbylgjur í 45° halla á móti straumátt vökvans. Sendir Beluga út á tíðni A en fær til baka tíðni B, vegna Dopplersáhrifa (e. Dopplers effect) [6]. En Doppleráhrifin er það þegar viðtakandi upplifir tíðni breytinga vegna færslu sendis eða viðtakanda. Getur viðtakandi upplifað bæði tíðni aukningu eða tíðni rýrnun. Svipað eins og þegar bíll með sírenu ekur fram hjá [7]. En Beluga neminn reiknar út hraðann út frá tíðni mismun á A og B, þar sem vökvinn ýtir endurkastinu frá A hraðar til baka.

Til að sjá hvernig uppsetning skynjara er hugsuð er hægt að skoða mynd 4.



Mynd 4: Hugmynd af uppsetningu á síritum í hlutfylltum lögnum [8].

2.3. Núverandi lausn Mannvits

Mannvit setur síritun í skólplögnum upp sem tvær tegundir að lausnum í dag, sem eru annars vegar skammtímamælingar sem telur í vikum eða langtímamælingar sem er sett upp sem varanlega stöð með það í huga að hún verði ekki tekin niður á næstu árum. Það eru notaðar mismunandi aðferðir við þessar tegundir mælinga, en þegar kemur að skammtímamælingum er notaður síriti sem heitir **Hach FL900** sem mælir bæði vatnshæð og vatnshraða til að finna rennsli. En við langtímamælingar er notaður síriti frá **Campell**, en þar er einungis mæld vatnshæð. Til að reikna út rennslið er fundið sambandið á milli vatnshæðar og rennslis með því að framkvæma skammtímamælingar á rennsli og borið það saman við síritaða vatnshæðarmælingu og út frá því er svokallaður rennslislykill fundinn, en hann segir til um samband vatnshæðar og rennslis.

Ástæða þess að einungis er mælt hæðina í langtímamælingum er að ýmislegt sem finnst í skólpi á það til að festast í hraðaskynjara, þar sem hann er ofan í skolpinu með þeim afleiðingum að hraðaskynjari hættir að mæla rétt.

Báðir síritar eru fremur dýrir en verð á þeim má sjá í töflu 1. Í töflu 1 er miðast við að 1 USD sé 141 ISK þann 25.11.2022 [9].

Tafla 1: Verð á núverandi síritum.

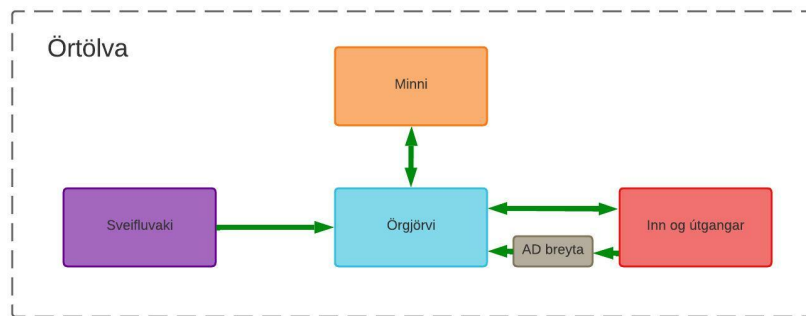
	Verð [USD]	Verð [ISK]
Campell	1100	155100
Hach FL900	2800	394800

3. Heili sírita

Þegar síritar eru smíðaðir er oft notað örtölvu (e. Microcontroller) sem sér um allt sem kemur að síritanum. Það er að lesa merkið frá skynjara, vinna með það ef þess þarf, og skrá það niður í minni.

3.1. Uppbygging örtölvu

Uppbygging örtölva er í grunninn samsetning örgjörva (e. Central processing unit), sveifluvaka (e. oscillator), minnis, inn- og útganga (e. I/O unit), hliðræn til stafræn breyta (e. AD converter) ásamt fleiru er sett saman í einn íhlut, sem myndar örtölvu. Örgjörvi, sem er skammstafað CPU er kjarninn í örtölvunni en hann sér um að tala við alla hina íhlutina, eins og sést á mynd 5 [10].



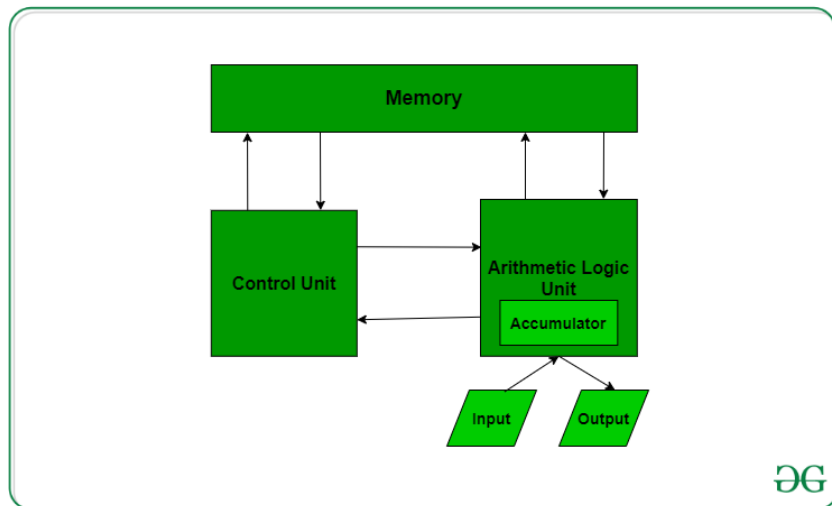
Mynd 5: Samband tenginga innan örtölvu.

3.1.1. Örgjörvi (CPU)

Örgjörvi eða CPU, er heilinn á bak við örtölvuna. Er það í honum sem allar aðgerðir eru framkvæmdar. Mynd 6 sýnir uppbyggingu örgjörva. Örgjörvi hefur síðan tvær megin einingar:

- Reikni og rökverk (e. Arithmetic logic unit) eða ALU.
- Stýriverk (e. Control unit) eða CU.

CU er undirstaðan á örgjörvanum, það sér um að upplýsingar frá minni fara á réttan stað í réttri röð í örgjörvanum og ALU sér um hvernig á að bregðast við inn- og útgagnsmerkjum. ALU sér einnig um að framkvæma allar aðgerðir eins og útreikninga og stjórnun á inn- og útgöngum [11] [12].



Mynd 6: Uppbygging CPU [12].

Við hvern örgjörva er tengdur sveifluvaki sem gefur út klukkupúlsa. Hraði klukkupúlsanna er það sem ræður því hve hratt örgjörvinn getur unnið. Þar sem hann getur einungis framkvæmt eina aðgerð við hvern klukkupúls, ein aðgerð getur verið til dæmis útreikningur, kveikja á útgögum eða lesa af útgöngum [13].

3.1.2. Minnið

Örtölvur hafa tvær gerðir af minni en þau eru:

- Lesminni (e. Read-only memory) eða ROM.
- Vinnuminni (e. Random-access memory) eða RAM.

ROM er það minni sem ekki er hægt að breyta auðveldlega þar sem þetta geymir upplýsingar um hvernig örtölvun á að haga sér. En geymir það upplýsingar um forritunarkóða örgjörvans. Upplýsingarnar sem eru geymdar í ROM hverfa ekki í straumleysi [12].

RAM er minnið sem geymir upplýsingar um stöðu á örgjörva á meðan örgjörvi er í keyrslu. Þar af leiðandi eyðist allt í þessu minni við straumleysi [12].

3.1.3. Inn- og útgangar

Til þess að örtölvur geti haft samskipti við umheiminn, þurfa þær að vera með einhverja inn og útganga. Allir inn- og útgangar eru stafrænir (e. digital) og ná því einungis að lesa stafræn merki en ekki hliðræn (e. analog), nema hafa einhverja breytu á milli. En stafrænt merki er annað hvort af eða á eins og 0V er af og 5V er á, á meðan hliðrænt getur verið allt á milli 0-5V.

Þá til að geta lesið hliðrænt merki þarf að nota hliðræna til stafræna breytu (e. analog to digital converter) eða ADC eins hún er skammstöfuð. En hún sér til að örgjörvinn geti lesið hliðræn merki. Alltaf er einhver skekkja í þessum ADC breytum en nákvæmni hennar ræðst af því hversu há safntíðnin (e. sampling rate) er ásamt því hver mikil upplausnin (e. Resolution) er [14].

Safntíðni segir til um hversu ört breyta les á hliðræna merkið á hverri sekúndu, sem er gefið upp í tíðni. Til að finna út hver safntíðnin er, er hægt að nota formúlu 2.

$$f_s = \frac{1}{T} \quad (2)$$

$f_s[Hz]$ = Safntíðni.

$T[s]$ = Tími milli lesningar.

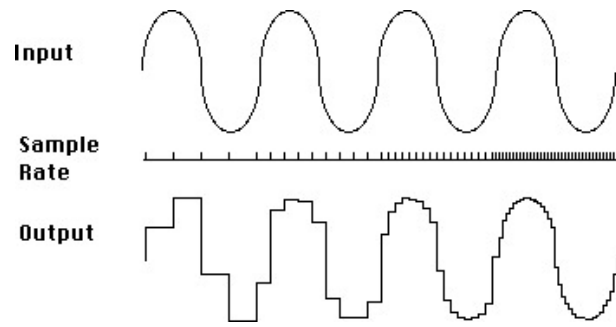
Til að tryggja að safntíðnin sé nægilega góð er hægt að beita Nyquist setningu (e. Nyquist Theorem). En hún segir að safntíðnin skuli vera að minnsta kosti tvisvar sinnum stærri en hæsta mögulega tíðni í merki, sem hún á að geta lesið. Sjá má jöfnu Nyquist setningu í jöfnu 3 [14].

$$f_s = 2 \cdot f_{max} \quad (3)$$

$f_s[Hz]$ = Safntíðni.

$f_{max}[Hz]$ = Hæðsta mögulega tíðni sem á að mæla.

Þannig því hærrí sem safntíðnin er, því betur er hægt að lesa inn upprunalega merkið. En ef mynd 7 er skoðuð má sjá hversu mikil áhrif það hefur að hafa safntíðnina nægilega góða til þess að örtölvun geti lesið merkið vel.



Mynd 7: Munurinn á mismunandi safntíðni á merkjagæðum [15].

Upplausn á ADC segir til um hversu mikil skekkja er á merki sem lesið er inn. Notaðir eru bitar (e. bits) til að segja til um nákvæmni ADC þar sem merkinu er deilt niður í 2^{bit} bita. Hægt er að setja þetta upp í einfaldri formúlu 4 [14].

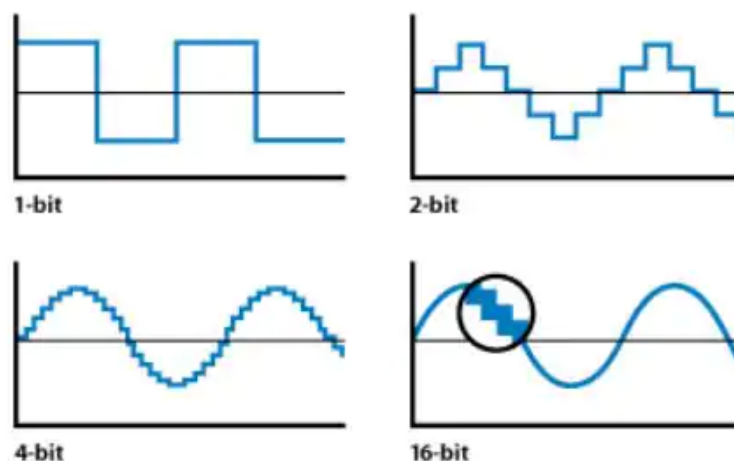
$$V_{bit} = \frac{V_{ref}}{2^N} \quad (4)$$

$V_{bit}[V]$ = Spenna sem einn biti tákna.

$V_{ref}[V]$ = Hæsta spenna sem ADC breyta ræður við.

N = Bitastærð ADC.

Virkar það þannig að merkið er bútað niður í 2^N gildi, því stærra sem N er því nákvæmnari er upplausn á merkinu. Eins og sést á mynd 8 þá er mikill munur á því að hafa 1 bit, 4 bit, 8 bit eða 16 bit ADC.



Mynd 8: Mismunandi upplausn merkis sem örtölva nær að lesa. Innmerki er alltaf sínus bylgja [14].

3.2. Valin örtölva

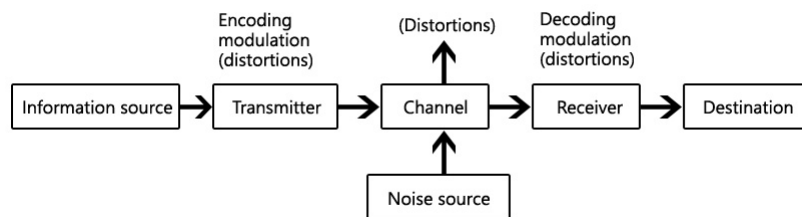
Til eru margar örtölvur sem henta í að smíða sírita. Hér eru tvær örtölvur sem henta miðað við hönnunar forsendur (kafli 1.2):

- **Arduino**, er opin búnaður (e. open-source), en í því fær maður bæði örtölvu ásamt því að fá hugbúnað til að forrita hana. Arduino er þægileg í notkun þar sem auðvelt er að koma forriti í hana ásamt því að hún er vinsæl. Einnig er mikið til af upplýsingum um hana. Takmarkandi þáttur er hve fá forritunar mál er hægt að nota í hennina [16]. Arduino getur verið að nota á 11 til 98 mA í keyrslu [17] .
- **Raspberry PI**, er opinn búnaður, en það er tölva sem er á stærð við kreditkort. Ekkert takmark er á hvaða forritun er notuð við að forrita búnaðinn, því er hægt að nota hana í fleiri tilvikum en Arduino. Hins vegar er straumnotkun meiri en er hún á bilinu 100 til 140 mA [18] [19].

Fyrir þetta verkefni var Arduino notað fyrir sírita fremur en Raspberry PI, vegna þess að Arduino notar minna afl þegar hún er í keyrslu. Einnig ef siritinn myndi missa allan straum myndi hann byrja stax aftur eins og frá var horfið. En það gerir Arduino tölvan alltaf en á Raspberry PI gæti það klikkað.

4. Fjarskipti

Til þess geta skilað inn mælingum frá mörkinni til gagnavers Mannvits þarf að nota fjarskipti. Fjarskipti eru í grunninn leið til að geta haft samskipti milli tveggja staða með skilvirkum máta. Uppbygging fjarskipta kerfa má sjá á mynd 9 en uppbygging þess er að á sendingarstað eru upplýsingar sem þarf að koma frá punkti A (mörkin) til punkts B (gagnavers Mannvits). Þessar upplýsingar fara því í sendi (e. transmitter) en hann undirbýr upplýsingar með kóðun (e. encoding) eða mótun (e. modulation) og sendir þær út á flutningsrás (e. channel). Þegar skilaboðin komast til móttakarans (e. receiver) sem er að hlusta á rásina afkóðar (e. decode) eða afmótar hann þá skilaboðin og skilar því á áfangastað. Á leiðinni milli A og B geta truflanir á formi suðs (e.noise) eða bjögunar (e. distortion) komist inn í merkið, með þeim afleiðingum að upplýsingarnar skili sér ekki [20].



Mynd 9: Uppbygging fjarskiptakerfa [20].

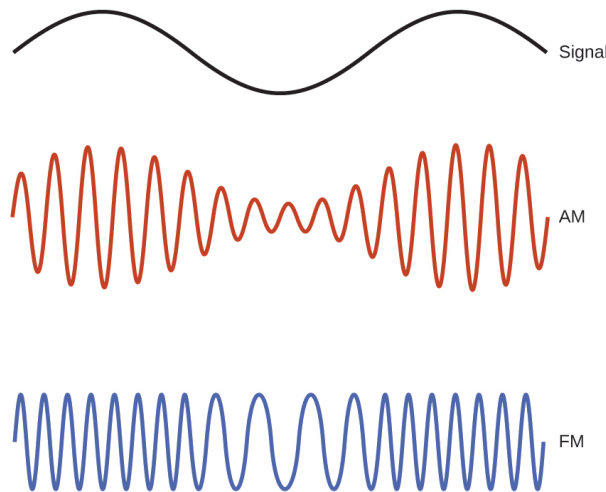
4.1. Mótun og kóðun

Mótun og kóðun í fjarskiptakerfum er leið til þess að þjappa upplýsingum við burðarbylgju, er það gert með því að breyta eiginleikum burðarbylgjunnar. En burðarbylgja er það sem er sent út til viðtakanda, eins og útvarpsbylgjur sem eru á sviðinu 3KHz til 300Gz. Dæmi á þessu sviði er WIFI á 2,4GHz, útvarp 88MHz - 104MHz og VHF talstöðvar á 30-299MHz. Munur á mótun og kóðun felst í því hvort að upplýsingar sem á að senda séu hliðrænar (e. analog) eða stafrænar (e. digital) upplýsingar. Þar sem mótun er notuð þegar merki er hliðrænt og kóðun ef merkið er stafrænt. Almennt er það því lægri sem tíðnin er, því lengra drifur það. Á móti kemur að á hærri tíðnum er hægt að koma hraðar upplýsingar til viðtakanda [21] [22] [23].

Hagkvæmmin í því að nota mótun er að hægt er að fara með upplýsingar lengri vegalengdir sem annars væri ekki hægt. Má nefna til dæmis talstöðvar, en erfitt er að reyna tala við einhvern sem stendur langt í burtu, er því hægt að nota til dæmis VHF talstöðvar þá er hægt að tala saman í meira en 50 km fjarlægð [23]. Dæmi um mótun:

- **FM mótun.** Þá er tíðni burðarbylgjunnar breytt í samræmi við merkið en styrkleiki er sá sami [21].
- **AM mótun.** Þá er tíðni á burðarbylgjunnar haldin sú sama en styrkleikanum er breytt í samræmi við merki [21].

Mynd 10 sýnir FM og AM mótun í mynd.

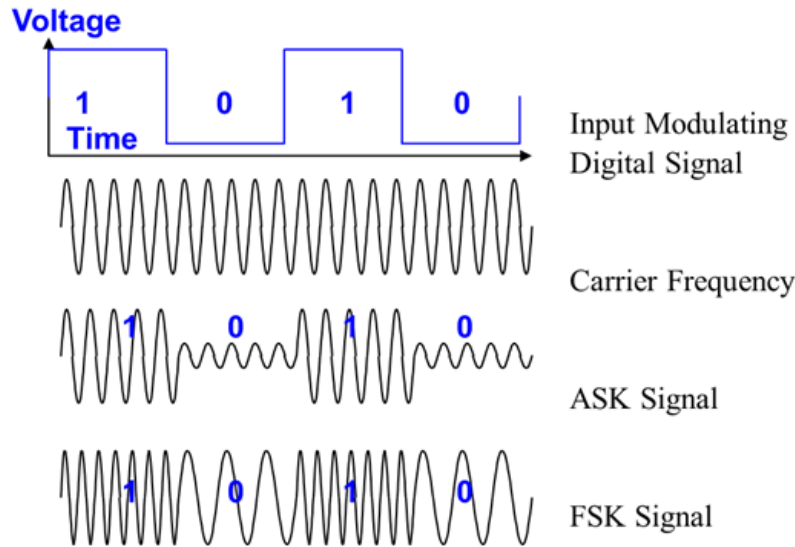


Mynd 10: Munur á milli AM og FM mótunar, burðarbylgjan er það sem er eins í FM og AM [24].

Á sama skapi er kóðun notuð til þess að upplýsingar komist lengri vegalengdir með því að breyta eiginleikum burðarbylgjunnar. Dæmi um kóðun:

- **Frequency-shift keying** eða FSK kóðun. Þar eru notaðar tvær tíðnir til að merkja hvaða stafræna merki er verið að senda [21].
- **Amplitude Shift Keying** eða ASK kóðun. Það virkar svipað eins og AM þar sem styrkur á til að tákna mismunandi merki [21].

Mynd 11 sýnir FSK og ASK kóðun í mynd.



Mynd 11: Munur á milli ASK og FSK [25].

4.2. Flutningsrás

Burðarbylgjan sem er búið að blanda upplýsingum við er síðan send út á rás eða á burðarmiðil. En burðarmiðill er það sem tengir sendi og móttakarann saman, þannig að samskipti geta verið á milli þeirra. Hægt er að skipta burðarmiðlinum upp í tvo flokka [26].

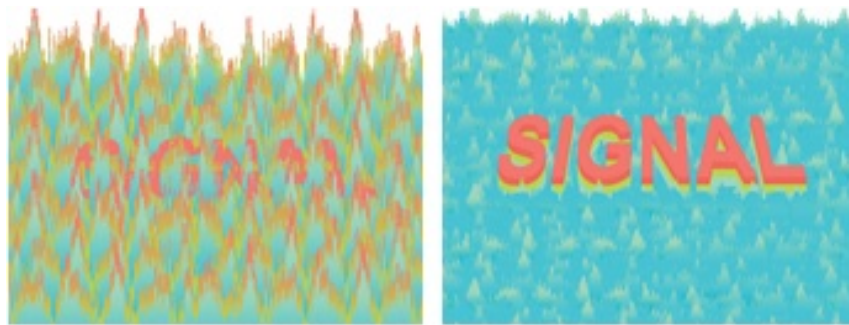
- **Práðbundin fjarskipti**, þar sem leiðari/þráður er lagður milli tveggja staða sem burðarmiðill. Til dæmis ljósleiðari, kóax kapall, kopar vír eða netstrengur [26].
- **Práðlaus samskipti**, þar sem loftið er notað sem burðarmiðill. Til dæmis Wifi, útvarp, símakerfið og talstöðvar [26].

Kostir og gallar eru við hvort tveggja en helst er að nefna í þráðlausum fjarskiptum er auðveldara að bæta við sendistöðum og móttökustöðum, þar sem loftið er notað til að tengja tæki saman. Aftur á móti hefur loftið takmarkandi flutningsgetu og ekki hægt að auka það, en hægt er að bæta endalaust af vírum til að auka flutningsmagn. Það er hins vegar tímafrekt og kostnaðarsamt, sérstaklega ef það þarf að fara langar vegalengdir [26].

4.3. Truflanir

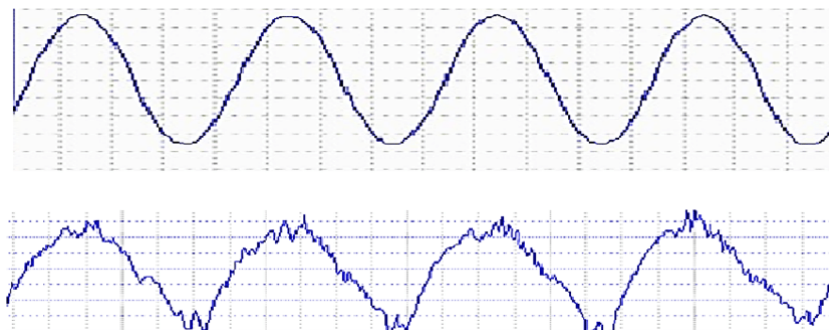
Allar þær truflanir sem verða á merkinu á meðan það er verið að senda það, hafa áhrif hvort allar upplýsingar skili sér alla leið. Tvær helstu truflanir eru **suð** og **bjögun** [27].

Suð er það þegar auka upplýsingar komast inn í merkið. Ef suðið verður of mikið getur það skyggt fyrir upplýsingarnar sem er verið að reyna senda, sem veldur því að viðtakandi áttar sig ekki á því hvað var sent. Má nefna til dæmis eins og í mikilli mannþröng þar sem er mikill kliður, getur reynst erfitt að heyra í félagamanns sem stendur við hliðina á sér vegna kliðurs. Má því flokka kliður sem suð [27]. Mynd 12 sýnir hvernig suð hefur áhrif á merki.



Mynd 12: Séð hvernig suð getur truflað, vinstri myndin sýnir mikið suð en nánast ekkert suð er á hægri mynd [28].

Bjögun er hins vegar þegar upplýsingarnar aflagast á einhvern hátt, til dæmis ef merki missir styrk eða tíðnibreyting verður á merkinu. Þetta veldur því að erfitt getur reynst að skilja þær upplýsingar sem koma til viðtakanda. Má hugsa til dæmis einhvern sem er á mútur tímabilinu en þá getur viðkomandi lent í því að rödd hans bjagast eitthvað og erfitt reynist að heyra hvað hann sagði [27]. Mynd 13 sýnir mismun á milli merki og merki sem hefur orðið fyrir bjögun.



Mynd 13: Hvernig bjögun hefur áhrif á merki, efri mynd sýnir upprunalega merkið og neðri mynd það bjagaða [29].

4.4. Fjarskiptalausnir sírita

Þar sem síritar eru oft staðsettir úti í mörkinni, langt frá öllum innviðum eru þráðbundin fjarskipti óhagkvæm. Því notast síritar mikið við þráðlaus samskipti. En að velja réttan fjarskiptamáta getur verið flókið þar sem til eru yfir 30 tegundir af þráðlausum fjarskiptamöguleikum [30]. Þeir fjarskiptamöguleikar sem geta hentað vel fyrir síritann sem á að hanna eru þráðlaus fjarskipti sem hafa drægni sem er meira en 5 km :

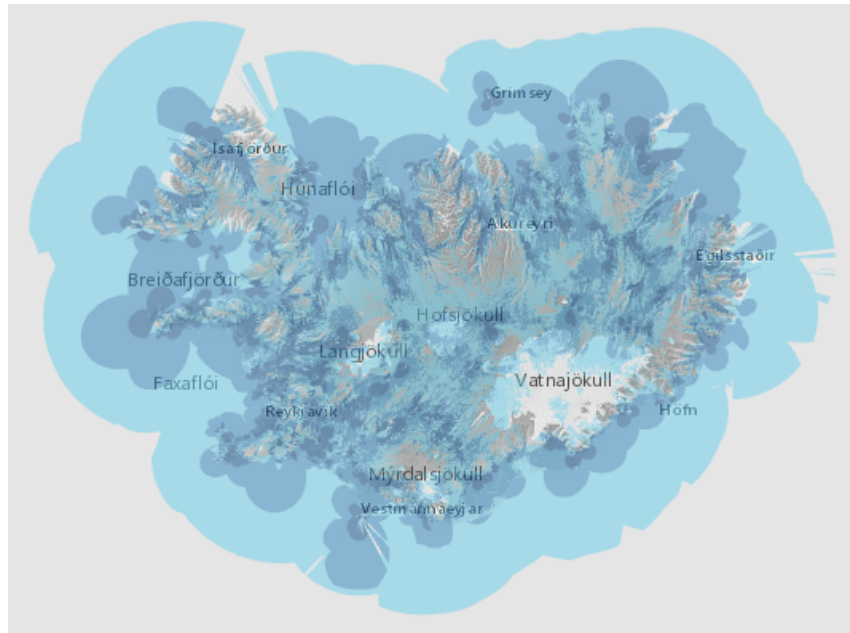
Dæmi um fjarskipti með langa drægni eru:

- **Símkerfi**
- **LoRaWAN**
- **SigFox**

Þegar valið er flutningskerfi þarf að huga að hentugleika og straumnotkun á kerfinu. Einnig þarf að hafa í huga að kerfið þarf að ráða við að senda skrá sem inniheldur klukkustundar löng gögn eða lengra. Skrá sem var gerð í tölvu með gervi gögnum fyrir tvo nema í klukkustund var um 500 bytes, fjarskiptalausnin þarf að geta minnsta kosti ráðið við það. Nánar er skoðað hvern og einn í köflum 4.4.1, 4.4.2 og 4.4.3 hér að neðan.

4.4.1. Símakerfið

Þegar það kemur að nota símakerfið til gagnaflutninga er hentugleiki í því þar sem miklir innviðir eru nú þegar til staðar, sem veldur því að mikil dreifing er á því um allt land. Því þarf síritinn einungis að vera með búnað sem getur tengst við símakerfið, til að geta flutt gögn á áfangastað. Mynd 14 sýnir hversu mikla dreifing bara símin hefur á Íslandi.



Mynd 14: Uppgefinn 4g dreifing Símans [31].

Það var árið 1993 sem fyrst var hægt að nota símkerfið til gagnaflytninga með komu 2g tækni oft kölluð GSM. Þó að fyrst um sinn var einungis hægt að senda SMS, en með komu 3g árið 2001 sem hægt var senda mun stærri skjöl, til dæmis tölvupóst. En með komu 4g árið 2009 var hægt að nota myndsímtöl til þess að tala saman þar sem hraðinn á gagnaflytningum var búinn að aukast það mikið. En munurinn var að 2g lofaði hraða upp á 0.065Mbps, 3g 8Mbps og 4g 50 Mbps [32].

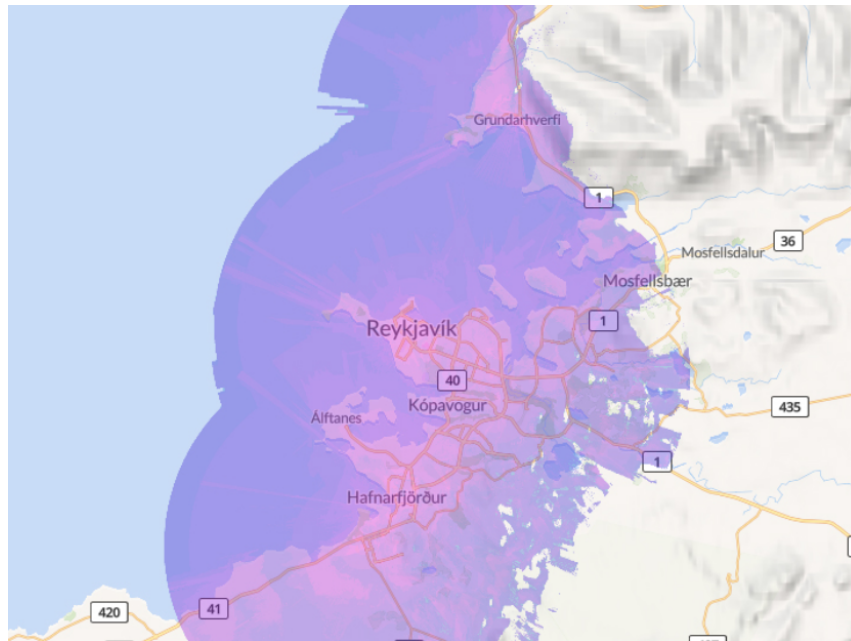
Ef ákveðið er að nota símkerfi þarf að hafa búnað sem notar 4g tæknina, þar sem það verður lagt niður 2g og 3g útsendingar á næstu árum hér innanlands [33]. Drægni á 4g tækni getur verið allt að 35 km í bestu aðstæðum [34].

Til þess að senda upplýsingarnar með 4g er oftast notaður búnaður sem tengir síritann við 4g kerfið svipað eins og 4g ráðar. Við að velja 4g verður ekkert takmark á því hversu stór skráin getur verið sem á að senda. Á móti kemur eyðir sá búnaður miklum straum, sem er óhentugt því síritar eru yfirleitt keyrðir á batteríum. En búnaðurinn getur verið að nota allt að 2500mA í sendingu þó að oftast er það í kringum 800mA [35].

4.4.2. LoRaWAN

LoRaWAN er fjarskiptakerfi sem er lágorku hádrægni kerfi. Er það mikið notast fyrir batterís nema sem eiga ekki að mæla mjög ört [36].

Lítið er verið að nota LoRaWAN á Íslandi en þó er einhver dreifing á því, en hún bundin við stór Höfuðborgarsvæðið. Til að geta notað LoRaWAN fyrir síritann þyrfti að setja upp innviði til þess að geta haft samband við búnaðinn. En búnaður sem notar LoRaWAN er gefinn upp að hafa drægni upp að 20 km við bestu aðstæður [37]. Mynd 15 sýnir uppgefna dreifingu sem Lýsir hefur.



Mynd 15: LoRaWAN dreifni frá Lýsi [38].

LoRaWAN hefur takmark á því hve stórar gagnasendingin getur verið í einni sendingu, en það er 243 bytes [37]. Fyrir stóra gagnasendingar þarf að senda marga pakka í einu með tilheyrandi straumnotkun en hver sending notar um 110 mA. Gagnahraði á LoRaWAN er hægur en hraðinn er upp að 27 kbit/s [39].

4.4.3. SigFox

SigFox er fjarskiptakerfi sem er lágorku hádrægni kerfi. Er það mikið notast fyrir batterísnema sem eiga ekki að mæla mjög ört [40].

SigFox er sú fjarskiptalausn sem er minnst notuð hérna á landi enda er enginn aðili að dreifa henni um landið, þannig til að nota SigFox þyrfti að fara í mikla innviða uppbyggingu. En búnaður sem notar SigFox getur haft drægni upp að 40 km við bestu aðstæður. SigFox hefur einnig takmörk á því hversu stór gagnasending getur verið við eina sendingu, en það er 12 bytes. Því þyrfti að senda margar sendingar í einu, með tilheyrandi straumkostnaði [37]. Ein sending á SigFox notar um 50 mA við hverja sendingu og er gagnahraði upp að 100 bit/s [39].

4.5. Valin fjarskiptalausn

Eins og sagt var í kafla 4.4 þá þarf fjarskiptalausnin sem valin er að geta sent gagnapakka langar vegalengdir, straumnotkun ekki of mikil og geta sent yfir um klukkustundar löng gögn. Einnig er gott að hafa í hug hverjir innviðir eru í landinu núna, þar sem það skapar þægindi að þurfa ekki að huga að innviðauppbyggingu.

Tafla 2: Eiginleikar fjarskiptalausna [35] [39] [41] [42] [43].

	Símkerfið	LoRaWAN	SigFox
Innviðir til staðar	Mikið	Lítið	Ekkert
Mesta fjarlægð	40 km	20 km	40 km
Mesta stærð á skrá	fræðilega óendaleg	222 bytes	12 bytes
Straumnotkun í sendingu	~ 800mA	~ 110mA	~ 50mA

Tafla 2 sýnir vel samanburð á þeim fjarskiptalausnum sem voru efnilegar í siritann. Símkkerfið var metið besti kostur þar sem mestir innviðir eru til staðar ásamt því að það getur sent klukkustunda löng gögn eða meira, án þess að þurfa brjóta þær upplýsingar niður í fleiri og minni sendingar. Vegur það þyngra heldur en straumnotkun þótt að símkkerfið kemur verst út þar.

Þarf því að finna einhvern búnað sem myndi þá nota símkerfið til að geta komið upplýsingum yfir á gagnaver Mannvits frá mörkinni. Búnaðurinn þarf annað hvort að geta verið stjórnað af Arduino, eða þá að geta lesið af skynjurunum og komið þeim á gagnaverið. En búnaður sem fannst sem hentug fjarskiptalausn á síritann var:

- **Sim7600CE Arduino skjöldur.** Það er búnaður sem hægt er að tengja við Arduino, sem hægt er að nota sem síma, GPS og tengjast símkerfinu til gagnaflytninga [44].
- **Arduino MKR NB 1500.** Það er Arduino tölva sem hefur innbyggðan búnað til þess að tengjast 4g símkerfinu [45].
- **nRF9160 DK.** Það er búnaður til þess að geta tengst 4g símkerfinu og lesið af skynjurum [46].

Þegar það kom að velja búnaðinn sem átti að nota, var gert mat á hversu auðvelt væri að nota búnaðinn. Færi það eftir reynslu á búnaði og hve miklar upplýsingar um búnaðinn væri hægt að finna á netinu. Einnig var lagt mat á hvort að búnaðurinn hefði einhvern aukabúnað sem gæti nýst vel í síritann.

Tafla 3: Samanburður á búnaði sem fjarskipta lausn [45] [46] [47].

	Opnar upplýsingar	Aukabúnaður	Straumnotkun
Sim7600CE	Mikið	SD, GPS	3.3 mA
Arduino MKR NB 1500	Mikið	Enginn	190 mA
nRF9160 DK	Lítið	GPS	.009 mA

Tafla 3 sýnir samanburð á þessum búnaði sem var skoðaður. Við samanburð á þessum þremur þá virðist Sim7600 henta best miðað við forsendurnar. Þar sem Sim7600 hefur auka möguleika sem geta nýst í síritann. En GPS eða Global Positioning System er hægt að nota til að leiðrétta rauntímaklukkuna í síritanum, þó að það sé mest notað til að staðsetja fólk eða búnað [48]. Sim7600 hefur líka þann eiginleika að geta lesið af SD korti þannig að hægt er að senda upplýsingar af því.

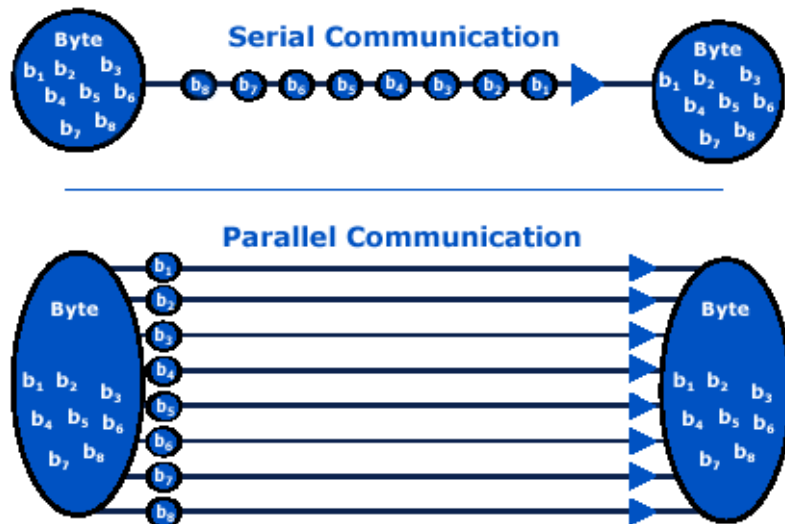
Sim7600 er módem búnaður til þess að geta tengst við símkerfið/Internetið hvort sem er með því að hringja eða senda upplýsingar í gegnum það. En módem er þá búnaður sem myndar tengingar fyrir Internetið á milli staða A og B. Á heimilum er módemíð það sem kemur með Internetið inn í hús og ráðerinn sér síðan um að tengja tækin á heimilinu við módem sem gerir það kleyft að tækin komist á Internetið [49].

5. Samskiptamátar raftækja

Til að geta komið upplýsingum á milli raftækja er notaðir staðlaðir samskiptamátar þannig að tækin geti haft samband við hvort annað, svipað eins og tungumál. Þessir samskiptamátar eru jafnólíkir eins og þeir eru margir [50].

Þegar það er verið að senda upplýsingar milli tækja er upplýsingum komið fyrir í röð sem eru annað hvort 0 eða 1 sem tækin geta þá skilið, oft talað um sem bitar (e. bits). En 8 bitar eru eitt bæti (e. byte). Hver biti stjórnast af því hver rökrásaspennan (e. Logic level) er, sem er yfirleitt skilgreind $V_{\text{logic level low}} = 0$ og $V_{\text{logic level high}} = 1$. Getur hún verið mishá á milli tækja, þarf því að athuga hvort að tækin hafi sama logic level áður en þau er tengd saman. Þar sem of hátt logic level getur skemmt tæki og of lág logic level getur misskilist. Til dæmis ef að tæki A notar 5V logic level en tæki B notar 3.3V. Ef tækin eru þá tengd saman getur A skemmt B, þar sem 5V er hærra en 3.3V. Tæki A getur lent í því að skilja ekki tæki B, þar sem A vill fá 5V sem 1 ekki 3.3V og misskilið það sem 0. En til þess að leysa það vandamál er hægt að setja rökrása spennubreyti (e. logic level shifter) á milli. Logic level shifter breytir spennunni á milli þannig að tæki A og B geta talað saman án vandræða [50].

Upplýsingar sem fara á milli tækja A og B er oftast bútað niður í eitt byte áður en það er sent eftir rásinni sem tækin eru tengd saman. Samskiptin sem tæki A og B geta notað má skipta upp í tvo megin flokka, sem eru annað hvort samsíða sendingar (e. parallel communication) eða raðsendingar (e. serial communication) samskipti. Munurinn á þeim er að í parallel eru notaðir átta víra til að senda eitt byte í einu en í serial samskiptum er einn vír og er byte-inu bútað niður í átta bita og er einn biti sendur í einu [50]. Mynd 16 sýnir munin á parallel og serial.



Mynd 16: Munur á serial og parallel samskiptun [51].

Kostir og gallar eru við bæði serial og parallel. Til dæmis má nefna að serial samskipti eru almennt mikið hægar en parallel þar sem í parallel er hægt að senda fleiri bita í einu. Á móti kemur að serial er yfirleitt mun hentugra þar sem það þarf oftast einn vír til að senda upplýsingar, en í parallel þarf marga víra til að senda fleiri upplýsingar í einu. Það veldur því að serial er yfirleitt mun ódýrara en parallel. Tafla 4 sýnir almennan samanburð á milli serial og parallel [52].

Tafla 4: Munur á serial og parallel [52].

	Serial	Parallel
Gagnahraði	Hægt	Hratt
Vírafjöldi	Einn	Margir
Verð	Látt	Hátt
Skalanleiki	Auðvelt	Erfitt

Oftast eru serial samskiptin notuð þar sem fjöldi pinna er takmarkaður á íhlutum og því hentar betur að nota serial. En sá búnaður sem notaður var við síritann var hannaður fyrir serial.

5.1. Serial samskipti

Til þess að serial samskipti geti virkað þarf að vera einhver leið til að greina í sundur hvenær hver og einn biti kemur. Er notast við tvær aðferðir til þess sem eru:

- **Samfasa** (e. **synchronous**).
- **Ósamfasa** (e. **asynchronous**).

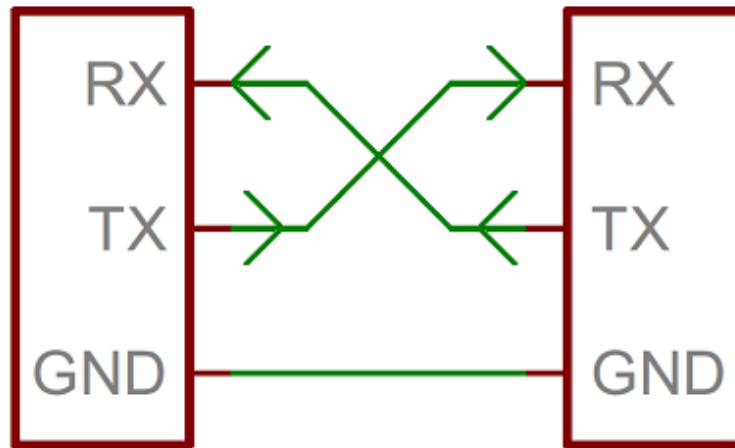
Helsti munurinn á þessum leiðum er að í samfasa er notaður klukkuvír til að greina bitanna á milli en í ósamfasa er fyrirfram ákveðnar reglur skilgreindar til þess að vita hvenær bitarnir koma. Hefur þetta allt sína kosti og galla en má nefna að samfasa eru hraðari en í samanburði við ósamfasa [53].

Þegar það er talað um serial samskipti er nær alltaf verið að tala um ósamfasa samskipti, en þegar það kemur að samfasa samskiptamátum fá þeir yfirleitt sér nafn þar sem þeir geta verið mjög mismunandi. Dæmi eru til dæmis I^2C (kafi 5.1.3) og SPI (kafi 5.1.4).

Þegar það kemur að ósamfasa samskiptum þurfa að vera reglur til staðar, þannig að tækin geti skilið hvort annað. Fyrirfram skilgreindar reglur eru:

- **Baud rate**, segir til hversu hratt á að senda gögnin [50].
- **Data bits**, er stærð gagnasendingar [50].
- **Synchronization bits**, er notað til að tryggja að tækin lesi bita á réttum tíma [50].
- **Parity bits**, mjög einföld villuathugun til að skoða hvort sending heppnaðist eða ekki [50].

Serial samskiptin nota alltaf tvo pinna sem heita receive eða **RX**, og transmit eða **TX**. Til þess að tæki A getur tala við tæki B með serial þarf að víxla tengingum á RX og TX á tækjum. Þannig að RX á tæki A að fara í TX á tæki B og öfugt. Sjá mynd 17. Einnig þarf að tryggja að tækin hafi sameiginlegan viðmiðunar punkt (GND) þar sem Logic level getur orðið mismunandi þó að tækin hafi sömu logic level en þau miða ekki við sama punkt[50].



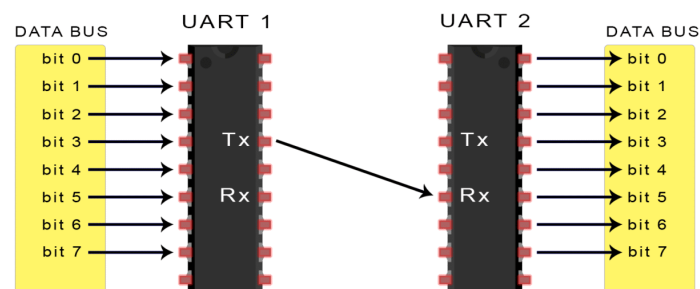
Mynd 17: Tenging á serial samskiptum [50].

Þessi samskipti geta síðan verið annað hvort einþátta sending (e. half-duplex) eða tvístefnu sending (e. full-duplex). Half-duplex samskipti er þegar tækin geta bara hlustað eða sent í einu, eins og talstöðvar. Full duplex er þegar tæki geta sent og hlustað á sama tíma, eins og sími [50].

5.1.1. Universal Asynchronous Receiver/Transmitter (UART)

Universal Asynchronous Receiver/Transmitter eða UART eru samskipti þegar parallel upplýsingum í tæki A er umbreytt yfir í serial upplýsingar til að geta sent á tæki B. Á stað A eru parallel upplýsingar til dæmis gildi mismunandi átta innganga sem þurfa að komast á stað B. Til að einfalda sendinguna er hentugt að umbreyta upplýsingum með UART. Þá sparast sex vólar sem annars þyrfti að leggja á milli. Á stað B er upplýsingunum aftur breytt í parallel með UART [54]. Sjá mynd 18.

UART er ósamfasa samskipti og notar því, eins og áður hefur verið nefnt, baud rate, data bits, synchronization bits og parity bits til að greina í sundur gagnasendingar [54].



Mynd 18: Myndræn útfærsla á UART [54].

5.1.2. RS485 og RS232

RS485 og RS232 eru vinsælir serial samskiptamátar sem hafa verið til í nær hálfa öld. Nota þeir svipaðar aðferðir til að senda upplýsingar á milli en munurinn á þessum samskiptamátum er [55]:

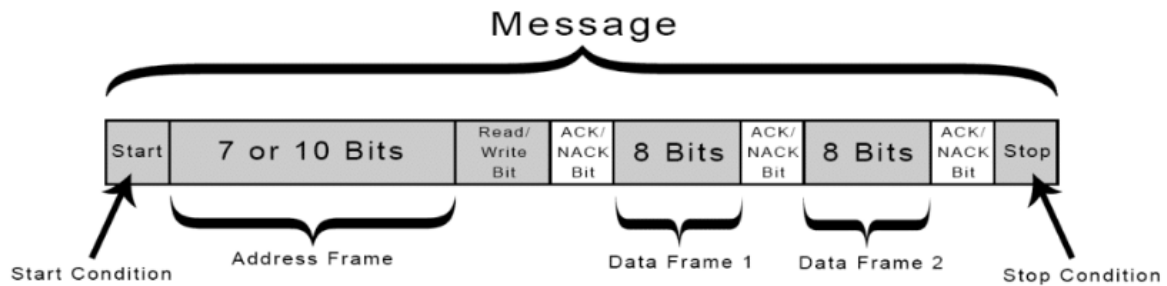
- **Lengsta mögulega vegalengd.** RS232 nær upp í 15 m en RS485 nær allt að 1200 m [55].
- **Vörn gegn truflunum.** RS232 er mjög veikt fyrir truflunum á meðan RS485 er það ekki [55].
- **Mögulegur fjöldi tækja á samskiptalínu.** RS232 getur einungis verið á milli tveggja tækja en RS485 ræður við allt upp að 30 tæki séu tengd saman [55].
- **Gagnahraði.** RS232 ræður við upp að 1 Mbit/s en RS485 upp að 10 Mbit/s [55].

Þó að þessir samskiptamátar séu mjög líkir er ekki hægt að tengja þá sama nema hafa breyttu á milli [55].

5.1.3. Inter-Integrated Circuit (I2C)

Inter-Integrated Circuit eða I^2C er samfasa samskiptamáti sem notar einungis tvo víra til að geta talað á milli. Hafa þessir vírar fengið nöfnin Serial Data eða **SDA**, og Serial Clock eða **SCL**. En þá eru gögnin send eftir SDA og klukkupúlsinn er á SCL. Hægt er að tengja allt að 1008 tæki við þetta kerfi, þar sem hvert og eitt tæki fær sitt vistfang (e. address). Virka þessi samskipti þannig að það er einn stjórnandi (e. Controller) og restin er jaðartæki (e. Peripherals) [56].

Virka samskiptin þannig að stjórnandi byrjar að senda út skilaboð um að hann sé að fara að byrja samskipti, með því að byrja klukkuna á SCL. Eftir það sendir hann út vistfangið á SDA og hvort hann sé að fara að skrifa eða lesa af tækinu. Það jaðartæki sem á vistfangið svarar með staðfestingar (e. acknowledge) bita eða ACK til að láta stjórnandann vita að jaðartækið sé viðstatt. Eftir það sendir stjórnandi hvað hann vill og jaðartækið svarar. En upplýsingarnar eru alltaf bútaðar niður í 8 bita og ACK bita er sendur á milli til þess að tryggja að allar upplýsingarnar hafi skilað sér [56]. Á mynd 19 sést hvernig sendingu er bútað niður.

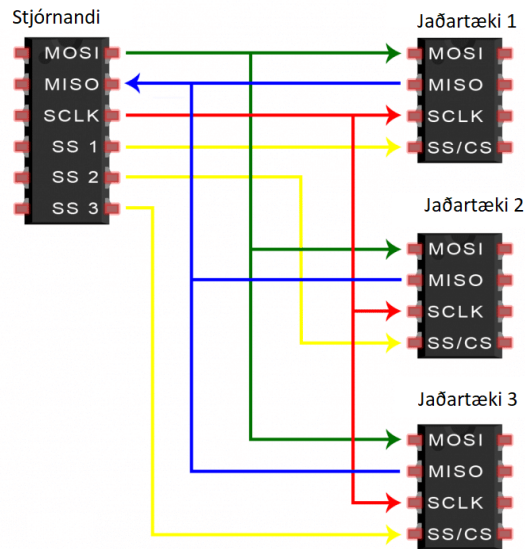
Mynd 19: *I²C* sending [56].

5.1.4. Serial Peripheral Interface (SPI)

Serial Peripheral Interface eða SPI er samfasa samskiptamáti sem notar 4 víra til að eiga samband á milli tækja. Hafa þessir víra fengið nöfnin: [54]:

- **MOSI**, það er að segja stjórnandi út og jaðartæki inn.
- **MISO**, það er að segja stjórnandi inn og jaðartæki út.
- **SCLK**, sem er serial klukkan.
- **SS/CS**, það er að segja pinni til að velja jaðartæki.

Þessi samskiptamáti virkar þannig að stjórnandi notar SS/CS pinnann til virkja það jaðartæki sem stjórnandinn vill tala við. Eftir að það eru upplýsingar frá stjórnandanum í jaðartækið með MOSI línunni og ef að jaðartækið þarf að svara notar hann MISO línu. Þetta veldur því að SPI er full duplex þar sem stjórnandi getur verið að senda upplýsingar á sama tíma og hann er að fá. Dæmigerð tengimynd má sjá á mynd 20 [54].



Mynd 20: Tengimynd af SPI [54].

Það sem gerir þennan samskiptamáta enn sérstæðan samskiptamáta miðað við aðra sem hafa komið fram, þá koma gagna sendingar ekki í bútum heldur er allt sent í einu. Þetta veldur því að SPI er næstum því tvöfalt hraðari en I^2C . Aftur á móti hefur SPI þann ókost að tækin vita ekki hvort að sendingar heppnuðust eða ekki. Þar sem ekkert er sent til að athuga hvort sending heppnaðist eða ekki [54].

5.2. Modbus

Modbus er samskiptamáti sem notar RS485 yfir UART til þess að tengja iðntölvur, skynjara, loka eða mótorá saman. Beluga hefur Modbus samskipta möguleika. Modbus-inn er byggður svipað og I^2C þar sem það er einn stjórnandi (e. Controller) og mörg jaðartæki (e. Peripherals), sem öll hafa sitt hvort vistfangið (e. address). Virkar það þannig stjórnandi sendir út á samskiptalínuna hvað hann vill fá og frá hvaða vistfangi og tæki með vistfangið svarar [57].

Upplýsingarnar í tækjunum eru geymdar í fjórum mismunandi töflum. En til að geyma af/á stöður eru notaðar tvær töflur sem kallast spólur (e. coils) og skiptast spólutöflurnar upp í inn- og útganga. Töluleg gildi eru geymd í hinum tveimur töflunum sem kallast gisti (e. register), register töflurnar eru einnig skipt upp í inn- og útganga. Til þess að lesa eða skrifa inn í töflurnar þarf að nota vistfang (e. address) til þess að geta farið beint í rétta gildið. Í hverju töflu fyrir sig getur verið 9998 gildi. En tafla 5 sýnir upplýsingatöfluna sem Modbus hefur um að ráða [58].

Tafla 5: Sýnir þær töflur sem geyma upplýsingarnar sem tækin geyma [58].

Coil/Register Numbers	Data Addresses	Type	Table Name
1-9999	0000 to 270E	Read-Write	Discrete Output Coils
10001-19999	0000 to 270E	Read-Only	Discrete Input Contacts
30001-39999	0000 to 270E	Read-Only	Analog Input Registers
40001-49999	0000 to 270E	Read-Write	Analog Output Holding Registers

Til þess að geta nálgast upplýsingarnar er send út skipun (e. function code) sem segir til um hvort það eigi að lesa eða skrifa í minnistöflunar en tafla 6 sýnir allar þær skipanir sem Modbus hefur að geyma [58].

Tafla 6: Fuction codes í Modbus samskiptum [58].

Function Code	Action	Table Name
01	Read	Discrete Output Coils
05	Write single	Discrete Output Coil
15	Write multiple	Discrete Output Coils
02	Read	Discrete Input Contacts
04	Read	Analog Input Registers
03	Read	Analog Output Holding Registers
06	Write	Analog Output Holding Register
16	Write multiple	Analog Output Holding Registers

5.3. AT skipanir

AT skipanir (e. AT commands) eru mikið notaðar til að stjórna módem (e. modem), en Sim7600 nota AT skipanir. En AT er stytting á attention, eða athygli sem er send á modem ásamt skipun, sem segir hvað modem á að gera. Við hverja skipun sem er framkvæmd kemur alltaf svar til baka en fer það eftir því hvernig skipun var send og hvaða svar kemur. Til eru fjórir megin flokkar af AT skipunum [59]:

- Skipanir til athugunar, notað til að skoða hvaða stillingar eru leyfilegar. Uppsetning `AT+<skipun>=?`
Dæmi: `AT+CREG=?` Svar: `+CREG:(0-2)`, sem þýðir hægt er að nota 0,1 eða 2 í gildi [59].
- Skipanir til að lesa núverandi gildi úr módem. Uppsetning er `AT+<skipun>?`
Dæmi: `AT+CREG?` Svar: `+CREG: 1,1` sem þýðir að núverandi módem hefur þessar stillingar [59].
- Skipanir til að skrifa í módem, notað til að breyta núverandi skipunum. Uppsetning skipunar er `AT+<skipun>=<gildi>`
Dæmi: `AT+CREG=1` Svar: í flestum til fellum einungis OK en stundum kemur nánara svar hvort skipun hefur heppnast. En OK þýðir að skipun er í lagi [59].
- Skipanir til að framkvæma. Uppsetning `AT+<skipun>`
Dæmi `AT+CFTPSPUT="test.csv"` Svar: í flestum tilfellum einungis OK en stundum kemur nánara svar hvort skipun hefur heppnast. En OK þýðir að skipun er í lagi [59].

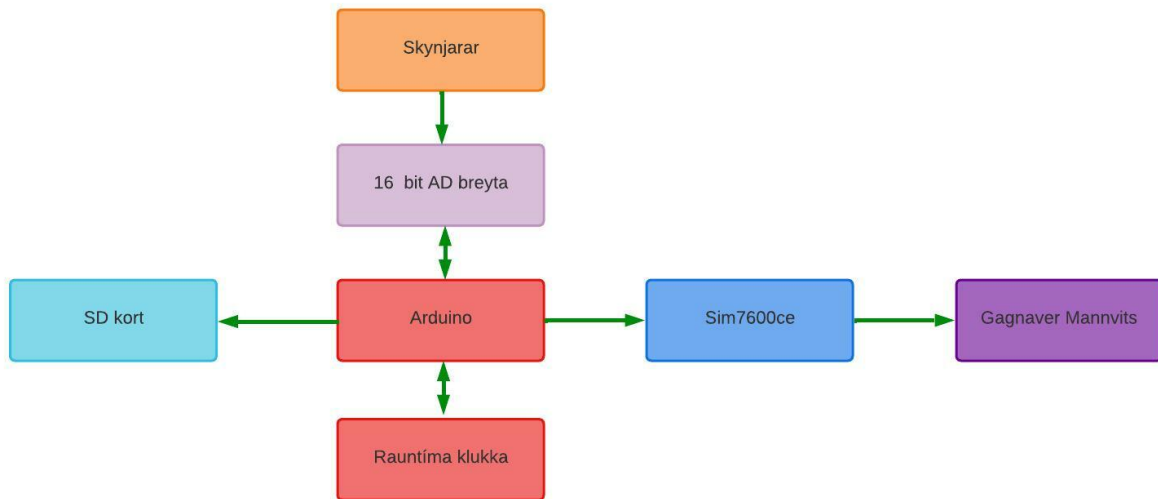
AT skipanir eru staðlar á milli módem í gegnum staðalinn **3GPP TS 27.005** og **3GPP TS 27.007**, en ekki öll módem hafa allar AT skipanir sem til eru því þarf að skoða í hverju tilfelli hvaða skipanir eru í boði í hvert sinn [59]. En allar skipanir fyrir Sim7600 má finna í skipana leiðbeiningunum (e. command manual) frá framleiðanda í heimild [60].

6. Samsetning síritans

Við smíðina á frumgerðinni á síritanum var notað:

- **Arduino uno R3**, heilinn í síritanum [61].
- **16bit AD breyta**, notað til að geta fengið nógu nákvæm gildi frá skynjurunum [62].
- **Sim7600CE skjöldur**, notað til að senda gögn á gagnaver Mannvits [44].
- **SD kort**, notað til gagnageymslu þar sem bæði Sim7600 og Arduino geta lesið af SD korti.
- **Logic level shifter**, þar sem logic spenna er mishá á milli Arduino, Sim7600 og SD korts.
- **DS3231 rauntíma klukka**, til að fá rauntímaklukku [63].
- **Skynjararnir**, Vegapuls C1 og Beluga frá Flowtronics.

Blokkmynd af síritanum sem var smíðaður sést á mynd 21. En rásamynd af sírita má finna í viðauka A.



Mynd 21: Blokkmynd af síritanum.

6.1. Tengjast við skynjara

Til að geta lesið gildið sem skynjararnir (sjá kafla 2.2) mæla þarf að skoða hvernig merki þeir senda út miðað við mæld gildi. Tafla 7 sýnir með hvaða hætti er hægt að fá mæligildið sem skynjarar mæla.

Tafla 7: Hvaða merki skynjarar gefa frá sér og hvaða spennu þeir nota [3] [5].

	Vinnuspenna	Útmerki
C21 skynjari	12-36Vdc	4-20 mA merki
Flow-tronic	4.8-24VDC	4-20 mA merki/Modbus

Vegapuls C21 er einungis með 4-20 mA sem notað var til að geta fundið hæðina. Beluga skynjarinn var með tvo möguleika á útgangsmarki Modbus og 4-20 mA, reynt var að nota modbus en gekk það illa (sjá kafla 7.2.3). Var því notað 4-20 mA merkið frá Beluga. Kostur hefði verið að nota Modbus merkið þar sem fleiri gildi frá skynjara, en sjá í heimild [5] hvaða gildi er hægt að fá aukalega.

Til þess að geta fengið út mæld gildi frá skynjurum út frá straumnum, þarf að reikna það út með formúlu 5.

$$P_v = \frac{\Delta P_v}{\Delta I} \cdot (I - I_{low}) + P_{v_{low}} \quad (5)$$

P_v [Mæligildi skynjarar] = Reiknaða gildið.

ΔP_v [Mæligildi skynjarar] = Mælisviðs skynjara.

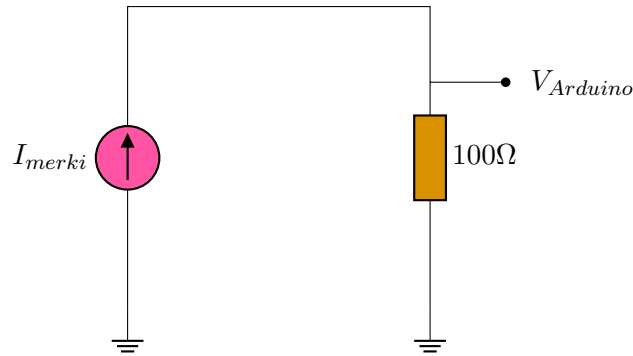
ΔI [A] = Mismunur á stærsta og lægsta straum.

I [A] = Mældur straumur.

I_{low} [A] = Lægri mörk straums.

$P_{v_{low}}$ [Mæligildi skynjarar] = Lægri mörk mælisviðs.

Arduino getur einungis átt við spennumerki, því þarf að breyta 4-20 mA merkinu í spennumerki. Var það gert með því að setja viðnám út frá straummerkinu þannig að Arduino gæti lesið merkið.



Mynd 22: Hvernig er hægt að lesa spennumerki út frá straummerki.

100 Ω viðnám var notað þar sem það var til á lager hjá Mannvit. En það viðnám er sérstaklega hannað fyrir sírita til að breyta 4-20 mA í spennumerki, þar sem hitastuðull (e. temperature coefficient) er minni en á venjulegum viðnámmum. En þetta viðnám er með hitastuðul upp á 0.8 ppm per gráðu en eðlilegt er að viðnám séu með 20 ppm per gráðu [64] [65]. En það sem hitastuðullin segir til um er hversu mikið viðnámið breytist við hita en því lægri sem stuðull er því minna breytist viðnám við hitabreytingu.

Til að reikna spennu út frá straumi getur maður notað jöfnu Ohms

$$V = I \cdot R \quad (6)$$

Breytist því jafna 5 frá því að straumur er notaður í spennu.

$$Pv = \frac{\Delta Pv}{\Delta V} \cdot (V - V_{low}) + Pv_{low} \quad (7)$$

Pv [Mæligildi skynjarar] = Reiknaða gildi.

ΔPv [Mæligildi skynjarar] = Mælisviðs skynjara.

ΔV [V] = Mismunur á stærstu og lægstu spennu.

V [V] = Mæld spenna.

V_{low} [V] = lægri mörk spennu.

Pv_{low} [Mæligildi skynjarar] = Lægri mörk mælisviðs.

Hentugt reyndist að báðir skynjaranir geta verið á 12VDC þar sem flest batterí sem Mannvit notar er 12VDC, og var því ákveðið að nota 12VDC til að keyra skynjaranna.

Arduino er með 10 bit ADC breytu. Var kannað hvort sú ADC breyta skili nægilega góðri upplausn frá skynjurunum, eða hvort að það þurfi að finna aðra ADC breytu með hærri upplausn. Til að geta reiknað út nákvæmnina er notuð formúla 8.

$$\text{Nákvæmni} = \frac{V_{ref} \cdot \text{Mælisvið}}{2^n \cdot \Delta V} \quad (8)$$

Nákvæmni [Mælisvið/bit] = Nákvæmni ADC breytunnar

$V_{ref}[V]$ = Hæsta spenna sem ADC breyta ræður við.

Mælisvið [Mæligildi skynjarar] = Mælisvið sem skynjari er gefinn upp með.

N = Bitastærð ADC.

$\Delta V[V]$ = Spennumunur á merki.

Tafla 8 sýnir hvers vegna farið var í að nota stærri ADC breytu, þar sést vel að ADC breytan á Arduino nær ekki að ráða við að lesa uppgefna nákvæmni frá skynjurunum.

Tafla 8: Mismunur á nákvæmni á milli 10 bit og 16 ADC [3] [5] .

	Mælisvið skynjara	10 bit AD breyta	16 bit AD breyta	Uppgefin skekkja skynjara
Vegapuls C21	0 til 15 m	± 46 mm	± 0.72 mm	± 2 mm
Beluga	-2 m/s til 6 m/s	± 0.02 m/s	± 0.0002 m/s	± 0.01 m/s

Notað var **ADS115** 16 bit ADC breytuna, vegna þess að hægt var að komast fljótt í hana. Hún hefur einnig þann kost að vera með fjóra innganga þannig hægt væri að tengja fjóra skynjara við hana. Til að geta náð gildum frá ADC breytunni er notað I^2C samskiptamátann [62].

6.2. Koma gögnum á gagnaver Mannvits

Til að geta stjórnað Sim7600 er sendur AT skipanir í gegnum serial samskiptamátann. En Sim7600 bauð upp á nokkra möguleika til að senda gögnin.

- **Secure File Transfer Protocol**, eða FTPS. Sem er staðall um hvernig hægt er að flytja skrá í gegnum Internetið [66].
- **MQ Telemetry Transport**, eða MQTT. Sem er staðall hvernig tæki geta sent upplýsingar á Internetið, þar sem tæki sendir rauntímamælingar beint í gagnaver [67].
- **Secure Soceks layer**, eða SSL. Sem er staðall sem býr til tengingu á milli tveggja staða til að geta sent gögn á milli þeirra [68].

Notast var við FTPS þar sem Mannvit er kunnugt notkun á FTPS. Hægt er að nota FTPS á Sim7600 á tvo vegu:

- Láta Sim7600 senda skrá sem er lesin af SD korti.
- Láta Sim7600 búa til skrá á FTPS svæði með upplýsingum sem koma í gegnum serial samskipti.

Í upphafi var áætlað að Arduino skrái á eitt SD kort og Sim7600 lesi skrána og sendir hana í gagnaver Mannvits. Ekki náðist að láta Arduino og Sim7600 lesa af sama SD korti. Þar af leiðandi var fjarskiptalausnin að Arduino skrifar á SD kortið og á klukkustunda fresti sendir Arduino gögn seinustu klukkustundar á Sim7600 sem sendir þau til gagnavers Mannvits. Þannig var hægt að fá gögnin á klukkustundar fresti og ef eitthvað færi úrskaiðis þá væru gögnin til á SD kortinu.

6.3. Skrifa á SD kort

Til að geta skrifað á SD kort með Arduino er notast við samskiptamátann SPI. En tryggja þarf að gögn tapist ekki ef að tæki verður straumlaust. Þannig þegar tækið fær aftur straum eftir straumleysi haldi það áfram að skrifa gögn.

Sniðmát (e. Format) á skránni er comma separated value eða **CSV** skrá. Það er gerð skráar sem notar kommu eða stafi til að aðskilja á milli gilda. Er það hentugt þar sem hægt er með einföldum hætti að vinna með gögnin. Titill var notaður til að vita uppruna mælingarstaðs.

Hægt er að skrá annað hvort mæligildin frá skynjurum eða reiknaða flæðið í lögnum. Þau gögn sem eru skrifuð á SD kort voru mæligildin frá skynjurum því ef villa kemur upp í reikningi á flæði er erfitt að leiðrétta ef einungis er skráð niður reiknað flæði.

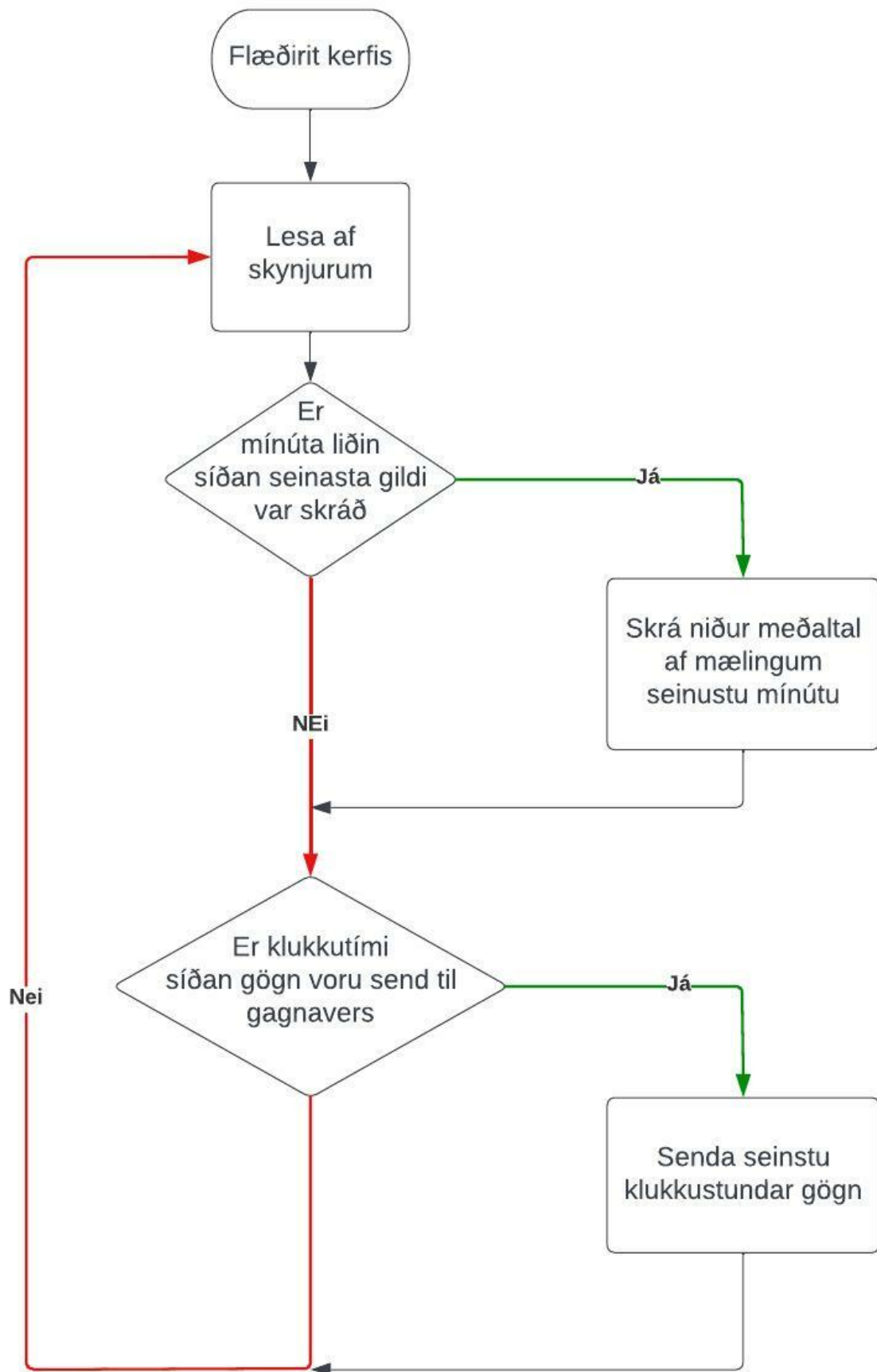
6.4. Forritun

Þegar Arduino var forrituð var notað forritunarmálið Arduino program language. Arduino forritunarmál sem er byggt ofan C++ og C. Er það gert til þess að hafa forritunarmálið auðlesanlegra [69]. Til þess að skrifa á Arduino þarf ná í Arduino IDE, en þar er skrifað forritið ásamt því að setja forritunarkóðann á Arduino-inn. En Arduino IDE breytir forritunarkóðanum í assembly kóða, og skrifar það á Arduino [69]. En assembly kóði er það tungumál sem flestir örgjörvar nota [70].

Þegar kemur að því að forrita fyrir síritann er heildarmyndin á forritinu fremur einföld þar sem forritinu er hægt að skipta upp í þrjár megin aðgerðir, þær eru:

- Lesa af skynjurum.
- Skrá niður mælingar á SD kort.
- Senda gögn í gagnaver.

Forritið les af skynjurunum á 5 sekúnda fresti. Reiknar Arduino meðaltal frá seinustu 12 mælingum, og skrifar það á SD kort. Á klukkustunda fresti sendir Arduino út skipun á Sim7600 um að senda seinustu klukkustund af gögnum í gagnaver. Flæðirit af sírita má sjá á mynd 23, ásamt því að finna forritið í viðauka B.



Mynd 23: Flæðirit forrits í síritanum.

7. Samantekt

Tókst að smíða frumgerð af sírita sem gat síritað mínútumédaltal frá skynjunum tveim, Beluga frá Flow-Tronics og C21 frá Vegapuls. Síritinn var látinn keyra í rúmar tvær klukkustundir hjá Mannviti að mæla í tilraunar pípu fyrir flæðinema. Skiluðust bæði gögnin á FTPS svæði Mannvits ásamt því að skrifa þau á SD kortið. Einnig sást að meira af gögnum voru komin á SD kortið heldur en FTPS svæðið, þar sem ekki var kominn tími á að senda fleiri gögn þangað. Mynd 24 sýnir uppsetningu mælingar.



Mynd 24: Uppsetning prófun á frumgerðar.

Í prófun á síritanum var ákveðið að stilla hann frekar á að senda á 10 mínútu fresti, fremur en á klukkustundar fresti á FTPS svæði Mannvits. Var það gert til að fá fleiri sendingar til gagnavers Mannvits á skemmri tíma til að sjá virknina betur.

Samkvæmt frumprófun náðist að uppfylla allar uppgefnar hönnunarforsendur. Frumgerðin er þó ekki tilbúin til þess að mæla og skrá úti á mörkinni en til þess krefst viðbótar hönnunar sem yrði annað tímafrekt verkefni. Þær hugmyndir af viðbótum má sjá í kafla 7.4. En vegna ýmissa vandræða við smíðina var ekki farið í það fyrir þetta verkefni. Helstu vandræði má sjá í kafla 7.2.

7.1. Verðsamanburður

Þegar verð er borinn saman við núverandi sírita Mannvits og þess sem smíðaður var er ekki tekið inn verð á skynjurum, þar sem skynjarar eru þeir sömu.

Tafla 9: Verð í frumgerð síritans [44] [71] [72] [62].

	USD	ISK
Arduino uno R3	\$27.60	3,892 kr
Sim7600CE	\$70.99	10,010 kr
SD breakout bord	\$3.50	494 kr
ADS115	\$14.95	2,108 kr
Samtals	\$117.04	16,503 kr

Í töflu 9 er miðast við að 1 USD sé 141 ISK þann 25.11.2022 [9].

Tafla 9 sýnir að með því að nota Arduino verður síritinn um 9 til 20 sinnum ódýrari en þeir síritar sem eru í notkun. Einnig má nefna að verð núverandi sírita í töflu 1 hafa ekki fjarskiptalaun innifalda þannig að raunverulegur verðmunur er meiri.

7.2. Vandamál sem komu upp við smíðina

Gekk þetta verkefni ekki áreynslulaust fyrir sig, heldur komu upp mörg vandamál sem þurfti að leysa. Sum af þessum vandamálum leystust en sum vandamálin eru ennþá til staðar. Þó virkar frumgerðinn eins og ætlast var til af henni. Við áframhaldandi hönnun þarf að leysa nokkur vandamál til frumgerðin verði fullhönnuð til að notkunar. Stærstu vandamálin voru þau sem tengdust:

- **SD kort**, nánar í kafla 7.2.1.
- **Sim7600**, nánar í kafla 7.2.2.
- **Modbus**, nánar í kafla 7.2.3.

7.2.1. SD kort

Eitt af þeim vandræðum var að illa gekk í fyrstu að skrifa á SD kortið. En til að skrifa á það þarf að hafa logic level shifter á milli SD korts og Arduino, þar sem SD notar 3.3 VDC en Arduino 5 VDC logic level. Við fyrri uppsetningu með Arduino logic level shifter sem var notað til að hafa samband við Sim7600 gekk allt vel. En þegar átti að skrifa á SD kortið gerðist ekkert. Var því miklum tíma eytt í að reyna að skilja forritið sem skrifaði á SD kortið og reynt að finna villur í því, en ekkert athugavert fannst þar. Við prófun á öðrum búnaði kom í ljós að Arduino gaf 0 VDC en ekki 3.3 VDC eins og hann áti að gera. Var því prófaður annar Arduino til þess að tengja við SD kortið. Til mikillar gleði virkaði þá stax að lesa og skrifa á SD kortið, enda voru 3.3VDC í lagi í þeim Arduino.

7.2.2. Sim7600

Dæmi um vandamál sem eru enn til staðar þótt frumgerðin virki eru vandamál sem tengjast Sim7600. Annað þeirra tengist að senda skipanir og hitt sem tengist við notkun FTPS.

Vandamálið við að senda skipanir á Sim7600 er að stundum hættir hann bara að samþykkja skipanir. Suma daga virkar að senda skipanir, en síðan eins og þruma úr heiðskíru lofti, hættir hann að taka við skipunum. Lagast það ekki nema aftengja straum frá Sim7600 og bíða í smá tíma, en eftir það virkar aftur að senda skipanir á Sim7600. Þetta er allt í lagi í frumgerð en ef það ætti að fara að hanna þetta til að vera út í mörkinni þarf meiri áreiðanleika. Mögulegt er að vandamálið liggi í því að sim7600 festist í skipun, það er að segja að bíða eftir tilviki sem kemur aldrei. Því virkar að taka strauminn af, þá gleymir hann því sem hann var að bíða eftir.

Hitt vandamálið sem tengist Sim760 er að stundum þegar það skráir sig inná FTPS server kemur upp villuboð "peer closed" frá Sim7600 eftir nokkrar sekúndur. En þessi villuboð segir til um að Sim7600 hefur verið skráður út af FTPS svæðinu. Til þess að komast fram hjá þessum villuboðum er hægt að flýta sér að vera á undan villuboðunum sem virkar í nær öll skipi. Prófað hefur verið að tala við tölvusvið Mannvits um ástæðu þess að vera skráð út FTPS af svæðinu en þeir segja að það sé ekkert hjá þeim sem eigi að skrá út Sim7600. Því er þetta vandamál ennþá óleyst.

7.2.3. Modbus

Vandamál var að ná að tengjast við Beluga skynjarann með Modbus. Arduino notar RS232 en Modbus er með RS485, var því sett breyta á milli sem virkaði þó ekki. En þegar skynjari var tengdur í gegnum hugbúnað frá tölvu sem sendi skipanir á Arduino fékkst svar í tölvu. Það er að segja Tölva->Arduino->RS232 til RS485 breyta->skynjari, því virðist breytan eða tengingar ekki vera vandamálið. Grunur liggur að vandamálið sé í kóðanum sem skrifað er í Arduino. Það gæti verið að Arduino sendi ekki rétta skipun á skynjarann og/eða bíði ekki eftir svári frá honum.

Vandamálið getur einnig leynst í því að 3.3 VDC voru ónýtt á Arduino, þegar tilraunir við að reyna tengjast við Beluga með Modbus. En íhluturinn sem býr til 3.3 VDC inn í Arduino er sami íhlutur og sér um RS232 samskiptin. Kom það ekki í ljós fyrr en seinna að 3.3 VDC voru ónýtt og á eftir að prófa með annari Arduino tölvu.

7.3. Straumnotkun búnaðarins

Skoða þarf straumnotkun á búnaði heildarlausnar til að áætla endanlega nýtingu á batteríi. Í töflu 10 má sjá uppgefna straumnotkun á þeim búnaði sem var notaður.

Tafla 10: Straumnotkun búnaðar [17] [47] [63] [73] [74]

	Straumur [mA]	Spenna [V]
Arduino	12-94	5-12
Sim7600	2.3-3.3	3,4-4,2
SD kort	0,75-30	3,3
16 bit ADC	0,15	2-5
Logic Level Shifter	NaN	1,8-10
DS3221	0,0005	2,3-5,5

Þar sem stendur NaN er ekki gefið upp frá framleiðanda hver straumnotkun á búnaði er. Ekki er hægt að áætla batterís endingu þar sem tölur eru ekki gefnar í tíma, en þetta gefur hugmynd af straumnotkun. Til að vita nákæma straumnotkun þarf að gera straummælingu af búnaði í keyrslu, sem var ekki gert í þessu verkefni.

7.4. Framtíðar viðbætur

Frumgerð síritans sýndi fram á að hægt sé að smíða ódýrari sírita en Mannvit er að nota núna. Þessi síriti er samt sem áður ekki hæfur til notkunar ennþá og þarf að fara í viðbætur áður en hann getur verið tekin í notkun. Hugmyndir að viðbættum sem þyrfti að fara í eru:

- Nota GPS sem leiðréttingu rauntíma klukku.
- Geta látið Sim7600 lesa af SD korti sem Arduino skrifar í.
- Slökkva á Sim7600 þegar ekki er verið að senda gögn.
- Laga vandmálin með Sim7600, sjá kafla 7.2.2.
- Nota Modbus til að fá gildin frá Beluga nema Flow-tronics.
- Hanna prentplötu sem allir íhlutir mundu raðast á.
- Hanna vatnspéttan kassa fyrir sírita.
- Skipta Arduino út fyrir aðra örtölvu sem getur keyrt margar skipanir í einu. En Arduino geta einungis gert eina skipun í einu þannig þegar það er verið að senda gögn til gagnavers þá hættir Arduino að lesa af skynjurunum. Það getur orðið vandamál ef Sim7600 festist í að senda gögn.

8. Lokaorð

Hannaður var síriti fyrir hlutfylltar lagnir eins og skolplagnir, hann mældi á fimm sekúndna fresti og skráði niður meðaltal mælinga á hverri mínútu. Á klukkustundarfresti notaðist síritinn við FTTPS tækni til þess að senda gögn til gagnavers Mannvits.

Þegar flæði í skolplögnum eru mæld þarf að mæla bæði hraða og hæð vökvans í lögnunum. Þetta er mikilvægt þar sem vatnshæð getur verið breytileg eftir tíma. Þessar mælingar voru gerðar með skynjara frá Vegapuls C1 og Belgua frá Flow-tronics.

Farið var yfir fræði við hönnun sírita ásamt því að skoða möguleika á nýrri lausn. Til þess að finna bestu lausnina var gerður samanburður á búnað til sírita, flutningi gagna o.s.frv. Þá var borið saman Arduino og Rasper PI til að mæla og skrá gögn en átti Arduino vinninginn. Jafnframt var borið saman símakerfið, LoRaWAN og SigFox við flutning upplýsinga. Út úr þeim samanburði kom símakerfið upp sem besti kostur. En til þess að flytja upplýsingar í gegnum símakerfið var borið saman Sim7600, Arduino NB 1500 og nRF9312 DK og þar stóð sim7600 upp úr.

Smíðin á frumgerð síritans gengu eftir áætlun miðað við uppgefnar forsendur og tókst það með Arduino örtölvu og Sim7600CE. Örtölvan var notuð til þess að lesa af skynjara og skráningu gagna en Sim7600 sá um gagnaflutninginn sjálfan. Núverandi hönnun síritans dugar ekki sem heildarlausna þar sem einungis var gerð frumgerð af honum. Aftur á móti er það efni í annað lengra verkefni að fullhanna síritann sem heildarlausn.

Heimildir

- [1] *Mannvit Web Portal*. vefslóð: <https://vmkerfi.mannvit.is/applications/#waterdata-mannvit/stationoverview> (skoðað 21.11.2022).
- [2] *Flow Rate Formula*, en-us, Section: Physics-MAQ, maí 2022. vefslóð: <https://www.geeksforgeeks.org/flow-rate-formula/> (skoðað 08.11.2022).
- [3] V. G. KG, *VEGAPULS C 21*, en. vefslóð: <https://www.vega.com/en/products/product-catalog/level/radar/vegapuls-c-21> (skoðað 08.11.2022).
- [4] T. Agarwal, *RADAR - Basics, Types, Working, Range Equation & Its Applications*, en-US, sep. 2013. vefslóð: <https://www.elprocus.com/radar-basics-types-and-applications/> (skoðað 08.11.2022).
- [5] *BELUGA*, en. vefslóð: <https://www.flow-tronic.com/products/beluga> (skoðað 08.11.2022).
- [6] Y. Takeda, „Velocity profile measurement by ultrasonic doppler method,“ en, *Experimental Thermal and Fluid Science*, Experimental methods in Thermal and Fluid Science, árg. 10, nr. 4, bls. 444–453, maí 1995, ISSN: 0894-1777. DOI: 10.1016/0894-1777(94)00124-Q. vefslóð: <https://www.sciencedirect.com/science/article/pii/089417779400124Q> (skoðað 08.11.2022).
- [7] O. CNX, *17.7 The Doppler Effect - University Physics Volume 1 | OpenStax*, en. vefslóð: <https://openstax.org/books/university-physics-volume-1/pages/17-7-the-doppler-effect> (skoðað 08.11.2022).
- [8] *Area-Velocity Flow Meter - AVFM 6.1*, de-DE. vefslóð: <https://montar-mt.com/produkt/area-velocity-flow-meter-greyline-avfm-6-1/> (skoðað 08.11.2022).
- [9] Seðlabankinn, *Seðlabankinn - Opinber gengisskráning*, is. vefslóð: <https://www.sedlabanki.is/hagtolur/opinber-gengisskraning/> (skoðað 28.11.2022).
- [10] T. Agarwal, *What is a 8051 Microcontroller with Basics?* en-US, apr. 2015. vefslóð: <https://www.elprocus.com/8051-microcontroller-history-and-basics/> (skoðað 08.11.2022).
- [11] admin, *What Is a CPU: Understanding the Basics*, en-US, jan. 2020. vefslóð: <https://www.hellotech.com/blog/what-is-cpu-cores-what-cpu-do-i-have> (skoðað 17.11.2022).
- [12] *Difference Between ALU and CU*, en-us, Section: Computer Organization and Architecture, jan. 2021. vefslóð: <https://www.geeksforgeeks.org/difference-between-alu-and-cu/> (skoðað 17.11.2022).

- [13] C. s. i. o. o. y. C. k. s.-b. w. d. i. r. mean?1, *CPU Speed: What Is CPU Clock Speed?* en. vefslóð: <https://www.intel.com/content/www/us/en/gaming/resources/cpu-clock-speed.html> (skoðað 25.11.2022).
- [14] M. Gudino, *Analog-to-Digital Converters: How Does an ADC Work? | Arrow.com*, en. vefslóð: <https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters> (skoðað 08.11.2022).
- [15] P. Elsea, *Digital Recording*, 1996. vefslóð: http://artsites.ucsc.edu/EMS/music/tech_background/TE-16/teces_16.html (skoðað 08.11.2022).
- [16] *What is an Arduino? - SparkFun Learn*. vefslóð: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all#re> (skoðað 21.11.2022).
- [17] cdaviddav, *Guide to reduce the Arduino Power Consumption*, en-US, Section: Microcontroller, júl. 2020. vefslóð: <https://diy10t.com/arduino-reduce-power-consumption/> (skoðað 28.11.2022).
- [18] H. Mafukidze, *Introduction to the Raspberry Pi*, en-US, nóv. 2021. vefslóð: <https://www.circuitbasics.com/introduction-to-the-raspberry-pi/> (skoðað 21.11.2022).
- [19] H. Zahid, *How Much Power Does Raspberry Pi Consume While Operating*, en-US. vefslóð: <https://linuxhint.com/power-consumption-raspberry-pi/> (skoðað 28.11.2022).
- [20] *Basic block diagram of communication system*, en-US, Section: Electronics Articles, apr. 2018. vefslóð: <https://www.polytechnichub.com/basic-block-diagram-communication-system/> (skoðað 10.11.2022).
- [21] *modulation | Definition, Types, & Facts | Britannica*, en. vefslóð: <https://www.britannica.com/technology/modulation-communications> (skoðað 10.11.2022).
- [22] C. Manning, *What are radio waves?* en, Publisher: Brian Dunbar, maí 2015. vefslóð: http://www.nasa.gov/directorates/heo/scan/communications/outreach/funfacts/what_are_radio_waves (skoðað 23.11.2022).
- [23] *How Far Can I Talk?* en. vefslóð: <https://quality2wayradios.com/store/radio-range-distance> (skoðað 23.11.2022).
- [24] Lyhor, *What is FM, AM, PM, Modulation ?* Vefslóð: <https://henglyhor.blogspot.com/2019/11/what-is-fm-am-pm-modulation.html> (skoðað 23.11.2022).
- [25] L. Ago, *Digital Modulation - Frequency Shift Keying(FSK)*, en, des. 2017. vefslóð: <https://steemit.com/steemstem/@lorenzor/digital-modulation-frequency-shift-keying-fsk> (skoðað 23.11.2022).

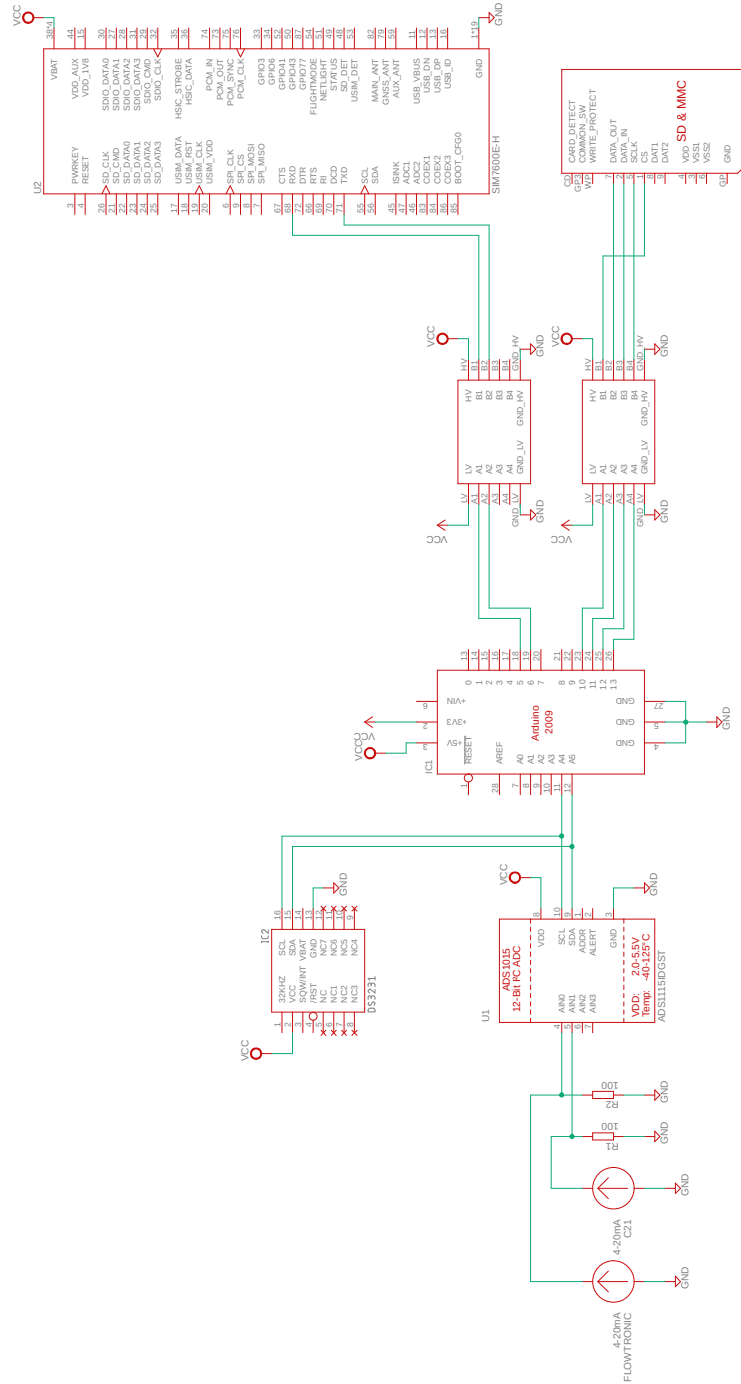
- [26] *Wired and Wireless Networking*, en-us, Section: Class 9, jún. 2021. vefslóð: <https://www.geeksforgeeks.org/wired-and-wireless-networking/> (skoðað 28.11.2022).
- [27] *noise | telecommunications | Britannica*, en. vefslóð: <https://www.britannica.com/technology/noise-telecommunications> (skoðað 10.11.2022).
- [28] *Signal-Noise Ratio - The Essence of Radio*, en-US, jún. 2017. vefslóð: <http://play.fallows.ca/wp/radio/ham-radio/signal-noise-ratio-essence-radio/> (skoðað 26.11.2022).
- [29] *Figure 4. Input harmonic signal and distorted output signal with...* en. vefslóð: https://www.researchgate.net/figure/Input-harmonic-signal-and-distorted-output-signal-with-additive-noise_fig1_283198559 (skoðað 23.11.2022).
- [30] *Choosing the Right Wireless Communication Protocol - Neuronic Works*, en. vefslóð: <https://neuronicworks.com> (skoðað 10.11.2022).
- [31] *Hvar er hægt að tengjast farsímaneti á Íslandi? | Dreifikerfi | Síminn*, is. vefslóð: <https://www.siminn.is/dreifikerfi> (skoðað 17.11.2022).
- [32] *Comparison of 2G 3G 4G 5G | 2G vs 3G vs 4G vs 5G | Rantcell*, en. vefslóð: <https://rantcell.com/blog.html> (skoðað 24.11.2022).
- [33] Fjarskiptastofa, *Samráð um áætlun um lokun GSM (2G) og 3G þjónustu*. vefslóð: <https://www.fjarskiptastofa.is/fjarskiptastofa/tolfraedi-og-gagnasafn/frettasafn/frett/fr%C3%A9ttir/samrad-um-aaetlun-um-lokun-gsm-2g-og-3g-thjonustu> (skoðað 08.11.2022).
- [34] *4G speed vs 4G range-What is the value of 4G speed,4G range*. vefslóð: <https://www.rfwireless-world.com/Terminology/4G-Speed-Vs-4G-Range.html> (skoðað 24.11.2022).
- [35] J. Barani, *Sigfox and LoRaWAN compared to GPRS/GSM/GPRS/LTE/NB-IoT in weather stations*, en-US. vefslóð: <https://www.baranidesign.com/faq-articles/2019/3/21/sigfox-lorawan-compared-to-gprs-gsm-lte-nbiot-in-weather-stations> (skoðað 24.11.2022).
- [36] T. C. Inc, *LoRaWAN - Most Common Applications and Use Cases*, en-US, des. 2021. vefslóð: <https://www.iotforall.com/lorawan-most-common-applications-and-use-cases> (skoðað 27.11.2022).
- [37] K. Mekki, E. Bajic, F. Chaxel og F. Meyer, „A comparative study of LPWAN technologies for large-scale IoT deployment,“ *ICT Express*, árg. 5, nr. 1, bls. 1–7, 2019, ISSN: 2405-9595. DOI: <https://doi.org/10.1016/j.icte.2017.12.005>. vefslóð: <https://www.sciencedirect.com/science/article/pii/S2405959517302953>.

- [38] *Hvað er LoRaWAN og hvernig er það að breyta heiminum? | Lysir.is.* vefslóð: <https://www.lysir.is/blog/hvad-er-lorawan-og-hvernig-er-thad-ad-breyta-heiminum> (skoðað 17.11.2022).
- [39] C. Gomez, J. C. Veras, R. Vidal, L. Casals og J. Paradells, „A Sigfox Energy Consumption Model,“ *Sensors (Basel, Switzerland)*, árg. 19, nr. 3, bls. 681, feb. 2019, ISSN: 1424-8220. DOI: 10.3390/s19030681. vefslóð: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6387435/> (skoðað 24.11.2022).
- [40] *SIGFOX.COM*, en. vefslóð: <https://www.sigfox.com/en> (skoðað 27.11.2022).
- [41] *Qualification | Sigfox build.* vefslóð: <https://build.sigfox.com> (skoðað 13.11.2022).
- [42] E. Bäunker, A. Miguel Garcia og P. Woias, „Minimizing power consumption of LoRa[®] and LoRaWAN for low-power wireless sensor nodes,“ en, *Journal of Physics: Conference Series*, árg. 1407, nr. 1, bls. 012092, nóv. 2019, ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1407/1/012092. vefslóð: <https://iopscience.iop.org/article/10.1088/1742-6596/1407/1/012092> (skoðað 24.11.2022).
- [43] *Achievable bit rates and payload sizes in LoraWAN | Download High-Resolution Scientific Diagram*, en. vefslóð: https://www.researchgate.net/figure/Achievable-bit-rates-and-payload-sizes-in-LoraWAN_tbl1_319315539 (skoðað 25.11.2022).
- [44] *SIM7600CE 4G HAT For Raspberry Pi, LTE Cat-4 4G / 3G / 2G Support, GNSS Positioning, For China.* vefslóð: <https://www.waveshare.com/sim7600ce-4g-hat.htm> (skoðað 27.11.2022).
- [45] Arduino, *Arduino MKR NB 1500*, en. vefslóð: <https://store.arduino.cc/products/arduino-mkr-nb-1500> (skoðað 27.11.2022).
- [46] *nRF9160 DK*, en. vefslóð: <https://www.nordicsemi.com/Products/Development-hardware/nRF9160-DK> (skoðað 27.11.2022).
- [47] *SIM7600X Module 4G Wireless Solutions | SIMCom Wireless Solutions Co.,Ltd.* vefslóð: <https://www.simcom.com/product/SIM7600X.html> (skoðað 27.11.2022).
- [48] *GPS Basics - SparkFun Learn.* vefslóð: <https://learn.sparkfun.com/tutorials/gps-basics/all> (skoðað 28.11.2022).
- [49] *Modem vs Router: What's the Difference?* en. vefslóð: <https://www.xfinity.com/hub/internet/modem-vs-router> (skoðað 25.11.2022).
- [50] *Serial Communication - SparkFun Learn.* vefslóð: <https://learn.sparkfun.com/tutorials/serial-communication> (skoðað 21.11.2022).

- [51] A. Zhu, *Serial Transmission vs. Parallel Transmission*, en, okt. 2016. vefslóð: <https://medium.com/@AriaZhu/serial-transmission-vs-parallel-transmission-3689e1af9200> (skoðað 21.11.2022).
- [52] *Difference Between Serial and Parallel Communication (with Comparison Chart)*, en-US, júl. 2019. vefslóð: <https://circuitglobe.com/difference-between-serial-and-parallel-communication.html> (skoðað 24.11.2022).
- [53] *Difference between Synchronous and Asynchronous Transmission*. vefslóð: <https://www.tutorialspoint.com/difference-between-synchronous-and-asynchronous-transmission> (skoðað 24.11.2022).
- [54] S. Campbell, *Basics of UART Communication*, en-US, feb. 2016. vefslóð: <https://www.circuitbasics.com/basics-uart-communication/> (skoðað 28.11.2022).
- [55] *Serial Communications: RS232 vs RS485 - What are their differences?* en-US, des. 2019. vefslóð: <https://www.seeedstudio.com/blog/2019/12/06/what-is-rs485-and-its-difference-between-rs232/> (skoðað 28.11.2022).
- [56] S. Campbell, *Basics of the SPI Communication Protocol*, en-US, feb. 2016. vefslóð: <https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/> (skoðað 25.11.2022).
- [57] *Modbus Tutorial from Control Solutions*. vefslóð: https://www.csimn.com/CSI_pages/Modbus101.html (skoðað 25.11.2022).
- [58] S. Electric, *What is Modbus and How does it work? | Schneider Electric USA*, en-US. vefslóð: <https://www.se.com/us/en/faqs/FA168406/> (skoðað 20.11.2022).
- [59] *AT Commands Guide for GSM | Developer Blog*, en. vefslóð: <https://www.emnify.com/developer-blog/at-commands-for-cellular-modules> (skoðað 25.11.2022).
- [60] SimCom, *SIM7500_SIM7600Series_AT CommandManual*. vefslóð: https://www.waveshare.net/w/upload/6/68/SIM7500_SIM7600_Series_AT_Command_Manual_V2.00.pdf.
- [61] *UNO R3 | Arduino Documentation*. vefslóð: <https://docs.arduino.cc/hardware/uno-rev3> (skoðað 27.11.2022).
- [62] A. Industries, *ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier*, en-US. vefslóð: <https://www.adafruit.com/product/1085> (skoðað 27.11.2022).
- [63] *Adafruit DS3231 Precision RTC Breakout*, en-US. vefslóð: <https://learn.adafruit.com/adafruit-ds3231-precision-rtc-breakout/overview> (skoðað 28.11.2022).

- [64] *CURS100 - 100 Ohm Current Terminal Input Module*, en. vefslóð: <https://www.campbellsci.com/curs100> (skoðað 24.11.2022).
- [65] Riedon, *Understanding Resistors and Temperature*. vefslóð: <https://riedon.com/technical/understanding-temperature> (skoðað 28.11.2022).
- [66] *Revisiting FTP Basics*, en, Section: IIS Support Blog, feb. 2019. vefslóð: <https://techcommunity.microsoft.com/t5/iis-support-blog/revisiting-ftp-basics/ba-p/343904> (skoðað 28.11.2022).
- [67] *MQTT Basics - MATLAB & Simulink - MathWorks Nordic*. vefslóð: <https://se.mathworks.com/help/thingspeak/mqtt-basics.html> (skoðað 28.11.2022).
- [68] *What is an SSL certificate – Definition and Explanation*, en, Section: Resource Center, feb. 2022. vefslóð: <https://www.kaspersky.com/resource-center/definitions/what-is-a-ssl-certificate> (skoðað 28.11.2022).
- [69] Kashif, *What Programming Language Does Arduino Use*, en-US. vefslóð: <https://linuxhint.com/programming-language-arduino-use/> (skoðað 26.11.2022).
- [70] *Assembly - Introduction*. vefslóð: https://www.tutorialspoint.com/assembly_programming/assembly_introduction.htm (skoðað 26.11.2022).
- [71] *Arduino Uno Rev3*, en, nóv. 2022. vefslóð: <https://store-usa.arduino.cc/products/arduino-uno-rev3> (skoðað 27.11.2022).
- [72] A. Industries, *Adafruit Micro SD SPI or SDIO Card Breakout Board - 3V ONLY!* en-US. vefslóð: <https://www.adafruit.com/product/4682> (skoðað 27.11.2022).
- [73] L. J. Bradley og N. G. Wright, „Optimising SD Saving Events to Maximise Battery Lifetime for Arduino™/Atmega328P Data Loggers,“ *IEEE Access*, árg. 8, bls. 214 832–214 841, 2020, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3041373.
- [74] *DS3231 RTC Module*, en. vefslóð: <https://components101.com/modules/ds3231-rtc-module-pinout-circuit-datasheet> (skoðað 28.11.2022).

A. Rásateikning



B. Forrit

```

/*
*/

#include "RTClib.h"
#include <SoftwareSerial.h>
#include "Adafruit_FONA.h"
#include <Adafruit_ADS1X15.h>

#define BUTTON 2
#define BUZZER 8
#define LED11 11
#define LED12 12
#define LED13 13

#define GSM_RX 6
#define GSM_TX 5

#define GSM_RST 20 // Dummy
#define GSM_BAUD 115200

#define timi_maeling 5

SoftwareSerial SIM7600SS = SoftwareSerial(GSM_TX, GSM_RX);
SoftwareSerial *SIM7600Serial = &SIM7600SS;
RTC_DS3231 rtc;
Adafruit_ADS1115 ads;

#define Filename "TEST4.CSV" // verður að vera caps
#include <SPI.h>
#include <SD.h>

File myFile;

Adafruit_FONA SIM7600 = Adafruit_FONA(GSM_RST);

////////////////////

unsigned long time_last;

String Stod = "_HMV_xxx";
int maelt =0;
int loggad=0;

int head=0;
int hradi=0;
String send_sting = "";
void setup()
{

  Serial.begin(9600);
  Serial.println("Interfacing SIM7600 GSM GPS Module with Maker UNO");
  Serial.println("Initializing... (May take a minute)");
  delay(15000);

  // Make it slow so its easy to read!
  SIM7600Serial->begin(GSM_BAUD);
  ckeak_on_sim7600();

  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    Serial.flush();
  }

  if (rtc.lostPower()) {
    Serial.println("RTC lost power, let's set the time!");
    // When time needs to be set on a new device, or after a power loss, the
    // following line sets the RTC to the date & time this sketch was compiled
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    // This line sets the RTC with an explicit date & time, for example to set
    // January 21, 2014 at 3am you would call:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
  }
  DateTime now = rtc.now();

```

```

time_last = now.unixtime();

if (!SD.begin(4)) {
  Serial.println("initialization failed!");
}
if (!ads.begin()) {
  Serial.println("Failed to initialize ADS.");
}
Makeheader();
}

void loop()
{
  DateTime now = rtc.now();

  if ((now.unixtime()-time_last) >=timi_maeling){
    hradi = hradi+read_sensor(1);
    head = head+read_sensor(1);
    time_last = now.unixtime();
    maelt++;
  }

  if (maelt >=12){
    maelt =0;
    int head_avg = head/12;
    int hradi_avg = hradi/12;
    hradi =0;
    head=0;
    String time_now = String(now.hour())+"."+String(now.minute())+"."+String(now.second());
    send_sting = send_sting + time_now + "."+ String(head_avg)+ "."+ String(hradi_avg)+ "."; \n";
    Write_To_file(time_now + "."+ String(head_avg)+ "."+ String(hradi_avg)+ "."; \n");
    loggad++;
  }

  if (loggad>=60){
    loggad = 0;
    String time_now = String(now.hour())+"."+String(now.minute())+"."+String(now.second());
    Upload_file(MakeCommed(time_now), send_sting);
    send_sting="";
  }
}

float read_sensor(int sensor){ //***** 1 er headanemi 2 er hradi
int adc = ads.readADC_SingleEnded(sensor);
float V_Sensor = ads.computeVolts(adc);

if (sensor==1){
  return 15/1.6 *(V_Sensor-0.4);
}
else if (sensor ==2){
  return 8/1.6 *(V_Sensor-0.4)-2;
}
}

void ckeak_on_sim7600(void){
if (!SIM7600.begin(*SIM7600Serial)) {
  Serial.println("Couldn't find SIM7600");
  SIM7600Serial->print("AT+CNMI=2,1\r\n"); // Set up the SIM800L to send a +CMTI notification when an SMS is received
}
}

String MakeCommed(String log_time){
  String first = "AT+CFTPSPUT=\\";
  String last = ".CSV\\";
  String command = first+ log_time+Stod+ last;
  return command;
}

uint8_t sendATcommand(String ATcommand, const char* expected_answer, unsigned int timeout)
{
  uint8_t x = 0, answer = 0;
  char response[100];

```

```

unsigned long previous;

memset(response, '\0', 100); // Initialize the string

delay(100);

while (SIM7600Serial->available() > 0) { // Clean the input buffer
  SIM7600Serial->read();
}

SIM7600Serial->println(ATcommand); // Send the AT command

x = 0;
previous = millis();

// This loop waits for the answer
do {
  if (SIM7600Serial->available() != 0) {
    // if there are data in the UART input buffer, reads it and checks for the answer
    response[x] = SIM7600Serial->read();
    Serial.print(response[x]);
    x++;
    // check if the desired answer is in the response of the module
    if (strstr(response, expected_answer) != NULL) {
      answer = 1;
    }
  }
  // Waits for the answer with time out
} while ((answer == 0) && ((millis() - previous) < timeout));

// SIM7600Serial->print("\n");
return answer;
}

void Upload_file(String sendcommand , String Send_data){

  delay(100);
  sendATcommand("AT+CFTPSSTART", "OK", 1000);
  delay(100);
  sendATcommand("AT+CFTPSLOGIN=\\"ftp.mannvit.is\\",21,\"username\\",\"password\\",0", "+CFTPSLOGIN: 0", 10000);
  sendATcommand(sendcommand, "> ", 2000);
  delay(50);
  SIM7600Serial->write("hmv_xxx , head, hradí \n ");
  SIM7600Serial->println(Send_data);
  SIM7600Serial->println();
  SIM7600Serial->write(0x1A);
  sendATcommand("AT+CFTPSLOGOUT", "+CFTPSSTOP: 0", 2000);
  delay(100);
  sendATcommand("AT+CFTPSSTOP", "OK", 1000);
  delay(100);
}

bool Loop_files(File dir, int numTabs) {
  bool Quit = false;
  while (! Quit) {
    File entry = dir.openNextFile();
    if (! entry) {
      // no more files
      Quit = true;
    }

    for (uint8_t i = 0; i < numTabs; i++) {
    }

    if (String(entry.name())== Filename){
      return false;
    }

    if (entry.isDirectory()) {
      Loop_files(entry, numTabs + 1);
    } else {
      // files have sizes, directories do not
    }
    entry.close();
  }
  return true;
}

void Makeheader(void){
  myFile = SD.open("/");
  bool Nofile =Loop_files(myFile, 0);
  myFile.close();
  if (Nofile){

```