
WASP

Wireless Arduino Sensor Protocol

GROUP SW513E15



Christian Lundtofte
Henrik Djernes Thomsen
Jonathan Hastrup

Bjørn Opstad
Morten Mandrup
Mathias Corlin



AALBORG UNIVERSITY
STUDENT REPORT

Department of computer science
Selma Lagerlöfs Vej 300
9220 Aalborg Ø

Title:

WASP - Wireless Arduino Sensor Protocol

Theme:

Embedded Systems

Project period:

02/08/2015

21/12/2015

Project group:

SW513E15

Members:

Christian Lundtofte Sørensen
Henrik Djernes Thomsen
Jonathan Hastrup
Bjørn Opstad
Morten Mandrup Hansen
Mathias Corlin

Synopsis:

Synopsis her!

Supervisor:

Hua Lu

No. printed Copies: ?

No. of Pages: ?

No. of Appendix Pages: ?

Total no. of pages: ?

Completed: 21/12/2015

The contents of this report is freely accessible, however publication (with source references) is only allowed upon agreement with the authors.

	<hr/>	Bjørn Opstad
<hr/>		
Christian Lundtofte		
	<hr/>	
		Morten Mandrup
<hr/>		
Henrik Thomsen		
	<hr/>	
		Matthias Corlin
<hr/>		
Jonathan Hastrup		

1 Project introduction	5
1.1 Initializing problem statement	5
I Analysis	6
2 Context	8
3 Use case	9
4 Technologies	10
4.1 Mesh networks	10
4.2 Wireless communication	10
4.3 Communication protocols	10
5 Problem Statement	11
5.1 Requirements	11
II Implementation	12
6 Theory	13
7 Design	14
8 Implementation	15
9 Test	16
III Conclusion	17
10 Reflection	18
10.1 What have we done!?	18
11 Summary	19
11.1 It ended like this	19
12 Future Work	20
12.1 To be done	20
IV Appendix	21

1. Project introduction

This is an introduction.

Communication technology can today be seen as a foundation for wide range of technologies. Devices today can communicate with each other in many different ways, and can take part in large and small networks alike.

This report revolves around embedded systems and how communication can be established between these. This focus point is quite broad in terms where it can be applied, so this report will make use of a specific usecase to showcase the communication between embedded systems its applied usage.

The usecase that will be in focus throughout the report is the moisture of a golf course, where multiple devices will relay information about the earth's moisture levels to a central unit. If a device is placed out of range from the central unit, it will need to send the its data reading to the central unit by using other devices within reach.

To ensure that the data will reach central unit, and ensuring that the data is still correct/usable(?), a sufficient protocol must be found/established to verify this. The protocol must be able executable on the embedded devices.

This have lead to the following initializing problem statement:

1.1 Initializing problem statement

How can a sensor network and a protocol be designed, so that data can be relayed throughout the network, enabling an endpoint device to receive the information without being within range of all sensors in the network?

It is a good question and we will analyze it. (lol)

Part I

Analysis

The analysis will discuss and look into the different aspects of the initializing problem formulation and the topics therein. The sections in this chapter blah-blah..

2. Context

3. Use case

The purpose of this project is to create a protocol that allows multiple Arduinos to share data to a single endpoint, but a use case is needed to test the protocol.

The chosen use case for this report is soil moisture sensors for use on golf courses. A golf course is usually very large, and covering an entire golf course with cords would be a big task. Furthermore this would make the system hard to extend and almost impossible to make hot pluggable.

This makes this project a good use case for golf courses, as soil moisture is important in determining where it is necessary to water the course.

4. Technologies

We shall look at some existing technologies now.

4.1 Mesh networks

4.2 Wireless communication

4.3 Communication protocols

5. Problem Statement

Very good problem statement for you, my friend. Special prize.

Make a good sending data network for arduino.

5.1 Requirements

There are some requirements to the system and its software. These are split in two categories: functional and non-functional. This is based on some smart guys work [keylist].

5.1.1 Functional requirements

The list of functional requirements:

1. Actually run is an important part to passing the exam

5.1.2 Non-functional requirements

List of non-functional requirements:

1. Looking good is not a bad thing.

Part II

Implementation

8. Implementation

Part III

Conclusion

10. Reflection

oh..

10.1 What have we done!?

11. Summary

ok..

11.1 It ended like this

12. Future Work

Here's what's missing..

12.1 To be done

Part IV

Appendix