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

Supervisor:
Hua Lu

Synopsis:

Synopsis her!

No. printed Copies: ?
No. of Pages: ?
No. of Appendix Pages: ?
Total no. of pages: ?
Completed: 21/12/2015

Bjørn Opstad

Christian Lundtofte

Morten Mandrup

Henrik Thomsen

Matihas Corlin

Jonathan Hastrup

Contents

1. Project introduction

This is an introduction.

Here is the 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.

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 blahblablah..

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

A mesh network use a wide variety of protocols, to manage the route data is transferred. In networking a protocol is a special set of rules and standards for how devices would interacts with each other. A well known protocols would be TCP/IP(Transmission Control Protocol/Internet Protocol), which today are used to communicate between anything with a internet connection. The mesh network we are looking at is a radio based network, and therefore some more relevant protocols will be examined. Few excising protocols will be presented in this section.

4.3.1 Time division multiple access

4.3.2 Ad hoc On-Demand Distance Vector Routing

4.3.3 Radio Link Protocol

Radio Link protocol(RLP) is a automatic repeat request(ARQ)[1] fragmentation protocol used over a wireless air interface. Most air interface protocols have a packet loss of up to 1% which is intolerable when handling sensitive data. RLP detects losses in packets and with a retransmission tries to bring down the losses. The retransmission can bring the loss down to 0.1% to 0.0001%. This loss rate is more tolerable when handling sensitive and precise data.

RPL cannot request a certain payload size from the air interface, the air interface scheduler instead determines the packet size, based on changing channel conditions constantly. Most of the other fragmentation protocols, such as 802.11b[2] and IP, determine a payload of a certain size by the upper layers, and call upon the MAC. These protocols are not as flexible as RLP, and sometime fail transition during small fades in a wireless environment.

[1] an error-control method for data transmission that uses acknowledgements and timeouts

[2] An wireless networking specification that extends throughput up to 11 Mbit/s

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

6. Theory

7. Design

8. Implementation

9. Test

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