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

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

Synopsis:

Synopsis her!

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 blah-blablah..

Arduino is an open source platform, which makes the designing of an interactive or general electronics system easier. This is also why the Arduino platform is often used at school for learning about electronics.

Arduino boards have a setup of input and output ports enabling it to read from a sensor, or a button and then maybe activate a motor or an LED light. How the Arduino handles or reacts to input is up to the designer which can program the Arduino board using the Arduino IDE.

"Arduino" covers a range of platforms with a lot of different boards varying in size, from Arduino Nano up to Arduino Mega. One of the more popular Arduino boards is the Arduino Uno which is somewhere in the middle of boards, considering size and power.

In the following sections, the Uno and Mega, which will be used in this project, will be described.

2.1 Arduino Uno

One of the most common boards is the Arduino Uno 2.1, which is based on the ATmega328 microcontroller. It has 14 digital input/output pins, and 6 analog inputs for connecting the different components. Considering specifications, which is shown on 2.1, the Uno is limited on its resources. Therefore it is needed to limit both program- and datasize, and also the amount of complex tasks.

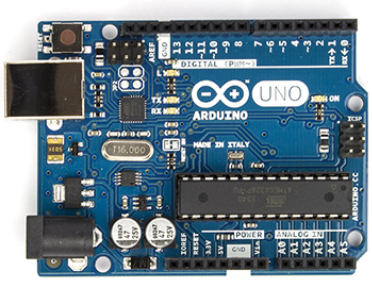

Figure 2.1: Arduino Uno board[1].

2.2 Arduino Mega

The Arduino Mega is a larger version of the Uno. The Mega has more memory and pins, which makes it better for handling larger programs or amounts of data. This also allows more components can be connected to the board. Since the clock speed is the same as the Uno, the Mega will not process data faster.

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB is used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Table 2.1: Specifications for Arduino Uno

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Table 2.2: Specifications for Arduino Mega

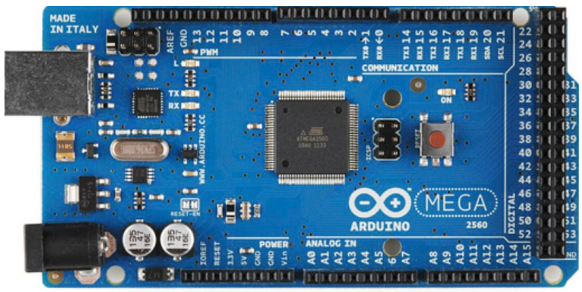

Figure 2.2: Arduino Mega board[2].

3. Context

4. 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 this protocol.

The chosen use case for this report is soil moisture 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 or hot pluggable.

That 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. Wasting water is not much of a problem in Denmark, but some places in the world, water is a sparse resource. Using less water on a large golf course could save money down the line too, and is good for the environment. The use of radiocommunication is well suited for large, open spaces, like a golf course.

4.1 Interview

An informal interview was performed with a greenkeeper at Aalborg Golfcourse, Kim Jensen. Kim provided insights on the requirements of the system, potential problems, and ideas for later iterations of the system. The interview questions and answers can be found in Appendix ##.

Some holes on Aalborg Golfklub is located in a swampy area, which often makes watering them obsolete. This project could help stop some of this water waste. Some holes are in a moor, where watering is often required. When it is required, this project could inform the greenkeepers, who could act on this. Making the product hot pluggable would be useful if there is suddenly more or less water than usually, in which case the product can be added to this location, and the water levels can be measured to determine if there is a problem, and in that case how to solve it. The same applies if an area is more dry than usual.

An important consideration is where to place the devices. The sensors needs access to the soil in order to measure the moisture levels, but they should not be placed above ground as golfers could hit and destroy the devices. The initial idea was to place it in the holes for the sprinklers, but this is bad since sprinklers often leak, and the sensor would therefore get inaccurate readings. This makes it necessary to bury the devices at known locations in the course. The depth the devices are buried at is important, due to the different types of grass. The greens use a 2cm. layer of sand at the top, which allows water to quickly go through. This makes it necessary to bury the devices 10-15cm. in the ground. These requirements changes based on the kind of grass and what is beneath.

Regarding future iterations of the project, Kim suggests adding a pH meter to the devices. This would allow the greenkeepers to create specific mixes of fertilizer for different parts of the course, depending on the pH value of the soil.

5. Technologies

We shall look at some existing technologies now.

5.1 Networks

This section will contain descriptions of networks and network theory, and connect to the established use case.

A computer network is a collection of computers and devices connected so that they can share information and services [3]. The way these devices are interconnected is called topology. The communication structure the devices use to exchange information over a medium is called protocol, which will be described in another section.

In this section the term node is used for a device connected to the network, to avoid binding to a specific device type.

There are different types of network topologies, and here are some examples:

- Ring
- Line
- Bus
- Tree
- Star
- Mesh
- Fully connected

These can be seen in figure xx, that will be put here somewhere.

The star network has one main node that the other nodes are directly connected to. An example of a star topology network is wifi, typically with a wireless router to which other devices connect to gain network access. The wireless router will handle all the network communication and redirect the information to the correct device. A limitation of the star network is that all network devices must have a connection to the main node, and therefore be clustered within the reach of the main node coverage. [**todo**]

A tree network also utilizes a main node, but the devices in the network do not necessarily connect directly to the main node, but rather connect to another node that relays to the main node. This can repeat over multiple levels, so that information is relayed through several nodes, before reaching the main node. The tree network has a fixed node structure, and the relay nodes will route the information towards the destination. [**todo**]

Another topology is the mesh network. It is a type of mobile ad hoc network. There are two kinds of mesh networks, the full-mesh and the partial-mesh networks. A full-mesh describes a network where all the nodes are interconnected, similar to a fully connected graph. A partial mesh is also a mesh network, but does not require all nodes to be connected, so that it's similar to a tree network with cycles. The mesh networks have the same limitation as a tree network, regarding the information transmission delay because the information transmits through up to several nodes. [4]

The best fitting network topology for the use case is a mesh network. It can transmit information through the network without limiting the connected devices to a certain distance from a main device, as with a star topology. It is also capable of multiple methods of distributing information. It does not rely on all nodes working at all times, as the network can reconfigure and find another path of information. This applies as long as there somehow exists another node that can relay the information towards the main node.

Move this paragraph? There are multiple methods of communicating through a mesh network, therein routing and flooding are two alternatives. Routing will transfer the information towards a destination node, whereas the flooding method will notify all nodes within reach to distribute the information forward, and this will repeat until all nodes has transmitted the information, and hence the destination node also has received the information.

5.2 Wireless communication

5.3 Communication protocols

A mesh network can 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 nodes would interacts with each other. A well known protocols could be TCP/IP(Transmission Control Protocol/Internet Protocol), which today are used to communicate between almost 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. A few excising protocols will be presented in this section.

5.3.1 Time division multiple access

Time division multiple access(TDMA) is protocol that divides a single channel into smaller time slots. Each time slot transmits one byte or a segment of a signal, in a sequential serial data format. Each slot is active of a small amount of

time, before the next slot in the queue get time to transmit.

TDMA is as an example used in the T1 telecommunication transmission system. Each T1 channels carry up to 24 voice telephone connections. Where each connection covers 300 Hz to 3000Hz and is digitized at an 8-kHz rate, which is two times the highest frequency component needed to retain all the analog content.

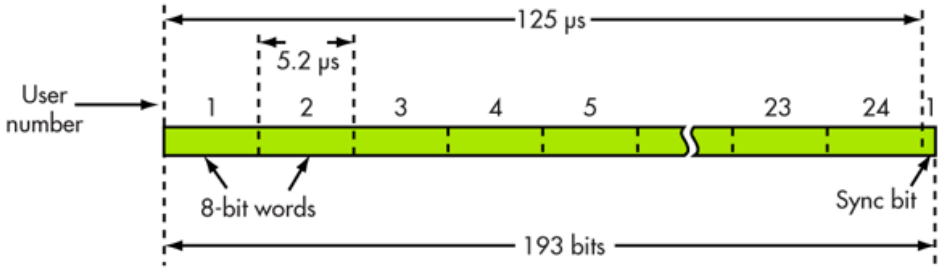

Figure 5.1: Illustration of the TDMA protocol

On Figure 5.1 it is seen how the channel, for T1, is split up into 24 smaller pieces. Each time slot is accountant for a user using a voice channel, to talk to some other user. Each time slot is of the size 8-bit, the user is unaware of this small data size, and that other are using the same channel, because the shift between each time slot happens so fast. This gives the illusion of each user talking with no interruptions, even though each user only is assigned 1/24 of the total bandwidth. A single bit is used to synchronization. The TDMA system can maximum achieved a data rate of 1.544 Mbit/second.[**TDMA**]

TDMA can be used for any system that require several device to use the same channel, without interfering with each other.

5.3.2 Ad hoc On-Demand Distance Vector Routing

Ad hoc On Demand Distance Vector(AODV) routing algorithm is a routing protocol designed for ad hoc networks. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes.[**AOVD1**]

AODV is a network where only the local nodes around a newly added node, is affected by the addition. If a link between nodes is broken, and it does not affect an ongoing transmission, no global notification occurs. This leads to a network where new nodes easily can be added and removed without overhead on the entire network. The transmission route in AODV is managed so, that only nodes in the direct route is active, which reduce the need for route maintenance and reduce idle nodes. The protocol can determine multiple routes between a source

and a destination, but only a single one is implemented. The route is not necessarily the shortest.

A downside to AODV is that if a single route breaks, due to a defect node, it is not possible to know whether other routes exists. But if a route breaks, the protocol discovers a new route, if possible, as and when necessary.

When a node is ordered to send a packet to a specified destination, it checks its routing table to determine if it has a current route to the destination. If there already exist a node, the packet will be delivered to the next node in the route, repeating until it arrives at the correct location. If there does not exist a route, the node will initiates a route discovery process.

A route discovery process begins with the source creating a Route Request (RREQ) packet. The packet contains the sources node's unique ID and IP, and the destination node's unique ID and IP. The packet is then transmitted out from the source to it's neighbours. The unique ID of each node is then stored in the packet, to ensure the same node is not transmitting more than once. The destination ID is to ensure it knows when the destination node is reached. A simple AODV network can be seen on Figure 5.2, where S and D represent the source and destination. It is visualized how the route discovery is invoked using the RREQ packet.[**AOVD2**]

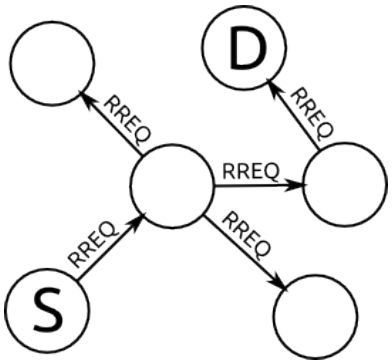

Figure 5.2: Illustration of the AODV route discovery

5.3.3 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

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

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.[**MobileComm**]

RLP is used to make a more fail-safe environment for the transmitted data, and to ensure nothing is lost on the way. The Radio Link Protocol is typically used in cellular transmission.

[2] An wireless networking specification

6. Problem Statement

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

Make a good sending data network for arduino.

6.1 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**].

6.1.1 Functional requirements

The list of functional requirements:

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

6.1.2 Non-functional requirements

List of non-functional requirements:

1. Looking good is not a bad thing.

Part II

Implementation

7. Theory

8. Design

9. Implementation

10. Test

Part III

Conclusion

11. Reflection

oh..

11.1 What have we done!?

12. Summary

ok..

12.1 It ended like this

13. Future Work

Here's what's missing..

13.1 To be done

Bibliography

[1] Arduino. *Arduino - Introduction*. Seen 16/09/2015. URL: http://arduino.cc/en/Guide/Introduction.

[2] Seen 17/09/2015. URL: https://www.robotics.org.za/image/data/Arduino/Arduino%20Boards/arduino_mega_r3_002_hd.jpg.

[3] K. Mansfield and J. Antonakos. *Computer Networking for LANS to WANS: Hardware, Software and Security*. Networking (Course Technology, Inc.) Cengage Learning, 2009. ISBN: 9781423903161. URL: https://books.google.no/books?id=VQvhAN9iBuMC.

[4] S. Misra, S.C. Misra, and I. Woungang. *Guide to Wireless Mesh Networks*. Computer Communications and Networks. Springer London, 2009. ISBN: 9781848009097. URL: https://books.google.no/books?id=etluRHnDAUQC.

Part IV

Appendix