
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 Use case	8
2.1 Interview	8
3 Problem Statement	9
3.1 Requirements	9
4 Technologies	10
4.1 Controller units	10
4.2 Sensors	11
4.3 Networks	11
4.4 Wireless communication	13
4.5 Communication protocols	13
4.6 RF24 transreciever analysis:	17
II Implementation	19
5 Theory	20
6 Design	21
7 Implementation	22
8 Test	23
III Conclusion	24
9 Reflection	25
9.1 What have we done!?	25
10 Summary	26
10.1 It ended like this	26
11 Future Work	27
11.1 To be done	27
IV Appendix	29

1. Project introduction

This is an introduction.

Communication technology can today be seen as a foundation for a 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 resolves 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.

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?

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..

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.

2.1 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.

3. Problem Statement

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

Make a good sending data network for arduino.

3.1 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].

3.1.1 Functional requirements

The list of functional requirements:

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

3.1.2 Non-functional requirements

List of non-functional requirements:

1. Looking good is not a bad thing.

4. Technologies

We shall look at some existing technologies now.

4.1 Controller units

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.

4.1.1 Arduino Uno

One of the most common boards is the Arduino Uno 4.1, 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 4.1.1, the Uno is limited on its resources. Therefore it is needed to limit both program- and datasize, and also the amount of complex tasks.

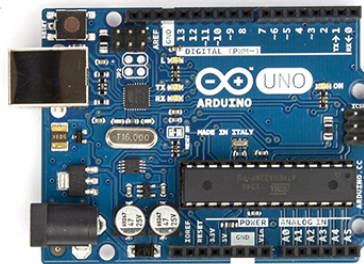


Figure 4.1: Arduino Uno board[1].

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 4.1: Specifications for Arduino Uno

4.1.2 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.

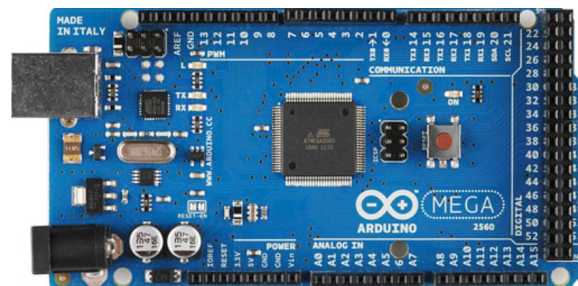


Figure 4.2: Arduino Mega board[2].

4.1.3 XBee

4.2 Sensors

4.3 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

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 4.2: Specifications for Arduino Mega

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.

4.4 Wireless communication

4.5 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 interact 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 exciting protocols will be presented in this section.

4.5.1 Time division multiple access

Time division multiple access (TDMA) is a 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 for a small amount of time, before the next slot in the queue gets time to transmit.

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

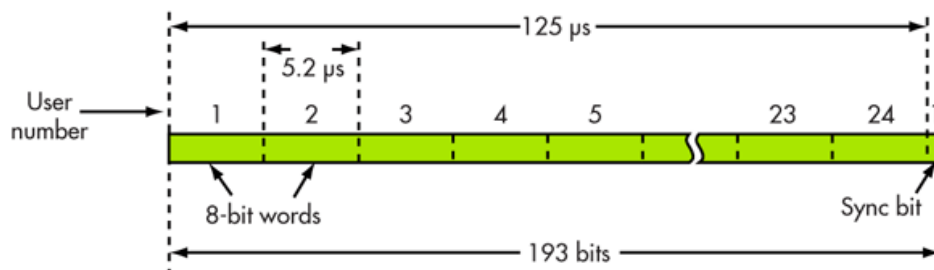


Figure 4.3: Illustration of the TDMA protocol

On Figure 4.3 it is seen how the channel, for T1, is split up into 24 smaller pieces. Each time slot is accountable 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 others 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 for synchronization. The TDMA system can maximum achieved a data rate of 1.544 Mbit/second.[5]

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

4.5.2 Dynamic Source Routing protocol

The Dynamic Source Routing protocol (DSR) is a simple routing protocol designed specifically for use in multi-hop wireless ad hoc networks. With DSR the network is completely self-organizing and self-configuring, requiring no administration or existing network structure. The nodes in the network work together to forward packets to destinations that can be outside of the sender's transmission range. As nodes can be added and removed ad hoc, the protocol automatically determines and maintains the routing of the packets throughout the network.

Since the number or sequence of nodes needed to reach any destination may change at any time, the resulting network topology may be quite rapidly changing. DSR is created to create a very low overhead yet being able to react quickly to change in the network environment. The protocol provides a highly reactive service to ensure successful delivery of data packets where node may be moving around, or other changes in the network occur.

The DSR protocol is composed of two main mechanisms that allow the discovery and maintenance of the routes in the network.

- Route Discovery is the mechanism by which a node S wishing to send a packet to a destination node D obtains a source route to D. Route Discovery is used only when S attempts to send a packet to D and does not already know a route to D.
- Route Maintenance is the mechanism by which node S is able to detect, while using a source route to D, if the network topology has changed such that it can no longer use its route to D because a link along the route no longer works. When Route Maintenance indicates a source route is broken, S can attempt to use any other route it happens to know to D, or it can invoke Route Discovery again to find a new route for subsequent packets to D. Route Maintenance for this route is used only when S is actually sending packets to D.

DSR is a source routing protocol where packets carries a header, a ordered list of nodes to determine the route. This explicit use of routing allows the sender to select and control the route for the packets it sends. Source routing allow for load balancing, since the sender can create different routes out though the network, avoiding "heavy traffic". It is also guarantee that the routes used are loop-free, since a generated route never use the same node twice. By including this source route in the header of each packet, other nodes forwarding or overhearing any of these packets can easily cache this information for future use.[DSR]

4.5.3 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.[6]

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 exist. 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 exists 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 initiate a route discovery process.

A route discovery process begins with the source creating a Route Request (RREQ) packet. The packet contains the source 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 its 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 4.4, where S and D represent the source and destination. It is visualized how the route discovery is invoked using the RREQ packet.[7]

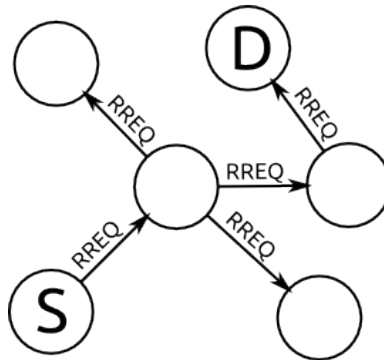


Figure 4.4: Illustration of the AODV route discovery

4.5.4 Better approach to mobile ad-hoc networking

Better approach to mobile ad-hoc networking (B.A.T.M.A.N) is a routing protocol for multi-hop ad-hoc mesh networks. B.A.T.M.A.N is said to solve some of the more typical problems with the classical routing protocols, which is not well suited for wireless ad-hoc networks. Some of the problems is, such networks are unstructured, are based on an inherently unreliable medium and dynamically change their topology.

When using the B.A.T.M.A.N algorithm, the approach is to divide the knowledge about the most reasonable path between nodes in a network to all of the participating nodes. Each node in the network only account for the best path to all

other nodes in the system. This results in, the need for a global knowledge about local topology changes becomes unnecessary. In addition, does B.A.T.M.A.N have an event-based, but timeless, flooding mechanism that prevents the occurrence of loops and limits the amount of topology messages flooding the network. The protocol is designed to handle network where links between nodes are not always reliable.[8]

4.5.5 Protocol comparison

Below is a comparison table with the protocol discusses in this chapter. Her skal skrives et eller andet.

[9][10][DSR][8]

4.6 RF24 transreciever analysis:

To confirm the usability of the RF24 transreciever, labeled as RF24 from this point, a short analysis with tests will be performed.

4.6.1 Range:

A basic test to test the range is useful to ensure that RF24 is usable in a practical application with a wide distance between units.

Testing is done by having two units set as transmitter and reciever units respectively, having the transmitter transmit a single signal repeatedly while increasing the distance between the units. When the reciever no longer recieves the signal, the distance can then be measured.

CODE GOES HERE!

4.6.2 Bit error rate:

what is?

how? code, setup etc.

Table 4.3: Protocol comparison

	Ad hoc	Route metrics	Loop Free	Load balancing	Reliability	Throughput
TDMA	Yes	Routes ensuring guaranteed bandwidth	Yes	Yes	High	Decreases as more nodes are added
DSR	Yes	Source routing	Yes	No	High	Decreases as more nodes are added
AOVD	Yes	Fastest & shortest path	Yes	No	High	Poor for more than 20 nodes
B.A.T.M.A.N	Yes	Fastest & shortest path	Yes	No	High	Good - scales well with more nodes

Transmitter \ Reciever	Unit 1	Unit 2	Unit 3
Unit 1	-	?	?
Unit 2	?	-	?
Unit 3	?	?	-

Datasheet	Unit 1	Unit 2	Unit 3
0,0%	?	?	?

Part II

Implementation

7. Implementation

Part III

Conclusion

oh..

9.1 What have we done!?

10. Summary

ok..

10.1 It ended like this

11. Future Work

Here's what's missing..

11.1 To be done

- [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=VQvvhAN9iBuMC>.
- [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>.
- [5] Lou Frenzel. *Fundamentals of Communications Access Technologies: FDMA, TDMA, CDMA, OFDMA, AND SDMA*. URL: <http://electronicdesign.com/communications/fundamentals-communications-access-technologies-fdma-tdma-cdma-ofdma-and-sdma>.
- [6] Samir Das Ian Chakeres Charles Perkins Elizabeth Belding-Royer. *AODV*. URL: <http://moment.cs.ucsb.edu/AODV/>.
- [7] Johns Hopkins Dr. Baruch Awerbuch Dr. Amitabh Mishra. *Ad hoc On Demand Distance Vector (AODV) Routing Protocol*. URL: <http://www.cs.jhu.edu/~cs647/aodv.pdf>.
- [8] Jean-Philippe Lang. *B.A.T.M.A.N. protocol concept*. URL: <http://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept>.
- [9] Stephane Rousseau Vania Conan. *Route Selection for Capacity Maximization in Multi-Rate TDMA-based Wireless Ad Hoc Networks*. Seen 16/09/2015. URL: <http://cnd.iit.cnr.it/rbruno/files/meshtech09.pdf>.
- [10] Piotr ZWIERZYKOWSKI Piotr OWCZAREK. *ROUTING PROTOCOLS IN WIRELESS MESH NETWORKS - A COMPARISON AND CLASSIFICATION*. 2009. URL: http://www.academia.edu/5756527/ROUTING_PROTOCOLS_IN_WIRELESS_MESH_NETWORKS_-_A_COMPARISON_AND_CLASSIFICATION.

Part IV

Appendix