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 **5**
1.1 Initializing problem statement 5

I Analysis **6**

2 Arduino **8**
2.1 Arduino Uno 8
2.2 Arduino Mega 8

3 Context **11**

4 Use case **12**
4.1 Interview 12

5 Technologies **13**
5.1 Networks 13
5.2 Wireless communication 14
5.3 Communication protocols 14

6 Problem Statement **15**
6.1 Requirements 15

II Implementation **16**

7 Theory **17**

8 Design **18**

9 Implementation **19**

10 Test **20**

III Conclusion **21**

11 Reflection **22**
11.1 What have we done!? 22

12 Summary **23**
12.1 It ended like this 23

13 Future Work **24**
13.1 To be done 24

IV Appendix **26**

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 used at school for learning about electronics. It is developed by the students David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis, and Massimo Banzi at the Interaction Design Institute Ivrea in Italy[**makingofarduino**].

Arduino boards have a set op 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 huge platform with a lot of different boards greatly 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 subsections, 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.

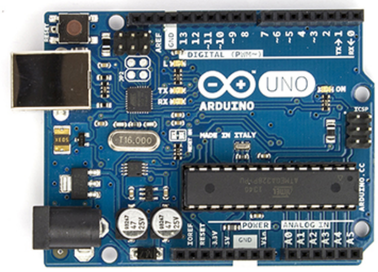

Figure 2.1: Arduino Uno board[**arduinointroduction**].

2.2 Arduino Mega

The Arduino Mega is a larger version of the Uno for some specifications. The Mega is larger on its memory and Pins, which makes it better for handling larger

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

programs or amounts of data. Also a lot 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.

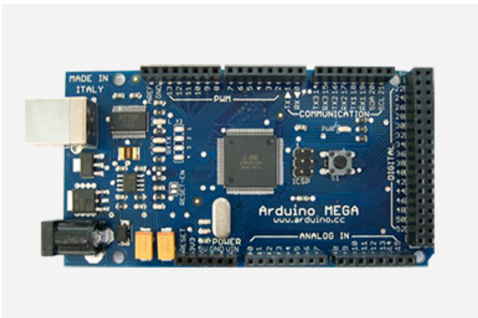

Figure 2.2: Arduino Mega board[**arduinomega**].

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

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 the 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 and make 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 moistore 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 was 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 are important, due to the differnet 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 [1]. 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.

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.

Another topology is the mesh 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.

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. Routing and flooding are two alternatives. Routing will transfer the information towards a destination node, whereas the flooding method will ask all nodes within reach to spread the information forwards, 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

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

9. Implementation

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] 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`.

[2] C.S.R. Murthy and B.S. Manoj. *Ad Hoc Wireless Networks: Architectures and Protocols*. Pearson Education, 2004. ISBN: 9780132465694. URL: `https://books.google.no/books?id=U-yLb-9nXyYC`.

Part IV

Appendix