

---

# 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    |       |                 |

|  |           |
|--|-----------|
| <b>1 Project introduction</b>                | <b>5</b>  |
| 1.1 Initializing problem statement . . . . . | 5         |
| <b>I Analysis</b>                            | <b>6</b>  |
| <b>2 Arduino</b>                             | <b>8</b>  |
| 2.1 Arduino Uno . . . . .                    | 8         |
| 2.2 Arduino Mega . . . . .                   | 8         |
| <b>3 Context</b>                             | <b>11</b> |
| <b>4 Use case</b>                            | <b>12</b> |
| 4.1 Interview . . . . .                      | 12        |
| <b>5 Technologies</b>                        | <b>13</b> |
| 5.1 Networks . . . . .                       | 13        |
| 5.2 Wireless communication . . . . .         | 13        |
| 5.3 Communication protocols . . . . .        | 13        |
| <b>6 Problem Statement</b>                   | <b>14</b> |
| 6.1 Requirements . . . . .                   | 14        |
| <b>II Implementation</b>                     | <b>15</b> |
| <b>7 Theory</b>                              | <b>16</b> |
| <b>8 Design</b>                              | <b>17</b> |
| <b>9 Implementation</b>                      | <b>18</b> |
| <b>10 Test</b>                               | <b>19</b> |
| <b>III Conclusion</b>                        | <b>20</b> |
| <b>11 Reflection</b>                         | <b>21</b> |
| 11.1 What have we done!?! . . . . .          | 21        |
| <b>12 Summary</b>                            | <b>22</b> |
| 12.1 It ended like this . . . . .            | 22        |
| <b>13 Future Work</b>                        | <b>23</b> |
| 13.1 To be done . . . . .                    | 23        |
| <b>IV Appendix</b>                           | <b>25</b> |

# 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-blah..

## 2. Arduino

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.

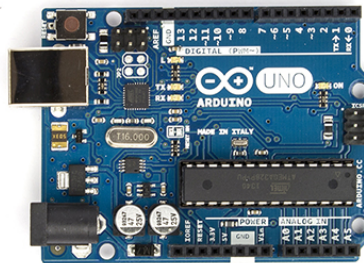


Figure 2.1: Arduino Uno board[1].

### 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

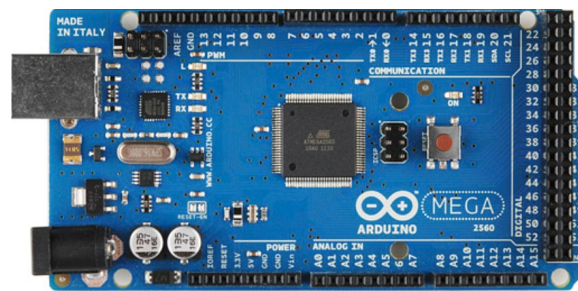


Figure 2.2: Arduino Mega board[2].



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 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 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 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

---

## 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**







## 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

---

- [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](https://www.robotics.org.za/image/data/Arduino/Arduino%20Boards/arduino_mega_r3_002_hd.jpg).



## **Part IV**

# **Appendix**