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 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[1].

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.

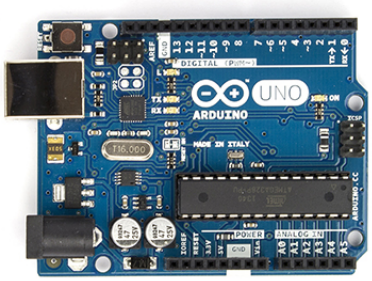

Figure 2.1: Arduino Uno board[2].

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.

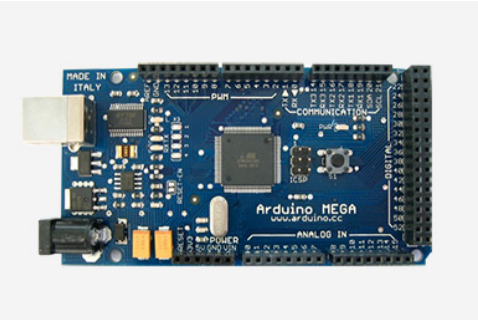

Figure 2.2: Arduino Mega board[3].

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 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.

5. Technologies

We shall look at some existing technologies now.

5.1 Mesh 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

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] IEEE Spectrum. *The making of Arduino*. Seen 16/09/2015. URL: http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino.

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

[3] Arduino. *Arduino - ArduinoBoardMega*. Seen 16/09/2015. URL: https://www.arduino.cc/en/Main/arduinoBoardMega.

Part IV

Appendix