UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science

Lauris Kruumäe

Thesis Tittle

Bachelor Thesis (6 EAP)

Supervisor: his/her name, degree
Co-supervisor: his/her name, degree

Author:..... "....."Month 201X

Supervisor:..... "....."Month 201X

Professor:..... "....."Month 201X

TARTU, 2013

Abstract

Nowadays, mobile applications are becoming more context aware due to technological achievements which enable the applications to anticipate users' intentions. This is achieved through using the device's own micromechanical artifacts that can be used to perceive the environment. However, this is constrained to the hardware limitations of devices.

A proposed solution for this has been made in the thesis "Context Sensor Data on Demand for Mobile Users Supported by XMPP" by Kaarel Hanson. The solution is to use XMPP for transporting sensor data from Arduino microcontroller to the cloud. Arduino provides low-cost hardware, while the cloud offers the reliable and high- availability means for storing and processing sensor data.

This solution shows that running on a 9V battery the microcontroller lasts for 101 minutes when using an Ethernet module for communications, and 161,5 minutes with a WiFi module. These results are not good enough for remote data collection with limited access to the microcontroller.

This thesis proposes an optimisation for the system so that instead of reading and sending sensor data every 10 seconds, the cloud server would notify the controller when to start sending data and when to stop. This means implementing an algorithm for detecting similar sensor data readings and notifying the microcontroller of needed operations. With similar readings, the microcontroller could be put to an idle state for limiting power consumption, which would prolong battery life.

The aim is to optimise the sensor reading process enough to prolong the Arduino microcontroller's battery life on a 9V battery.

Sisukord

1

Introduction

1.1 Introduction

Briefly summarize the question (you will be stating the question in detail later), and perhaps give an overview of your main results. (it is not just a description of the contents of each section)

1.1.1 Motivation

Some of the reasons why it is a worthwhile question.

1.1.2 Contributions

Solution developed - (e.g. algorithm, tools, etc.)

1.1.3 Outline

Brief introduction of each chapter

2

State of the Art

The state of the art used in the thesis highlighted the advances in the cloud computing domain and the mobile domain...

2.1 Jabber and XMPP

Description of Jabber and XMPP protocol and its usage.

2.2 XMPP to Cloud

2.3 Arduino

Arduino (1) is an open-source electronics prototyping platform based on a simple microcontroller board and a development environment for writing software for the board. It is intended for anyone interested in creating interactive solutions. Arduino can take inputs from a variety of sensors and control various actuators and lights. The microcontroller is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino IDE enables to choose between different board models, microcontroller programmers and communication ports.

1. add an image of the IDE here... Programs (called sketches) are written and uploaded to the board using the Arduino IDE. Each sketch must have two functions-

setup() and *loop*(). *setup*() is the first function called after Arduino is started or rebooted. It is called once and afterwards the function *loop*() is called consecutively until the board is stopped, restarted or crashes. When a crash occurs, the program is restarted, which means calling *setup*() again.

Since programs are written in C/C++, there are a lot of libraries available for use.

2.3.1 Arduino Mega ADK

Arduino Mega ADK (2) is one of the most capable boards available. The Arduino ADK is a microcontroller board based on the ATmega2560. Similar to the Mega 2560 and Uno, it features an ATmega8U2 programmed as a USB-to-serial converter. It has a USB host interface to connect with Android based phones, 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, USB B, micro B connections and a 2.1mm center-positive power jack. The ADK is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove

The ADK has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

The Arduino ADK can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter or battery. The board can operate on an external supply of 5.5 to 16 volts. The recommended range is 7 to 12 volts.

2. format the text a bit, remove describe what is EEPROM, add an image

2.3.2 Wireless SD Shield

The Wireless SD shield (3) allows an Arduino board to communicate using a wireless module. The module can communicate up to 100 feet indoors or up to 300 feet outdoors.

The shield has an on-board switch which allows to select between USB and Micro modes. In USB mode, the shield bypasses Arduino board's microcontroller

and communicates directly to the USB-to-serial converter. In Micro mode, data sent from the microcontroller will be transmitted to the computer via USB as well as being sent wirelessly by the wireless module. The microcontroller will not be programmable via USB in Micro mode.

2.3.3 RN-XV WiFly Module

The RN-XV module (4) by Roving Networks is a certified Wi-Fi solution especially designed for customer who want to migrate their existing 802.15.4 architecture to a standard TCP/IP based platform without having to redesign their existing hardware. In other words, if your project is set up for XBee and you want to move it to a standard WiFi network, you can drop this in the same socket without any other new hardware.

The RN-XV module is based upon Roving Networks' robust RN-171 Wi-Fi module and incorporates 802.11 b/g radio, 32 bit processor, TCP/IP stack, real-time clock, crypto accelerator, power management unit and analog sensor interface.The module is pre-loaded with Roving firmware to simplify integration and minimize development time of your application. In the simplest configuration, the hardware only requires four connections (PWR, TX, RX and GND) to create a wireless data connection.

3. Add an image of the wireless shield and rn-xv wifi module

2.3.4 TinkerKit

TinkerKit (5) is a tool used to build interactive products using Arduino boards. It consists of modules (sensors, actuators) and a sensor shield. The tool greatly simplifies product assembly, because instead of building circuits out of low level components, all the modules can be attached to the TinkerKit sensor shield with a snapping cable.

Here is the mega sensor shield used in this project... **4. Add a figure of the mega sensor shield**

The modules are divided into sensors and actuators.

5. Thermistor Module, Light Dependent Resistor Module,Hall Sensor Module

2.4 Fuzzy Logic

2.4.1 Fuzzy Set

2.4.2 Fuzzy Control Systems

Description of fuzzy logic and its uses.

3

Problem Statement

3.1 Current Solution

The current solutions uses three main components:

1. Arduino sensor module
2. OpenFire XMPP server
3. Data collection web server in the cloud

Both the data collection server and Arduino module are XMPP clients. The XMPP OpenFire server runs in the cloud and provides XMPP communications to both clients. Clients connect to the same chat where the server listens for messages from the sensor module. When a message is received by the server, sensor data is parsed from it and saved in a database.

3.1.1 Arduino

6. describe the current code The current Arduino implementation initializes the connection to the OpenFire server and XMPP connection in the *setup*() function. In the *loop*() function, connections are checked (and reconnected if dropped). The implementation checks if 10 seconds has passed since the last transmission and send the data again if needed.

The problem with this implementation is that *loop*() is continuously called and the amount of time since last transmission is checked to determine when a new message should be sent. Since the transmission interval is 10 seconds, the sensor module consumes needless power for that time period.

3.1.2 XMPP Communication

7. describe the communication between arduino and xmpp server Description of the communication between the Arduino board, XMPP server and data collection server.

8. xmpp initialization time

9. problems with the wifly module, etc

3.1.3 Power Consumption

10. graphs with power consumption with different configurations Show how much power is consumed with the current implementation.

4

Problem Solution

Transition - Several are the issues that were discussed in previous chapter, regarding the use of cloud services from the mobile...

4.1 Arduino Improvements

Description of improvements made to the current solution

4.1.1 Software

List of software improvements and their effects on power consumption

4.1.2 Hardware

Might not include this...

4.2 Communication

Description of improvements made to the communication between the Arduino board and the server.

4.2.1 Fuzzy Logic Implementation

Description of the fuzzy logic implementation used.

4.2.2 Communication Interval

Effects of the improvements on power consumption, etc

5

Conclusions

5.1 Conclusions

5.2 Summary of Contributions

6

Related Work

Compare your solution with existing projects. How your solution is better than the others?, why to use your solution?, etc.

7

Future Research Directions

Briefly indicate how your current research can be extended, some improvements, etc.

8

Sisukokkuvõte

Eesti abstract...

Kirjandus

[1] Arduino - HomePage, http://www.arduino.cc/. URL http://www.arduino.cc/ 2

[2] Arduino - ArduinoBoardADK, http://arduino.cc/en/Main/ArduinoBoardADK. URL http://arduino.cc/en/Main/ArduinoBoardADK 3

[3] Arduino - ArduinoWirelessShield, http://arduino.cc/en/Main/ArduinoWirelessShield. URL http://arduino.cc/en/Main/ArduinoWirelessShield 3

[4] RN-XV WiFly module - wire antenna - SparkFun electronics, https://www.sparkfun.com/products/10822. URL https://www.sparkfun.com/products/10822 4

[5] 4

ToDo