Group 6
Pond Environmental Measurement
SSNS - Smart Sensor Network Systems

Alexander K., Sabrina B., Rozana A., Alexander V.D.

Fachhochschule Frankfurt am Main - University of Applied Sciences

June 19, 2013

Table of Contents

1. Project Describtion

1.1 Idea

1.2 Requirements

2. Function Point Analysis

2.1 Defining the Unadjusted Function Point Count

2.2 Determinig the Value Adjustment Factor

2.3 Determining Function Points

3. System Architecture

3.1 The Sensor

3.2 ZigBee Network

3.3 The Collecting Node

3.4 The Measuring Nodes

3.5 Application

Project

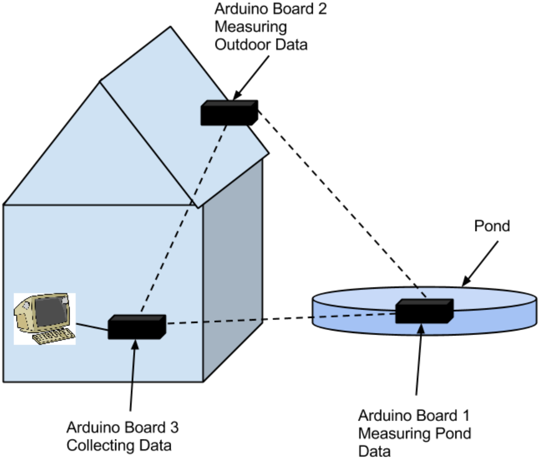


Figure: Model

System Requirements

- reliable 24/7 data Aquisition
- Data being storaged on the collecting Node
- Graphical visualization on the PC

Function Points

1. Defining the Unadjusted Function Point Count
2. Determining the Value Adjustment Factor
3. Determining Function Points

Defining the Unadjusted Function Point Count

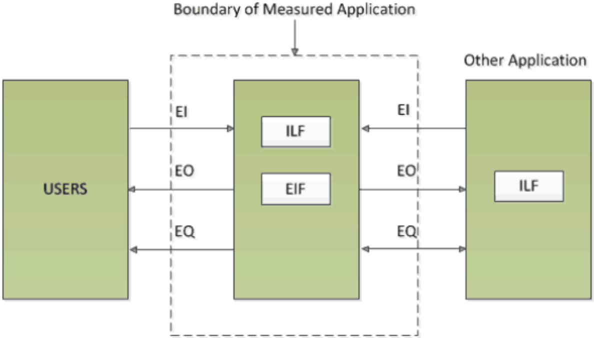
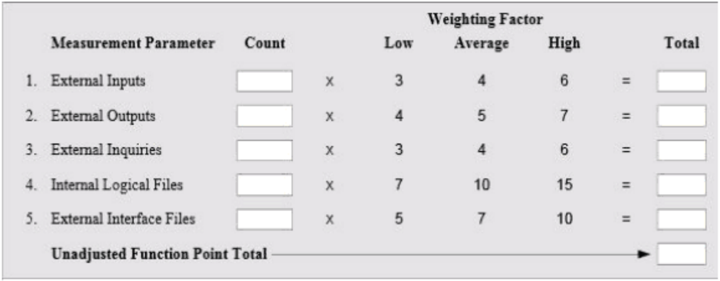


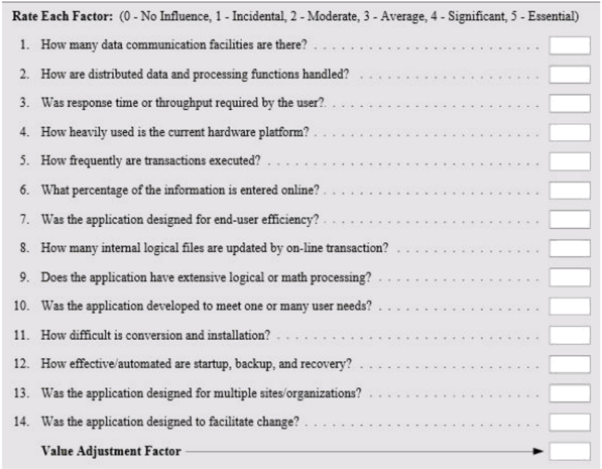
Figure: Boundary of MA

Unadjusted Function Point Count and Multipliers



				Weighting Factor			
Measurement Parameter	Count		Low	Average	High		Total
1. External Inputs		x	3	4	6	=	
2. External Outputs		x	4	5	7	=	
3. External Inquiries		x	3	4	6	=	
4. Internal Logical Files		x	7	10	15	=	
5. External Interface Files		x	5	7	10	=	
Unadjusted Function Point Total →							

Determinig the Value Adjustment Factor



Rate Each Factor: (0 - No Influence, 1 - Incidental, 2 - Moderate, 3 - Average, 4 - Significant, 5 - Essential)

1. How many data communication facilities are there? . . . . □
2. How are distributed data and processing functions handled? . . . . □
3. Was response time or throughput required by the user? . . . . □
4. How heavily used is the current hardware platform? . . . . □
5. How frequently are transactions executed? . . . . □
6. What percentage of the information is entered online? . . . . □
7. Was the application designed for end-user efficiency? . . . . □
8. How many internal logical files are updated by on-line transaction? . . . . □
9. Does the application have extensive logical or math processing? . . . . □
10. Was the application developed to meet one or many user needs? . . . . □
11. How difficult is conversion and installation? . . . . □
12. How effective/automated are startup, backup, and recovery? . . . . □
13. Was the application designed for multiple sites/organizations? . . . . □
14. Was the application designed to facilitate change? . . . . □

Value Adjustment Factor ――――――――――→ □

Figure: Total Degree of Influence

Determining Function Points

Project	Function Points	Man-Months
ASD	11	1
KWO	24	2
RMD	53	5
WBO	72	6

Determining Function Points

Project	Function Points	Man-Months
ASD	11	1
Arduino	22	1.2
KWO	24	2
RMD	53	5
WBO	72	6

The Sensors

We are using two Sensors:

- Temperature Sensor TMP36
- Light Dependant Resistor GL5528

Temperature Sensor

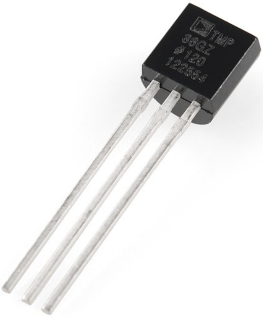


Figure: TMP36

Temperature Sensor

Following Specification:

- outputs voltage depending on the temperature
- relation is linear
- Temperature Range: $-40°C$ to $125°C$
- scalefactor of $10mV/°C$
- Accuracy of $\pm 1°C$ at $25°$ and $\pm 2\%$ in the range of $-40°C$ to $125°C$

Temperature Sensor

To calculate the Temperature in °Celsius we use the formula:

$$Temp = \frac{Voltage - 500}{10}$$

Light Dependant Resistor

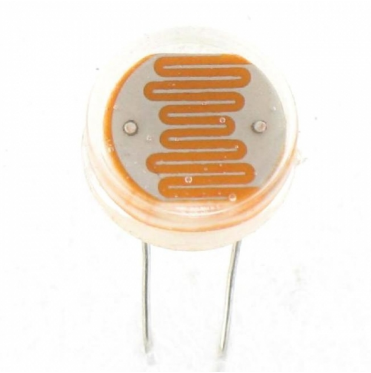


Figure: GL5528

Light Dependant Resistor

Following Specification:

- not precise enough to measure the light level
- only measure darkness from lightness
- Reliable perfomance
- linear relation

Light Dependant Resistor

To get the light value in Lux you have to do following steps:

1. Get Voltage of Resistor
2. Get Resistor Value with formula:

$$\frac{5.0 - LightVoltage}{lightv} * 10000$$

3. Get Lux with formula:

$$10 * \frac{14000}{LightResistor}^{\frac{1}{0.7}}$$

Error Calculation

- calculate quantisation error
- Temperature: ± 1 Degree
- Light: too high → only measure dark or bright of light

ZigBee Network

- The Nodes in our WSN communicate in a ZigBee Network
- ZigBee Networks need a coordinator. The Collector will be the coordinator.
- The measuring Nodes will function either as End-Nodes or Routers
  - For the programming of the Nodes, this is irrelevant

- As Radio Modules we are using XBee Modules from Digi
- They are attached to the Arduino's using Wireless Shields.
- To Adress the XBee Modules in the Software, we use the xbee-arduino Library
- The ZigBee adress of the coordinator will be hardcoded into the measuring Nodes Software
- The ZigBee Adress of the Measuring Nodes will be hard coded into the Coordinators Software

Communication

- The coordinator will request Measurements from the Measuring Nodes
- The Message looks like this:

Byte(s)	content	meaning
1	'R'=0x52	identifier for Measurement Request

- On Request, the Pond Measuring Node will Respond by this Message:

Byte(s)	content	meaning
1	'W'=0x57	identifier for Measurement response
2-5	float	float for temperature measurement in Celsius
6-9	float	float for light intensity in Lux

- The Weather Measuring Node Responds with this Message:

Byte(s)	content	meaning
1	"P'= 0x50	identifier for Pond measuremnt response
2-5	float	float for temperature measurement in Celsius

The Collecting Node

- The collecting Node will take the following Responsibilities
  - ZigBee coordinator Role
  - Know Time by using NTP
  - Request and Receive Measures and store them
  - act as TCP Server, providing stored Data to clients
- The collector consists of an Arduino Ethernet, with the same Wireless Shield and XBee Module like the other Nodes, and an attached SD Card

The Measuring Nodes

- Measurement Kit → Arduino Uno + Wireless Shield + XBee Module + Sensor Module
- Sensor Module is different for Pond and Weather Measurement Station
- act as an ZigBee EndNode
- check if Coordinator has sent request
- if true send response to Coordinator

The Measuring Nodes

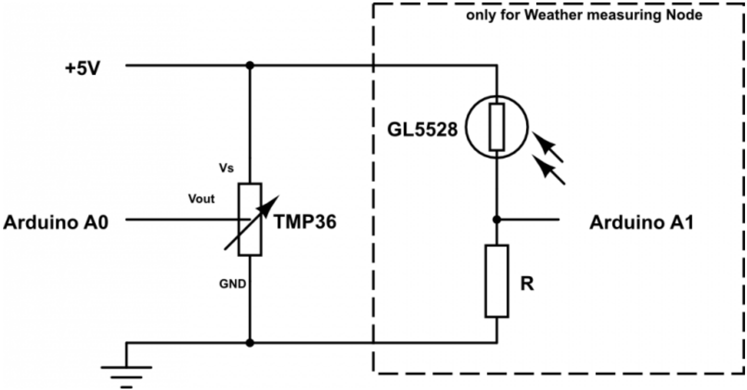


Figure: Measuring Node Circuit

Behaviour

- on startup, and every 24 hours, the collector will synchronize its time via NTP
- every full and half hour, the collector will request a Measurement from the Measuring Nodes
- The measuring Nodes have 30 Seconds to respond, otherwise their measurement is ignored
- After 30 seconds, or when the Measuring Nodes all answered, the measurements get stored.

Data Storage

- The collector stores the Data in a File on its SD Card
- The File is a CSV-File (Coma separated Values) with the following Format:
- **hh,mm,ss,dd,mm,yyyy,pppp,aaaa,llll**
- Each Line represents a complete measurement of the system
- New measurements add lines to the File

- missing values will be left out (but comas stay)
- Example:
- **00,30,15,13,05,2013,11.5,9.7,**
- Means: At 00:30:15 on the 13th of May 2013, the Pond Temperature was 11.5C, the air temperature was 9.7 C, and the light level was unknown

Application

- The application is used to access the collected data from the WSN
- The Data is displayed in a Table
- The Data can then be exportet as the same CSV File as stored on the collector
- The exportet File can then be used in other applications like GNUPlot or SciLab.

Thank you for your attention!