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

July 7, 2013

Table of Contents

Project

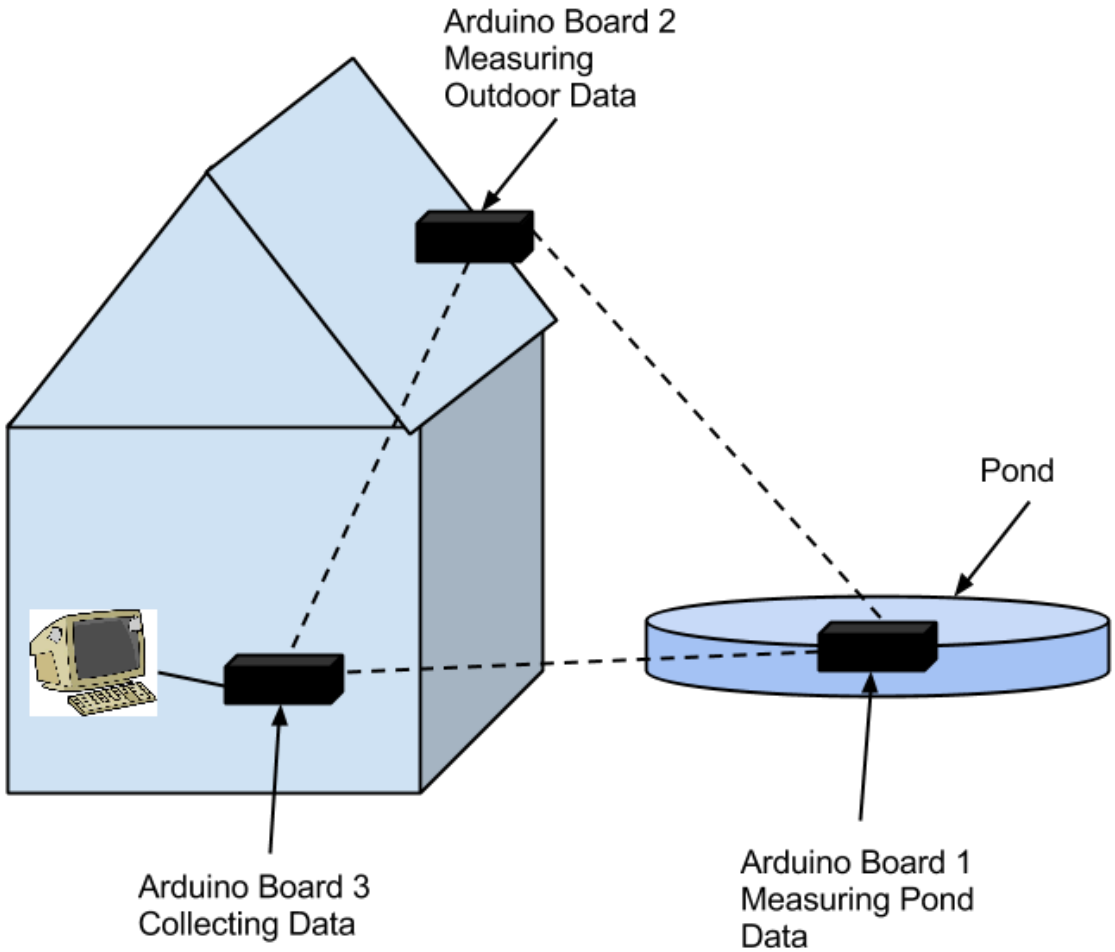


Figure: Model

System Requirements

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

Project Plan

Hardware

Hardware

- Ardiuno UNO module
- Arduino Wireless SD
- XBee ProSeries 2XBP24-Z7 module
- The Sensors

IDE installation

Arduino UNO module

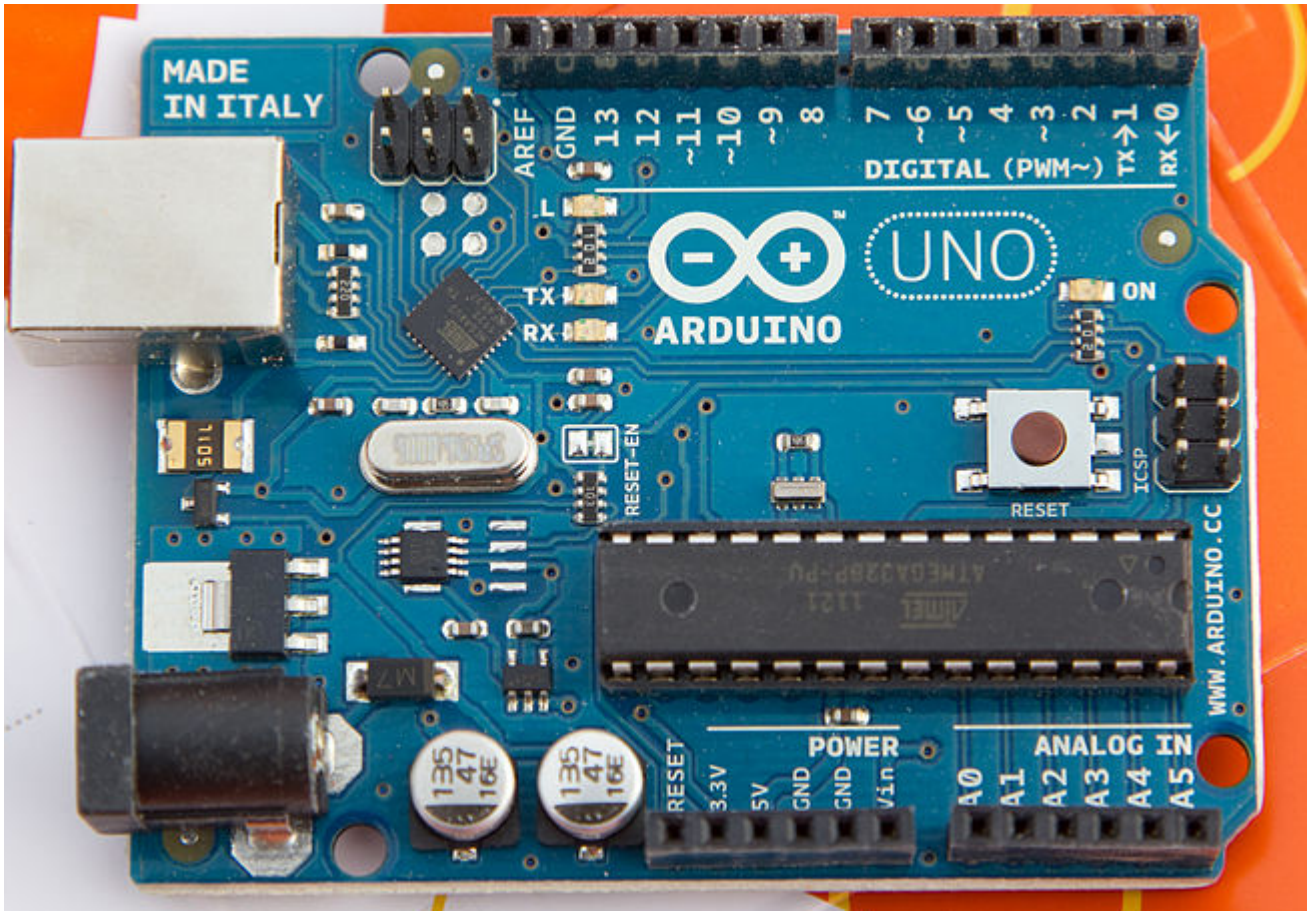


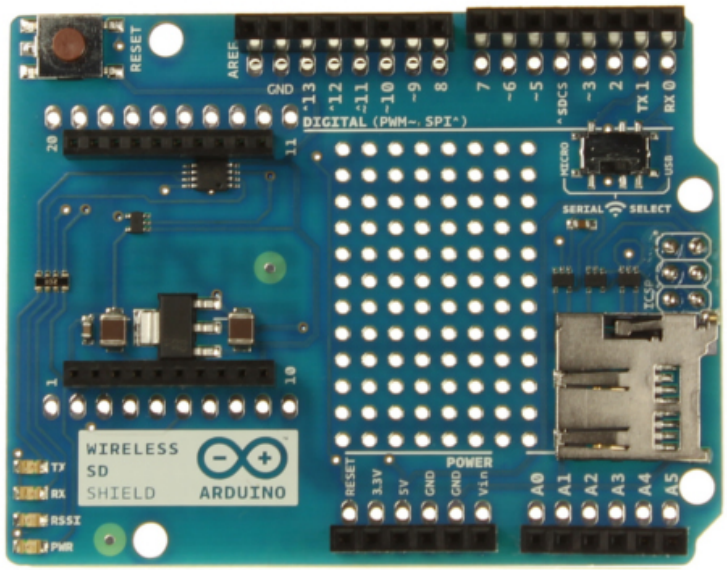
Figure: Arduino Uno

Arduino UNO module

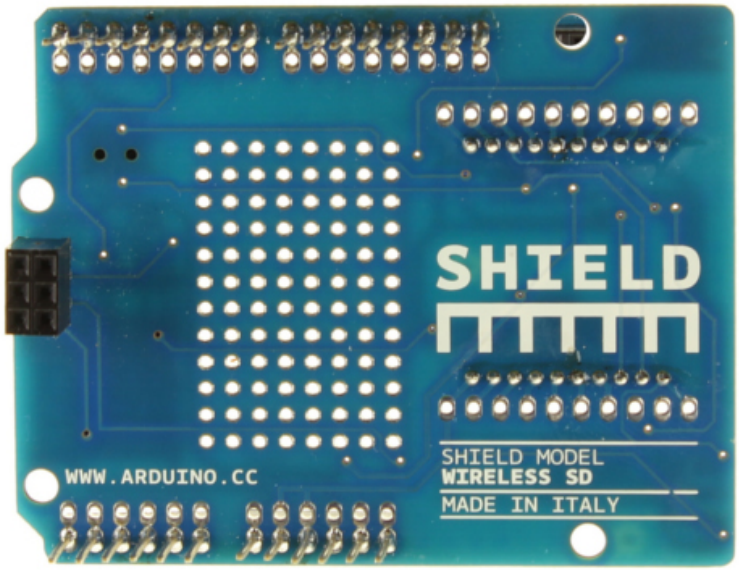
- An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits.
- An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as shields.

Wireless SD Shield

Wireless SD Shield



Wireless SD Shield Front



Wireless SD Shield Back

Figure: Arduino Wireless SD Shield

Wireless SD Shield

- The Wireless SD shield allows an Arduino board to communicate wirelessly using a wireless module.
- It is based on the Xbee modules from Digi, but can use any module with the same footprint.
- The module can communicate up to 100 feet indoors or 300 feet outdoors (with line-of-sight).
- It can be used as a serial/usb replacement or you can put it into a command mode and configure it for a variety of broadcast and mesh networking options.
- The shields breaks out each of the Xbee's pins to a through-hole solder pad.

XBee ProSeries 2XBP24-Z7 module

(a) Xbee Module front

(b) Xbee Module back

XBee ProSeries 2XBP24-Z7 module

1. Allow to complex mesh networks based on the XBee ZB ZigBee mesh firmware.
2. These modules allow a very reliable and simple communication between microcontrollers, computers, systems, really anything with a serial port!
3. Point to point and multi-point networks are supported.

Integrated developement environment

The Arduino IDE is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects.

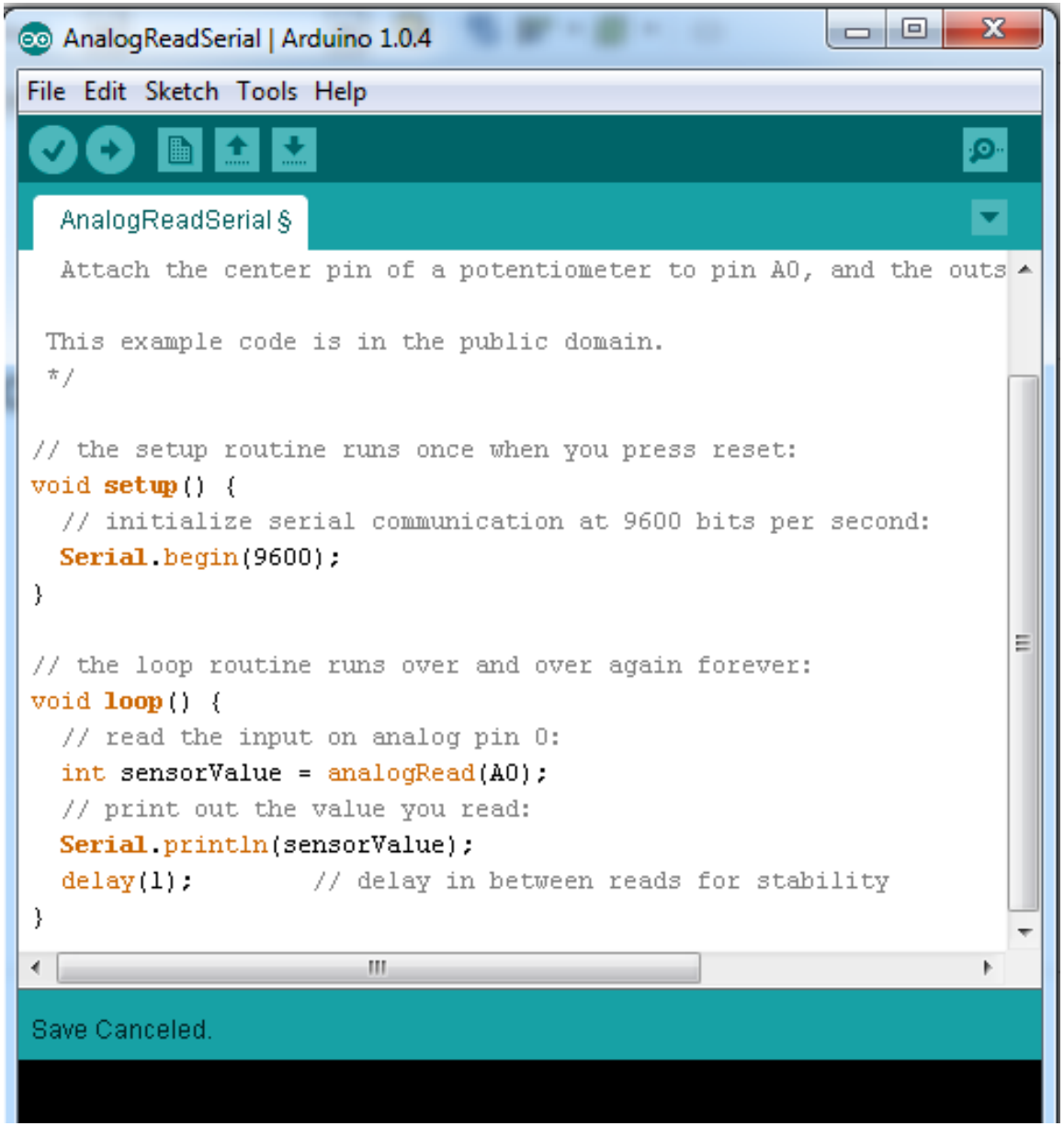
Integrated developement environment

It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit makefiles or run programs on a command-line interface.[citation needed] A program or code written for Arduino is called a sketch.

Integrated developement environment

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier.

Integrated developement environment



Integrated developement environment

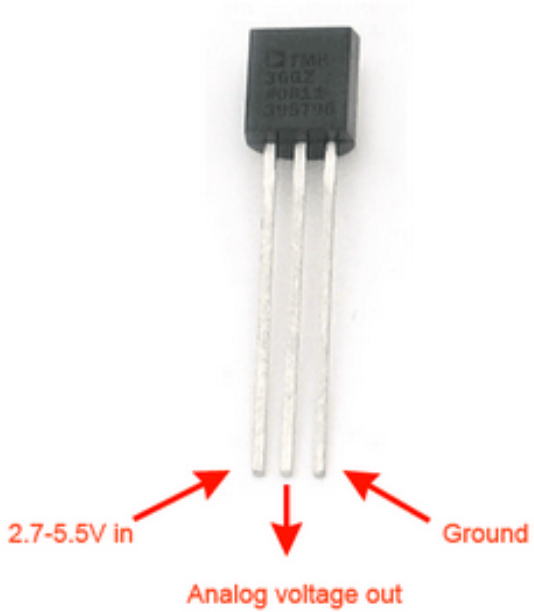
Users only need define two functions to make a runnable cyclic executive program:

- setup(): a function run once at the start of a program that can initialize settings
- loop(): a function called repeatedly until the board powers off

The Sensors

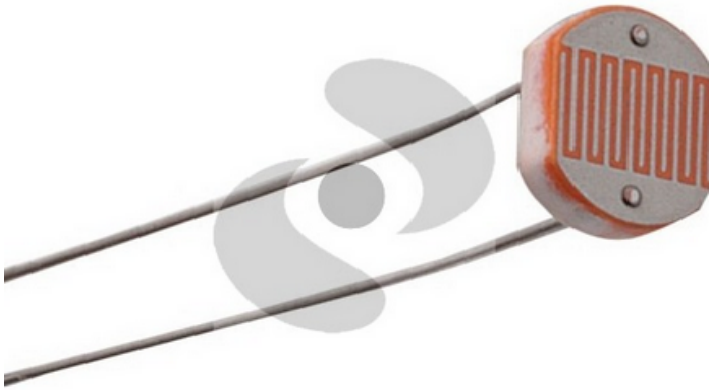
- Temperature Sensor TMP36
- LDR GL5528

Temperature Sensor TMP36



1. These sensors use a solid-state technique to determine the temperature.
2. They use the fact as temperature increases, the voltage across a diode increases at a known rate.
3. By precisely amplifying the voltage change, it is easy to generate an analog signal that is directly proportional to temperature.

LDR GL5528



- GL5528 Photo Resistor (Light Depend Resistor). Photoresistor is a resistor which made of semi-conductor material,and the conductance changes with luminance variation.
- Applications:Photometry, Photoelectric Control, Light control lamp, Light control switch, Electronic Toy

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

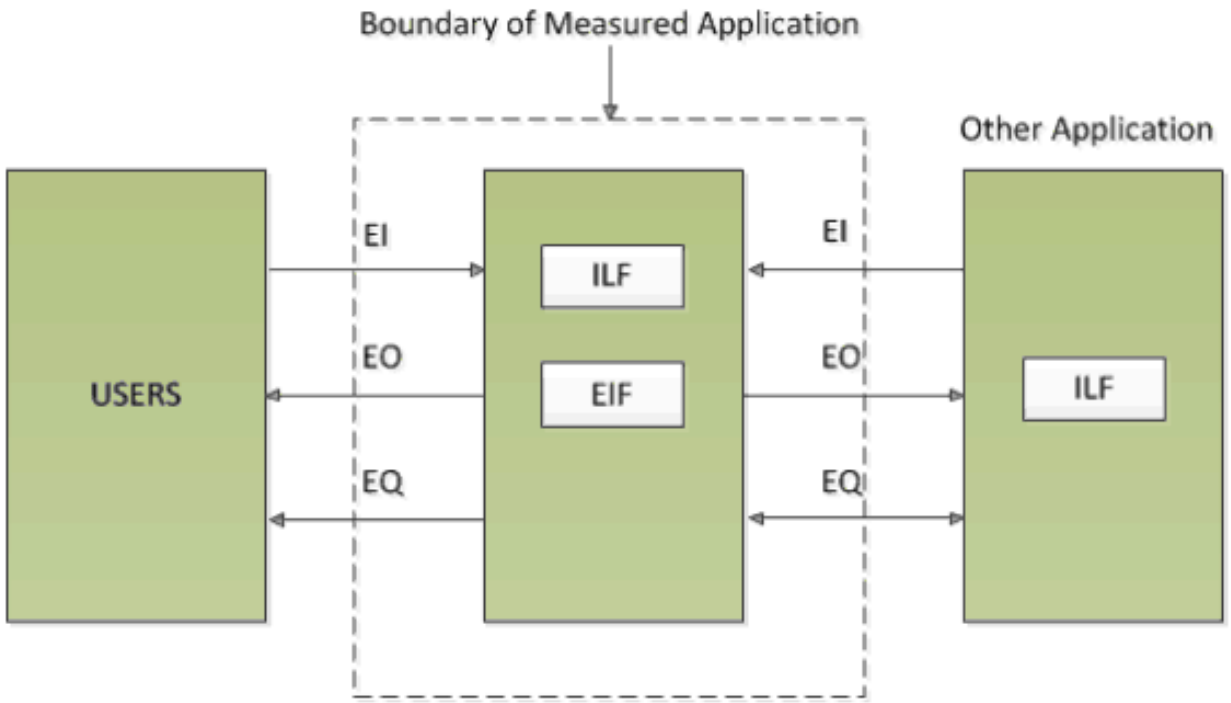


Figure: Boundary of MA

Unadjusted Function Point Count and Multipliers

	Measurement Parameter	Count		Weighting Factor 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

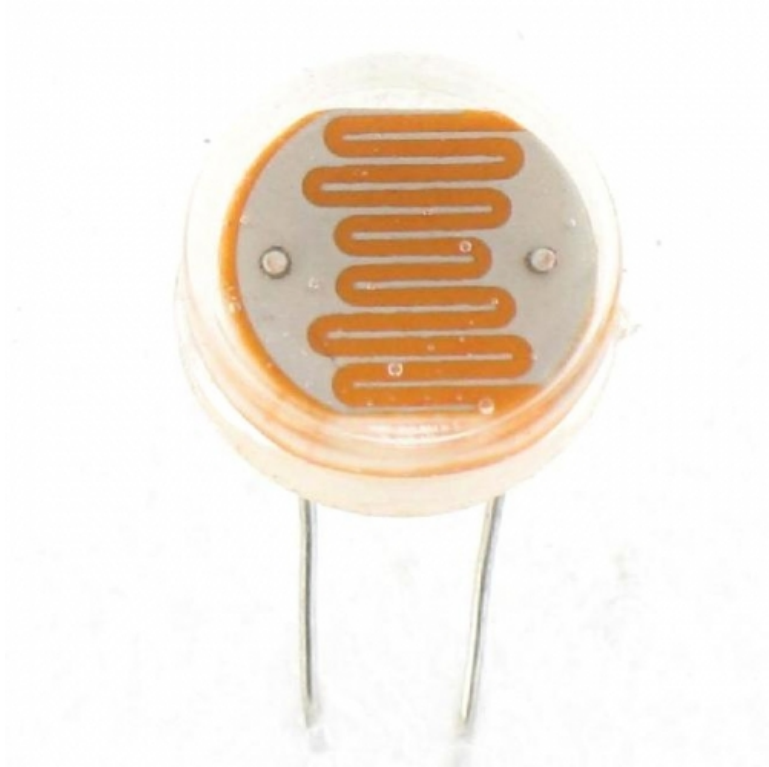


Figure: GL5528

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 $\rightarrow$ 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 takes the following Responsibilities:
  - ZigBee Coordinator Role
  - Know Daytime and Date by using NTP
  - Request and receive Measurements
  - Store all Data
  - act as TCP Server, providing stored Data to Clients
- The collector consists of:
  - Arduino Ethernet
  - Wireless Shield
  - XBee Module
  - micro SD-Card

Behavior

- on startup, and every 24 hours, the collector will synchronize its stored daytime vie 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 answer will be ignored
- After 30 seconds, or when both Measuring Nodes answered, the measurements get stored

Software Architecture

- Since the Collecting Node takes several responsibilities, it is necessary to partition the Software into components
- This components group specific kinds of functionalities.
- Those components are realised by implementing so called Arduno Libraries
  - The Arduini Plattform allows the developer to write his own Libraries.
  - A Library consists of a class, that can be instantiated in the main sketch
  - Arduino Libraries are a way of using OOP-Techniques in the otherwise simplified Arduino Programming Language

Software Components

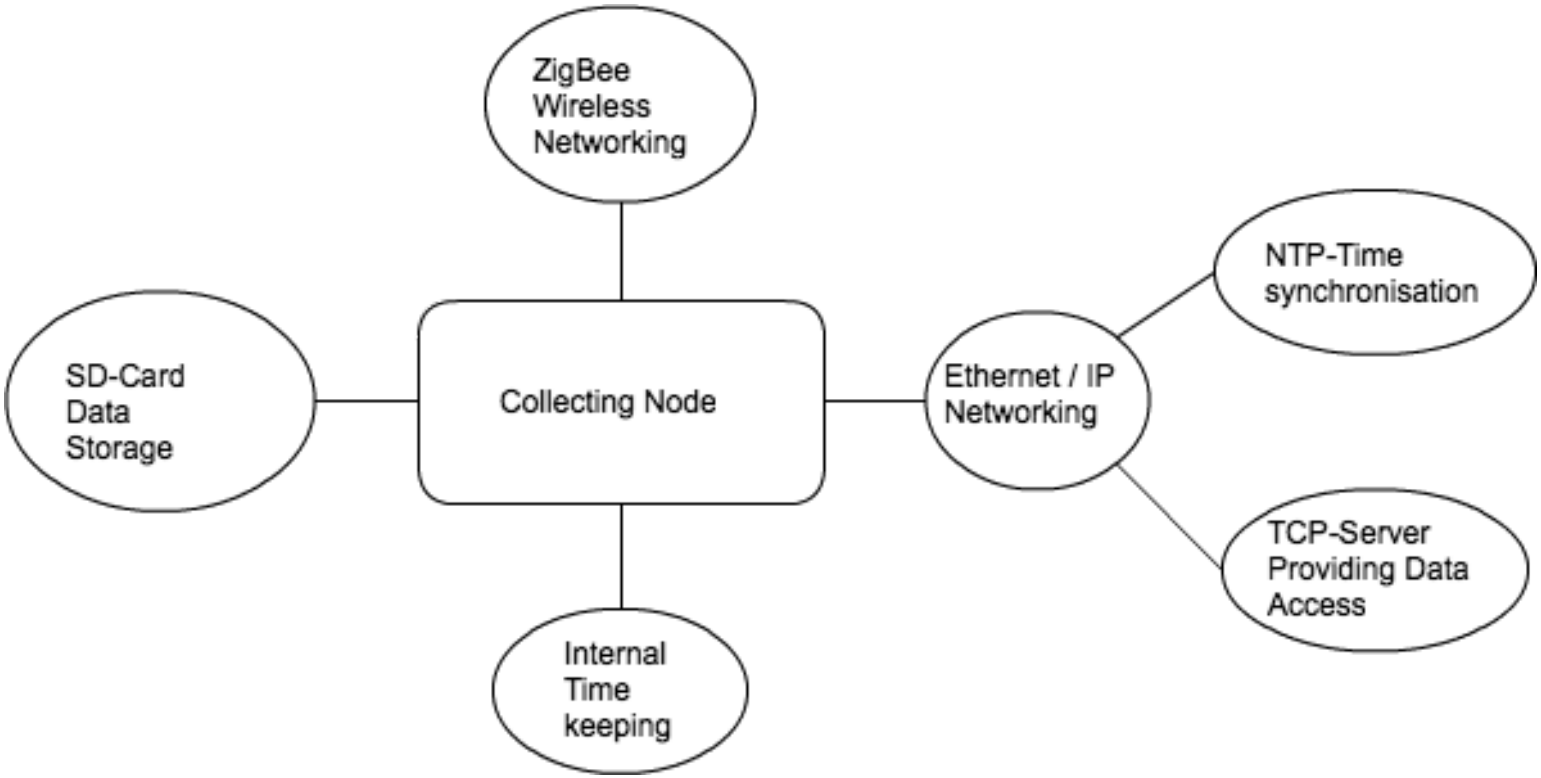


Figure: Collector: Software Components

Data Storage

- The collector stores all 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 in this file represents a complete measurement of the System at a certain point in time
- New Measurements add lines to the file

Data Storage

- A missing Value might be left out (But all comas stay to preserve the format)
- 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 9.7C, and the light level was unknown

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

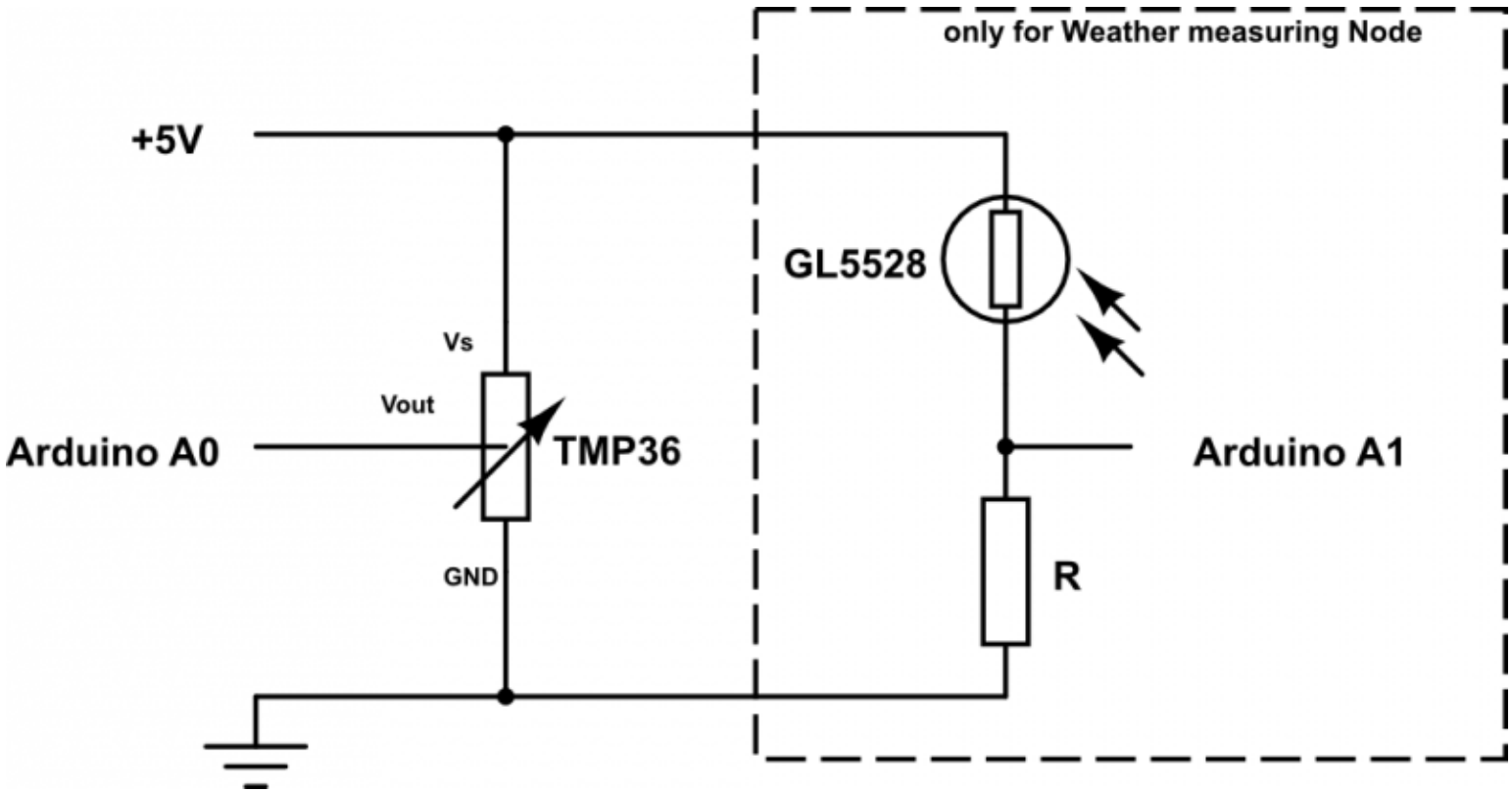


Figure: Measuring Node Circuit

Weather Node

- outside node
- measure Temperature and Light Intensity
- Send data via payload in a tx packet to Collector Node

Pond Node

- inside node
- measure Temperature in a pond
- Send data via payload in a tx packet to Collector Node

Application

- The application is used to access the collected data from the WSN
- The Data is displayed in a Textarea and Graphs
- 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.

Collecting Node

- The Collecting Node has the most complex architecture
- unfortuantly the Collecting Node could not be finished on time
  - some of the Collecting Nodes Software Components are finished (SD-Card Handling, Time and Date Awareness)
  - but other Components could not be implemented
  - The problem with implementing those Components is related to the Arduino Developement Environment

Problems with the Arduino IDE

- The Arduino Language & IDE are made to be simple and easy to learn for Beginners
- But those simplifications also enforce limitations for the developer
  - Writing Libraries by using other libraries is problematic $\rightarrow$ IDE does not link in the necessary object files
  - there is no way of debugging Arduino sketches

Problems with the Arduino IDE

- Professionell IDEs for Microcontrollers provide possibilities to debug code that is running on the hardware
- This allows access to the processors for example registers or memory
- It is possible to comprehend what is going wrong
- Without Debugging, the possibilities to understand not working code are very limited
- It took too much time, so the Collecting Node could not be finished on time

Demonstration

Thank you for your attention!