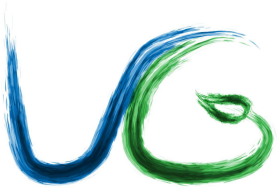
Sprint 1



Urban Green

Matthias Schwebler
Ramin Bahadoorifar
Samuel Schober
Konrad Kelc

	Name	**Datum**	**Unterschrift**
Erstellt:	M. Schwebler, R. Bahadoorifar	TT.MM.YYYY	
Geprüft:	S. Schober, K. Kelc	TT.MM.YYYY	

Contents

Datum	Version	Autor	Status
TT.MM.YYYY	0.1	Matthias Schwebler, Ramin Bahadoorifar, Konrad Kelc, Samuel Schober	Erstentwurf
TT.MM.YYYY	1.0	A B, X Y	Fertigstellung
TT.MM.YYYY	1.1	A B, X Y	Überarbeitung

1 Datenmanagement

1.1 Verwaltung der lokalen Daten - User Story 1571 u. 2097

Um alle Daten der Sensoren Abzulesen wird die Serielle Schnittstelle verwendet. Der C-Code der auf dem Arduino ausgeführt wird, sendet alle Informationen an den Raspberry Pi. Dieser muss lediglich auf der Schnittstelle auf eingehende Daten warten, was mit einem Python Script realisiert wird. Hierbei werden die Libraries "redis", "serial" und "time" verwendet. Damit es zu keinen Problemen beim Auslesen kommt, muss nach jedem Lesevorgang 0.1 Sekunden pausiert werden. Falls dies nicht eingehalten wird, wird beispielsweise "temperature" zu "temp" und "erature". Der folgende Code ist ein Beispiel, wie dauerhaft Daten ausgelesen und in die Datenbank gespeichert werden können.

```
1  import redis
2  import serial
3  import time
4
5  # init serial
6  # TODO 'ttyACM0' needs to be changed to whatever the
       raspberry Pi's USB-port is called
7  ser = serial.Serial(
8    port='/dev/ttyACM0',
9    baudrate=115200
10 )
11 # init redis
12 r = redis.StrictRedis(host='localhost', port=6380,
       db=0)
13
14 data = ""
15 while True:
16   bytesToRead = ser.inWaiting()
17   data = ser.read(bytesToRead)
18   if data != "":
19     kv = data.split(' ',data)    # kv... KeyValue
20     r.hset('system', kv[0], kv[1])    # write data to
         redis
21     time.sleep(0.1)
```

1.2 Übertragung zum zentralen Server - User Story 1984

Um die lokal gespeicherten Daten zum zentralen Server zu übertragen, wird wieder ein Python Script verwendet. Da auf dem Server MongoDB verwendet wird, muss zuerst auf dem lokalen SBC die entsprechende Library installiert werden. Dies geschieht mit folgendem Befehl:

```
pip install pymongo
```

Im Python Script muss zuerst die Verbindung zu den beiden Datenbanken hergestellt werden:

```
# init mongodb and redis
mongoClient = MongoClient('vps336521.ovh.net', 27017)
r = redis.StrictRedis(host='localhost', port=6380,
    db=0)

# get database
db = mongoClient.urbangreen

# get collection
collection = db['sensor-data']
```

Danach kann mit folgendem Code ein neuer Datensatz angelegt werden:

```
newData = {
"temperature": r.hget("system", "temperature"),
"ph": r.hget("system", "ph"),
"ec": r.hget("system", "ec")
}
collection.insert_one(newData)
```