

Geodätisches und Geotechnisches Monitoring mit Low-cost-Komponenten

PROF. WERNER STEMPFHUBER

Seit der Markteinführung der zielverfolgenden Tachymeter Anfang der 90iger Jahre und der etwa gleichzeitigen Verfügbarkeit von GPS- bzw. GNSS-Empfängern haben sich die geodätischen und geotechnischen Monitoringanwendungen stetig weiterentwickelt. Neben den kommerziellen Systemen der bekannten Hersteller existieren verschiedene Soft- und Hardwarelösungen einzelner Büros, Organisationen und Hochschulen. Heutzutage werden verstärkt geodätische und geotechnische Low-cost-Sensoren und -Komponenten für permanente Überwachungsaufgaben verwendet. In Kombination mit kleinen Micro-Controllern und Micro-Prozessoren (z.B. dem Raspberry Pi oder die verschiedenen Typen der Arduino-Boards) ergeben sich individuelle Möglichkeiten in der Ingenieurvermessung und Geotechnik. Modular vernetzte Sensoren in unterschiedlichen Netzwerken, meist als Geosensornetzwerke bezeichnet (HEUNECKE ET AL. 2013), arbeiten dabei mit geringen Ressourcen optimal abgestimmt zusammen. Dabei kann eine große Anzahl von Subsystemen in einem zentralen Mess-, Steuer- und Auswertesystem für individuellen Anforderungen zusammengestellt werden. Dieser Beitrag stellt einen holistischen Ansatz für flexible und kostengünstige Anwendungen zur automatisierten Überwachung von technischen Anlagen und Bauwerken nach Vorgaben des Bauingenieurwesens sowie die Durchführung von Überwachungsprojekten in den verschiedenen Bereichen der Geowissenschaften dar. Zusätzlich werden verschiedene Implementierungen mit einzelnen Sensoren mit individueller Konfiguration des gesamten Geosensornetzwerkes erläutert. Der Schwerpunkt des Beitrags liegt neben den geodätischen Komponenten wie Tachymetrie, Low-Cost-GNSS, Neigungssensoren und Nivellement in der Verwendung der geotechnischen Sensoren und deren Echtzeitverarbeitung mit dem Raspberry Pi und dem Arduino Uno.

Schlüsselwörter – Geodätische Messtechnik, Geotechnik, Geosensornetzwerke, Raspberry Pi, Linux, Python, Modulare Komponenten, Client-Server-Setup, Low-cost, Open-Source

1. Einleitung

Durch die technischen Entwicklungen der geodätischen Vermessungsgeräte, wie zielverfolgende Tachymeter, GPS-/GNSS-Empfänger, automatisierte Digitalnivelliergeräte, Neigungssensoren, 3D-Scansystem etc. wurden seit über 25 Jahren automatisierten Überwachungsmessungen stetig weiterentwickelt. Klassische Anwendungsbereiche der Überwachung sind technische Anlagen und Gebäude, geowissenschaftliche Anwendungsbereiche wie Hangrutschungen, Erdbebenmonitoring oder Gletscherprojekte. Heute existieren neben den Systemen der bekannten Gerätehersteller viele individuelle Monitoringlösungen (STEMPFHUBER 2009). Dabei fokussieren sich einige Anbieter bzw. Systemhersteller auf Spezialgebiete wie z.B. GNSS basierte Monitoringsysteme (ALBERDING & STEMPFHUBER 2017). Standard-Monitoringsysteme haben gewisse Vorteile und gewährleisten einen stabilen Dauerbetrieb. Die hohen Anschaffungs- und Betriebskosten für solche Systeme und die Notwendigkeit von Zusatzmodulen sind stehen einer großen Verbreitung entgegen. Beim tatsächlichen Betrieb werden häufig nur wenige Funktionen benötigt. Die Flexibilität dieser Systeme ist häufig begrenzt und lässt Realisierung

von Spezialaufgaben nur eingeschränkt zu. Außerdem ist das Gesamtsystem in der Regel mit einem hohen Ressourcenverbrauch konzipiert (z.B. hoher Stromverbrauch, teure Hard- und Softwaremodule). Auch durch die Serveranbindung und durch die Supportleistungen entstehen weitere Betriebskosten. Deshalb sind alternative Monitoringansätze entstanden, die diese Lücken schließen. Die Hochschule Brandenburg betreibt z.B. ein Monitoring-Forschungsprojekt mit dem Projektnamen DaBaMos (EN- GEL ET AL. 2015). Das System ist flexibel einsetzbar und ebenfalls modular aufgebaut. Bei diesem Ansatz wird ebenfalls als Recheneinheit ein Raspberry Pi und unterschiedliche Low-cost-Komponenten verwendet.

Geodätische Monitoringanwendungen in der Praxis erfordern neben den administrativen Aufgaben immer auch einen hohen Aufwand an Installationsarbeiten. Diese fordert vom Monitoringsystem eine hohe Flexibilität. Zusätzlich muss der Auftraggeber auch die Datenanalyse durchführen. Das in diesem Beitrag vorgestellte Verfahren wurde in enger Zusammenarbeit mit dem Ingenieurbüro gim (Gesellschaft für Informations-Management mbh, in Berlin-Spandau) entwickelt. Das Ingenieurbüro gim von Herrn Schiefelbein ist seit vielen Jahren im Bauwerksmonitoring aktiv und hat viele Referenzprojekte im gesamten Bereich der automatisierten Bauwerksüberwachung durchgeführt (www.gim-vermessung.com).

Das Consulting-Büro Stempfhuber-Engineering bearbeitet hierbei die Basismodule der Softwaresteuerung. Diese modularen Komponenten können als intelligentes Interface (IIT) betrachtet werden und haben den Softwarenamen **RaMonA** für **Raspberry Pi Monitoring Application**. Das modular aufgebaute System kann in verschiedenen Konfigurationen installiert werden. Bei vielen Installationen werden auch Anpassungen auf die jeweilige Aufgabenstellung implementiert. Die Basis bilden immer verschiedene Raspberry Pi's als Geosensorknoten und Kommunikationskanal. Die Aufgabenzuteilung, Intelligenz des Systems und die Auswertungen sind beliebig konfigurierbar (in der Regel mit den Modulen der gim als IIT abgestimmt). Zunächst werden zwei Setups als Grundsystem im Abschnitt 1.1 skizziert.

2. RaMonA Grundsysteme

RaMonA-TS++

Für ein einfaches Monitoringsystem, z.B. zwei Tachymeter mit Temperatursensoren sind nur wenige Hard- und Softwarekomponenten erforderlich. Die RaMonA-Basismodule werden im nachfolgendem Abschnitt schematisch aufgezeigt und erläutert. Der Aufwand für ein einfaches System ist neben den Anschaffungen der Tachymeter sehr gering. Der Raspberry Pi als Einplatinencomputer mit der Pythonsoftware RaMonA-TS++ (TS für Totalstation mit Zusatzkomponenten) wird in der Abbildung 1 zusammen mit den häufig verwendeten digitalen Temperatursensoren DS18B20 als Basiskomponente dargestellt. Die gesamte Software kann auf dem Raspberry Pi installiert werden. Die Ausgleicheung der Beobachtungen kann ebenfalls als Komponente aktiviert werden. Dabei ist immer auf die Vollständigkeit der Beobachtungen zu achten.

Die zentrale Grundlage bei diesem Ansatz ist der Raspberry Pi mit dem OpenSource Linux-Betriebssystem Rasbian. Auf dem Einplatinencomputer ist bereits Python 2 und Python 3 vorinstalliert (dies ist bei nahezu allen Linux-Betriebssystemen und den IOS-Betriebssystemen so und zeigt damit die Bedeutung dieser Skriptingsprache). Die Tachymeter der Fa. Leica Geosystems haben eine geeignete Programmierschnittstelle (Application Programming Interface, API) mit dem Namen GeoCOM (die API-Dokumentation ist u.a. in Leica myWorld verfügbar). Mit dieser API können über 40.000 verschiedene Befehle an den Tachymeter gesendet werden. Diese bidirektionale Kommunikation über die serielle Schnittstelle (Optional via Bluetooth) steuert den Tachymeter und führt die individuellen Kommandos aus (STEMPFHUBER 2010). Für die GeoCOM-Schnittstelle wurden eigene Python-Bibliotheken erstellt. Neben den Schnittstellenparametern und den Grundeinstellungen sind anschließend wenige Befehle zum Anzielen und Messen erforderlich. Die Daten können in verschiedenen Datensystemen oder Datenbanken abgespeichert werden. Ein einfacher Setup wäre die Übertragung der Daten in

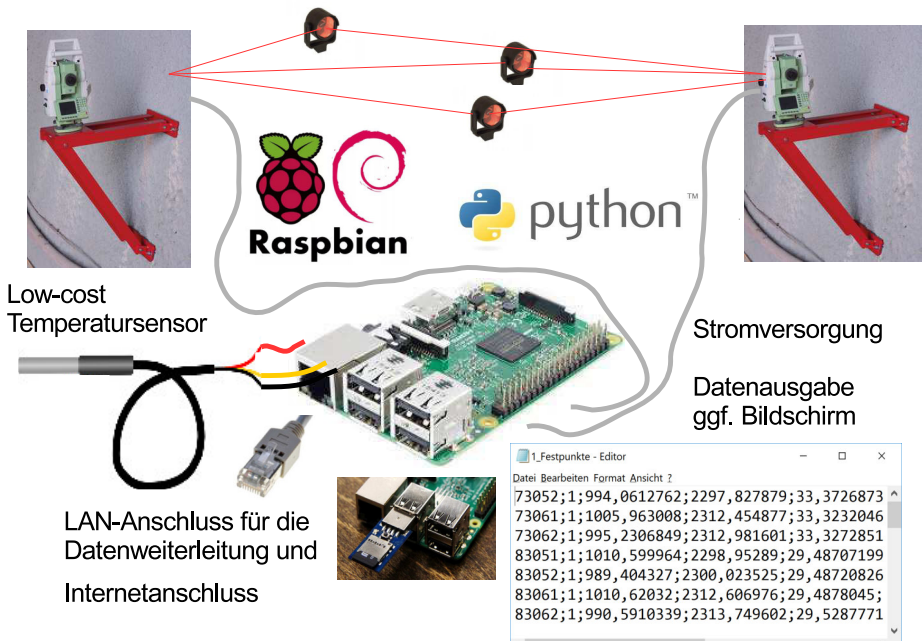


Abbildung 1: RaMonA-TS++ (einfaches Tachymetersystem mit Temperatursensoren)

einem vordefinierten Zeitabschnitt auf einen Datenserver (kommerzieller Server oder ein Raspberry Pi Server, z.B. durch die Komponente RaMonA-Fileserver). Zusätzlich sind in diesem Beispiel BUS-fähige Temperatursensoren aufgeführt. Diese können z.B. einfach ins Mauerwerk des Messobjektes eingeführt und via der GPIO-Schnittstelle (vgl. Kap. 3) angesprochen werden. Bei der Verwendung gebrauchter Tachymeter wäre die Hardwareanschaffung für dieses 24/7-System unter 10.000 Euro exklusiv des RaMonA-TS++ Moduls und der Datenverarbeitung.

RaMonA-GNSS++

Ein ähnliches System mit L1 Low-cost GNSS Empfängern und einer gesamten Wetterstation wird durch folgenden Ansatz für automatisierte Überwachungsmessungen eingesetzt. Als GNSS-Board sind vor allem die beiden Hersteller uBlox (www.u-blox.com) mit den Empfängerboard Lea M8T und Lea M8P und die Empfängerboards des Herstellers NVS www.nvs-gnss.com) des Typs NV08C geeignet. Beide besitzen neben der Rohdatenschnittstelle der Code und Trägerphase im firmenspezifischen Rohdatenformat auch einen implementierten bzw. integrierten RTK-Algorithmus. Die GNSS-Messdaten werden zur Auswertung automatisiert in das standardisierte RINEX-Format konvertiert. Als gute und effiziente Low-cost-Wetterstation bietet sich für den Micro-Controller Arduino Uno das Aufsatzshield EXP-R05-1020 der Fa. Sparkfun an (Beispiel einer Anwendung, cactus.io). Hierbei werden durch

einen entwickelten Arduino-Sketch die Sensorwerte *Temperatur, Luftfeuchtigkeit, Druck, Niederschlag, Tageslicht, Windstärke und Windrichtung* im Sekundentakt digital über die USB-Schnittstelle an den Raspberry Pi gesendet und in die Gesamtauswertung integriert. Das GNSS-System kann mit Low-cost Funkmodulen (z.B. XBee) auch als RTK-System konfiguriert und eingesetzt werden. Hierbei kann ein beliebiger Datenstring mit der transformierten GNSS-Position und z.B. mit den gesamten Messwerten der Wetterstation mit bis zu 10Hz ausgegeben werden. Eine automatisierte Basislinienberechnung der konvertierten GNSS-L1-Rohdaten erfolgt u.a. vollständig automatisiert durch die OpenSource-Software rtklib ((www.rtklib.com) ebenfalls auf den Raspberry Pi als Serverlösung im Büro. Optional können alle GNSS-Daten bei sehr kurzen Abständen auf einen Raspberry Pi am Messort prozessiert werden. Es können aber auch verteilt die GNSS-Rohdaten gesammelt und an einem RaMonA-GNSS Berechnungsserver (u.a. Übertragung via SFTP an den Raspberry Pi Datenserver) berechnet werden. Die Konfigurationsdatei gibt hierbei die Länge und Häufigkeit der Rohdatenaufzeichnung und der Prozessierung an.

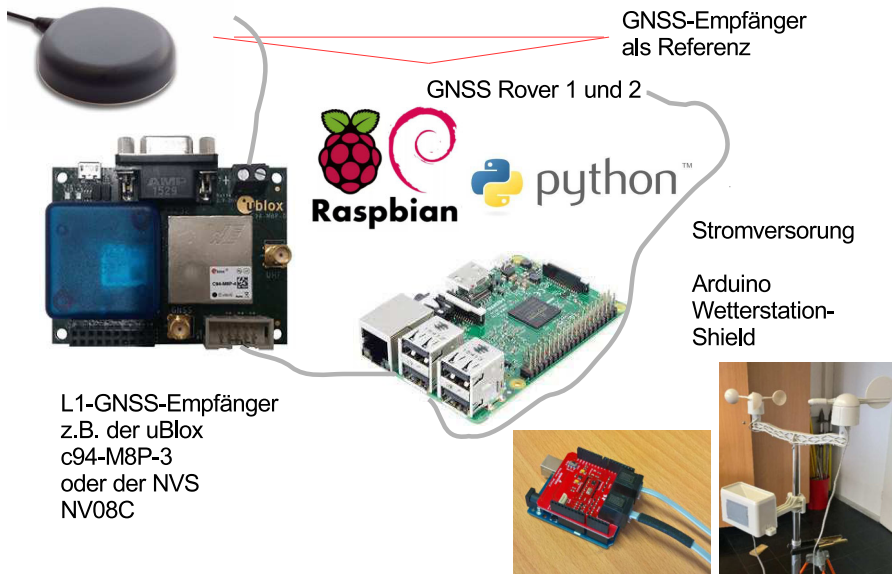


Abbildung 2: RaMonA-GNSS++ (Low-cost L1-GNSS-System mit Arduino Wetterstation)

Ein solches System mit vier Einheiten ist u.a. bei einem Forschungsprojekt an der Verklärungskirche in Kishi (Koordinaten: 62° 4' 4.0" N, 35° 13' 24.6" E) installiert. Hierbei werden gegenwärtig die Einflüsse der Holzfeuchtigkeit über den Jahresverlauf bei der ca. 37m hohen Hauptkirche analysiert. Das Projekt startet im Frühjahr 2019 (KICKLER 2017). Die Auswertung und Messdauer dieses modularen Systems kann individuell angepasst werden.

Die optionalen Bestandteile sind:

- Beobachtungsdauer
- Auswertalgorithmus
- GNSS-Antenne und
- Empfängertyp.

Dieser Setup des GNSS-Systems in Kishi liegt mit drei Rover und einer Basisstation bei etwa 750,- Euro Hardwarekosten exklusiv RaMonA-GNSS-++. Die Ausgleichung der Basislinien kann ebenfalls angewendet werden.

Für weitere geodätische Anwendungen können die Module RaMonA-Niv mit Zweiachsneigungssensoren unterschiedlicher Hersteller (RESNIK 2016), das RaMonA-Level mit z.B. dem Leica DNA03 oder den Geräte TS15 verwendet werden. Alle geotechnischen Sensoren sind ebenfalls als Sensorknoten in entsprechender Konfiguration und einsetzbar.

3. Raspberry Pi als zentraler Monitoringbaustein

Der Raspberry Pi (www.raspberrypi.org/) ist ein Einplatinencomputer der britischen **Raspberry Pi Foundation**. Dieses Projekt wurde an der Universität Cambridge gestartet und hatte das Ziel, mehr Studierende für das Informatikstudium zu gewinnen. Seit 2012 sind solche Ein-Chip-Systeme mit dem ARM-Prozessor erhältlich. Gegenwärtig ist das Modell 3 Modell B+ erhältlich (Frühjahr 2019). Mittlerweile werden jährlich mehrere Millionen Raspberry Pis weltweit verkauft. Der Rechner hat alle Schnittstellen, Netzwerkkomponenten und Graphikkartenoptionen (auch BT und CSI) eines herkömmlichen PC und liegt bei ca. 32 Euro Anschaffungspreis inklusive der microSD Karte. Zusätzlich sind am Raspberry Pi 40 GPIO-Pins (General Purpose Input and Output-Pins, PWM, SPI, ICS, Serial, (www.raspberrypi.org/documentation/usage/gpio/) für die Verwendung von Elektronikkomponenten implementiert. Als Betriebssystem stehen verschiedene Varianten als OpenSource Systeme zur Verfügung. Microsoft hat hierzu ein kostenloses Windowssystem IOT (Internet of Things) entwickelt. Für RaMonA wird i.d.R. das Linuxsystem Raspian verwendet (www.raspberrypi.org/downloads/). Hierzu wird die microSD-Karte mit dem System geflasht. Nach dem Einrichten aller Hard- und Softwarekomponenten kann dieses System ganz einfach geklont werden. Somit ist eine günstige und effiziente Backup-Version bei evtl. Hard- oder Softwareproblemen vorhanden. Das System kann auch mit dem Raspberry Zero (ca. 15 Euro, aber ohne GPIO, USB und HDMI-Anschluss) aufgebaut werden. Die kleine Dimension und der geringe Stromverbrauch (Betriebsspannung 2,3-5V) sind weitere Stärken des Raspberry Pi's. Das Linuxsystem läuft enorm robust und ist für den 24/7-Einsatz konzipiert. Das gesamte Betriebssystem läuft auf der microSD-Karte und kann einfach zwischen den Rechnern getauscht werden. Hierzu muss lediglich die microSD Karte geflasht und ausgetauscht werden. Alle Anwendungen werden durch Python 2- und/oder Python 3-Skripts realisiert. Im folgenden Abschnitt werden wesentliche Grundkomponenten des Linuxsystems kurz aufgeführt und erläutert.

RPI Crontab

Anforderungen hinsichtlich des Dauerbetriebs können in unterschiedlicher Weise umgesetzt werden. Zum einen können z.B. durch einen definierten Reboot mittels eines Crontabs (System zum automatischen Einrichten von Programm- oder Systemaufgaben auf dem Raspberry Pi) betriebssystembedingte Speicherüberläufe abgefangen werden. Eine effizientere Vorgehensweise ist die Überprüfung des RaMonA-Pythonskript, z.B. durch den Eintrag

Listing .1: Raspberry Pi Crontab-Skriptüberwachung

```
sudo pgrep -f Testprogramm.py ||
python3 /home/pi/Ueberwachung/Testprogramm.py
```

Idealerweise kann dies als Linux-Shell (z.B. *bash sudo Shellname.sh*) in einem definierten Zeitablauf (wieder als Crontab) erfolgen. Diese Routine überprüft die fehlerfreie Anwendung des Skriptprogramms oder verschiedener Skriptprogramme (im Beispiel RaMonA-TPS++ oder -GNSS++ inkl. der Wetterdatenerfassung). Grundsätzlich werden diese Programme durch das Linux Terminal gestartet und überwacht. Die sequenzielle Ausführung des Programms erfolgt ebenfalls in einem Crontab-Job.

RPi Watchdog

Eine eingebaute Hardware im Raspberry Pi (BCM2708 Broadcom Prozessor) kann als Watchdog-Funktion das Betriebssystem überwachen (z.B. via heartbeat = 15 sec). Bei Störungen oder einem kompletten Systemabsturz kann dieser Dienst den Problemfall selbstständig erkennen und einen vollständigen Reboot ausführen. Hierzu wird ein Kernelmodul und ein Daemon benötigt. Dieser Hardware-Watchdog ist als eigenständiges Device implementiert (Path: */dev/watchdog*).

RPi globale Fernwartung

Die Fernwartung kann über verschiedene Varianten oder Dienste erfolgen. Grundsätzlich steht immer ein für kommerzielle Anwendungen ausgelegter Teamviewerdienst oder VNC (Virtual Network Computing) zur Verfügung (kostenpflichtige Dienste bei kommerzieller Nutzung). Zusätzlich kann über den Windowsdienst Remote-Desktop auf den Raspberry Pi zugegriffen werden. Diese Dienste müssen auf dem System aktiviert und eingerichtet werden. Hierzu muss ein DNS zur Verfügung stehen und die Portweiterleitung aktiviert werden.

Eine elegante Lösung ist der kostenfreien Dienst Dataplicity (www.dataplicity.com). Hiermit kann der Raspberry Pi über das Webportal durch das Terminal ferngewartet werden. Bei der Verwendung dieser Webdienste muss natürlich immer auch die Datensicherheit beachtet werden. Eine weitere Kommunikationsvariante (lokales oder globales Netzwerk) ist durch das Netzwerkprotokoll Secure Shell (SSH) möglich. Diese verschlüsselte Netzwerkverbindung stellt die Kommunikation zwischen den einzelnen Raspberry Pi und ggf. einem Windowsrechner her. Das Terminalprogramm Putty bietet alle erforderlichen Funktionen zur Anwendung. So können sehr geschickt Daten unterschiedlicher Filesysteme im Netzwerk ausgetauscht werden.

RPi FTPS und SFTP

Grundsätzlich können die Messdaten via FTP oder via der Secure Version SFTP auf einen Datenserver geschrieben werden. Einen solchen Fileserver kann auch ein eigener (S)FTP-Raspberry Pi Server sein (z.B. ProFTPD Paket).

Listing .2: Raspberry globaler SSH Zugriff

```
from ftplib import FTP, TLS
#Verbindung mit dem TLS FTP-Server, Beispiel
ftpserver = ftplib.FTP_TLS('RaMonA_Datacloud.dnshome.com')
ftpserver.ssl_version = ssl.PROTOCOL_TLSv1_2
#ftpserver.set_debuglevel(1)
ftpserver.set_pasv(True)
ftpserver.connect(port=21, timeout=80)
ftpserver.login('client4ftp', 'myPassword')
ftpserver.prot_p()
```

Vor allem der SFTP-Datenaustausch läuft bei verschiedenen Installationen absolut zuverlässig. FTP-Unterbrechungen haben meist mit Einschränkungen des Providers zu tun. Dieses Verfahren kann auch zum Austausch von Messdaten im lokalen Netzwerk (z.B. zu einem Windowsrechner) verwendet werden.

RPi MQTT

Die Kommunikation zwischen einem Raspberry Client und -Server (Broker) kann gut durch das MQTT-Protokoll (Message Queuing Telemetry Transport) erfolgen. Dieses verschlüsselte M2M Nachrichtenprotokoll ermöglicht die effiziente Übertragung von Sensordaten (Maschine2Maschine). Eine Basisrealisierung des MQTT-Client sieht etwa im lokalen Netz wie folgt aus:

Listing .3: Raspberry MQTT Client

```
import paho.mqtt.client as mqtt
def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("CoreElectronics/test")
    client.subscribe("CoreElectronics/topic")

def on_message(client, userdata, msg):
    print(msg.topic+" "+str(msg.payload))
    if msg.payload == "Hello":
        print("Received message #1, do something")
        # Do something
    if msg.payload == "World!":
        print("Received message #2, do something else")
        # Do something else
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect("test.mosquitto.org", 1883, 60)
client.loop_forever()
```

Somit können Messdaten, Pythonprogrammabfragen, Funktionsaufrufe der oder allgemeine Befehle via MQTT im Geonetzwerk ausgetauscht werden. Abbildung 3 zeigt dieses Konzept in einer Anwendung. Die Technologie kann auch über verschiedene Kommunikationsmedien realisiert werden. So sind auch Funklösungen (wie z.B. LoRaWAN (Long Range Wide Area Network) mit dem Frequenzbereich zwischen 433 MHz und 870 MHz, bei einer Datenrate von beträgt 0,3 - 50 kbit/s mit maximal 25 mW Sendeleistung) einfach realisierbar.

Ein solches Kommunikationskonzept ist in RaMonA implementiert und dient zum flexiblem Datenaustausch und zur internen Kommunikation.

RPi Windowsdatenaustausch

Durch die Verwendung des Samba-Servers auf dem Raspberry Pi kann mittels eines konfigurierbaren Zugangs der Datenaustausch mit verschiedenen Filesystem effizient und stabil umgesetzt werden. So ist u.a. eine Kommunikation oder ein Datenaustausch mit einem Hersteller eines bestehenden Monitoringsystems möglich. Auch geschlossene Netzwerke sind über einen Raspberry LAN Adapter

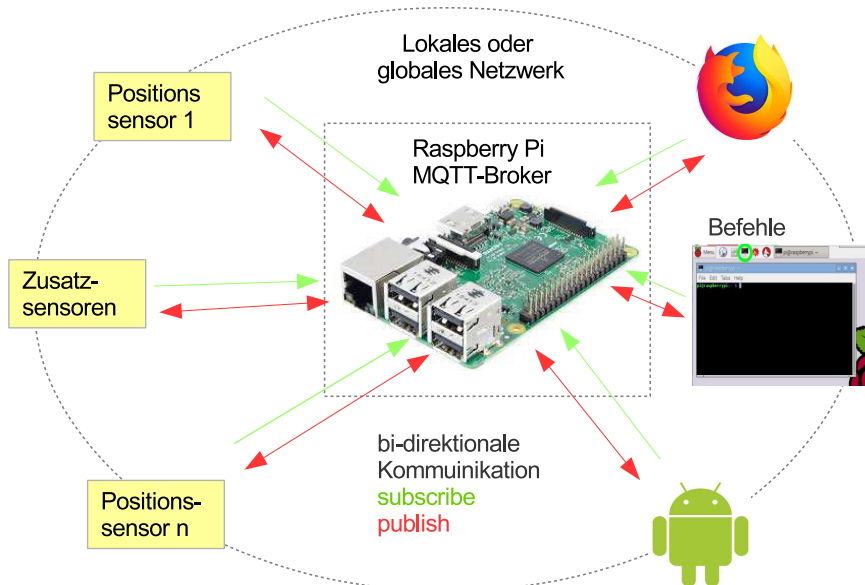


Abbildung 3: Client-Server Kommunikation via MQTT

einfach zu verknüpfen. Hierzu kann RaMonA mit bestehenden Monitoringlösungen ideal kommunizieren (gegenwärtig mit dem gim System). Alle IIF werden als Sensor oder Auswerteknoten betrachtet.

RPi Datenformate

Die Anpassung der Datenformate ist für die Auftraggeber ein sehr wichtiges Kriterium. Die einfachste Möglichkeit ist die Verwendung von Textfiles. Hier ist u.a. die Auswertung mit Python sehr effizient realisierbar. Eine angepasste Ausgabe im CSV-Format mit Kommatrennung findet häufig Anwendung. Das kompakte Java Script Object Notation (kurz JSON) ist ein oft verwendetes ASCII-Format. Auch XML, ShapeFile oder DXF-File sind einfach auszugeben. Entsprechende Parser (Aufteil- oder Analysefunktionen) können diese Files lesen und weiterverarbeiten. Auch die Konfiguration des Systems ist in JSON-Format realisiert.

Listing .4: RaMonA Konfigurationsbeispiel im JSON-Format

```
{
  "Projekt": {
    "TPSNr": "01",
    "FlagLoop": true,
    "HomePath_Programme": "/home/pi/share/Programme",
  }
}
```



```

    "Comment": "Projekt Kishi"
  },
  "ComPort": {
    "Parameter": [
      8,
      "none",
      1,
      1000
    ],
    "Baudrate": 115200,
    "Port": "/dev/ttyUSB0"
  }
}

```

4. RaMonA Monitoringkomponenten

In den beiden vorhergehenden Abschnitten wurden die Basiskomponenten dargestellt. Die Implementierungen setzen viele Details für die entsprechenden Anforderungen um. Sie sind häufig nur Einzelösungen für eine Monitoringaufgabe. Neben den beiden RaMonA Monitoring Basiskomponenten TS und GNSS sind Neigungssensoren und Digitalnivellierverfahren implementiert. Die weiteren RaMonA-Komponenten sind neben den bereits genannten Temperatur- und Wetterstationsensoren folgende Module:

- Reflektorlose Distanzmesser (z.B. Leica Disto)
- Ultraschall Distanzmesser (Sharp GP2Y0A710K0F mit variablen Reichweiten)
- Hochgenaue Distanzmessung im 1/10 mm Messbereich
- Bewegungsmelder
- IMU (z.B. der 10Dof Sensor www.adafruit.com/product/1604)
- unterschiedliche Kamerasysteme
- Dual LAN Adapter für verschiedene Netzwerkverbindungen
- große Auswahl an Umweltsensoren
- große Auswahl von Warnsignalausgaben
- etc.

Die Monitoring-Auswertungen sind oftmals für die einzelnen Fragestellungen anzupassen. Dabei sind automatisierte Berichte mit den Monitoringparametern und den Ausgleichsergebnissen auszugeben. Warnmeldungen können in den klassischen Varianten durch SMS- oder Whatsapp-Nachrichten bzw. via Emails verbreitet werden. Hierfür stehen ebenfalls kostenlose Dienste zur Verfügung. Warnsignale können durch den integrierten Lautsprecheranschluss problemlos ausgegeben werden. Mobile Monitoringanwendungen sind ebenfalls möglich (STEMPFHUBER & KICKLER 2015).

5. Ausblick

Das aufgezeigte Verfahren soll die Möglichkeiten der Anwendung von geodätischen und geotechnischen Monitoringaufgaben mit Low-cost Komponenten sowie effizienten Microcontrollern und Microprozessoren darstellen. Neben den konventionellen Verfahren ist ein solches System sehr effizient auf Anforderungen einzelner Sonderaufgaben anzupassen. Natürlich müssen diese modularen Systeme immer ausreichend in ausführlichen Szenarien getestet werden. Dies ist sicherlich die kritischste und zeitintensivste Komponente des Ansatzes. Ansonsten bietet diese Monitoringvariante eine kostengünstige, schnelle und zuverlässige Methode zur Bearbeitung der Anforderungen hinsichtlich der Messkomponenten, der Messfrequenz, der Auswertung und der Redundanz. Geplante Funktionalitäten sind die Integration der Scanfunktionalität und die Implementierung von einzelnen SDK verschiedener Gerätehersteller.

Literatur

- ALBERDING, J. & STEMPFHUBER, W., (2012): Geodätische Monitoringsysteme mit RTK Low-Cost-GNSS, AVN 4/2012, VDE Verlag GmbH.
- ALBERDING, J. & STEMPFHUBER, W., (2017): Leistungsfähigkeit von Einfrequenz-GNSS-Empfänger, AVN 4/2017, VDE Verlag GmbH.
- ENGEL, P., ET AL. (2015): DABAMOS - freie Software für das automatische Deformationsmonitoring, Beitrag zum Tagungsband Geomonitoring 2015, Clausthal-Zellerfeld, 5.-6. März 2015.
- HEUNECKE, O., ET AL. (2013): Handbuch Ingenieurgeodäsie, Auswertung geodätischer Überwachungsmessungen, 2. neu bearbeitete und erweiterte Auflage. Wichmann-Verlag Berlin.
- RESNIK, B. (2016): Qualitätssicherung bei der Anwendung von Neigungssensoren für geodätisches Monitoring – Ansätze und Erfahrungen. DVW-Seminar Qualitätssicherung geodätischer Mess- und Auswerteverfahren. Zeit: 23. und 24. Juni 2016, Hannover.
- KICKLER, J. (2017): Anwendungsmöglichkeiten von 3dreidimensionaler Vermessungstechniken im Holzbau Möglichkeiten und Grenzen, 19. Internationale Geodätische Woche Obergurgl 2017, Wichmann Verlag Berlin.
- STEMPFHUBER, W. (2010): Grundlagen zur Entwicklung von Monitoringsystemen, In: Weisbrich S. und Kaden, R. [Hrsg.], Entwicklerforum Geodäsie und Geoinformationstechnik 2010, Shaker Verlag, Aachen.
- STEMPFHUBER, W. (2009): *Geodätische Monitoringsysteme – Stand der Technik und Abgrenzung der gegenwärtigen Systeme*, DVW Schriftenreihe 59/2009, Zeitabhängige Messgrößen - Verborgene Schätze in unseren Daten, Beiträge zum 85. DVW-Seminar am 7. und 8. September 2009.
- STEMPFHUBER, W., & KICKLER, J. (2015): Automatisierte Netzmessungen an der Verklärungskirche in Kizhi (Russland), 18. Internationale Geodätische Woche Obergurgl 2015, Wichmann Verlag Berlin, ISBN 978-3-87907-554-6.

Webzugriff der eingefügten Links vom Jan. 2019