Geodätische Automatisierungsverfahren mit Low-Cost-Komponenten und Minicomputern

Geodetic Automation Concepts with Low-Cost Components and Microcomputers

Werner Stempfhuber

In den letzten Jahren haben sich kostengünstige Mikroprozessoren in vielen Natur- und Ingenieurwissenschaften verbreitet. Sie werden in vielfältigen Anwendungsbereichen in unterschiedlichen Konfigurationen eingesetzt. In Kombination mit entsprechenden Programmier- bzw. Skriptingsprachen können in der geodätischen Messtechnik und in der praktischen Geoinformatik viele Aufgaben automatisiert und optimiert werden. Dieser Beitrag soll die Thematik in dem gesamten Anwendungsspektrum darstellen und Anregungen für den praktischen Einsatz aufzeigen. Effiziente Messsysteme mit zielgenauen Auswerteprozessen inkl. einer intelligenten Datenhaltung und -visualisierung verbinden diese beiden geodätischen Anwendungsbereiche ideal.

Schlüsselwörter: Low-Cost, Mikrocontroller, Mikroprozessor, Geosensorik, Raspberry Pi, Automatisierung, Skriptingsprachen

In the last few years, low-cost microprocessors have become accepted in applied sciences and engineering applications. These components are spread in many different applications with independent configurations. In combination with programming languages, many geodetic metrology and practical geoinformatics applications can be automated and optimized. This article illustrates several topics in various geodetic sub-areas. Efficient measuring systems with targeted data analyzing functions, including intelligent data storage concepts, ideally combine these two geodetic subareas.

Keywords: Low-cost, microcontroller, microprocessors, geosensors, Raspberry Pi, automation, scripting language

1 EINLEITUNG

Gegenwärtig werden in vielen Bereichen der Ingenieurwissenschaften Digitalisierungs- und Automatisierungsprozesse umgesetzt. Auch das Vermessungs- und Geoinformationswesen eignet sich hierfür sehr gut, sodass alle Teilbereiche diesem globalen Trend mit unterschiedlichen Zielvorgaben und Lösungsansätzen folgen. In der geodätischen Mess- und Auswertetechnologie bestehen zahlreiche Möglichkeiten der Umsetzung zur Steigerung der Leistungsfähigkeit, zur Gewährleistung der geforderten Messgenauigkeit und Zuverlässigkeit sowie der Optimierung der Gesamtprozesse. Die rasanten Entwicklungen der Vermessungsgeräte in Kombination mit einer Vielzahl von neuartigen Messsensoren beschleunigen diese Entwicklung zusätzlich. Hinzu kommt eine große Anzahl von MEMS-Sensoren (Micro-Electro-Mechanical Systems) zur Bestimmung von Geometrie- und Bewegungsgrößen. Außerdem existieren unzählige Open-Source-Entwicklungen und es sind kostengünstige und leistungsstarke Mikrocontroller bzw. Mikroprozessoren verfügbar. Diese Minicomputer, als Gegenstück von PC-Lösungen, können in vielen geodätischen und geoinformationstechnischen Aufgaben zielgenau eingesetzt werden. Die Kombination der Auswertung, der Datenhaltung und der Datenanalyse mit entsprechender Datenvisualisierung leistungsstarker Serverplattformen bilden eine ideale Ergänzung. Die Entwicklungen von Container helfen zur Integration

der Technologie auf unterschiedlichen Plattformen. Eine weitverbreitete Software stellt der Docker-Container dar. So können mit diesem Fundament betriebssystemunabhängige Systeme zielgenau definiert, spezifiziert und implementiert werden.

Dieser Fachbeitrag soll die Grundlagen zusammen mit konkreten Anwendungsbeispielen als Überblick erörtern und die technischen Möglichkeiten für das Vermessungs- und Geoinformationswesen mit den entsprechenden Schnittstellen zum Bauwesen beschreiben. Der Schwerpunkt liegt nicht auf der Einbindung von kommerziellen Lösungen bzw. Diensten, sondern auf der Umsetzung von zielgenauen Individuallösungen. Die große Flexibilität ist dadurch ein wesentliches Merkmal dieses Ansatzes. Hierzu sind verschiedene Kompetenzen aus den Teilbereichen

- Grundlagen der Anwendung von Mikrocontrollern und Mikroprozessoren,
- Grundlagen der eingesetzten Betriebssysteme,
- Programmierung in Skriptingsprachen,
- Grundverständnis von Bus-Systemen zur Steuerung der Prozesse und Sensoren,
- Datenbankgrundlagen,
- Client-Server-Anwendungen,
- Kommunikationsvarianten,
- Definition der Software-Architektur und
- Abstimmung der einzelnen Basiskomponenten

erforderlich. Alle Teilbereiche können modular aufgebaut und dadurch in unterschiedlicher Systemarchitektur verwendet werden. Somit besteht ein sehr flexibler Ansatz für unterschiedliche Anwendungsszenarien.

2 GRUNDLAGEN DER MIKROCONTROLLER UND MIKROPROZESSOREN

Geodätische Messaufgaben müssen in der Regel neben den technischen Voraussetzungen wirtschaftlich und konkurrenzfähig betrieben werden. Durch den Einsatz unterschiedlicher Hard- und Softwarekomponenten können die Digitalisierungsanforderungen für individuelle Messaufgaben in der Ingenieurgeodäsie und deren Weiterverarbeitung mit der Geoinformationstechnologie realisiert werden. Das geodätische Monitoring zur permanenten Überwachung von technischen Objekten und Bauwerken ist hierfür ein ideales Anwendungsbeispiel /Alberding 2012/, /Heunecke 2013/. Dieser Ansatz lässt sich aber auch für viele Bereiche der geodätischen Messtechnik (vgl. u. a. /Keller et al. 2015/), der allgemeinen Geoinformation, des autonomen Fahrens oder bei Arbeiten des Precision Farmings übertragen.

Um Digitalisierungs- und Automatisierungsaufgaben im Bereich der Geodäsie und Geoinformation umzusetzen, werden zunächst Recheneinheiten benötigt. Sie bilden zusammen mit dem verwendeten Betriebssystem und der Anwendungssoftware die Basis der Realisierung. Generell werden unterschiedliche Bezeichnungen wie Minicomputer, Mikrocontroller, Mikroprozessoren oder Einplatinenrechner verwendet /Flik 2005/. Neben den Großrechensystemen und den Smartphones/Tablets kann folgende Unterteilung erfolgen:

- Mikrocomputer
  z. B. Personal Computer (PC).
- Mikroprozessor
  Ein-Chip-System (Einplatinenrechner mit integriertem Schaltkreis, Bus-System, Speicher, I/O-Bausteine); eine sehr bekannte Vertretergruppe für diese Kategorie sind die unterschiedlichen Modelle des Raspberry Pi (http://www.raspberrypi.org).
- Mikrocontroller
  Mit Bus-Controller aus Funktionseinheiten für die selbstständige Durchführung von Teilaufgaben, z. B. Arduino (http://www.arduino.cc, https://de.wikipedia.org/wiki/Liste_von_Arduinoboards).
- Embedded Control
  „Eingebettet System“, oft direkt im Messsensor verbautes System.

Dabei erledigt der Mikroprozessor mit einem eigenen Betriebssystem viele individuelle Aufgaben. Diese können beim Raspberry Pi u. a. die Varianten

- Linux-Distributionen Ubuntu oder Debian (z. B. via Raspbian, NOOBS (New Out of the Box Software),
- Windows 10 IoT oder
- Third-party-Betriebssysteme (siehe https://www.raspberrypi.org/downloads)

sein. Diese Betriebssysteme sind alle kostenlos in unterschiedlichen Varianten verfügbar. Das Standard-Betriebssystem für viele Raspberry-Pi-Anwendungen stellt die auf der Linux-Distributionen Debian basierende Variante Raspbian (https://www.raspberrypi.org/downloads/raspbian) dar. Sie kann mit und ohne Anwendungssoftware als Desktoplösung oder als schlanke Lite-Version ohne Desktop installiert und verwendet werden. Das Linux-Betriebssystem bietet viele nützliche Zusatzfunktionen wie den Watchdog, Crontab, Shellskripting sowie die schnelle Installation von unterschiedlichen Serverkomponenten (alternativ die Verwendung des Linux-Ubuntu-Servers als Raspberry-Pi-Lösung). Einige dieser im Betriebssystem integrierten Funktionen helfen enorm bei der Umsetzung der Automatisierung. Für die Schnittstelle der Mess- und Auswertungsaufgaben werden häufig Skriptingsprachen wie Python eingesetzt. Datenbanksysteme auf dem Mikroprozessor verwalten zusätzlich die Geodaten. Die verschiedenen Hardware-Schnittstellen wie USB, BT, HDMI, LAN, WLAN, CSI und DSI ermöglichen die Kommunikation des Mikroprozessors mit den Sensorknoten. Das gesamte Betriebssystem inkl. aller Anwendungen und Konfigurationen befinden sich beim Raspberry Pi sich auf der Micro-SD-Karte. Dies ermöglicht dadurch ein einfaches und schnelles Spiegeln (Klonen) und erzeugt somit ein 1:1-Abbild des Gesamtsystems.

Ein Mikrocontroller hat in der Regel kein Betriebssystem und nur eine reduzierte Schnittstelle. Bei den unterschiedlichen Varianten der Arduino-Boards können Zusatzkomponenten durch Module hinzugefügt werden. Solche Komponenten sind u. a. WLAN/LAN-Adapter, Speicherkartenadapter, GNSS-Module usw. Durch diese Aufsatzshields zusammen mit der Schnittstellensoftware werden Sensorknoten für das Gesamtsystem erstellt (*Abb. 1*). Mikrocontroller arbeiten grundsätzlich nur eindeutig definierte Prozesse als Endlosschleife ohne entsprechendes Betriebssystem ab. So liefert z. B. ein einzelner Temperatursensor, ein Wetterstationsshield oder ein GNSS-Board in Kombination mit dem Mikrocontroller die berechneten Messwerte aus den Rohmesselementen an eine Schnittstelle (standardmäßig die USB-Schnittstelle). Dazu müssen die unterschiedlichen Typen der Mikrocontroller und Mikroprozessoren genau

Kategorie	Mikrocontroller Arduino Uno (R3)	Mikroprozessor Raspberry Pi 4 Modell 4 B
Preis	ca. 30 Euro	ca. 40-60 Euro
Prozessor	ATmega328 8-Bit-Mikrocontroller	Broadcom BCM2711, bis zu 4 GB RAM
Größe	7,6 cm × 6,4 cm	8,5 cm × 5,6 cm
Datenspeicher	nein	Micro-SD-Karte inkl. Betriebssystem
GPIO	14 (analog und digital)	40 (digital)
RAM	32KB (Flash Speicher)	Micro-SD-Karte (beliebige Größe)
Taktfrequenz	16 MHz	1,5 GHz
Multitasking	nein	ja
Flash Memory	32 KByte	Micro-SD-Karte (beliebige Größe)
Internet	ohne Aufsatzshield kein LAN/WLAN	WLAN und LAN (Gigabit)
Bluethooth	ohne Aufsatzshield keine BT	BT 5.0
Integrated Development Environment	Arduino IDE	IDLE, Linux-Funktionen, beliebige Shell- und Skriptingspakete (u.a. Python2/3)
Betriebssysteme	nein	verschiedene Linux-Distributionen, Windows IoT etc.
Videoausgabe	nein	2 x Mikro-HDMI-Ports (4K)

Tab 1 | Vergleich des Raspberry Pi Modells 4 mit dem Arduino Uno R3 (Stand Herbst 2019)

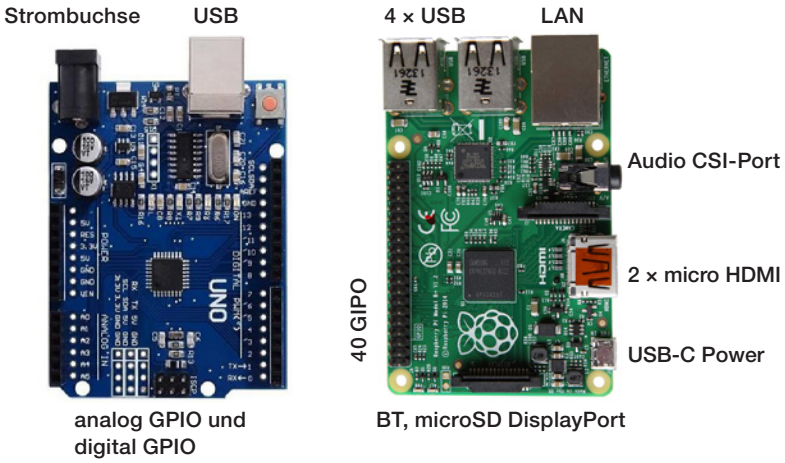


Abb. 1 | Mikrocontroller versus Mikroprozessor (Stand Herbst 2019)

bewertet werden. Eine komplette und aktuelle Auflistung aller Systeme ist nicht möglich. Trotzdem werden in *Tab. 1* die Leistungsunterschiede der zwei wichtigen Vertreter und in *Abb. 1* die wichtigsten Hardwareschnittstellen dieser beiden Vertreter gegenübergestellt.

3 KOMPONENTEN VON SENSORKNOTEN

Beide Ansätze haben für unterschiedliche Anwendungen ihre Berechtigung. Aufsatzplatinen (Aufsatzshields) erweitern den Funktionsumfang. Ideal ist beispielsweise die Erstellung eines eigenständigen Messsensors zur Erfassung von Messdaten als Sensormodul mit bidirektionaler Kommunikation und Konfigurationsparameter (GNSS-Einfrequenz- oder -Mehrfrequenz-Receiver, unterschiedlichen Distanzsensoren, INS-Boards, meteorologischen Messsensoren etc.). Die Möglichkeiten der Verwendung von MEMS-Sensoren für die Bearbeitung von Geodäsie- und Geoinformationsfragestellungen sind nahezu unbegrenzt. Ein wesentlicher Vorteil ist die Verfügbarkeit von analogen Pins beim Arduino-Board. So können u.a. analoge Sensoren (z.B. der Temperatursensor TMP36 oder verschiedene Sensoren zur Bestimmung der Bodenfeuchte) für die Erstellung von Sensorknoten verwendet werden. Beim Mikroprozessor Raspberry Pi ist hierfür eine A/D-Wandlung (z.B. durch den MCP3008 für 8-bit-, 10-bit- oder 12-bit-Chip für die digitale Abtastung) erforderlich. Bei digital arbeitenden Messkomponenten kann der gleiche Ansatz eines Mikrocontrollers mit einem Mikroprozessor umgesetzt werden. Hierfür ist ein einfaches Beispiel der Temperaturerfassung in *Abb. 2* gegenübergestellt. Eine Einbindung des Arduino-basierten Sensors beim Raspberry Pi ist ebenfalls durch die Installation der Arduino-Entwicklungsumgebung inkl. der Treiber realisierbar. Die Sensorprogrammierung kann dabei direkt am Raspberry Pi erfolgen. Durch die Verwendung der seriellen Schnittstelle ist somit der direkte Datenfluss in Echtzeit gegeben. Durch die Bibliothek drawnow kann der Echtzeitplot realisiert werden. Die Installation der Module erfolgt an der Terminalkonsole durch die Befehle

sudo apt-get install drawnow

oder

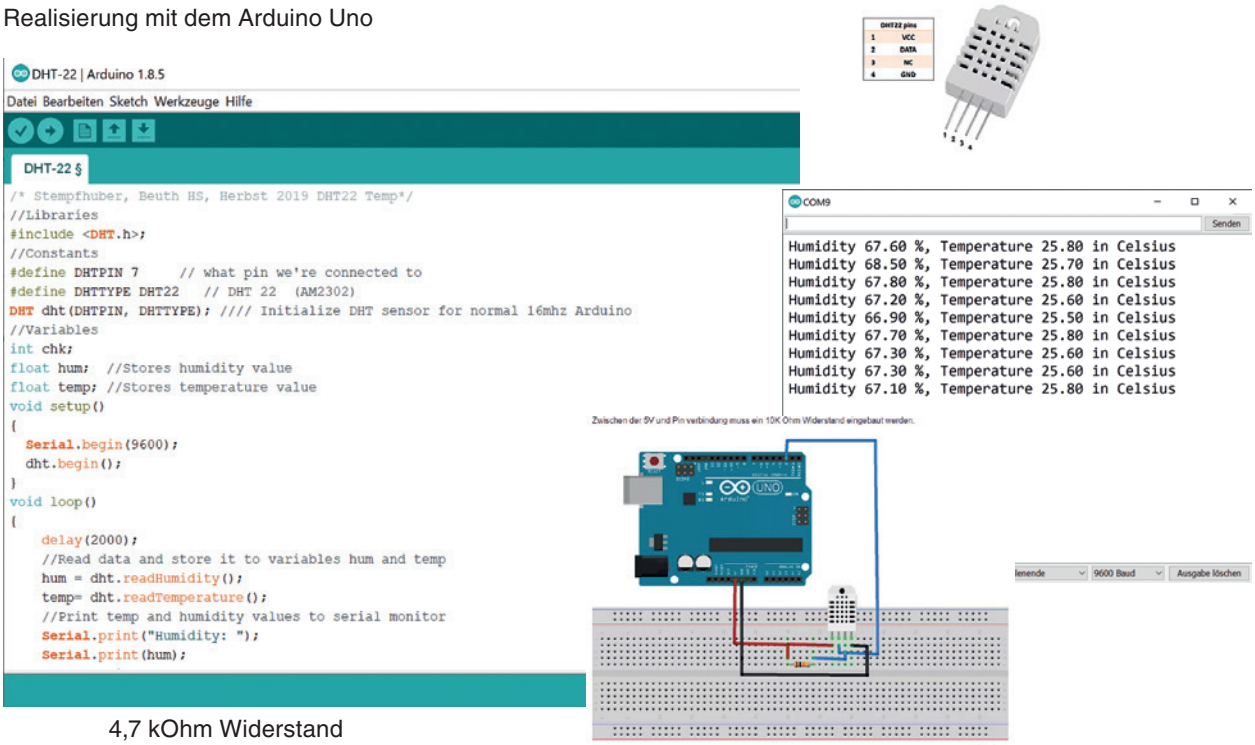
sudo apt-get install serial

(bei der Windowsvariante wäre dies der Befehl

pip3 install

drawnow).

Realisierung mit dem Arduino Uno



4,7 kOhm Widerstand

Realisierung mit dem Rasperry Pi

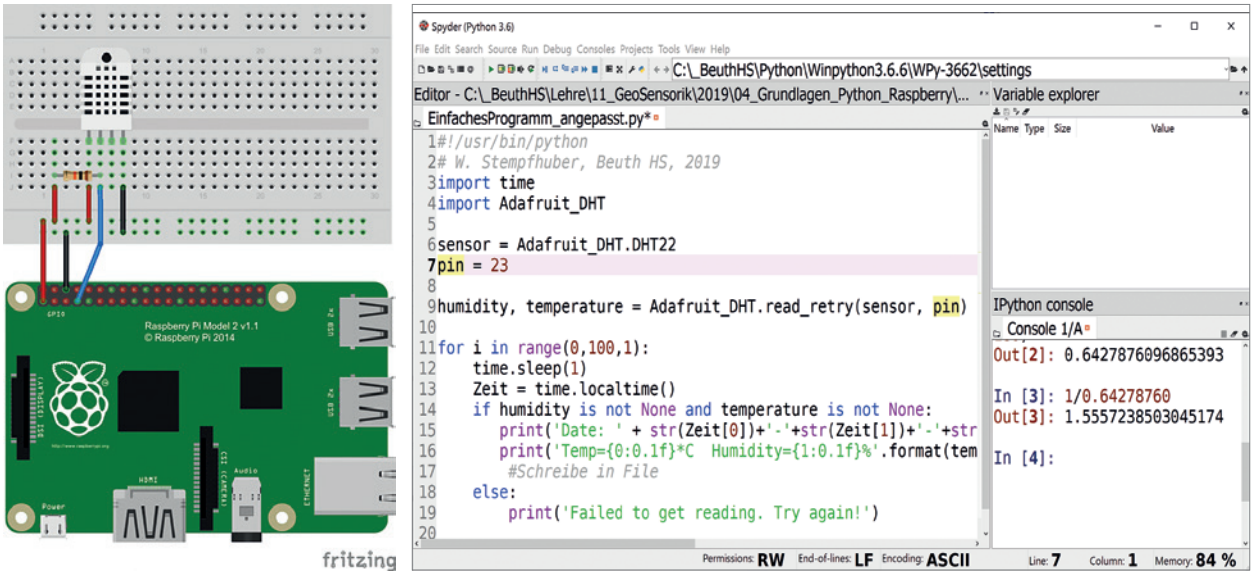


Abb. 2 | Anwendungen von Mikrocontroller und Mikroprozessor zur Erfassung von Temperatur- und Luftfeuchtigkeitswerten (als Beispiel mit dem Sensor DHT22)

Ein praktischer Lösungsansatz ist beispielsweise die Verwendung des Arduino-Weathershields mit den Komponenten Si7021 als Temperatur- und Luftfeuchtigkeitssensor, MPL3115A2 als Luftdrucksensor, ALS-PT19 als Lichtsensor und optional noch Zusatzkomponenten für die Erfassung der Windgeschwindigkeit, Windrichtung und Regenmesser. Diese Messdaten werden z. B. im Sekundentakt durch den entsprechenden Arduino-Sketch an den Bus-Systemen abgegriffen und in einer entsprechenden Baudrate an die USB-Schnittstelle übertragen. So kann dieser Sensorknoten durch das gleichzeitige Lesen der Messwerte an der RS232-Schnittstelle durch das entsprechende Python-Programm bzw. der Funktion in das Geosensornetzwerk eingebunden werden.

Zusätzlich existieren bei den Mikroprozessoren (Einplatinencomputer) unbegrenzte Möglichkeiten der effizienten Datenhaltung, der Datenanalyse, der Datenvisualisierung und Weiterverarbeitung. Die filebasierte Messdatenspeicherung kommt bei Zeitreihen über einen langen Zeitraum schnell an die Grenzen. Eine intelligente Datenhaltung in einer Datenbank kann direkt auf dem Mikroprozessor umgesetzt werden. Ein weiterer Punkt zur Optimierung von automatisierten Systemen ist die Notwenigkeit der Fernwartung bzw. des Fernzugriffs

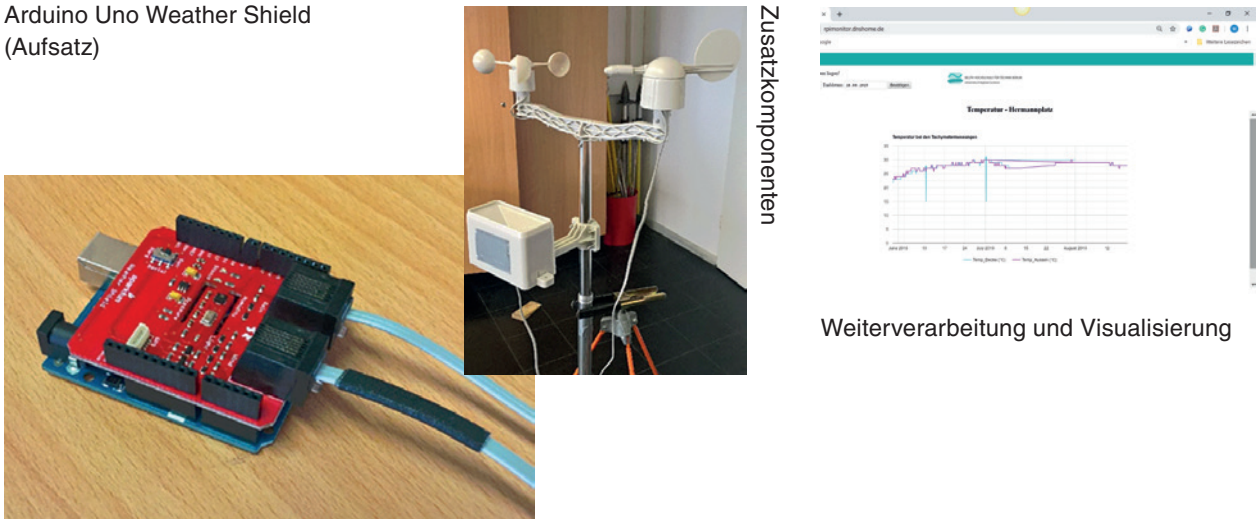


Abb. 3 | Arduino-Weathershield als Sensorknoten im Raspberry-Pi-Geosensornetzwerk

auf einzelne Komponenten der Sensorknoten, der Auswertemodule oder des Steuerprozesses. Bei Raspberry Pi existieren viele Realisierungsoptionen. Neben kommerziellen Diensten wie TeamViewer oder dem VNC-Viewer existieren webbasierte Dienste wie Dataplicity. In einem lokalen Netzwerk ist der Zugriff einzelner Mikroprozessoren untereinander sehr effizient umzusetzen, z. B. als Client-Server-Lösung oder als Socket-Kommunikation /Stempfhuber 2019/. Auch der Zugriff bzw. der Datenaustausch mit anderen Betriebssystemen ist realisierbar. So ermöglich z. B. die Verwendung eines Samba-Servers mit der entsprechenden Konfiguration den Zugriff auf unterschiedliche Filesysteme. Via Remote Desktop oder VNC ist das Einloggen von einem Windows-PC auf den Raspberry Pi umsetzbar. Hierzu muss die lokale oder globale IP-Adresse bekannt sein. Die Verwendung eines Secure-Shell-Client SSH (z. B. via Tool PuTTY) setzt auf der Linux-Terminalkonsole auf und kommuniziert durch das Netzwerkprotokoll Telnet. Dies ermöglicht eine manuelle Steuerung der einzelnen Module von außen sowie die Verwendung der Linux-Befehle auch von anderen Rechnerplattformen aus.

Sensoransteuerung

Die Umsetzung der Rohdatenabtastung inkl. der entsprechenden Weiterverarbeitung kann idealerweise mit der Skriptingsprache Python 2 oder 3 erfolgen. Die Installation des Linux Betriebssystems Raspbian beinhaltet bereits diese beiden Versionen (idealerweise können noch zusätzliche Editortools wie der Spyder installiert werden). Selbstverständlich können aber auch andere Umgebungen gewählt werden. Die Architektur der Rohmessdatenerfassung am Messsensor kann direkt durch das Pythonskript implementiert werden. Dazu müssen sowohl bei der Entwicklungsumgebung am Mikrocontroller oder am Mikroprozessor Module importiert und verwendet werden. Dies setzt zusätzlich die Aktivierung der Schnittstellen SSH, Kamera, VNC, I^2C, seriell, Eindraht-Bus und GPIO am Betriebssystem voraus. Die Steuerung des Messzeitpunkts kann durch einen externen Impuls, vordefinierte Messzyklen im Pythonskript oder die Definition eines Crontabs (Verwendung von Crontab-Daemon) implementiert werden. Ebenfalls lassen sich ganze Pythonskripte oder Linuxskripte via Crontab-Dienst zielgenau steuern. Durch die Verwendung eines Shebang-Eintrags im Skriptheader (bei Bash = #!/bin/bash, bei Python = #!/bin/python) erspart man sich die Angabe des Skriptinterpreters beim Aufruf im Terminal oder in Crontab. Gegebenenfalls müssen noch die entsprechenden Schreib- und Leserechte gesetzt werden. Dieser wichtige Schritt kann aber auch im Skript erfolgen.

Einsatz als Dauerbetrieb

Permanentanwendungen werden häufig als 24/7-Anwendungen mit 100 % Zuverlässigkeit bezeichnet. Diese Systeme müssen Systemausfälle durch intelligente Zusatzfunktionen eliminieren. Durch einen täglichen Autostart inkl. dem Start alle Geosensorknoten können verschiedene betriebssystemabhängige Störungen weitgehend abgefangen werden. Der im Raspberry Pi eingebaute Hardware-Watchdog überwacht das Betriebssystem und die Anwendungen und stellt somit den 24/7-Betrieb sicher. Dieses eigenständige Device kontrolliert alle definierten Prozesse des Raspberry-Pi-Minirechners. Zusätzlich kann eine intelligente Verwendung von Relais auf dem Raspberry Pi noch zusätzlich als Prozessüberwachungsmodul angewendet werden. Verschiedene Anwendungen im Monitoringbereich /Stempfhuber 2019/ bestätigen diesen Ansatz über lange Anwendungsbereiche.

Datenhaltung

Die Messdatenübertragung vom Messsensor zum Speichermodul kann in unterschiedlicher Art und Weise umgesetzt werden. Bei einer GNSS-Anwendung für Monitoringaufgaben werden z. B. im Stundentakt die Einfrequenz- oder Mehrfrequenz-Rohdaten am Mikrocontroller oder Mikroprozessor aufgezeichnet. Nach der Anwendung der herstellerspezifischen Binär-zu-ASCII-Konvertierung und Ausgabe in das RINEX-Format kann eine GNSS-Auswertesoftware (z. B. RTKLib) für gängige GNSS-Boards wie u-blox, NVS, Trimble RT17, Hemisphere, NovAtel, JAVAD /Takasu 2009/ auf dem

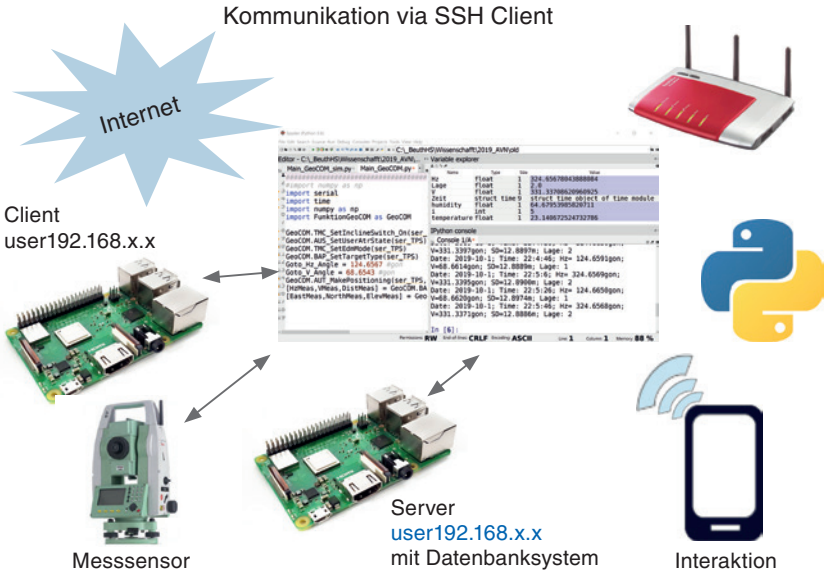


Abb. 4 | Messung via Client-Server (Beispiel Tachymetermessung)

Raspberry Pi die Basislinienberechnung übernehmen und die Ergebnisfiles inkl. des Berechnungsprotokolls mit den entsprechenden statistischen Auswerteparametern auf einen Datenserver speichern. Dabei wird häufig das File-Transfer Protocol (FTP) oder die sichere Variante Secure File-Transfer Protocol (SFTP) verwendet. Dabei können die Datenfiles auch als Messpakete versendet werden.

Client-Server-Kommunikation

Eine geeignete Systemarchitektur bei entsprechender Netzverfügbarkeit ermöglicht einen leistungsstarken Client-Server-Ansatz. Dabei werden auf dem Raspberry-Pi-Client (dies kann auch ein Zero-Modul sein) oft nur wenige Codezeilen benötigt. Der Raspberry-Pi-Server sendet eine solche Anfrage (z. B. Durchführung einer Zweilagenmessung inkl. Datenspeicherung).

Ein praxistaugliches Monitoringsystem erfordert natürlich einen großen Funktionalitätsumfang. Hierzu wurden u. a. in der Raspberry-Pi-basierten Monitoringplattform RaMonA /Stempfhuber 2019/ mit einer umfassenden Bibliothek für tachymetrische und GNSS-basierte Überwachungsmessungen bereitgestellt. Das Grundprinzip lässt sich effizient erweitern. Neben den Kommandobefehlen des SecureShell-Tunnels via Terminal oder einer TCP/IP-Kommunikation kann auch mittels Message Queuing Telemetry Transport Protocol (MQTT) die Nachrichtenübertragung bidirektional erfolgen /Engel 2015/. Diese Machine-to-Maschine-(M2M)-Kommunikation ist für die Client-Server-Architektur bei Geosensornetzwerken ideal einsetzbar. Die vom Broker (Server) abgesendeten Nachrichten werden vom Client ausgeführt und zurückgesendet. Dies ist auch über Funknetze (z. B. LoRaWAN) realisierbar.

Datenbankmanagementsystem

Bei großen Datenmengen (z. B. Wetterdaten in Form von Werten für Temperatur, Luftfeuchte, Luftdruck, Helligkeit, Niederschlagsmenge, Windgeschwindigkeit und Windrichtung) mit hochauflösenden Datenraten (z. B. im 10-Minuten-Takt über viele Monate Aufzeichnungszeitraum) versagen einfache Datenhaltungsansätze. Die gesamten Messreihen des jeweiligen Sensorknotens müssen auf einer einheitlichen Zeitbasis in den relationalen Datenbanksystemen verwaltet werden. Hierfür eignet sich pro Sensorknoten eine eigene Tabelle. Die Verwendung von Unix-Zeitstempeln ermöglicht die Darstellung der Messdaten in einer Zeitreihe mit Datum und Uhrzeit. Die Vorteile der Verwendung von Datenbanksystemen bietet sich entsprechend an. Auf dem Raspberry Pi können direkt Datenbanksysteme installiert und verwendet werden. Das relationale Datenbankmanagementsystem MariaDB ist eine Abspaltung aus MySQL als Open-Source-System und kann auf den Plattformen Linux (optional auch auf Microsoft, OS und Solaris) betrieben werden. Hiermit können solche Projekte ideal auf dem Raspberry Pi ohne Zusatzkosten umgesetzt werden. Die Datenbank kann entweder direkt auf dem Mikroprozessor zusammen mit dem Messwerterfassungsprogramm betrieben werden. Bei größeren Projekten bietet sich im lokalen, oder aber auch im globalen Geosensornetzwerk die Verwendung eines eigenen Datenbank-Mikroprozessors an. Dies kann auch auf angemietete Serverdienste implementiert werden. Das Anlegen und Vorführen der zeitbasierten Messdaten in die Datenbank wird z. B. ebenfalls durch ein Pythonskript ausgeführt. Diese Aufgabe könnte auch auf einem Webserver optional ein PHP-Skript übernehmen. Neben dem freien Webanwendungstool phpMyAdmin bei MySQL-Datenbankanwendungen existieren verschiedene Freeware-Tools zur Administration dieser Datenverwaltungssysteme. Das Tool Heidi-SQL ermöglich beispielsweise den Zugriff über ein globales Netzwerk via dem SSH-Tunnel von jeder Rechnerplattform aus.

Visualisierung und Analyse der Datenbankinhalte

Die Verwendung eines Webservers auf dem Raspberry Pi ermöglich zusätzlich die Auswertung und Darstellung der Messergebnisse (ggf. auch durch interaktive Zeitraumauswahl via SQL-Abfragen). Hierzu ist die Installation des Webservers Apache2 (beispielsweise mit dem Terminal Befehl sudo apt-get install apache2) erforderlich. Die gesamte Benachrichtigung, das Reporting und die Warndienste können in diesem System durch geeignete SQL-Abfragen angewendet werden. So können automatisierte Nachrichten, E-Mails, Whatsapp-Nachrichten oder Pushdienste implementiert werden. Die Umsetzung von grafischen Auswertungen der Datenbankeinträge aller Messwerte kann u. a. durch den interaktiven Webservice Google Chart erfolgen. Somit können die Datenbankeinträge und Tabellen idealerweise in Line-, Bar-, Pie-, Area-Charts, Tree Maps oder Dashboards visualisiert werden (https://developers.google.com/chart/). Dabei übernimmt ein PHP-Skript die Abfragen der Datenbankeinträge und stellt es im HTML-Teil als interaktive Webseite dar. Ein solcher Daten- und Webserver kann problemlos mit einem handelsüblichen Router betrieben werden. Hierzu müssen lediglich die Standardports (z. B. 21 für FTP, 22 für den SSH-Tunnel, 80 für HTTP, 443 für HTTPS und 3389 für das Remote Desktop) geöffnet werden. Der Betrieb eines Mikroprozessors mit einem

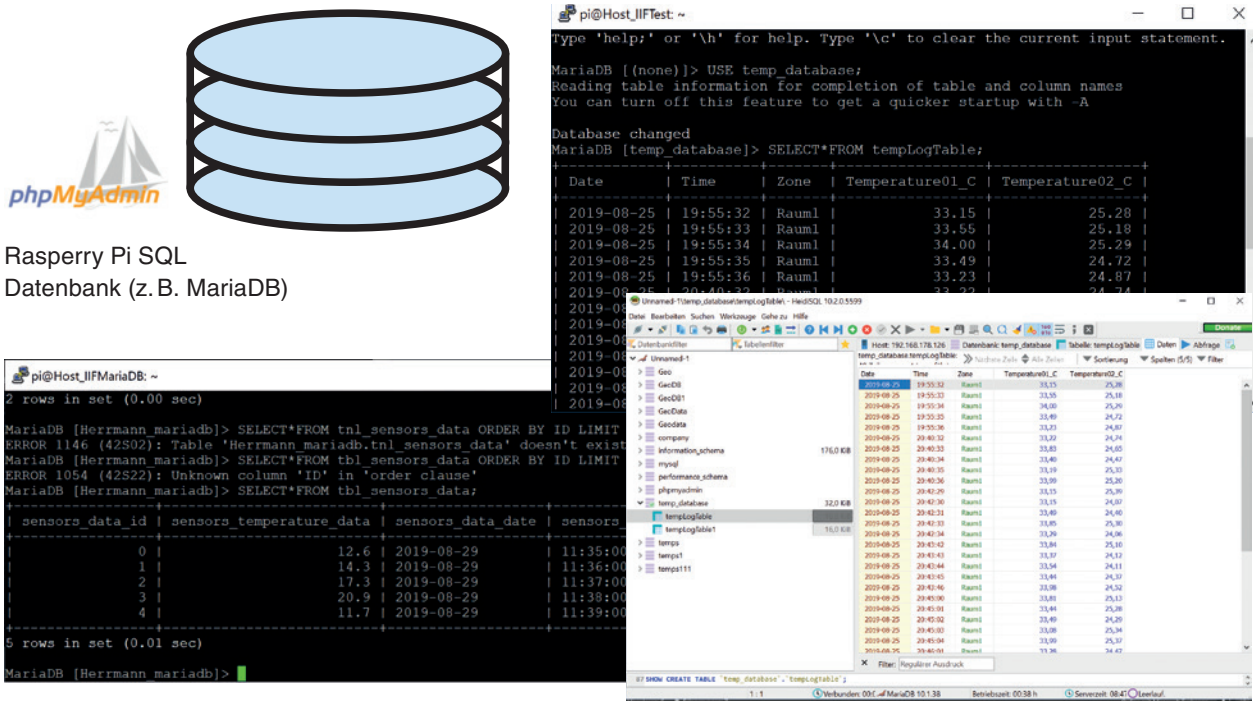


Abb. 5 | Datenbankverwaltung am Raspberry Pi

Router zusammen mit dem entsprechenden Webanbieter kann nur im globalen Netz durch die globale IP-Adresse des Device und der entsprechenden Portöffnung erfolgen. Mit der Webseite http://www.myglobalip.com kann diese globale IP-Zahlenfolge des Sensorknotens abgerufen und verwendet werden. Durch die Verwendung eines DNS-Dienstes (z. B. http://www.dnshome.de) kann eine eindeutige Webadresse auch bei wechselnden IP-Adressen umgesetzt werden. Die Auswerteplattform Grafana (http://www.grafana.com) stellt einen Server zur optimalen Auswertung von Zeitreihen aus Datenbanken zur Verfügung.

4 UMSETZUNG VON GEOSENSORANWENDUNGEN

Als vereinfachtes Beispiel wird abschließend ein mobiles System zur Erfassung von Umweltparametern vorgestellt. Die Sammlung von vielen Umweltinformationen kann die Echtzeiterfassung der stationären Erfassungssysteme, wie das von der Senatsverwaltung für Umwelt, Verkehr und Klimaschutz in Berlin betriebene Messnetz BLUME, ideal ergänzen. In verschiedenen Projekten an der Beuth-Hochschule für Technik Berlin wurden hierzu Entwicklungen umgesetzt /Krebs 2018/. Durch eine Powerbank kann das System über

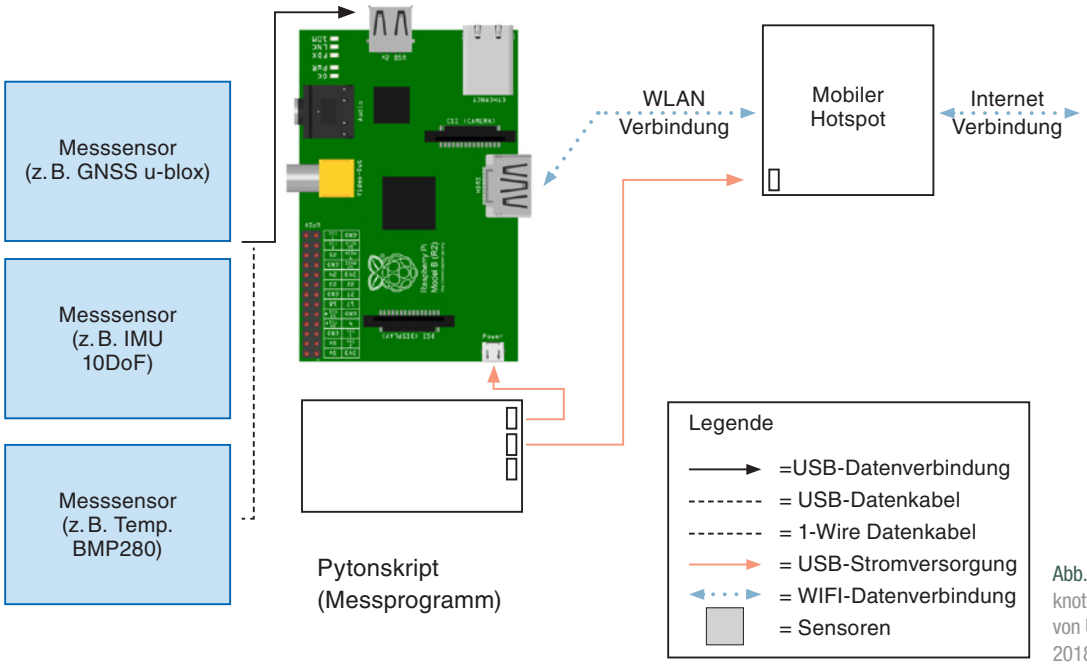


Abb. 6 | Geosensorknoten zur Erfassung von Umweltdaten /Krebs 2018/

viele Stunden mobil betrieben werden. Bei diesem Ansatz ist der Auswertungs- und Datenserver über eine Fritzbox mit dem Internet verbunden. Dies kann natürlich auch ein Firmen- oder Hochschulnetz sein.

Für den Raspberry Pi, welcher für die Erfassung der mobilen Daten als Clientsystem eingesetzt wurde, wird eine Eingangsleistung von mindestens 5 V und 1,8 A angegeben. Durch einen Akkupack (beispielsweiser der Firma ANKER mit ~ 27 Amperestunden (Ah)) konnte dieses System etwa 60 Stunden mit einem Verbrauch von 4,5 Watt aller Komponenten betrieben werden. Die Basis des Erfassungssystems wird durch das Client-Server-System gebildet. Einzelne Komponenten lassen sich beliebig ergänzen. So wird gegenwärtig dieses Verfahren für Anwendungen der Bauwerksüberwachung (u. a. im Forschungsprojekten zur Permanentüberwachung an der Christi-Verklärungskirche in Kizhi/Russland /Stempfhuber & Kickler 2015/) und beim Gletscherforschungsprojekt Juneau Icefield Research Program /Stempfhuber 2015/ eingesetzt und den jeweiligen Anforderungen entsprechend ergänzt. Hierzu müssen zusätzlich die Sensorkomponenten

- GNSS (Basislinienauswertung und RTK und PPP),
- Tachymeter,
- Neigungssensoren,
- Distanzsensoren unterschiedlichster Genauigkeitskategorien,
- Nivellement und/oder Rotationslaser,
- Low-Cost-IMU und
- Scanning-Komponenten

integriert werden. Zusätzlich ergänzen die Low-Cost-Komponenten eines Kameramoduls über die CSI-Schnittstelle, Relais zur Steuerung von Stromschaltkreisen, Feuchtigkeitssensoren, Ozonsensoren, Feinstaubsensoren das in *Abb. 6* aufgezeigte Gesamtsystem.

5 AUSBLICK

Die Motivation des vorliegenden Beitrags ist die Zusammenstellung der individuellen Lösungsmöglichkeiten, basierend auf modernen Mikrocontrollern und Mikroprozessoren, für geodätische und geoinformationsrelevante Fragestellungen. Zusätzlich lassen sich diese Low-Cost-Module durch Low-Cost-Messsensoren und Open-Source-Softwaremodule in ein modulares Gesamtsystem vereinen. Dadurch kann eine große Anzahl von neuen Anwendungen umgesetzt werden. Der Automatisierungsgrad kann zu 100 % realisiert werden. In Zusammenarbeit mit dem Ingenieurbüro gim (https://www.gim-vermessung.com) laufen in Berlin hierzu verschiedene geodätische Monitoringprojekte über einen sehr langen Zeitraum absolut stabil und zuverlässig. Die Kombination mit bereits bestehenden Komponenten (z. B. windowsbasierten Steuerungsalgorithmen) ist durch eine klare Definition von Schnittstellenparametern rechnerübergreifend realisiert. Der Raspberry Pi als Repräsentant der Familie der Mikroprozessoren stellt neben den geodätischen Messelementen das zentrale Herzstück dar. Die rasanten Entwicklungen werden auch in den nächsten Jahren im gleichen Tempo weiterlaufen und bilden so eine innovative Alternative für den Einsatz dieser Technologie.

LITERATUR

Alberding, J.; Stempfhuber, W. (2012): Geodätische Monitoringsysteme mit RTK Low-Cost-GNSS. In: allgemeine vermessungs-nachrichten (avn) 119(2012)5, 132–139.

Engel, P. et al. (2015): DABAMOS – freie Software für das automatische Deformationsmonitoring. Beitrag zum Tagungsband Geomonitoring 2015. Clausthal-Zellerfeld.

Flik, Th.; Liebig, H. (2005): Microprozessortechnik. 7. Auflage. Springer, Heidelberg/Berlin.

Heunecke, O.; Kuhlmann, H.; Welsch, W.; Eichhorn, A.; Neuner, H. (2013): Handbuch Ingenieurgeodäsie – Auswertung geodätischer Überwachungsmessungen. 2. Auflage. Wichmann, Berlin/Offenbach.

Keller, F.; Kagerah, S.; Sternberg, H.: Automatisierung eines Low-Cost-360°-Reflektors für die genaue Positionierung einer 3D-Trajektorie. In: allgemeine vermessungs-nachrichten (avn) 122(2015)8-9, 259–268.

Krebs, F. (2018): Erstellung eines Low-cost-Messsystems zur Erfassung und Verarbeitung von Umweltparametern. Bachelorarbeit, Beuth-Hochschule für Technik Berlin (unveröffentlicht).

Pink, S. (2017): Entwicklung und Erprobung eines multifunktionalen Geo-Sensornetzwerkes für ingenieurgeodätische Überwachungsnetze. Schriftenreihe 83(2008). Universität der Bundeswehr München.

Stempfhuber, W. (2015): Gletschermonitoring am Juneau Icefield. In: Hanke, K.; Weinold, Th. (Hrsg.): 18. Internationale Geodätische Woche Obergurgl 2015. Wichmann, Berlin/Offenbach, 233–238.

Stempfhuber, W.; Kicker, J. (2015): Automatisierte Netzmessungen an der Verklärungskirche in Kizhi (Russland). In: Hanke, K.; Weinold, Th. (Hrsg.): 18. Internationale Geodätische Woche Obergurgl 2015. Wichmann, Berlin/Offenbach, 124–134.

Stempfhuber, W. (2019): Geodätisches und Geotechnisches Monitoring mit Low-cost-Komponenten. Beitrag zum Tagungsband Geomonitoring 2019. Hannover. https://www.repo.uni-hannover.de/handle/123456789/4549 (09/2019).

Takasu, T. (2009): RTKLIB: Open Source Program Package for RTK-GPS. In: FOSS4G 2009, November 2, 2009, Tokyo, Japan.

u-blox White Paper (2016): Achieving Centimeter Level Performance with Low-Cost Antennas, UBX-16010559. www.u-blox.com (09/2019).

Prof. Dr.-Ing. Werner Stempfhuber

BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
FACHGEBIET PRAKTISCHE GEODÄSIE
FACHBEREICH III

Luxemburger Str. 10 | 13353 Berlin
stempfhuber@beuth-hochschule.de