



V27 – Mikrocontroller

Ort: F-Praktikum C60.007

Betreuer: Dr. M. Dehnert

Die automatisierte Erfassung und Verarbeitung von Messdaten sind aus der modernen experimentellen Physik nicht mehr wegzudenken. Nahezu jedes wissenschaftliche Experiment wird heute durch Mikrocontroller oder eingebettete Computersysteme gesteuert, die Sensoren auslesen, Messgeräte ansteuern, Daten speichern und Messabläufe automatisieren. Im Gegensatz zu klassischen Messgeräten besitzen Mikrocontroller zunächst keine fest vorgegebene Funktionalität. Erst durch die Programmierung wird festgelegt, welche Sensoren ausgelesen werden, welche Daten verarbeitet werden und wie die Messergebnisse dargestellt oder gespeichert werden.

In diesem Versuch wird anhand verschiedener Sensor- und Aktormodule des Tinkerforge-Systems die grundlegende Programmierung eines Mikrocontrollers mit Python erlernt. Dazu werden zunächst verschiedene Sensoren zur Messung von Temperatur, Luftfeuchtigkeit, Luftdruck sowie elektrischer Leistung ausgelesen und deren Messwerte verarbeitet. Anschließend werden Messdaten kontinuierlich erfasst, auf einem Display dargestellt, in Dateien gespeichert und während der Messung grafisch visualisiert. Abschließend werden die erzeugten Messdaten automatisiert in ein elektronisches Laborbuch übertragen.

Die Temperaturmessung während des Erwärmens von Wasser dient dabei als Beispiel für ein vollständig automatisiertes physikalisches Experiment. Neben der Erfassung der Temperatur wird gleichzeitig die elektrische Leistungsaufnahme des Wasserkochers gemessen, sodass aus den aufgezeichneten Messdaten die aufgenommene elektrische Energie sowie der Wirkungsgrad des Wasserkochers bestimmt werden können.

Basisprogramm Python:

1. Lesen Sie die Messwerte für Temperatur, Luftfeuchtigkeit und Luftdruck mit dem Air Quality Bricklet aus und geben Sie diese mit Einheiten auf der Konsole aus.
2. Lesen Sie die Temperatur des Thermocouple Bricklet 2.0 aus und geben Sie den Messwert mit Einheit auf der Konsole aus.
3. Schreiben Sie einen Text auf das OLED 128x64 Bricklet 2.0. Stellen Sie die aktuelle Temperatur auf dem OLED dar.
4. Messen Sie mit dem Energy Monitor Bricklet die Netzspannung, den Strom, die Wirkleistung und den Leistungsfaktor eines elektrischen Verbrauchers und geben Sie die Werte auf der Konsole aus.



Messdatenerfassung:

5. Wetterstation: Schreiben Sie die Messwerte der Temperatur, der Luftfeuchtigkeit und des Luftdrucks auf das OLED 128x64 Bricklet. Aktualisieren Sie die Anzeige einmal pro Sekunde, ohne dass das Display flackert.
6. Temperaturmessung: Speichern Sie die Temperatur des Thermocouple Bricklet zusammen mit einem Zeitstempel in einer Datei. Optional: Verwenden Sie das Dateiformat HDF5. Speichern Sie zusätzlich Metadaten (Versuchsname, Datum, Sensor, Einheit, Abtastrate und Experimentator).
7. Führen Sie eine Temperaturmessung während des Erhitzens von 1 Liter Wasser in einem Wasserkocher durch. Erfassen Sie gleichzeitig die elektrische Leistung mit dem Energy Monitor Bricklet. Bestimmen Sie aus den Messdaten die aufgenommene elektrische Energie und berechnen Sie den Wirkungsgrad des Wasserkochers. Berechnen Sie die gesamte vom Wasser aufgenommene Wärmeenergie sowie die aufgenommene elektrische Energie. Vergleichen Sie beide Größen.
8. Schreiben Sie ein Programm, welches die aktuellen Messdaten, ohne die Messung zu unterbrechen, in eine Grafik einzeichnet. Nutzen Sie dafür Matplotlib. Aktualisieren Sie die Grafik während der Messung fortlaufend.

Upload der Daten:

9. Laden Sie die erzeugte Messdatei einschließlich der Metadaten automatisch in das elektronische Laborbuch hoch.
10. Laden Sie alle geschriebenen Skripte in das elektronische Laborbuch hoch.

Tinkerforge, Python-Funktionen, Dateiausgabe, Abtastrate, HDF5-Dateien, Thermoelement, Wirkleistung, Stromwandler

1. Tinkerforge GmbH: Python Bindings Documentation https://www.tinkerforge.com/de/doc/Software/API_Bindings_Python.html
2. Tinkerforge GmbH: OLED 128x64 Bricklet 2.0 – Python API https://www.tinkerforge.com/de/doc//Software/Bricklets/OLED128x64V2_Bricklet_Python.html
3. Tinkerforge GmbH: Thermocouple Bricklet 2.0 – Python API https://www.tinkerforge.com/de/doc//Software/Bricklets/ThermocoupleV2_Bricklet_Python.html
4. Tinkerforge GmbH: Air Quality Bricklet – Python API https://www.tinkerforge.com/de/doc//Software/Bricklets/AirQuality_Bricklet_Python.html
5. Tinkerforge GmbH: Energy Monitor Bricklet – Python API https://www.tinkerforge.com/de/doc/Software/Bricklets/EnergyMonitor_Bricklet_Python.html
6. h5py Documentation (Quick Start Guide) <https://docs.h5py.org/en/latest/quick.html>
7. Matplotlib Documentation <https://matplotlib.org/stable/index.html>
8. PyBis Documentation <https://pypi.org/project/pybis/>



- ▶ Tinkerforge Masterbrick 3.2
- ▶ Tinkerforge OLED 128x64 Bricklet 2.0
- ▶ Tinkerforge Air Quality Bricklet
- ▶ Tinkerforge Thermocouple Bricklet 2.0
- ▶ Thermoelement Typ K
- ▶ Tinkerforge Energy Monitor Bricklet
- ▶ 30 A Stromwandler
- ▶ Wasserkocher

- ▶ Zur Unterstützung der Programmierung kann die lokale KI des FP genutzt werden.
- ▶ Mit dem Programm Brick Viewer können Sie die ID der einzelnen Bricklets auslesen.
- ▶ Achten Sie darauf, dass das heiße Wasser nicht in Kontakt mit Ihrer Haut kommt.