

Pflichtenheft: SnowHeight - Sensoraufbau für räumliche Schneehöhenmessungen inklusive Handy-App

1. Einleitung

1.1 Projektziel

Das Projektziel ist ein funktionsfähiges Schneehöhenmessgerät mit mehreren Sensoren, das Daten eigenständig aufnimmt und mittels Handy-App ausgelesen werden kann.

1.2 Projektscope

Projektdauer: 9 Wochen, ungefähr 32 Stunden

Projekthintergrund: Fortführung eines vorherigen Projekts, Motivation zur Unterstützung von Fachleuten der Universität IBK

1.3 Referenzen

Atmel Atmega 2560 Datasheet: Enthält das Datenblatt des Atmel Atmega 2560 Mikrocontrollers. Bietet technische Informationen über den Mikrocontroller, einschließlich Pin-Belegungen, elektrischer Eigenschaften und Funktionen.

Bosch (2017). BME 280 Sensor: Beschreibt den BME 280 Sensor von Bosch. Das Datenblatt enthält Informationen über den Sensor, wie z.B. seine Funktionsweise, technische Spezifikationen und Anwendungshinweise.

Datalogger (2017). SD Memory Reader Datasheet: Im Datenblatt werden Informationen über den SD-Speicherkartenleser für Datalogger bereitgestellt. Es enthält technische Spezifikationen, Anschlussdiagramme und Hinweise zur Verwendung des Lesegeräts.

ElecFreaks (2017). Ultrasonic Ranging Module HC - SR04 Datasheet: Beschreibt das HC-SR04 Ultraschall-Entfernungsmessmodul von ElecFreaks. Es enthält technische Details wie Messbereich, Genauigkeit und Betriebsanleitung für das Modul.

1.4 Referenz-Links

Die entsprechenden Links zu den Datenblättern:

http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

https://cdnshop.adafruit.com/datasheets/BST-BME280_DS001-10.pdf

<http://datalogger.pbworks.com/w/file/attach/89507207/Datalogger%20-%20SD%20Memory%20Reader%20Datasheet.pdf>

<http://www.electfreaks.com/store/hcsr04-ultrasonic-sensor-distance-measuringmodule-ultra01-p-91.html>

https://www.robotshop.com/media/files/pdf2/pli03_specifications_and_hardware.pdf

2. Systemübersicht

2.1 Beschreibung des Projekts

Als Hauptcontroller wird ein Atmel ATmega2560 auf einem Arduino-Board verwendet.

Anstelle von Arduino-Bibliotheken wurde von den Vorgängern bereits in Atmel Studio in reinem C++ programmiert.

Die Versorgungsspannung jedes Moduls wird im Standby-Modus abgeschaltet, um Energie zu sparen. Vor jeder Verwendung der Laser-Servomotoren wird der Batteriestatus überprüft, um eine vollständige Entladung während des Betriebs zu verhindern.

Die erforderlichen Einstellungen können über die Android-App eingegeben und dann an den EEPROM des μ Controllers übertragen werden.

2.2 Systemarchitektur

1. Hardwarekomponenten:
 - Atmel ATmega2560 auf einem Arduino-Board als Hauptcontroller
 - Verschiedene Sensormodule wie der BME280-Sensor für Umgebungsdaten, ein SD-Kartenleser für die Datenspeicherung und ein HC-SR04 Ultraschallmodul für die Schneehöhenmessung
 - Laser-Servomotoren zur Steuerung der Sensoren
 - Stromversorgungseinheit und Batterie zur Energieversorgung des Systems
2. Softwarekomponenten:
 - Programmierung in C++ mit Atmel Studio für die Steuerung des Hauptcontrollers und der Sensormodule
 - Entwicklung einer Android-App zur Benutzerinteraktion und Konfiguration der Systemeinstellungen
 - Verwendung von EEPROM-Speicher des μ Controllers zur Speicherung der Systemkonfigurationen
3. Kommunikation:
 - Kommunikation zwischen dem Hauptcontroller (ATmega2560) und den Sensormodulen über entsprechende Schnittstellen (z. B. I2C, SPI)
 - Kommunikation zwischen dem Hauptcontroller und der Android-App über eine drahtlose Verbindung (z. B. Bluetooth, WLAN)
4. Systemsteuerung:
 - Überwachung des Batteriestatus vor der Verwendung der Laser-Servomotoren, um eine vollständige Entladung während des Betriebs zu verhindern
 - Einsatz eines Watchdog-Timers zur Erkennung von schwerwiegenden Fehlfunktionen und Auslösung eines Systemresets

3. Anforderungen

3.1 Funktionale Anforderungen

3.1.1 Datenerfassung

- Erfassung von Schneehöhe, Temperatur, Luftfeuchtigkeit, Druck usw.
- Periodische Messungen

3.1.2 Datenübertragung

- Bluetooth-Kommunikation zwischen Mikrocontroller und Handy-App
- Senden der gesammelten Daten an die SD-Karte über den Mikrocontroller

3.1.3 Datenabfrage

- Möglichkeit, auf die gespeicherten Daten auf der SD-Karte zuzugreifen
- Anzeige von Wetterdaten und Schneehöhenverlauf in der Handy-App

3.2 Nicht-funktionale Anforderungen

3.2.1 Leistung

- Schnelle und zuverlässige Datenerfassung und Übertragung
- Energieeffizienz

3.2.2 Benutzerfreundlichkeit

- Intuitive Benutzeroberfläche in der Handy-App
- Klare Anweisungen zur Bedienung der Station

3.2.3 Zuverlässigkeit

- Robustheit gegenüber extremen Wetterbedingungen
- Stabile Bluetooth-Verbindung

3.2.4 Wartbarkeit

- Möglichkeit zur Aktualisierung der Firmware des Mikrocontrollers
- Einfache Erweiterbarkeit der Funktionalität

4. Benutzeroberfläche der Handy-App

- Startbildschirm mit Verbindungsoptionen und grundlegenden Informationen
- Anzeige der aktuellen Wetterdaten und Schneehöhenverlauf
- Einstellungen für Messintervalle und Bluetooth-Kommunikation
- Datenabfrage und Anzeige von auf der SD-Karte gespeicherten Daten

5. Hardwareanforderungen

- Mikrocontroller für die Datenverarbeitung und Bluetooth-Kommunikation
- Sensoren zur Erfassung von Schneehöhe, Temperatur, Luftfeuchtigkeit, Druck usw.
- SD-Karte für die Datenspeicherung

6. Schnittstellenanforderungen

- Bluetooth-Kommunikation zwischen der Handy-App und dem Mikrocontroller
- Schnittstelle zwischen dem Mikrocontroller und der SD-Karte

7. Zeitplan

- Genaue Zeitplanung für die Entwicklung der Hardware und Software:

9 Wochen - entspricht 32 Schulstunden

8. Projektorganisation

8.1 Teamzusammensetzung

Teamleiter

- Fabian Kessler: Verantwortlich für die Koordination und Überwachung des Projekts
- Claire Pramstaller: Zuständig für die Entwicklung der Mikrocontroller-Firmware
- Jakob Moser: Zuständig für die Entwicklung der Handy-App (Frontend) & (Backend)
- Christian Kaufmann: Zuständig für etwaig notwendige Dokumentationen/sonstiges

8.2 Verantwortlichkeiten und Aufgabenbereiche

- Fabian Kessler: Koordination des Projekts, Kommunikation, Überwachung des Zeitplans, sowie Hardwaretechnisch notwendige Anpassungen.

- Claire Pramstaller: Programmierung der Mikrocontroller-Firmware, Integration der Hardwarekomponenten

- Jakob Moser: Entwicklung der Benutzeroberfläche und Interaktionslogik in der Handy-App, Implementierung der Datenübertragung und -speicherung in der Handy-App

- Christian Kaufmann: Ergänzung jeglicher anfalligen Dokumentationen, Erstellung des Pflichtenheftes, Instandhaltung des Lastenheftes