! Fork From IOT-Messwagen - Ronny140778

This is Forked From https://github.com/Ronny140778/IOT-Messwagen

My Changes Are:

- Added Support for vsCode
- Reichelt.de Shopping Cart
- ebay.de: original Part TLE4906L
- amazon.de: USB-C Buchse

IOT-Messwagen

IOT-Messwagen für die Modellbahn

Wichtig 1

Wer das Projekt unterstützen/honorieren möchte, den bitte ich das Geld an folgende goFundMe-Kampange zu spenden.
https://www.gofundme.com/f/hilfe-fr-eine-therapie-fr-unseren-sohn-dennyric

Wichtig 2

Ich bin ein Java-Entwickler, es kann also sein, dass einige Dinge nicht so programmiert wurden wie es ein echter C-Entwickler machen würde..

Umfang des Projektes

Dieses github-Projekt bildet zusammen mit einem Arduino nano 33 IOT und einem Hallsensor einen Messwagen für die Modellbahn. Die ermittelten Daten werden im Browser angezeigt und die ermittelten Werte können mittels Schaltflächen zurück gesetzt werden. Der Messwagen kann sich entweder mit dem heimischen Netzwerk verbinden oder als Access-Point arbeiten.
Folgende Daten werden ausgegeben:

- Steigung/Gefälle sowie die erreichten Min/Max-Werte
- Neigung sowie die erreichten Min/Max-Werte
- Geschwindigkeit (diese wird anhand des Radumfangs und der Anzahl der verwendeten Magnete ermittelt)
- Geschwindigkeit im Original (anhand des konfigurierten Maßstabes)
- Strecke (Modell und Original)

Benötigte Bibliotheken

- Arduino_LSM6DS3
- WiFiNINA
- FlashStorage_SAMD

Konfigurationsmöglichkeiten

arduino_secrets.h

- hier können SSID und Passwort hinterlegt werden, damit sich der Messwagen mit dem eigenen Netzwerk verbinden kann. Der Arduino erwartet dann per DHCP eine IP zugewiesen zu bekommen.

Messwagen.h

- const int scale = 160; (Maßstab der Modellbahn, dient zur Ermittlung der Geschwindigkeit im Original)
- const float wheelCircumFerence = 1.948; (Umfang des Rades, zur Ermittlung der Geschwindigkeit und der zurückgelegten Strecke)
- const byte magnets = 1; (Anzahl der Magnete die den Hallsensor auslösen)
- const byte interruptPin = 2; (Pin an dem der Hallsensor angeschlossen ist)
- const IPAddress ip(192,168,1,1); (Wenn der Arduino als AccessPoint arbeitet ist die Website unter dieser IP zu erreichen)
- const int sizeAvg = 10; (legt fest aus wie vielen Werten Steigung & Neigung ein Mittelwert gebildet wird / Glättung der Werte)

Hinweise

- Der Hallsensor wird an Pin D2 angeschlossen.
- Wenn keine SSID konfiguriert ist oder das Netzwerk nicht in Reichweite ist, startet der Messwagen als AccessPoint. Es wird ein offenens Netzwerk "Messwagen" gestartet. Unter der konfigurierten IP kann dann im Browser die Website geöffnet werden. Ist eine SSID konfiguriert, kann man entweder im Rounter schauen welche IP der Messwagen bekommen hat (gibt sich beim Router auch als Messwagen aus) oder wenn man beim Start des Messwagens die serielle Schnittstelle geöffnet ist, wird auch dort die zugewiesene IP ausgegeben.
- Die Antenne des Arduino nanos liegt in Fahrtrichtung vorn (wichtig für die Ermittlung der Steigung).
- Da der Arduino nano nur 3,3 V an den analogen und digitalen Eingängen verträgt habe ich einen unipolaren Hallsensor genutzt, der bereits mit 3,3 V arbeitet.
- Für die Stromversorgung nutze ich aktuell einen Li-Ion-Akku (14500) in Verbindung mit einem Lade/Entladeregler. Dieser ist relativ schwer, es ist sinnvoll hier nach einer anderen Möglichkeit zu suchen (Mini-Lipo-Akku).

Thread bei 1zu160.net

IOT-Messwagen bei 1zu160.net

uwe-magnus.de

IOT-Messwagen
Platinenhalter 3D Druck