



Flugroboter mit CAN Interface

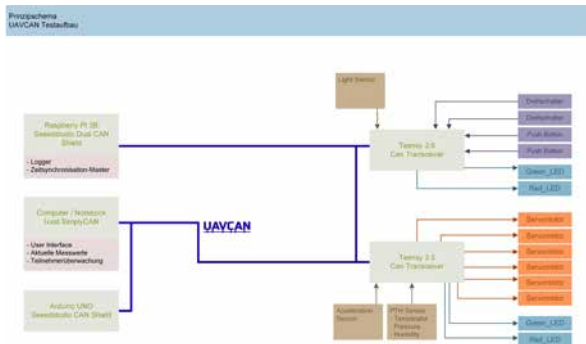


Abb. 1 Prinzipschema des UAVCAN Testnetzwerk mit fünf Teilnehmern.

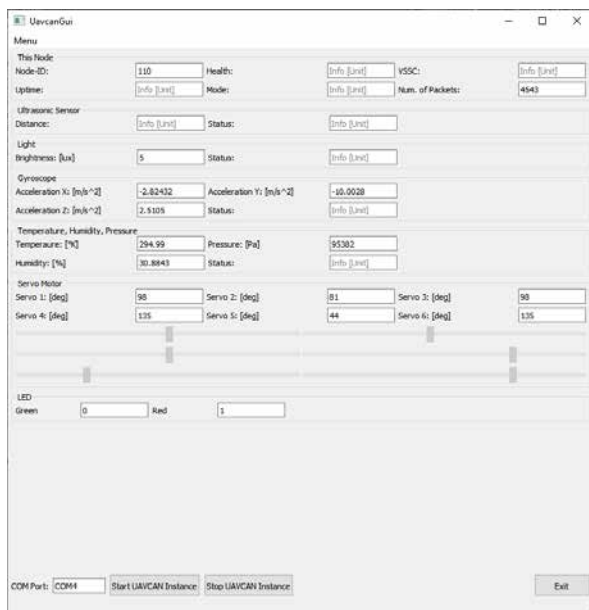


Abb. 2 GUI Applikation zur Anzeige von Statusinformationen und der Netzwerkteilnehmerüberwachung

Problemstellung

Autonome Flugsysteme und Drohnen bestehen aus mehreren, stark voneinander abhängigen Teilsystemen und Komponenten. Die Kommunikation zwischen den Komponenten erfolgt meist über direkte Punkt-zu-Punkt Verbindungen. Eine Vereinfachung der Netzwerkstruktur und zudem noch eine Reduktion des Verkabelungsaufwands, könnte durch den Einsatz von CAN erreicht werden. Die Arbeit klärt Fragen über die Verwendung von CAN in einer Flugdrohne.

Lösungskonzept

CAN als robustes, stabiles und sehr fehlertolerantes Übertragungsprotokoll eignet sich gut für die Anwendung in einem autonomen Fluggerät. Als höheres Protokoll, aufbauend auf CAN, empfiehlt sich UAVCAN. Die high-level Angelegenisse zur Definition von Datentypen vereinfacht eine Kommunikationsintegration ungemein.

Realisierung

In einem Testaufbau bestehend aus mehreren Microcontrollern, Servomotoren, Sensoren, einem Raspberry Pi und einem durch einen Adapter eingebundenen Computer, wurde die Umsetzung, Funktionalität, Verwendung und Stabilität von CAN und UAVCAN konkret getestet. Die daraus gewonnenen Erkenntnisse werden für den Entwurf eines möglichen Netzwerks in einer Drohne verwendet. Durch die automatische Umwandlung objektorientierter Datentypen in einfachen C Code kann ein Netzwerk auf einfache Weise umfangreich aber doch sehr logisch gestaltet und erweitert werden.

Ergebnisse

CAN ist ein äusserst robustes, schnelles und fehlertolerantes Übertragungsprotokoll. UAVCAN als höheres Protokoll vereint die Vorzüge von objektorientierter Datentypdefinition mit der Schlantheit einer hoch integrierten und echtzeitfähigen Protokollimplementierung. In einer Flugdrohne kann durch einen Bus ein grosser Aufwand an Verkabelung und die Menge an Stichleitungen reduziert werden. Die fehlende Kompatibilität mit CAN von bestehenden Teilnehmern, stellt jedoch insoweit ein Problem dar, dass für einen kompletten Aufbau diverse Geräte erneuert werden müssen.

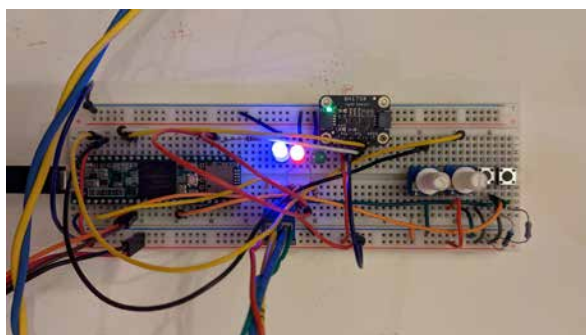


Abb. 3 Prinzipschema der Schaltung des Teensy 3.5 zur Steuerung von sechs Servomotoren und zwei Sensoren.

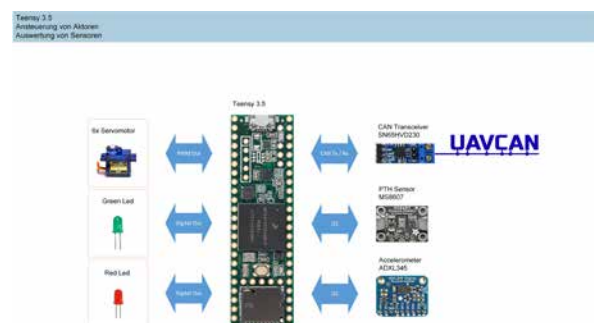


Abb. 3 Schaltung des Teensy 3.6 mit UAVCAN Netzwerk-anbindung, einem Sensor und vier Eingabemöglichkeiten.