



Microcontroller

Intelligenz in Projekte einbauen

Warum Microcontroller nutzen?

Ein kleiner Computer für alle Lebenslagen: Microcontroller lassen sich flexibel für unterschiedliche Aufgaben programmieren - ob es nun darum geht ein LED blinken zu lassen oder komplexe Sensorsysteme abzufragen, auszuwerten und miteinander zu kommunizieren. Auch wenn der Controller alleine manchmal etwas kompliziert erscheint, Modulplattformen wie Arduino erleichtern die Nutzung enorm.

Arten von Microcontrollern:

Atmega: Arduino, Standardeinsteigermodell im Arduino Uno

MSP430: Launchpad/Energia System (Arduino in Rot)
Etwas bessere Hardwaremöglichkeiten & preiswerter

ARM Cortex-M: Leistungsfähiger als der Atmega

ESP32: per Arduino-IDE programmierbar, WLAN und Bluetooth, preiswert

Eine andere Art Intelligenz einzubauen sind weniger weit verbreitete FPGAs, wo statt Programmierung des Ablaufs eine elektrische Schaltung einprogrammiert wird.

Produktionsprozess:

1. Prototyping am besten erstmal mittels einer Plattform wie Arduino, Breadbord und Modulen
2. Immer eine Funktion/Modul nach dem anderen integrieren
3. Schliesslich eventuell Umbau auf eigene PCB (Kleiner und Zuverlässiger)

Microcontroller

Intelligenz in Projekte einbauen

CheatSheet

Tips und Tricks

- Für fast jede Teilaufgabe wird es schon eine ähnliche Sache geben, die schon mal jemand mit dem Arduino gemacht hat. Der Vorteil von Open-Source-Systemen ist ja das man auf den Schultern seiner Vorgänger stehen darf (und deren Entwicklungen weiterführen)
- Im wesentlichen nutzt man entweder I2C, die serielle Schnittstelle (zum debuggen) und analogen bz. digitalen Eingang oder Ausgang um Sensoren und Aktuatoren anzusteuern
- Einmal die Arduino-Beispielbibliothek durchspielen hilft um für fast alle Komponenten die kommen können gewappnet zu sein. Und wenn nicht: Es wird mit hoher Wahrscheinlichkeit auch dazu eine Bibliothek mit Beispielen geben - was meist einfacher ist als Datenblätter zu wälzen und selber zu lösen
- Bluetooth-Debugging statt serieller Schnittstelle? - ein HC05-Modul einfach an die serielle Schnittstelle hängen und mit dem Rechner koppeln, danach kann man da wie zuvor kabelgebunden kommunizieren
- Processing ist das ideale Gegenstück auf Rechnerseite (tatsächlich ist die Arduino-Programmierungumgebung danach entwickelt) um einfach eigene Programme zu schreiben. Und mittels serieller Schnittstelle kann man auch gut mit dem Mikrocontroller reden

Beachte:

- Spannungslevel der Controller: Wenn ein 3.3V Controller mit einem 5V System zu kommunizieren versucht können Verständnisschwierigkeiten auftreten.
- Noch fieser: wenn man die gleiche Spannungsversorgung nutzt kann der der weniger braucht schon mal in Rauch aufgehen

Sensoren, Aktuatoren und andere nutzbare Module:

- LED, Motoren, Taster, Entfernung, Gesten, Bewegung, GPS, Beschleunigung, Bluetooth, WLAN, Temperatur, Gas, Feuchtigkeit, Druck,...

Troubleshooting:

- Verbindungen prüfen - Wackelkontakte machen keine Freude
- Spannungen und Stromverbrauch überprüfen
- Digitale, Analoge Werte und auch serielle Protokolle lassen sich messen und ausgeben

www.makercheatsheets.de



Personal
Photonics