ArduSmartPilot – High Tech Mechatronik von Schülern

Einen Spielzeug-Wurfgleiter zu einem Motorflugzeug umzurüsten, um ihn mit einer selbst programmierten Smartphone-App zu steuern, klingt nach „Rocket Science“ - absolute Hochtechnologie.

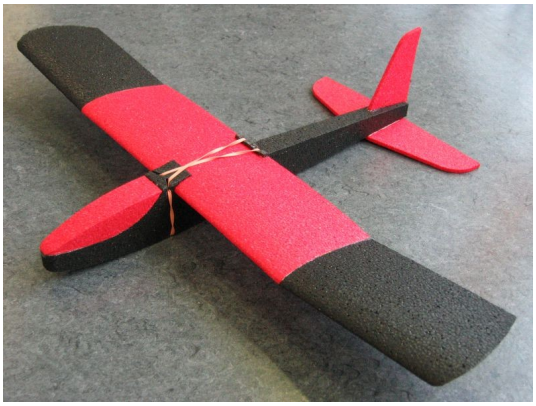
Ist es aber nicht: Wir haben bewiesen, dass Schüler in der Lage sind, dafür die komplette Mechatronik von der App-Software bis zum Antrieb zu realisieren.

Und dies überwiegend mit Open Source Soft-/Hardware bei Materialkosten unter 60 € für das komplette Flugzeug.

Projekt ArduSmartPilot:

Was in der Kindheit unserer Väter Dampfmaschinen waren, sind für unsere Kinder heute komplexe mechatronische Systeme wie automatische Fertigungen oder moderne Automobile.

Unsere Väter lernten die Technik ihrer Zeit über Modell-Dampfmaschinen kennen. Das Projekt ArduSmartPilot macht die heutigen Kinder mit mechatronischen Systeme vertraut, indem sie bei einem Modellflugzeug - dem sogenannten „ArduSmartPilot“- die Mechatronik inklusive Fernsteuerung selbst planen, bauen und in Betrieb nehmen.



Die Basis hierbei ist der Spielzeug-Wurfgleiter Felix80 des Hersteller Miniprop in Röthenbach (siehe Foto), den Schüler in ein intelligentes mechatronisches System umbauen.

Das Herzstück des ArduSmartPilot ist eine Arduino-Mikrocontroller Platine. Diese wird mit einem Bluetooth (BT) Sender versehen, damit sie mit einem Android Smartphone kommunizieren kann. Der Arduino steuert die Höhen- und Seitenruder des Flugzeugs sowie den Elektromotor für den Propeller. Dazu wird eine Android App von den Schülern erstellt, mit der das Flugzeug via Bluetooth über den Arduino gesteuert wird. Der Arduino empfängt zudem Daten mitfliegender Sensoren und überwacht die Qualität der Funkverbindung bzw. den Akkuladezustand des Flugzeugs.

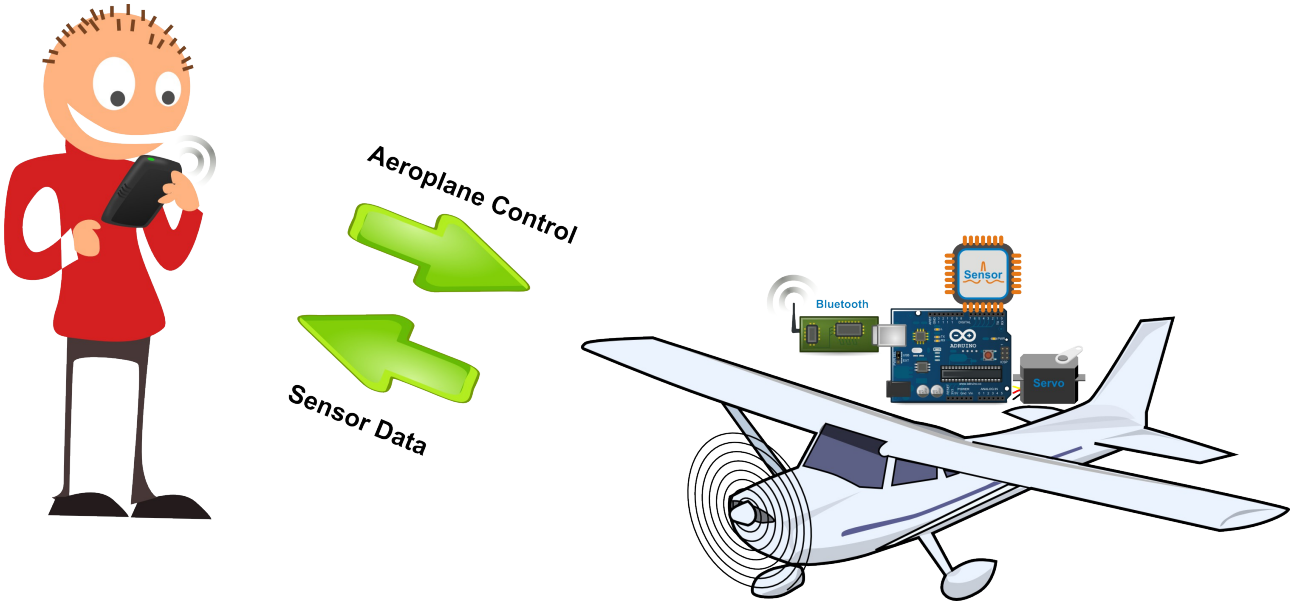
Die Schüler erfahren dabei hautnah die Kerngebiete der Mechatronik: Mechanik, Antriebe, Sensorik, Mikrocontroller, Elektronik, Informatik und Kommunikationstechnik.
Typisch für eine Hochschule der angewandten Wissenschaften steht beim ArduSmartPilot-Projekt der Praxisbezug im Mittelpunkt:
Die Schüler erstellen wie Mechatronik-Studenten Software und elektronische Schaltkreise. Sie entwerfen Mechanikteile, bauen Kinematiken und integrieren alle diese Komponenten mechanisch, elektrisch und algorithmisch zu einem funktionsfähigen System.

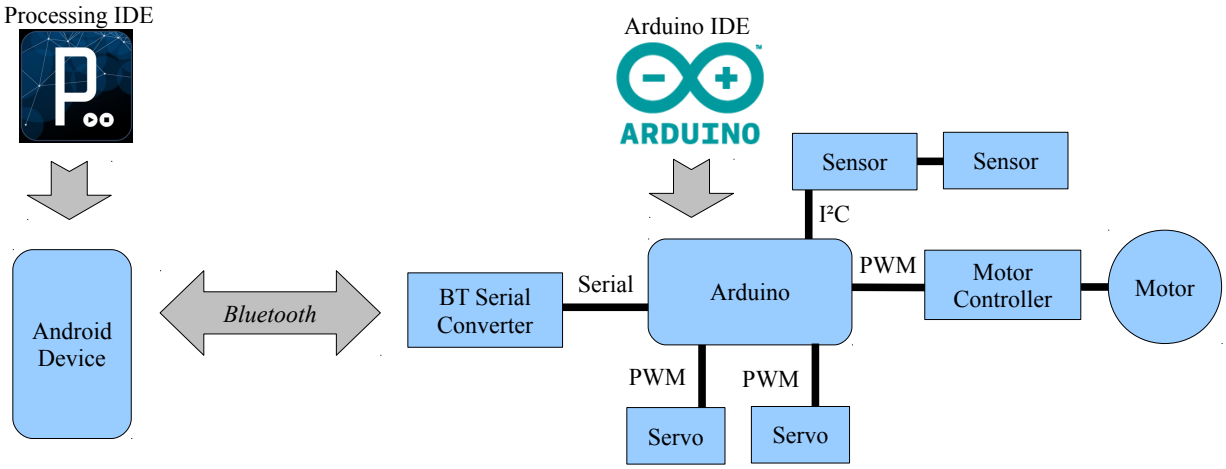
Anders als noch vor zehn Jahren ist dieses abenteuerliche Vorhaben heutzutage als Schülerprojekts realisierbar:
Das liegt an den geringen Komponentenkosten, einfach zu bedienender Software-Entwicklungsumgebungen (IDE) und speziell an der Arduino-Plattform. Hinzu kommt folgende Tatsache: Nahezu jeder Schüler hat in seiner Hosentasche ein ideal geeignetes Endgerät, um das Flugzeug zu

steuern und dessen Sensorsignal zu visualisieren: Das Smartphone.

Systemdesign:

In den nachfolgenden Grafiken ist das Systemdesign des ArduSmartPilot dargestellt:
Mit der Open Source IDE Processing wird eine Android App erstellt. Diese App sendet die Steuerbefehle über Bluetooth an einen Bluetooth-Seriell-Konverter, der an einen Arduino angeschlossen ist. Die Steuerbefehle können z.B. eine auf dem Bildschirm gedrückte Taste, das Betätigen der seitlichen Lautstärketasten oder ein Kippen des Smartphones sein. Hierbei entscheiden die Schüler selbst, auf welche Weise sie das Flugzeug steuern möchten.
Der Arduino setzt diese Steuerbefehle in Pulsweiten-Signale (PWM) um, mit denen die beiden Servos und der Motorregler gesteuert werden.





Neben der Kommunikation vom Smartphone zum Flugzeug ist aber auch die Kommunikation in die Gegenrichtung notwendig:
Der Arduino zählt die empfangenen Steuerbefehle pro Sekunde und meldet diesen Wert an das Smartphone zurück. Über diesen Wert kann die App des Smartphones erkennen, wenn die Funkverbindung an ihre Grenzreichweite kommt. (Der Arduino erkennt dies ebenfalls, und schaltet dann z.B. den Motor aus.)

Ein Pilot möchte aber noch weitere Informationen über den Zustand des Flugzeugs erfahren: Dazu

wird vom Arduino die Akkuspannung gemessen. Weiter sind ein Drucksensor, ein 3-Achsen Beschleunigungs- und Drehratensensor an den Arduino angeschlossen.
Alle diese Werte werden ebenfalls vom Flugzeug an das Smartphone zurück gesendet.

Verwendete Hardware:

Leider ist es nur mit sehr hohem Aufwand möglich, die Smartphone App unabhängig vom Betriebssystem (Windows, Apple, Android) zu erstellen. Aufgrund der kostenlosen Verfügbarkeit der Android Entwicklungssoftware sowie der Dominanz von Android Geräten bei Schülern werden die Apps zur Steuerung nur für Android erstellt.

Die Hardware des ArduSmartPilot ist nach folgenden Gesichtspunkten ausgewählt:

- Gesamtkosten kleiner 60 € pro Flugzeug.
- Plattformunabhängige Open Source Programmiersoftware.
- Geringes Gewicht.
- Mechanische und elektronische Robustheit.
- Weite Verbreitung in der Do It Yourself (DIY) Szene.

Diesen Vorgaben entspricht am besten die Arduino-Plattform: Hier existieren eine Vielzahl untereinander kompatibler Mikrocontroller-Platinen, mit unterschiedlicher Größe/Gewicht und Schnittstellen zum Programmieren. Aufgrund des Open Source Hardwaredesigns gibt es weltweit viele Hersteller dieser Platinen. Dies führt zu geringen Verkaufspreisen ab ca. 5 € für einen Arduino.

Arduino Platinen sind robust und weit verbreitet. Daher existieren zu allen auftretenden Problemen entsprechende Internetbeiträge. Zudem existieren für die Peripherie (Servos, Bluetooth-Modul usw.) fertige Bibliotheken.

Aus Gewichts- und Kostengründen wurde für dieses Projekt ein „Arduino Pro Mini“ (Atmega 328, 3,3 V, 8 MHz) ausgewählt. Bis auf die Prozessortaktrate ist dieser Arduino fast identisch zu einem „normalen“ Arduino Uno. Er besitzt aber keinen USB auf Seriell Konverter, den man benötigt, um ein Programm aus der IDE heraus via USB auf den Mikrocontroller zu übertragen. Dafür ist ein externer Konverter (z.B. eine FTDI oder eine Foca Platine) notwendig. Die 3,3 V Variante des Arduino Pro Mini wurde gewählt, da die Bluetooth-Module wie auch die Sensoren diese Betriebsspannung verwenden.

Ebenfalls aus der DIY-Szene stammen die beiden Bluetooth-Module HC-05 und BTM222, die alternativ eingesetzt werden können. Sie unterscheiden sich in der Reichweite und im Antennenanschluss: Das HC-05 Modul ist ein Class 2 Modul (nominelle Reichweite 10 m) mit integrierter Antenne. Das BTM222-Modul ist ein Class 1 Modul (nominelle Reichweite 100 m) mit einer externen Antenne.

Da immer eine Sichtverbindung zum Flugzeug existiert, wird selbst mit dem Class 2 Modul eine effektive Freifeld-Reichweite von ca. 80 m erreicht. Beide Module haben die Funktion die Daten der seriellen Schnittstelle des Arduino in Bluetooth-Signal zu wandeln bzw. umgekehrt.

Die Servos, der Akku, der Motorregler und der Motor sind wie die weiteren Kleinteile Standardbauteile aus dem Flugzeugmodellbau.

Das Flugzeug selbst ist ein EPP (extrudiertes Polypropylen) Wurfgleiter des deutschen Herstellers Miniprop. Dieses Flugzeug besitzt ein sehr stabiles Flugverhalten und ist äußerst robust. Das EPP Material lässt sich sehr einfach mit einem Cuttermesser bearbeiten und - wenn es doch einmal bricht - mit Epoxykleber leicht wieder reparieren.

Neben den Motoraufnahme, den Aussparungen für den Akku und die beiden Servos erstellen die Schüler das Höhen- und Seitenruder inklusive der Mechanik für die Ruderanlenkung. Für die Ruderflächen wird z.B. leichtes Sperrholz oder EPP-Material in Plattenform verwendet.

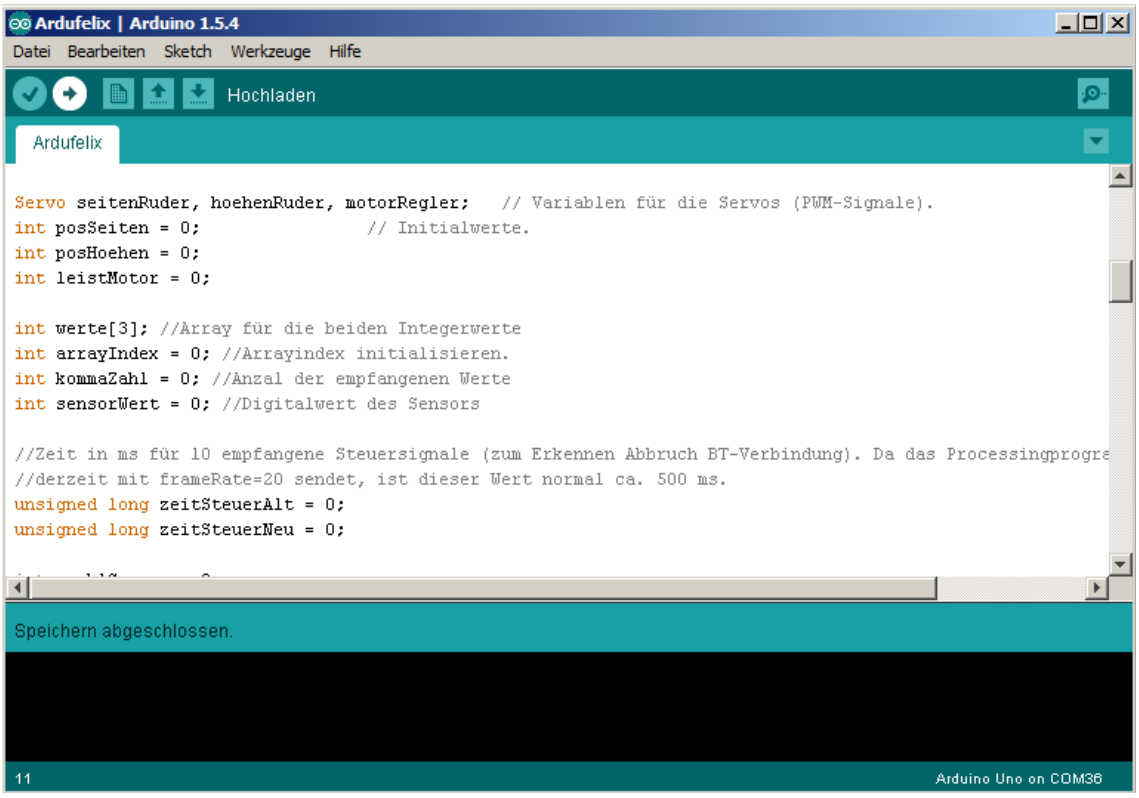
Verwendete Software:

Für den ArduSmartPilot sind zwei getrennte Softwareentwicklungen nötig: Erstens wird eine Android App und zweitens ein Arduino-Programm erstellt.

Die Arduino-Plattform ist nicht nur eine Hardwareplattform sondern beinhaltet auch eine Open Source IDE (Entwicklungsumgebung), die sowohl auf Mac, Linux und Windows läuft. Die Software des Mikrocontrollers wird hier über eine einfache C-artige Sprache geschrieben, für die es eine große Anzahl von Tutorials im Internet und entsprechende Lehrbücher gibt.
Die „Sketches“ genannten Arduino-Programme sind unverändert auf jede Arduino-Platine übertragbar, auch wenn sich darauf unterschiedliche Mikrocontroller befinden. Die Sketches werden daher zuerst auf einem handlichen Arduino Uno entwickelt und nach erfolgreichen Tests auf einen Arduino Pro Mini übertragen.

Die Möglichkeiten der Programmierung sind durch die C-ähnliche Sprache der Arduino IDE beschränkt. Dies lässt sich aber sehr leicht umgehen, da man innerhalb eines normalen Quelltextes auch Maschinencodebefehle und C++ Befehle verwenden kann. Dadurch erreichen die Schüler/Studenten einen fließenden Übergang in das professionelle Programmieren von Mikrocontrollern.

Nachfolgend ist ein Screenshot der Arduino IDE dargestellt.



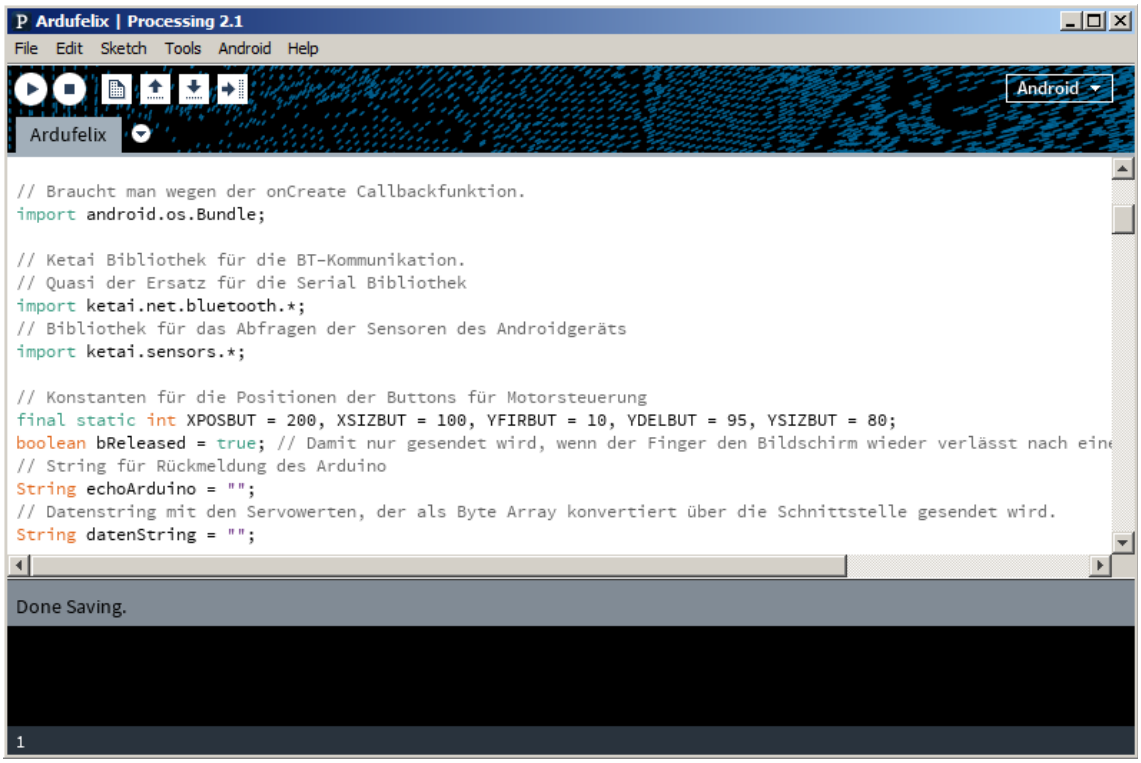
Das Erstellen einer Android App ist generell ein sehr komplexer Prozess, der gute Java-Kenntnisse und einen Umgang mit professionellen IDEs wie Eclipse voraussetzt. Die Programmierung wird wegen der hohen Diversität der Android-Endgeräte zusätzlich komplexer. Schließlich ist der Zugriff auf das Bluetooth-Modul im Android-Endgerät keine triviale Programmieraufgabe.
Die bis hierhin genannten Tatsachen machen die Android-Programmierung innerhalb eines Schülerprojekts eigentlich zu einem unmöglichen Unterfangen.

Doch glücklicherweise gibt es hierfür auch eine Open Source IDE, die nicht nur aus dem selben Ursprung wie die Arduino IDE stammt, sondern mit Hilfe entsprechender Bibliotheken eine ebenso einfache Programmierung erlaubt.

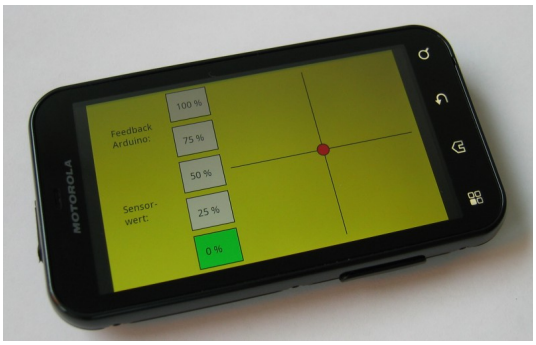
Diese plattformunabhängige IDE heißt „Processing“ und verwendet eine vereinfachte dem Java

ähnliche Programmiersprache. Es ist also kein Wunder, dass die Benutzeroberfläche von Processing fast identisch mit der der Arduino IDE ist.

Nachfolgend ist ein Screenshot der Processing IDE dargestellt.



Processing besitzt mehrere Modi, von denen in diesem Projekt der Java- und der Android-Modus verwendet werden: Im Java-Modus erstellt man Java-Programme, die sich auf einem PC/Mac ausführen lassen. Dabei handelt es sich um interaktive Grafikanimationen, d.h. der Programmablauf wird vom Benutzer über dessen Mausklicks gesteuert.



In diesem Sinne wird die App zur Steuerung des ArduSmartPilot zuerst als Java-Programm auf einem PC entwickelt, wobei der PC via USB (statt über Bluetooth) mit dem Arduino kommuniziert.

Damit lernen die Schüler die Programmierkonzepte wie Schleifen oder if-Verzweigungen nicht textbasiert sondern anhand von Grafikanimationen, was die Motivation und den Lernerfolg wesentlich erhöht. Auch kommen hier künstlerisch kreative Ansprüche an das Programmierergebnis weitaus mehr zur Geltung.

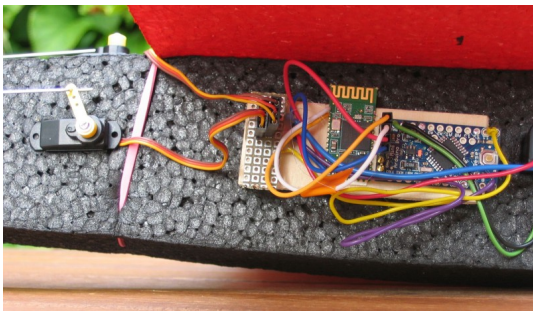
Ist das Java-Programm zur Steuerung des Flugzeugs erfolgreich am PC getestet, dann kann es im Android-Modus fast 1:1 auf das Androidgerät übertragen werden. Den Mausklicks entsprechen dann ein Tippen auf den Touchbildschirm. Das Foto oben zeigt die Benutzeroberfläche einer solchen App.

Mit Hilfe der „Ketai“-Bibliothek wird bei der Android App die Bluetooth-Verbindung realisiert sowie interne Sensoren des Smartphones ausgelesen. Damit wird es möglich, das Flugzeug z.B. mit Hilfe des Lagesensors über ein Kippen des Smartphones zu steuern.

Bevor direkt mit dem Arduino Pro Mini im Flugzeug kommuniziert wird, testen die Schüler ihre App an einem Arduino Uno, der mit einem Bluetooth-Shield ausgestattet ist (= aufsteckbares funktionsfertiges Bluetooth-Modul). Die hier geschilderte Vorgehensweise ermöglicht nicht nur den Schülern

einen schrittweisen und damit frustrationsarmen Einstieg in die Programmierung, sondern lehrt sie auch ein professionelles systematische Vorgehen beim Erstellen und Testen von Software für eingebettete Systeme.

Elektronische Schaltung

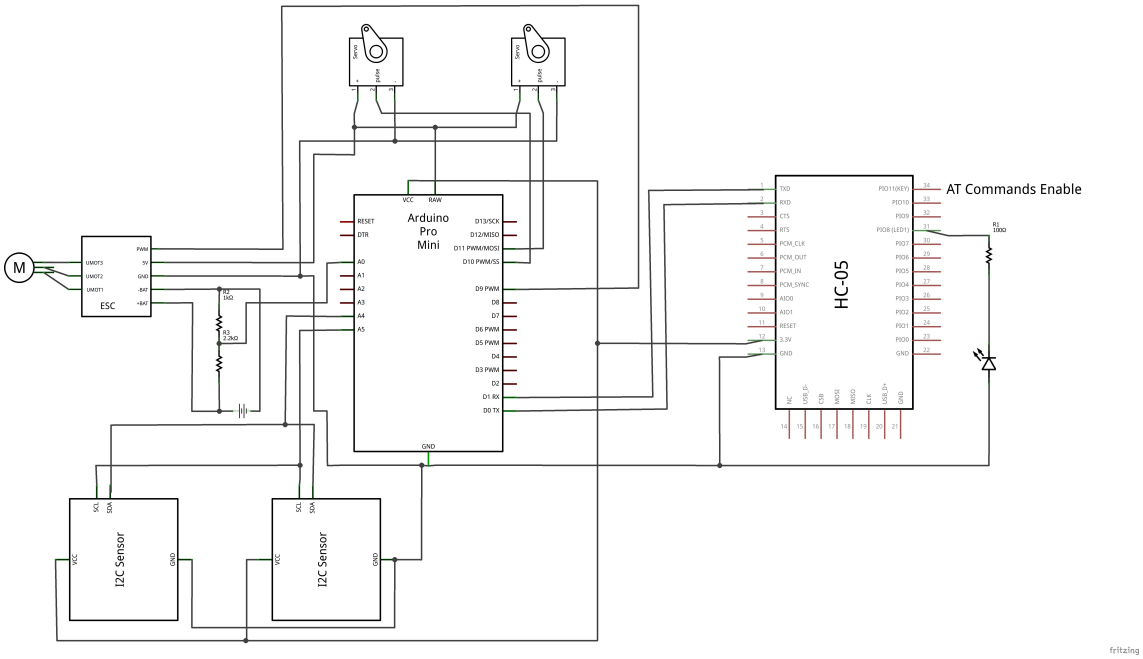


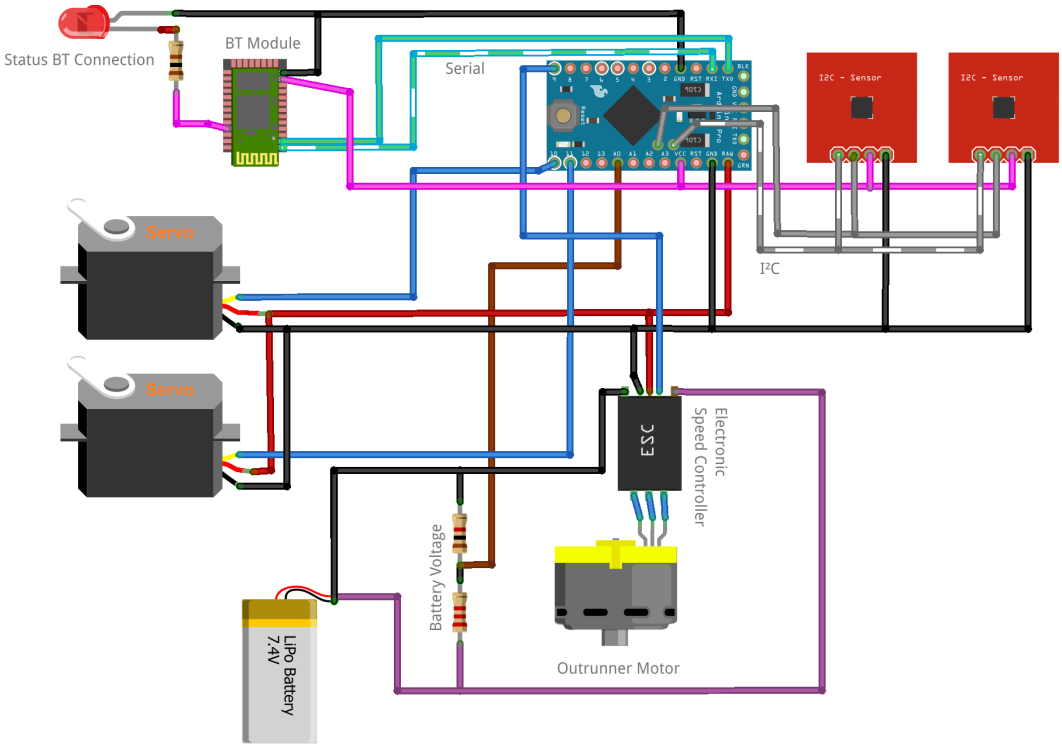
In einem konkreten Schülerprojekt wurde von der Fa. Robert Bosch GmbH in Reutlingen eine Adapterplatine erstellt, die mit Sensoren sowie mit LEDs, Tastern, Schaltern, einem Spannungsteiler und einem Summer vorbestückt ist. Auf diese Platine löten die Schüler das Bluetooth-Modul auf, kontaktieren den Arduino über Steckerleisten und stellen die restliche Schaltung über Lötverbindungen her.

Dieser Ansatz war alleine deshalb nötig, weil die Sensorbausteine nur in SMD-Gehäusen verfügbar sind.

Die elektronische Schaltung lässt sich aber auch ohne eine solche Adapterplatine z.B. mit Freidrahtverbindungen herstellen. In diesem Fall werden SMD-Sensoren auf Adapterplatinen (sogenannte „Breakouts“) verwendet, um sie einfacher kontaktieren zu können. Im Foto ist ein solcher Schaltungsaufbau beispielhaft (ohne Sensoren) dargestellt.

Der Verdrahtungsplan und der Schaltplan hierzu sind in den folgenden Abbildungen dargestellt. Hier wurde das HC-05 Bluetooth-Modul verwendet. Im Fall es BTM222-Moduls ändert sich die Pinbelegung etwas und eine externe Antenne kommt hinzu.





Ausblick:

Technisch, inhaltlich wie auch gestalterisch ist das ArduSmartPilot Projekt in der bis hierhin geschilderten Form noch sehr weit ausbaubar. In Zukunft werden folgende Ideen angegangen werden:

- Es werden Funktechniken mit höherer Reichweite wie z.B. WLAN verwendet.
- Sensoren im Flugzeug werden dazu verwendet, um das Flugverhalten aktiv zu stabilisieren.
- Der ArduSmartPilot wird autonom mit Hilfe eines GPS Sensors und eines Laserabstandssensors fliegen und landen können.
- Es werden Luftaufnahmen mit einem ArduSmartPilot möglich sein.
- Mehrere ArduSmartPilot werden einen Flugroboterschwarm bilden können.
- Die Steuerung des ArduSmartPilot wird über neue Smartphonefunktionen wie die Gestenerkennung realisiert.
- …

Diese Projektideen schließen den Kreis zwischen den Studenten der Mechatronik und den Schülern in den ArduSmartPilot-Projekten:
Im Mechatronikstudium arbeiten Studenten innerhalb ihrer Abschlussarbeiten solche neuen Projekte aus. Dabei lernen die Studenten das Design, den Aufbau, die Dokumentation und die Inbetriebnahme dieser intelligenten mechatronischen Systeme, analog zu einer späteren Berufstätigkeit in der Industrie.

Danksagung:

Ich danke den nachfolgend genannten Personen und Organisationen, die zum Erfolg dieses Projektes mit beigetragen haben:

Den Studenten des Studienbereichs Mechatronik der Hochschule Reutlingen für die Unterstützung dieses Projektes in der Form von Abschlussarbeiten und Tutorien in der Schülerarbeit.

Den Lehrern des Friedrich-Schiller-Gymnasiums in Pfullingen und des Gymnasiums im Bildungszentrum Nord in Reutlingen für die Mitarbeit an diesem Projekt.

Dipl.-Ing. Bernd Rothenberger für die Modellbauberatung.

Prof. Dr.-Ing Jürgen Schwager für die Initiierung und fachlich/organisatorische Unterstützung dieses Projektes.

Dem O'Reilly Verlag GmbH & Co. KG in Köln für die Unterstützung durch Fachbücher.

Der Fa. Robert Bosch GmbH in Reutlingen für die technische und materielle Unterstützung dieses Projektes.

Der Fa. Leuze electronic GmbH + Co. KG in Owen für die finanzielle Unterstützung dieses Projektes.

Prof. Dr. rer. nat. Stefan Mack

Hochschule Reutlingen, Fakultät Technik
Studienbereich Mechatronik

Alteburgstr. 150

72762 Reutlingen

Kontakt: stefan.mack@hochschule-reutlingen.de

Rechtliche Hinweise:

Selbst für kleine Modellflugzeuge wie den ArduSmartPilot besteht seit 2005 Versicherungspflicht. Erkundigen Sie sich bitte bei Ihrem Haftpflichtversicherer, ob Ihr ArduSmartPilot in Ihrer Privat-Haftpflichtversicherung mit eingeschlossen ist.
Fliegen Sie den ArduSmartPilot nur dort, wo Sie niemand/nichts gefährden können. Halten Sie insbesondere einen großen Abstand zu Straßen, Flughäfen oder Bahnanlagen.

Lizenz