FlightCtrl_1

15

LotharF
MikroKopter.de

Contents

Contents

1 FlightCtrl V1.0

Die FlightCtrl V1.0 gab es nur als unbestückte Platine und musste selber bestückt werden. Diese FlightCtrl wird nicht mehr vertrieben.

Hinweis:
Siehe auch: FlightCtrlAnleitung

(Beispielbild FC_V1.0)

1.1 Technische Daten Version V1.0

- Controller: AVR Atmel ATMEGA644 @20MHz.
- Sensoren: 3x Gyros (ENBC-03JA), 3-Achs-Beschleunigungsssensor (LIS3L02AS4), Luftdrucksensor (MPX4115).
- Statusanzeige: 2 LEDs (rot,grün), Elektromagnetischer Summer (z.B. für Unterspannung, Ortung, Funkausfall,...).
- Ausgänge: 2 Transistorausgänge (z.B. für zusätzliche LEDs).
- Anschlüsse: Eingang für RC-Empfänger, I2C-Bus für Motor-Regler, ISP-Stecker, universeller Erweiterungsstecker (Debugging, GPS,...).
- Sonstiges: unbenutzte Portpins auf Lötpads geführt (für eigene Erweiterungen).
- Abmessungen: ca. 50 x 50mm.
- Lochabstand: 45mm (63mm in der Diagonale).
- Gewicht (bestückt): 23g.
- weitere Komponenten extern anschließbar (z.B. GPS, Datenlogger,...).
- Umbau FlightCtrl auf v1.1 möglich. (Eine Beschreibung des Umbau findet man hier: Umbau).

INFO: Der LuftdruckSensor ist optional und für die Grundfunktion nicht erforderlich.

1.2 Schaltplan

2 FlightCtrl V1.1/V1.2

Die FlightCtrl V1.1 gab es nur als unbestückte Platine und musste selber bestückt werden. Diese FlightCtrl wird nicht mehr vertrieben.

Die FlightCtrl V1.2 gab es als bestückte Platine. Diese FlightCtrl wird nicht mehr vertrieben.

(Beispielbild FC_V1.1)

2.1 Technische Daten Version V1.1/V1.2

- Controller: AVR Atmel ATMEGA644**P** @20MHz.
- Sensoren: 3x GyroScope **(ENBC-03JA,ENBC-03JB oder ENBC-03JR)**, 3-Achs-Beschleunigungsssensor (LIS3L02AS4), Luftdrucksensor (MPX4115).
- Statusanzeige: 2 LEDs (rot,grün), Elektromagnetischer Summer (z.B. für Unterspannung, Ortung, Funkausfall,...).
- Ausgänge: 2 Transistorausgänge (z.B. für zusätzliche LEDs).
- Anschlüsse: Eingang für RC-Empfänger, I2C-Bus für Motor-Regler, ISP-Stecker, universeller Erweiterungsstecker (Debugging, GPS,...).
- Sonstiges: unbenutzte Portpins auf Lötpads geführt (für eigene Erweiterungen).
- Abmessungen: ca. 50 x 50mm.
- Lochabstand: 45mm (63mm in der Diagonale).
- Gewicht (bestückt): 23g.
- weitere Komponenten extern anschließbar (z.B. GPS, Datenlogger,...).
- Umbau FlightCtrl auf v1.1 möglich. (Eine Beschreibung des Umbau findet man hier: Umbau).

2.2 Schaltplan V1.1

2.3 Schaltplan V1.2

Stückliste der FlightCtrl V1.2: Download

3 FlightCtrl V1.3

Die FlightCtrl V1.3 gab es als bestückte und unbestückte Platine. Diese FlightCtrl wird nicht mehr vertrieben.

(Beispielbild FC_V1.3)

3.1 Technische Daten Version V1.3

(Es werden nur Unterschiede zur FC V1.2 aufgeführt)

- Automatischer Gyroabgleich, d.h. kein manuelles Anpassen von Widerständen mehr erforderlich.
- Einseitige Bestückung - es ist jetzt möglich, die FlightCtrl wesentlich flacher aufzubauen.
- Externe Anschlussflächen (Versorgung, I2C usw.) wurden vergrößert.
- Optional: zweiter Spannungsregler 7805 für Servoversorgung bestückbar (z.B. für Kamera-Tilt-Servo, Ausrichtung wie der erste 7805 mit Kühlfläche nach außen, nicht im Lieferumfang).
- Servo und ext. Beleuchtung über Stiftleiste anschließbar.
- Neuer (kompatibler) ACC-Sensor.
- Jetzt 4-lagiges Design.
- ⚠ Die Transistoren (PDTC143) an den Schaltausgängen SV2 der Version 1.3 können nur noch 100mA schalten! Das entspricht beispielsweise einem halben LED-Streifen (3 Stück der 2x3 LED).

3.2 Schaltplan V1.3

Stückliste der FlightCtrl V1.2: Download

4 Unterschiede zwischen FC 1.x und 2.x ME

- Die ME-Variante verwendet hochwertigere Gyroskope. Sie haben keine Temperaturdrift mehr, so dass man den MK bspw. nicht mehr nach der "Abkühlphase" erneut kalibrieren muss.
  Ebenso berichten Anwender im Forum, dass die Gyroskope der ersten Variante in einigen wenigen Fällen empfindlich auf Wechsel zw.
  Schatten und Sonnenlicht reagieren und damit das Fliegen instabiler machen.

- Die FlightCtrl ME kann fünf Servos steuern, die FlightCtrl 1.x nur einen.
- Ein paar Meinungen dazu finden sich auch in diesem Forumsbeitrag:
  http://forum.mikrokopter.de/topic-post177790.html

- Info ab Firmware V0.76: Die Servos werden erst nach dem Kalibrieren des Mikrokopter angesteuert!
  Ausnahme: Wenn keine BL-Ctrl gefunden wurden, werden die Servos sofort aktiviert (z.B. für Stand-alone Anwendungen der FC)

5 Hilfen zum Aufbau, Einstellungen und Inbetriebnahme

Hilfen zur FlightCtrl V1.x können hier eingesehen werden: FlightCtrl V1.x

6 Allgemeine Infos zur FlightCtrl

6.1 Aufgaben der FlightCtrl

- Messen der Drehgeschwindigkeiten der drei Achsen
- Messen von Beschleunigungswerten der drei Achsen
- Messen des Luftdrucks für die Höhenstabilisierung (Optional)
- Messen der Batteriespannung mit Unterspannungserkennung
- Auswerten des Fernbedienungssignals (RC-Signal)
- Verarbeiten der Sensordaten und Berechnung der aktuellen Fluglage
- Ansteuern der BL-Regler zur Motoransteuerung

6.2 Auswahl der Bauelemente

- Mikrocontroller: Die Kriterien bei der Auswahl des Controllers waren:
    - ausreichende Performance
    - gute Verfügbarkeit
    - geringer Preis
    - gut zu löten
    - kostenlose Entwicklungssoftware verfügbar

Dies alles trifft auf den ATMega 644 zu. Deshalb fiel die Wahl auf einen ATMega 644(P).

6.3 GyroScope

Die GyroScope (Gyros) messen die Winkelgeschwindigkeit (Drehgeschwindigkeit) um jeweils eine Achse. Es werden drei dieser Sensoren benötigt, um alle drei Achsen zu stabilisieren.
Diese Sensoren sind die elementarsten Bauelemente. (--> GyroScope)
Die Gyros arbeiten intern mit einer Frequenz. Gleicher Index = gleiche Frequenz.
Damit Gyros auf engem Raum keine "Schwingungen" durch Frequenzmischung erzeugen, kann man Gyros unterschiedlicher Frequenzen verwenden, z.B. Typ A + Typ B.
Ideal wären 1x Typ A, 1x Typ B, 1x Typ C. Der A-Typ arbeitet mit 22kHz und der B-Typ mit 24kHz.

Beim MikroKopter ist designbedingt keine Beeinflussung nachweisbar. Daher ist es nicht nötig unterschiedliche Gyros zu verwenden.

6.3.1 Hinweise Gyros

- Bezüglich der GyroScope gibt es verschiedene Bauformen und Versionen. Von den bedrahteten Gyros gibt es den **ENBC-03JA** und den **ENBC-03JB**.
  Der ENBC-03JR ist in **SMD**-Bauform. Elektrisch unterscheiden sie sich nicht voneinander.
  Allerdings ist die innere Taktfrequenz der Gyros leicht unterschiedlich.
  Aufgrund des großen mechanischen Abstandes zueinander sind unterschiedliche Gyrosfrequenzen nicht unbedingt erforderlich - stören aber auch nicht.

6.4 Beschleunigungs-Sensor (ACC-Sensor)

Die Hauptfunktion der Beschleunigungssensoren ist, die aktuelle Neigung des MikroKopters zu messen und die Höhenregelung zu unterstützen. Hier wird ein Drei-Achsen-Sensor verwendet. Theoretisch kann auf diese Sensoren verzichtet werden, wenn der Quadrokopter im sog. Heading-Hold-Modus (TODO: Link) betrieben

werden soll. Weitere Informationen: BeschleunigungsSensor

6.5 Luftdrucksensor

Er dient zur Stabilisierung der Flughöhe. Dieser Sensor ist optional. Die Drucköffnungen können mit Klebeband abgeklebt werden, in das mit einer kleinen Nadel ein kleines Loch gestoßen wird. Das schützt vor Wind und Licht. (--> LuftdruckSensor)

6.6 Signalisierung

- Zur akustischen Anzeige wird ein Mikrolautsprecher verwendet. Dieser enthält einen Magneten. Sollte ein Kompass integriert werden, muss der Summer möglichst weit entfernt am Rahmen angebracht werden. Sonst beeinflusst dieser die Messung des Kompasses.

7 Schnittstellen an der FlightCtrl

7.1 PPM-Anschluss

Hier wird der Empfänger angeschlossen. Über zwei Leitungen wird der Empfänger versorgt und über die dritte liefert er das RC-Summensignal zurück.
Im Gegensatz zu einem normalen Servo-PPM-Signal sind in dem Summensignal alle von der Fernbedienung gesendeten Kanäle enthalten.
In jedem Empfänger ist dieses Signal vorhanden, allerdings liefern nur wenige dieses Signal zum direkten Abgriff an einen Stecker (z.B. der RX3 Multi von ACT). (--> RC-Empfänger)

7.2 I2C-Bus

An diesem Bus werden die BL-Ctrl Regler angeschlossen, über den sie die Steuerbefehle erhalten.
Die Flight-Ctrl erfordert unseren speziellen Brushless-Motor-Regler, damit eine schnelle Kommunikation per I2C-Bus möglich ist.
Standard-Motor-Regler können **nicht** verwendet werden, weil sie nicht schnell genug angesteuert werden können.
Der I2C-Bus verfügt über eine Taktleitung (SCL) und eine Datenleitung (SDA). Im Bus werden alle SCL-Leitungen und alle SDA-Leitungen miteinander verschaltet.

7.3 Serielle Schnittstelle (SIO)

Hier wird zum Testen und Parametrieren z.B. ein PC angeschlossen. Der Pegel ist TTL-Pegel(0/+5V) und nicht V24(-10/+10V).
Aus diesem Grund muss ein Schnittstellenkonverter angeschlossen werden, falls mit der Seriellen Schnittstelle des PCs kommuniziert werden soll.
Später kann diese Schnittstelle auch zur Kommunikation (asynchron) mit anderen Controllern verwendet werden. Dies geschieht normalerweise über die SerCon.

7.4 ISP-Schnittstelle (synchron)

Der ATMEL-Controller wird darüber mittels eines ISP-Interfaces programmiert. Später kann diese Schnittstelle auch zur schnellen Kommunikation (synchron Seriell) mit anderen Controllern verwendet werden. Hierzu ist die SerCon hilfreich. Achtet bitte auf die von der ATMEL Belegung abweichende Pin Belegung beim Anschluss eines ISP Programmierers.

8 Aufbau

Der Aufbau ist abhängig, ob die Platine bestückt oder unbestückt ist. Bei der bestückten Version müssen nur wenige Komponenten aufgelötet werden.
Die Software ist bereits aufgespielt. Weitere Informationen sind hier zu finden: FCAufbauBestueckt.

Der Aufbau der unbestückten Platine dauert länger und ist fehleranfälliger, allerdings auch kostengünstiger. Und man lernt die Platine, den Aufbau und die Funktion besser kennen.
Es lohnt sich also für die technisch Interessierten. Der genaue Aufbau wird hier beschrieben: FCAufbauUnbestueckt.

9 Verkabelung

Zum Thema Verkabelung gibt es eine eigene Wiki-Seite:
ElektronikVerkabelung

10 Software

Die Software ist für alle Versionen geschrieben. Die Hardwareversionen werden anhand der Polarität der roten LED (LED2) erkannt. Bei der Version 1.0 ist die Anode der roten LED direkt am AVR angeschlossen, bei der Version 1.1 hingegen die Kathode.
Informationen zur Software findet man hier: Software

11 Softwareentwicklung

Das Downloaden und Compilieren eines im SVN abgelegten Quelltextes wird im Folgenden beschrieben.

11.1 unter Windows

Zuerst wird das komplette Projekt incl. aller Dateien auf deinen lokalen Computer übertragen, um es selber kompilieren zu können. Es hat sich folgende Vorgehensweise als vorteilhaft erwiesen:

1. Download und Installation von WinAVR (LINK) (Alle neueren Versionen als WinAVR-20060421 zeigen Performanceverluste). Siehe auch Forum

2. Download und Installation von einem Subversion-Client und Einrichten des Zugriffs auf den Source-Code. Das alles steht hier: MikroKopterRepository

Weiter gehts mit dem Kompilieren, was jetzt nur noch ein paar Klicks sind. Zum Kompilieren startet man "Programmers Notepad", welches mit "WinAVR" mit installiert wurde. Im "Programmers Notepad" öffnen wir unter "File" -> "Open Project(s)..." die jeweilige Projektdatei der Quellen, welche kompiliert werden sollen.

Angenommen, es sollen Jokos Sourcen kompiliert werden, dann wählt man

```
C:\MKSVN\FlightCtrl\branches\Flight-Ctrl_V0_xx_GPS_Joko\Quellen_Flight-Ctrl_V0_
```

aus.

Nun muss im "Programmers Notepad" nur noch unter "Tools" zuerst der Punkt "Make Clean" und dann "Make All" ausgewählt werden. Nachdem der Compiler fertig ist und im "Output" Fenster (unten) ein

```
Errors: none¶
kore-------- end --------¶
```

zeigt, steht das neue Hexfile zur Verfügung. Die erzeugten Hexfiles werden üblicherweise im Stammverzeichnis der jeweiligen Sourcedateien erzeugt, bei manchen Sourcen extra Verzeichnisse wie z.B. "Hex-Files" vorhanden sind. Wer nicht sicher ist, prüft das Datum der Files. Ein "Make Clean" räumt das Sourcen-Verzeichnis auf, in dem es sämtliche nicht benötigten, beim Kompilieren erstellten Temporär-Dateien löscht. Die Hex Datei bleibt dabei erhalten.

⚠ Falls die Kompilierung unter Windows 7 x64 aufgrund von nicht gefundenen Pfaden fehlschlägt ist ein einmaliges Starten von "Programmers Notepad [WinAVR]" mit Administratorrechten notwendig.

Zum Upload des Compilats in die FC benötigt man den gepatchen AvrDude.

11.2 unter Linux

Bevor man loslegen kann, benötigt man den "avr-gcc"-Compiler. Unter Debian-basierten Betriebssystemen wie etwa Ubuntu installiert man diesen mit

```
apt-get install gcc-avr avr-libc
```

. Zum Übertragen des übersetzten elf-files über die SerCon benötigt man AvrDude. Das aktuelle AvrDude aus Debian unterstützt dies nicht, man muss erst ein eigenes AvrDude übersetzen. Wie dies funktioniert, steht hier im Wiki unter AvrDude.

Man kann den Mikrokopter-Quellcode auch unter Linux übersetzen. Ein beherztes:

```
svn co http://mikrokopter.de/mikrosvn/FlightCtrl
```

bringt auch unter Linux die Sourcen auf die Platte. Will man nur die aktuelle Version haben, reicht auch ein

```
svn co http://mikrokopter.de/mikrosvn/FlightCtrl/tags/<Versionsnummer>
```

Wobei man die Versionsnummer durch die aktuelle ersetzt (z.B. V0.71h).

(Mehr zu Subversion und dem Repository findet sich hier: MikroKopterRepository)

Der Befehl `svn co` (Subversion checkout) sorgt dafür, dass eine lokale Kopie des FlightCtrl-Quellcodes auf die Festplatte des lokalen Rechners kopiert wird.

Sobald SVN den Kopiervorgang beendet hat, wechselt man einfach in das Unterverzeichnis mit dem Quellcode, den man übersetzen möchte und führt den Befehl `make` aus. Daraufhin wird eine .elf-Datei erstellt. Diese kann man nun mittels AvrDude an den Quadcopter übertragen.

Falls es bei `make` den Übersetzungs-Fehler gibt (Ubuntu 8.10 intrepid):

```
uart.c: In function '__vector_20':¶
uart.c:133: warning: asm operand 0 probably doesn't match constraints¶
uart.c:133: error: impossible constraint in 'asm'¶
```

dann hilft es, im File /usr/avr/include/avr/wdt.h nach `#define wdt_enable(value)` zu suchen und dort nach der Zeile

```
|| defined(__AVR_ATmega644__) \
```

die folgende Zeile einzufügen:

```
|| defined(__AVR_ATmega644P__) \
```

12 Einbau

Der Pfeil auf der einen Ecke der Platine zeigt in Flugrichtung. Dementsprechend muss die Platine in den Rahmen eingebaut werden. Zur Befestigung haben sich Stehbolzen aus Kunststoff bewährt. Sie isolieren die Platine gegen den Rahmen und verhindern somit Kurzschlüsse und Störungen.

13 Akustische Signale, Fehlermeldungen (Summer, Piepen)

Der an der FC angeschlossene Summer kann vielfältige Arten von Informationen vor und während des Fluges abgeben.

Zuallererst wären da mal die Fehlersignale. Weiterhin kann auch der aktivierte Höhensensor so konfiguriert werden, dass er in bestimmten Zuständen piept. Außerdem gibt es bei eingeschaltetem Position Hold (PH) der GPS Funktion des Navi-Ctrls regel- und unregelmäßige kurze Piepsignale, die den aktuellen Zustand anzeigen.

14 Allgemeine Sicherheitshinweise:

Wir garantieren nicht für fehlerfreies Verhalten der Elektronik oder Software. Trotz sorgfältiger Erstellung und Überprüfung übernehmen wir keinerlei Garantie oder Haftung (direkter oder indirekter Art) für die Fehlerfreiheit der Software, der Hardware oder Informationen. Sie benutzen die Elektronik auf eigene Gefahr (dies gilt auch für dazugehörige PC-Programme). Weiterhin übernehmen wir keinerlei Haftung für Folgeschäden an Sachwerten oder Personen, die durch Anwendung entstehen. Es liegt in ihrer Verantwortung, einen vollständigen Systemtest durchzuführen.

Der MikroKopter ist kein Kinderspielzeug! Dafür ist er zu teuer und zu gefährlich. Nicht über Personen fliegen! Eine Modellbauversicherung ist auf jeden Fall vor dem ersten Flug abzuschließen, dies ist gesetzlich vorgeschrieben! Hausratversicherungen decken Schäden durch Flugmodelle nicht ab! Näheres dazu unter VersicherungsPflicht.

Siehe auch: Sicherheit und Checkliste

15 Nutzungsbedingungen

Es gilt für das gesamte MikroKopter-Projekt (Hardware, Software und Dokumentation), dass eine Nutzung (auch auszugsweise) nur für den privaten (nicht-kommerziellen) Gebrauch zulässig ist. Sollten direkte oder indirekte kommerzielle Absichten verfolgt werden, ist mit uns Kontakt bzgl. der Nutzungsbedingungen aufzunehmen.

16 Weiterführende Links

- FC Aufbauanleitung für unbestückte Platinen
- FC Aufbauanleitung für vorbestückte Platinen
- Versions-Historie
- Umbau 1.0 auf 1.1
- BL-Ctrl_Anleitung
- 3V_Update
- MikroKopterEinstieg
- NaviCtrl
- Umbau1.1auf1.3
- Umbau 1.2 auf ME (2.0)
