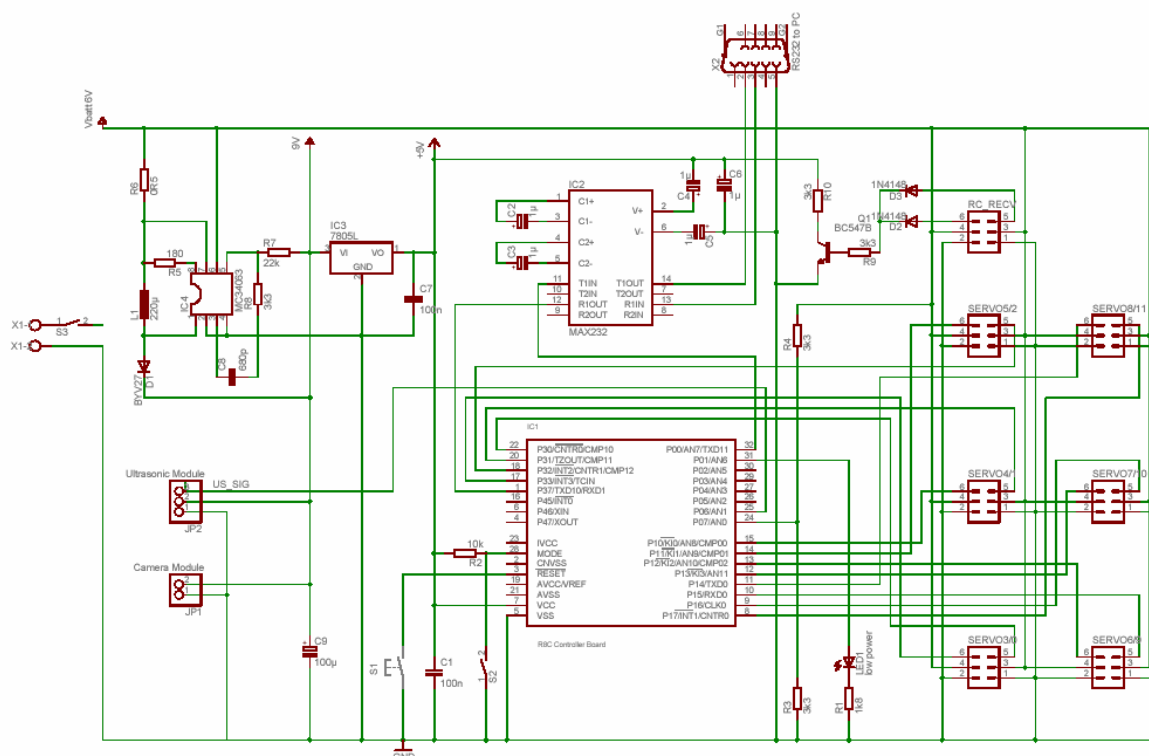




Beitrag zum Elektor R8C-Wettbewerb
von
Heiko Ott (heisa25@web.de)
und
Bernhard Schloß (deadlock@schwaben.de)

(Übersicht)

Die Schaltung wird von einem 5-Zellen-Akku (6V) versorgt, der über zwei Schraubklemmen angeschlossen wird. Die Akku-Spannung geht über den Hauptschalter direkt an die 12 Servo-Anschlüsse und den RC-Empfänger. Da 12 gleichzeitig arbeitende Servos einen Strom von 4A oder mehr ziehen können, wurde auf entsprechend starke Leitungen geachtet.



Über einen Step-Up-Regler mit dem Integrierten Schaltkreis MC34063 wird die Akkuspannung auf ca. 9V hochgeregelt, um das Kameramodul und das Ultraschall-Sensor-Modul versorgen zu können. Über einen 78L05 werden wiederum 5V zur Versorgung des R8C und des MAX232-Bausteins erzeugt. Da deren Stromverbrauch gegenüber den Servos vernachlässigbar ist, wird der schlechte Wirkungsgrad des Längsreglers in Kauf genommen.

Die Servos werden über Standard-Pfostenstecker (2x3) kontaktiert. Diese wurden auf der Platine so platziert, dass sie jeweils nahe der jeweiligen Beinposition liegen und die Servokabel direkt gesteckt werden können. Nur für zwei der hinteren Servos mussten die Kabel verlängert werden.

Zur Steuerung der Bewegung über einen RC-Empfänger werden zwei RC-Kanäle benötigt, und zwar jeweils für links und rechts, vorwärts und rückwärts, genau wie bei einem Raupenfahrzeug. Somit ist auch das Wenden auf der Stelle möglich. Um Eingangspins zu sparen, werden die Kanalausgänge des RC-Empfängers über Dioden und einen Transistor verodert, so dass nur ein Eingang belegt ist. Das geht jedoch nur, weil der Empfänger die Impulse an seinen Servo-Ausgängen nicht gleichzeitig startet, sondern die Impulse immer vollständig hintereinander ausgibt,



einen nach dem anderen. Die Software muss dann die beiden nacheinander eintreffenden Impulse trennen und der korrekten Seite zuordnen. Damit die beiden Impulse der Kanäle einen ausreichend grossen Abstand voneinander haben, dürfen nicht zwei direkt nebeneinanderliegende RC-Kanäle verwendet werden, also nicht z.B. Kanal 1 und 2, sondern z.B. Kanal 1 und 3, sonst würden die beiden Impulse als ein langer Impuls fehlinterpretiert.

Etwas ungewöhnlich ist, dass der Empfänger nicht über den sonst üblichen Batterie-Anschluss, sondern über seine Servo-Ausgänge versorgt wird. Dies ist möglich, da bei den üblichen Empfängern alle Versorgungsleitungen miteinander verbunden sind. Als Ultraschallsensor wird ein preiswerter, fertig erhältlicher Bausatz verwendet (Pollin), der über eine LED anzeigt, ob ein Gegenstand im Sensorbereich liegt. Hier wird ein Signal an der Platine angezapft und an einen Analogeingang des Mikrocontrollers gelegt.

Für die Steuerung findet das R8C-Board von Glyn (Elektor 12/2005) Verwendung, das über einen MAX232-Baustein direkt mit einem PC kommunizieren kann, also ohne spezielles Adapterkabel. Sollte ein solches Kabel schon vorhanden sein, kann natürlich auf den MAX232-Baustein verzichtet werden.

Das Schaltbild wurde mit Eagle erstellt und liegt hier als PDF vor:
aNt_sch.pdf



Software

Die Software erzeugt die Steuerimpulse für alle 12 Servos über den TimerY. Der Bewegungsablauf von aNt wird in einzelne Phasen eingeteilt, bei denen die Servos koordiniert von einer Stellung zur nächsten bewegt werden, und zwar entweder mit einer festen (autonomen Modus) oder einer variablen (Steuer-Modus über RC-Fernsteuerung) Geschwindigkeit. Dadurch können auch Servos mit unterschiedlichen Geschwindigkeiten verwendet werden.

Im Autonomen Modus läuft der Roboter zunächst geradeaus bis er vom Ultraschall-Sensor eine Kollision gemeldet bekommt. Danach kehrt er die Laufrichtung um bis die Kollisionsmeldung verschwindet und zusätzlich 2 Sekunden vergangen sind. Nun dreht er sich nach rechts. Auch während der Drehung testet er auf Kollision und dreht so lange bis keine Kollision mehr erkannt und 2 Sekunden vergangen sind. Dann setzt er seine Vorwärtsfahrt fort. Um beim Richtungswechsel schnelle Zuckungen der Beine zu vermeiden, erfolgt der Wechsel immer in der Nähe der Mittelstellung der Beine.

Empfängt der Controller Impulse vom RC-Empfänger, wechselt er in den Gesteuerten Modus. Hier setzt er die Stellung der Steuerknüppel für die linke und rechte Antriebsseite in entsprechende Beinbewegungen um. Die sechs Beine durchlaufen immer synchron denselben Bewegungsablauf, nur die Auslenkung der Servos für vor/zurück wird je nach Geschwindigkeit und Bewegungsrichtung angepasst. Die maximale Auslenkung eines Steuerknüppels bestimmt dabei die Geschwindigkeit in welcher der Bewegungszyklus abläuft.

Bei RC-Empfängern ist darauf zu achten, dass sie auch dann Impulse abgeben, wenn der Sender ausgeschaltet ist, was zu sehr ruckartigen, chaotischen Bewegungen führt. Wie beim RC-Modellbau üblich muss daher immer zuerst der Sender eingeschaltet werden. Soll der Roboter im Autonomen Modus laufen, müssen die beiden Empfänger-Leitungen ausgesteckt werden.

Da sich die Mechanik nie exakt so zusammenbauen lässt, dass die Servos ihre Mittellage immer an derselben Stelle haben, muss die Software an jedes aNt Individuum angepasst werden. Dazu kann mit einem beliebigen Terminal-Programm über Serielle Schnittstelle durch Drücken von "Enter" ein Service-Menü aufgerufen werden, das es erlaubt die einzelnen Servopositionen abzugleichen, und zwar jeweils die beiden Endstellungen. Die so ermittelten Werte können dann im Programm eingetragen werden.

Die Software ist für den GNU-Compiler GCC geschrieben, kann jedoch leicht auf andere Compiler wie z.B. HEW portiert werden. Dazu wurden schon einige Präprozessor Direktiven (`#ifdef`) vorgesehen.

Das vollständige Programmlisting und das Makefile liegen bei:

aNt.c
sfr_r813_gcc.h
Makefile
aNt.hex