

Roboter-Nachrichten 02 / 2015

Vorwort:

Liebe Leser, die sechste Ausgabe der Roboter-Nachrichten wird wieder einiges aufarbeiten, dass wir bereits in 2014 versprochen hatten, aber aus Zeitgründen nicht liefern konnten. Uns fehlt bei der Fülle der Ideen, die wir nur im Kopf entwickeln, einfach die Zeit. Beruf und Hobby – selbsterklärend, wo das der Schwerpunkt liegt, oder liegen muss!

Einer der Zeitkiller ist allerdings auch der neue 3D-Drucker, der uns viel Freude bereitet, aber uns auch immer wieder zur Verzweiflung bringt. Früher hat man einen Prototyp gebaut, in dem man etwas Holz, Messing oder Kunststoff nahm und anschließend ein neues Chassis oder Getriebe gebastelt hat. Heute läuft das anders: Man setzt sich an einen Computer und ist verwirrt von den vielen Möglichkeiten, die ein CAD Programm bietet. Dann sucht man nach einer Zeichenlösung, welche man früher z.B. mit einer Feile in zwei Minuten erledigt hätte!

Einen großen Teil dieser Ausgabe möchte wir dem Thema In-door Navigation widmen und einen möglichen Lösungsansatz zeigen. Auf der Basis von Infrarot-Baken, also mit fest installierten IR-Sendern wird mittels drehbarer Antenne eine Ortsbestimmung (terrestrische Navigation) durchgeführt.

Zu guter Letzt in diesem Vorwort: Die Ankündigung der Modellbaufreunde in Mainz ist da und wir sind fest entschlossen auch dieses Jahr an der Modellbaumesse: Inspiration Modellbau 2015 www.inspiration-modellbau.de teilzunehmen.

Die Roboternachrichten stehen auch als Download auf den Webseiten des VTH Verlages zur Verfügung www.vth.de oder auf den Webseiten von Schneider Engineerings www.ps-robotics.de.

Inhaltsangabe:

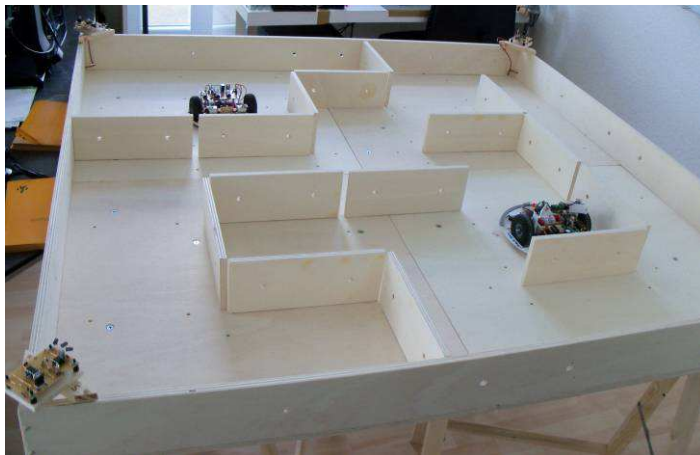
- **Projekt: In-door Navigation (ein Lösungsansatz)**
(War eigentlich schon für Ausgabe 04 / 2014 geplant)
- **Unser „neues“ Spielzeug der 3D-Ducker (Fortsetzung)**
- **Kolumne von Klaus Wellmann**

Projekt: In-door Navigation

Eines, der immer noch ungelösten Probleme, ist die Navigation in geschlossenen Räumen, in denen kein GPS Signal die genaue Positionsbestimmung ermöglicht.

Es gibt viele Ansätze, die erfolgversprechend klingen. Hier ein paar Beispiele, gemischt mit meinen Kommentaren. Auf der CeBit, die ich regelmäßig besuche, frage ich immer nach den neuesten Trends bzgl. In-door Navigation. Ein Beispiel, das ich erklärt bekommen habe, beruhte auf der genauen magnetischen Vermessung des Bodens in den Räumen, um so eine magnetische Landkarte zu erhalten. Das ist bestimmt eine Möglichkeit, aber überlegen Sie mal, wenn Sie Ihre HiFi Box oder sogar nur einen Stuhl verstellen. All das hat direkte Auswirkungen auf das Magnetfeld. Merke : Idee gut - aber ich meine, nicht umsetzbar!

Die Bundeswehr hat daran gearbeitet, dass GPS Signal auch In-door zu empfangen. Sie meinen, das geht nicht? Falsch gedacht: Das GPS Signal könnten Sie sogar in Ihrem Keller messen, falls Sie eine entsprechende Antenne mit Verstärkern haben. Wir haben das natürlich nicht! Eine weitere Idee beruhte darauf, einen RFID Sender im Boden zu verankern, sprich RFID Chips unter den Teppichboden oder das Laminat und folglich hätten wir genaue Bezugspunkte. Hört sich gut an, bis auf das Problem, seinen Lebenspartner/in von Aufreißen des Laminates zu überzeugen. Noch ein weiteres Problem mit RFID, schauen Sie sich mal die Größe der Antennen an. Genau, das sind die riesigen Dinger an den Ausgangstüren von Kaufhäusern! Verstanden, warum die In-door Navigation ein enormes Problem darstellt. Jetzt zu unserem Lösungsansatz und bitte beachten Sie, nur ein Ansatz und nichts was wirklich praktikabel wäre. Der Vorteil unserer Lösung, Sie können diese „☺ einfach ☺“ nachbauen! Unsere Idee ist nicht neu und kann auch im Internet nachgelesen werden. Wir haben das Rad also nicht neu erfunden, aber wir liefern eine all umfängliche Lösung, welche

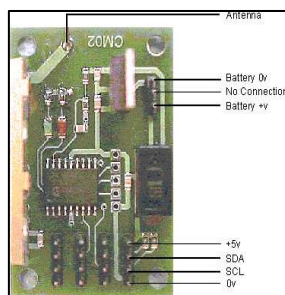


so mit ein NOVUM ist. Das System beruht auf der klassischen Navigation der Seefahrt mit festen Positionspunkten, die genau vermessen sind. In der Seefahrt sind das spezielle Landmarken, wie Leuchttürme, Kirchtürme, die Sie anpeilen können. Da wir aktive Signale benötigen, nutzen wir sogenannte IR-Baken. Das sind Sender, die ein kodiertes Infrarot

Signal aussenden. Anhand der Kodierung wissen wir welche, Bake von der Antenne erkannt worden ist. Für unsere Testumgebung nutzen wir, den zuvor schon beschriebenen Messestand. Deutlich erkennbar sind die IR-Baken an den Ecken des Testaufbaus. Die Baken nennen wir NORD, OST, SUED und WEST. Der Roboter, auf der Basis des R2PT3 Chassis wird mit einer Antenne ausgerüstet, welcher die kodierten IR Signale empfängt. Als Empfänger eignet sich ein TSOP1738 o.ä. Dieser Baustein filtert aus der Trägerfrequenz von 38KHz die IR-Baken Kodierung heraus. Der Sender, also die IR-Bake, sendet mit Hilfe eines NE555 eine Trägerfrequenz von 38KHz aus. Die Kodierung, sowie die Dekodierung, erfolgt mit den IC-

Bausteine IC14026(Codierer) und IC14027(Decodierer). Die Antenne auf dem Roboter muss 360 Grad drehbar sein und zu diesem Zweck, wird die Antenne über ein Getriebe mit einem Schrittmotor verbunden. Die Ansteuerung des Schrittmotors, als auch die Auswertung der Signale und einschließlich des Versands über den I2C Bus an die Steuerplatine des R2PT3 erfolgt, mittels eines weiteren ATmega8 Mikrocontrollers. Im Laufe der Entwicklung hat sich gezeigt, dass die Steuerplatine auf Basis eines ATmega8 nicht mehr ausreicht, um ein so mächtiges Programm aufzunehmen. Acht Kilobyte FLASH RAM ist dann doch ein bisschen wenig! Die Steuerplatine wurde neu aufgebaut auf der Basis des ATmega32 Mikrocontrollers. Auf die Ansteuerung (Programmierung) der Antenne und die neue Steuerplatine, incl. Programmierung möchten wir in diesem Newsletter nicht eingehen, das würde den Rahmen völlig sprengen. Vielmehr möchte ich die Mathematik für die Navigation erklären, auch wenn ich fairerweise sagen muss, dass Frau Christina Pagels, als erfahrene Mathematikerin die Formeln für die Berechnung zusammengestellt hat.

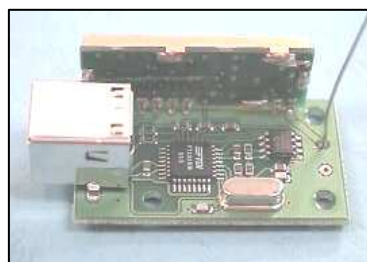
Zur Theorie: Die IR-Baken senden ein eindeutiges Signal mit einer Keule oder Kegel von 90 Grad aus. Die Sendeleistung muss so hoch sein, dass jeder Punkt der Testumgebung erreicht werden kann. Zur Ortsbestimmung reichen eigentlich zwei feste Punkte, aber es wurden immer alle IR-Baken von der Antennensteuerung ausgewertet. Die aufbereiteten Ergebnisse der Messung werden über den I2C Bus an die Steuerplatine des R2PT3 gesendet und dort



CM02

verarbeitet. Das eigentliche Signal, wo der Roboter hinfahren soll (Kurs) erhält der Roboter über ein RF-Modul „Empfänger CM02“, das über I2C Bus an die Steuerplatine angeschlossen wird. Der Sender „RF04“ wird von einem PC über die USB Schnittstelle angesteuert und mit Positionsdaten versorgt. Bitte beachten Sie; mit vier IR-Baken ist eine erste Ortsbestimmung auf dem Testaufbau nicht möglich. Es fehlt zur Dreiecksberechnung eine Seite. Noch mal zu

Veranschaulichung: Der Roboter wird einfach irgendwo auf dem Versuchsaufbau abgesetzt. Mit der Antenne kann der Roboter nur die relativen Winkel zu



RF04

den IR-Baken erfassen. Es gibt leider keine uns bekannte Lösung, mit der eine Positionsbestimmung nur über Winkel und ohne eine Seite eines Dreiecks, möglich wäre. Okay, Sie sagen beim GPS klappt das ja auch. Richtig, bei der GPS Messung werden die Laufzeiten der Signale von Satellit zum GPS-Empfänger gemessen zu der Position wo man sich gerade

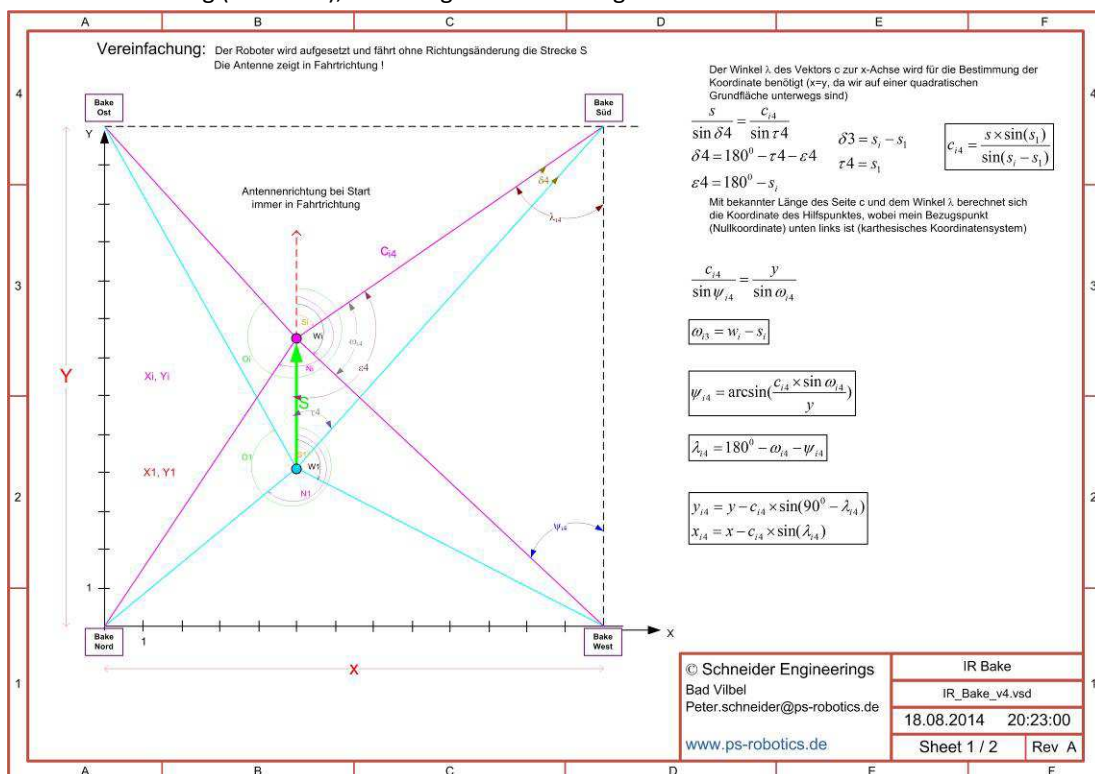
auf der Erde befindet. Zur Messung der Laufzeiten eines Signales, also wie lange das Signal vom Sender zum Empfänger benötigt, braucht man sehr, sehr genaue Uhren, die ein Mikrocontroller nun mal nicht hat. Außerdem benötigt das Signal von einer geostationären Position im Weltraum wesentlich länger zur Erde als in unserem Testaufbau mit sage und schreibe max. 130cm. Im Vergleich, die geostationäre Bahn der GPS-Satelliten befindet sich bei 36000km. Die Laufzeit des Signales wäre als noch geringer und müsste mit noch genaueren Uhren bestimmt werden.

Zurück zu unserem Versuch: Der Testaufbau wurde von uns wie ein x-y-Koordinatensystem aufgebaut mit Unterteilungen von 10cm, bei einer Gesamtgröße von 130 x 130cm. Die IR-Bake NORD ist dabei genau im Nullpunkt des Koordinatensystems. Über den PC kann man eine Koordinate z.B. 40, 70 an den Roboter senden. Das heißt, fahre von der jetzigen Position (noch über Messungen und Berechnung zu bestimmen) zu der Position X = 40cm und Y = 70cm. Hat der Roboter bereits die Info, wo er sich gerade befindet ist es wesentlich einfacher eine neue Koordinate anzufahren. Das wäre dann Teil zwei der Positions-berechnung. Die Schwierigkeit zu Anfang besteht in der ersten Ortsbestimmung. Diese sollte auch sehr exakt erfolgen da, alle weiteren Berechnungen auf die erste ermittelte Position aufbauen.

Erste Positionsbestimmung:

Ansteuern einer Koordinate in der Testumgebung (noch Roboter kennt seine erste Position nicht)

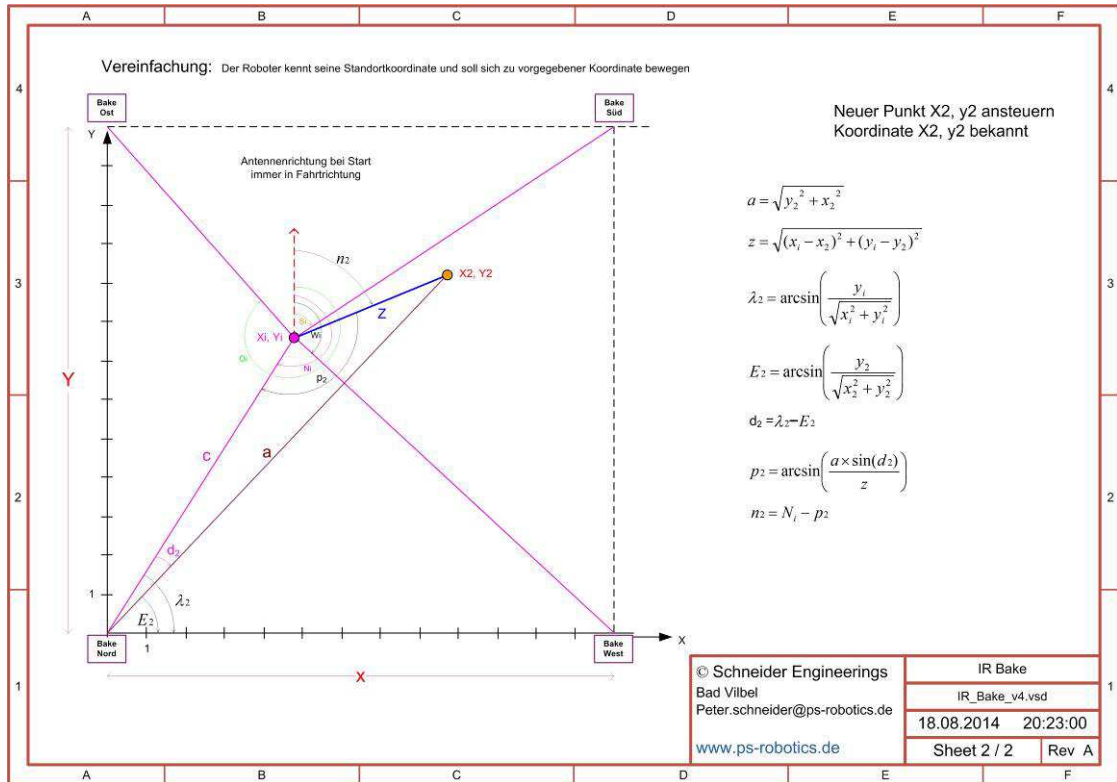
- 1) Signal über RF-Sender „Sender RF04“ von PC an den Roboter senden.
- 2) Signal wird vom RF-Modul „CM02“ auf dem Roboter empfangen und gespeichert.
- 3) Signal an die Antennensteuerung: Beginn, Start erste Messung.
- 4) Ein Rundlauf (360 Grad) der Antenne entspricht 700 Schritten des Schrittmotors, somit kann die Gradzahl pro IR-Bake ermittelt werden. Achtung, dass ist noch eine relative Gradzahl, da kein Kompass an Bord des Roboters ist.
- 5) Senden der aufbereiteten Daten, Gradzahlen pro IR-Bake über I2C Bus an die Steuerplatine des Roboters.
- 6) Speichern der Daten auf der Steuerplatine und Befehl, fahre exakt geradeaus 30cm.
- 7) Signal an die Antennensteuerung, Start zweite Winkelmessung.
- 8) Senden der zweiten Messung an die Steuerplatine des Roboters .
- 9) Speichern der zweiten Messung auf der Steuerplatine und Auswertung der Daten mittels Winkelberechnung (Sinussatz), siehe folgende Zeichnung.



Zweite Positionsberechnung:

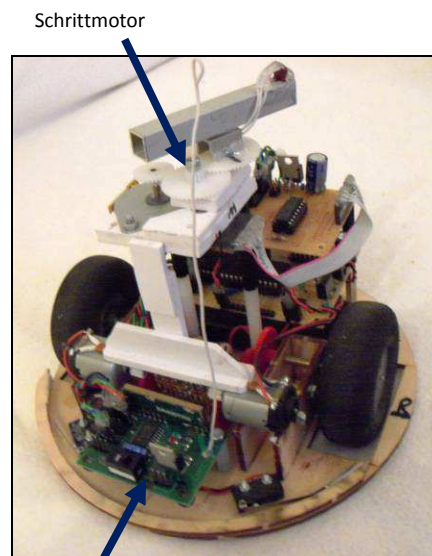
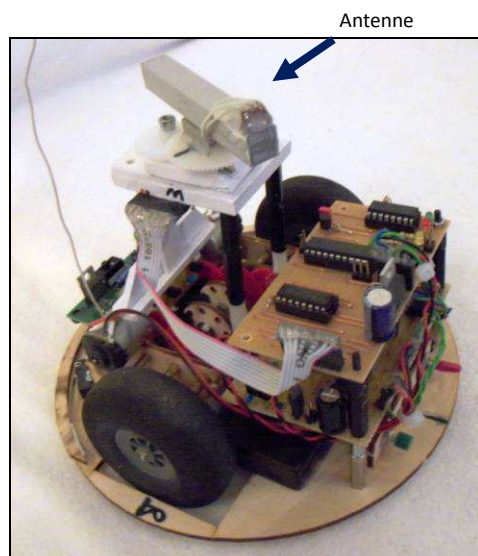
Die Position des Roboters ist nun bekannt und der Weg zur eigentlichen Zielkoordinate kann berechnet werden

- 1) Der Roboter kennt nun die eigene Position (X_i, Y_i) in der Berechnung. (Standortbestimmung)
- 2) Als nächstes wird die vom PC übertragene Position mit der ermittelten Position verglichen und ein Weg berechnet. Es wird der Drehwinkel zum Zielpunkt berechnet und die zufahrende Wegstrecke (geradeaus Fahrt), siehe folgende Zeichnung.



Hier zwei Bilder des Roboters auf Basis des Chassis R2PT3.

Erkennen Sie noch das einfache Grundmodell, wie in meinem Buch beschrieben?



RF-Empfänger Modul

Bitte beachten, je kleiner die Öffnung an der Spitze der Antenne und je länger die Antenne (Vierkantrohr) ist, umso genauer die Messung. Das Problem ist allerdings, dass die IR-Strahlung der Baken nicht so hoch ist und die Antenne ist bei der Messung in Bewegung. Das heißt, kann der IR Strahl den Sensor am Ende nicht erreichen, da sich das Rohr ja weitergedreht hat.

Ha, Ha, Ha, vollkommender Blödsinn, das Licht und IR-Strahlung ist Licht, nur in einer anderen und uns nicht sichtbaren Frequenz, ist 300000km pro Sekunde schnell.

Also daran kann es nicht liegen. Aber wenn das Rohr zu lang ist, dann lässt die Intensität, des durch die kleine Öffnung gelangenden Lichtes sehr stark nach und ist zu schwach den Sensor anzusprechen.

Beispiel: Sie nehmen die Fernbedienung Ihres Fernsehers und entfernen sich immer weiter vom Gerät. Irgendwann reicht die Kraft, bzw. die Intensität des IR Senders nicht mehr aus und der Fernseher reagiert nicht mehr. Gleiches Phänomen!

Es hat sich gezeigt, dass zwei Messungen ausreichen. Also jeweils eine Messung auf dem Hin- und Rückweg der Antenne und anschließend den Mittelwert bilden. Mehr Messungen sind natürlich besser, aber diese dauern halt lange, da die Antenne nicht sehr schnell dreht und die Übertragung der Daten über den I2C-Bus auch seine Zeit dauert.

Wenn Sie mich jetzt fragen, wie lange der Bau der Komponenten und die Programmierung gedauert haben → ...sehr, sehr lange...

Auf der Modellbaummesse konnten wir zeigen, dass es auch wirklich funktioniert. Allerdings hatten wir mit der Sonne zu kämpfen, da es sich um ein Messezelt handelte und die Infrarotstrahlung, der Sonne sich negativ auf die Messungen auswirkte.

Merke: In-door Navigation heißt In-door weil man drinnen ist, im Gebäude und nicht draußen unter einem Zelt.

Falls Sie Interesse haben an den Schaltplänen und oder Programmierung, dann bitte eine Email senden an: info@ps-robotics.de

Anstatt RF-Module könnte man auch XBee oder WLAN Module benutzen. Oder Sie basteln sich einen IR-Sender z.B. auf Basis der bereits im Text beschriebenen IC14027 und IC14026. Es würde sogar ein RS232-Kabel funktionieren. Nicht sehr elegant, wenn der Roboter ein Kabel hinter sich zieht. Wichtig ist nur, dass ein Datenaustausch zwischeneinem PC / IPAD / ANDROID oder Smartphone stattfindet.

Unser „neues“ Spielzeug, der 3D-Ducker

Wir wollen jetzt nicht weiter darüber reden, wie frustrierend das Wrapping sein kann, sondern wollen einfach ein paar Bilder von Rädern, Zahnrädern und unser erstes Getriebe zeigen, das komplett selbst gedruckt ist.

Zahnrad mit acht Löchern für die Impulsmessung mit einer Gabellichtscharanke.



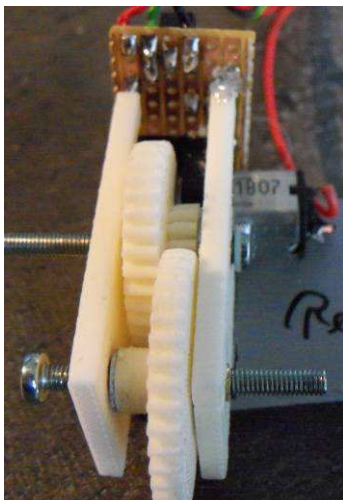
Antriebsrad für einen Roboter mit direkt aufgesetztem Zahnrad und Gummidichtung, gekauft im Baumarkt, als Reifen.



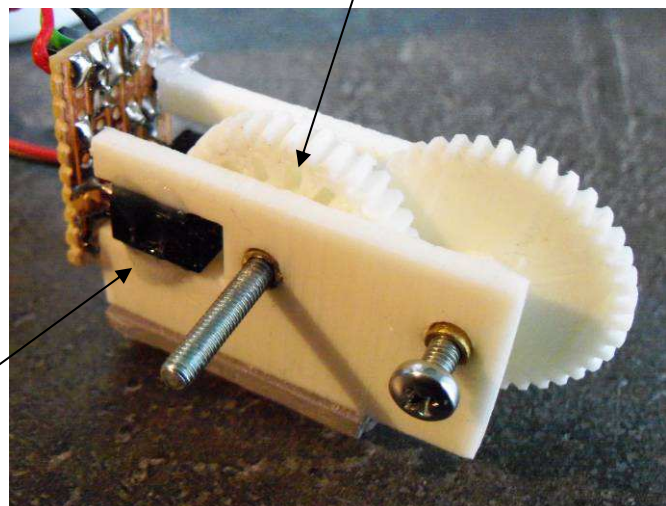
Schnecke für einen Schneckenantrieb



Unser erstes selbst gedrucktes Getriebe (Testaufbau) mit angeflanschem Impulsgeber auf Basis einer Gabellichtscharanke



Etwas schwer zu erkennen sind die Schlitze im Zahnrad für die Gabellichtscharanke.



Gabellichtscharanke

Kolumne von Klaus Wellmann

...das Blinken geht weiter....

ich weiß nicht, wie Sie an die (komplexe) Materie der C-Programmierung von Mikrocontrollern herangehen, aber ich muss einfach den Erfolg meiner Programmierung als „geglückt“, im Testaufbau sehen oder hören können! Für Einsteiger gibt es Unmengen von Literatur zum (Selbst) Erlernen der C-Programmierung. Im Grunde kann ich nur abraten, denn ohne praktischen Bezug zu einem Mikrocontroller mit seinen vielen Möglichkeiten der technischen Anwendung, finde ich diese Bücher für den Hobbybastler und Autodidakten sehr ermüdend und eigentlich wertlos....

Ich habe mir in den letzten Monaten viel Literatur aus der Bibliothek und aus dem Internet zum Thema AVR-Mikrocontroller ausgeliehen, bzw. durchstöbert und festgestellt:

Alle Autoren fangen mit dem Versuch an: **„Wir bringen eine LED zum Blinken“**

Schön, aber damit hört schon die Gemeinsamkeit wieder auf, denn jeder will sich natürlich von den Veröffentlichungen des Kollegen abgrenzen. Im Klartext: Der Anfänger ist mal wieder perfekt verwirrt und kann die verschiedenen Vorgehensweisen nicht zuordnen und wo überhaupt finde ich die „komischen“ Spezialzeichen auf der Tastatur?

Ich will einfache, aber nützliche Hinweise geben, um dann anschließend zum anspruchsvolleren „Blinken“, d.h. Vorgeben einer Frequenz high/low = oder einfach gesagt (LED an + LED aus) an einem Port überleiten.

Mein Ziel ist es für den Leser eine Zusammenstellung der Möglichkeiten (= wie eine Art Formelsammlung) darstellen und gebe deshalb bewusst keine Erklärung der einzelnen Programmschritte und überlasse das den vers. Autoren, die das so wie so besser können als ich !!!

Einschub:

Wer wissen will, wie man überhaupt ein Programm in einen Atmega8 überträgt und welche Grundschaltungen es gibt, dem empfehle ich den Kleinen BASCOM-Kurs am Beispiel des Atemga8. Internet : www.halvar.at. Oder im Buch von Peter Schneider „Der Weg zum eigenen Roboter“. Ab Seite 32.

Die gezeigten Methoden können mit der Test- und Programmierplatine, beschrieben im Buch „der Weg zum eigenen Roboter“ erhältlich beim VTH-Verlag, nachvollzogen werden.

Erste Methode: von der Schreibweise die einfachste Methode für eine LED zum blinken zu bringen:

```
#define F_CPU 8000000 // Frequenz mit der der Mikrocontroller arbeiten soll
#include <avr/io.h> // Basisinformationen zum AVR-Controller
#include <util/delay.h> // mit delay.h bindet man eine Headerfile in die
// Wartefunktion
// ein, das sind sog. Basisinformationen an den IC

int main (void) // Programmstart bzw. Hauptprogramm
{
    DDRD 0b10000000; // Datenrichtungsregister setzen (Port als Ein oder
// Ausgang)
// init-code 1000 0000 Port PD7 als Output setzen
// oder in Hex-Schreibweise optional DDRD = 0x80;
// Beginn einer Endlosschleife

    while (1)
    {
        PORTD = 0b10000000; // Bit PD7 gesetzt, somit Port auf high = LED an
        _delay_ms (200);
        PORTD = 0b00000000; // Bit gelöscht, somit Port PD7 auf low = LED aus
        _delay_ms (200); // die Zeiten sind variabel, mal andere Werte einsetzen!
    } // Ende der Schleife Sprung zum Beginn der Endlosschleife
    return (0); // Rückgabewert Null senden
}
```


Zweite Methode (2.0): schon komplizierte Schreibweise für eine LED, aber am häufigsten genutzt.
Der Vorteil ist, dass der Port als sprechender Name genannt ist und man keine Nullen mehr zählen muss.

```
#define F_CPU 8000000 // CPU-Takt
#include <avr/io.h>
#include <util/delay.h> // mit delay.h bindet man eine Headerfile in die
//Wartefunktion // ein

int main (void)
{
    DDRD |= 1<< PD7) // init-code 1000 0000 Port PD7 als Output gesetzt
// das „ | „ Zeichen ist links unten auf der Tastatur
// neben der Shift-Taste
// Beginn der Endlosschleife
    while (1)
    {
        PORTD |= (1<<PD7);
        _delay_ms (80); // Port PD7 auf high für 80 ms

        PORTD &= ~(1<<PD7);
        _delay_ms (80); // Port PD7 auf low
// alle Bits werden mit „~“ invertiert (=gedreht)
// und mit „&“ verknüpft
// das „~“ rechts oben und Strg + ALT (linke Hand
// und ~ Taste mit der rechten Hand)
    }
    return (0);
}
```

Zweite (unter) Methode (2.1): schon komplizierte Schreibweise für zwei oder mehr LEDs
Lernziel: Achten Sie auf das Setzen der Doppelklammer !!!

```
#define F_CPU 8000000 // CPU-Takt
#include <avr/io.h>
#include <util/delay.h> // mit delay.h bindet man eine Headerfile in die
//Wartefunktion // ein

int main (void)
{
    DDRD |= ((1<< PD6) | (1<<PD7)); // init-code 1100 0000 willkürlich PD6 und PD7 als Output
// das „ | „ Zeichen ist links unten auf der Tastatur
// neben der Shift-Taste
// Beginn der Endlosschleife
    while (1)
    {
        PORTD |= ((1<<PD6) | (1<<PD7));

        _delay_ms (80); // Port PD6 und PD7 auf high

        PORTD &= ~((1<<PD6) | 1<<PD7));

        _delay_ms (80); // Port PD6 und Pd7 auf low
// alle Bits werden mit „~“ invertiert (=gedreht) und mit
// „&“ verknüpft „~“ rechts oben auf der Tastatur und
// Strg + ALT (linke Hand und ~ Taste mit der rechten
// Hand)
    }
    return (0);
}
```

Dritte Methode , in Kurzform mit dem „Toggle“ - Befehl

```
#define F_CPU 8000000 // CPU-Takt
#include <avr/io.h>
#include <util/delay.h> // mit delay.h bindet man eine Headerfile in die
//Wartefunktion ein

int main (void)
{
    DDRD |= 1<< PD7; // init-code 1000 0000 Port PD7 als Output
                    // das „ | „ Zeichen ist links unten auf der Tastatur
                    // neben der Shifttaste
    while (1) // Beginn der Endlosschleife
    {
        PORTD ^= (1<<PD7); // ^ bedeutet toggle = an/aus oder high/low Potential
                           // das „ ^ „ Zeichen ist links oben auf der Tastatur
                           // neben der 1
        _delay_ms(100);
    }
    return (0);
}
```

Schlusswort:

So das war es erstmal. Die Robotnics gehen in die wohlverdiente Sommerpause und werden dann von den Erlebnissen der Modellbaummesse im Herbst berichten. Also bitte keinen Newsletter Q3 in 2015 erwarten. In der Zukunft werden wir eventuell die Themen von Lesern aufgreifen, die uns zugetragen werden.

Mit freundlichen Grüßen

Die Robotniks

(Christina, Gerhard, Klaus und Peter)

www.ps-robotics.de

Dieser Newsletter (Roboternachrichten) enthält Links zu externen Webseiten Dritter, auf deren Inhalte wir keinen Einfluss haben. Deshalb können wir für diese fremden Inhalte auch keine Gewähr übernehmen. Für die Inhalte der verlinkten Seiten ist stets der jeweilige Anbieter oder Betreiber der Seiten verantwortlich. Die verlinkten Seiten wurden zum Zeitpunkt der Verlinkung auf mögliche Rechtsverstöße überprüft. Rechtswidrige Inhalte waren zum Zeitpunkt der Verlinkung nicht erkennbar. Eine permanente inhaltliche Kontrolle der verlinkten Seiten ist jedoch ohne konkrete Anhaltspunkte einer Rechtsverletzung nicht zumutbar.