



Jugend Forscht 2015

# Kafu-Mat

Katzenfütterungsmachine



Hannah Wagner-Gillen  
Lena Hegemann

# SCHRIFTLICHE ARBEIT

## Inhaltsverzeichnis

Kurzfassung .....	1
Einleitung .....	1
Vorgehensweise.....	2
Ergebnisse .....	5
Ergebnisdiskussion .....	5
Zusammenfassung.....	5
Quellenverzeichnis.....	6
Personenverzeichnis (Unterstützer) .....	6
Anlage: Steuerungsprogramm Ka-Fu-Mat.....	7

## Kurzfassung

Ziel unserer Arbeit ist es, einen Automaten zu bauen, der eine Katze mit Trockenfutter versorgt. Er sollte das Tier nur dann füttern, wenn es das alte Futter aufgegessen hat und eine Frist von 24 Stunden vorbei ist. Dazu haben wir mit Lego Mindstorms NXT einen Roboter gebaut, den wir mit NXC programmierten. Das Programm rechnet anhand des Katzensgewichtes die optimale Futtermenge pro Tag aus und sendet dann ein Signal an den Motor, der eine Zentralschleuse dreht, die das Futter in den Fressnapf befördert.

## Einleitung

Wir haben dieses Thema gewählt, weil wir eine Methode entwickeln wollten, mit der Katzenbesitzer ihre Haustiere in den Ferien versorgen können, ohne dafür eine Pflegehilfe organisieren zu müssen. Deshalb haben wir einen Roboter mit Hilfe von Lego Mindstorms NXT gebaut.

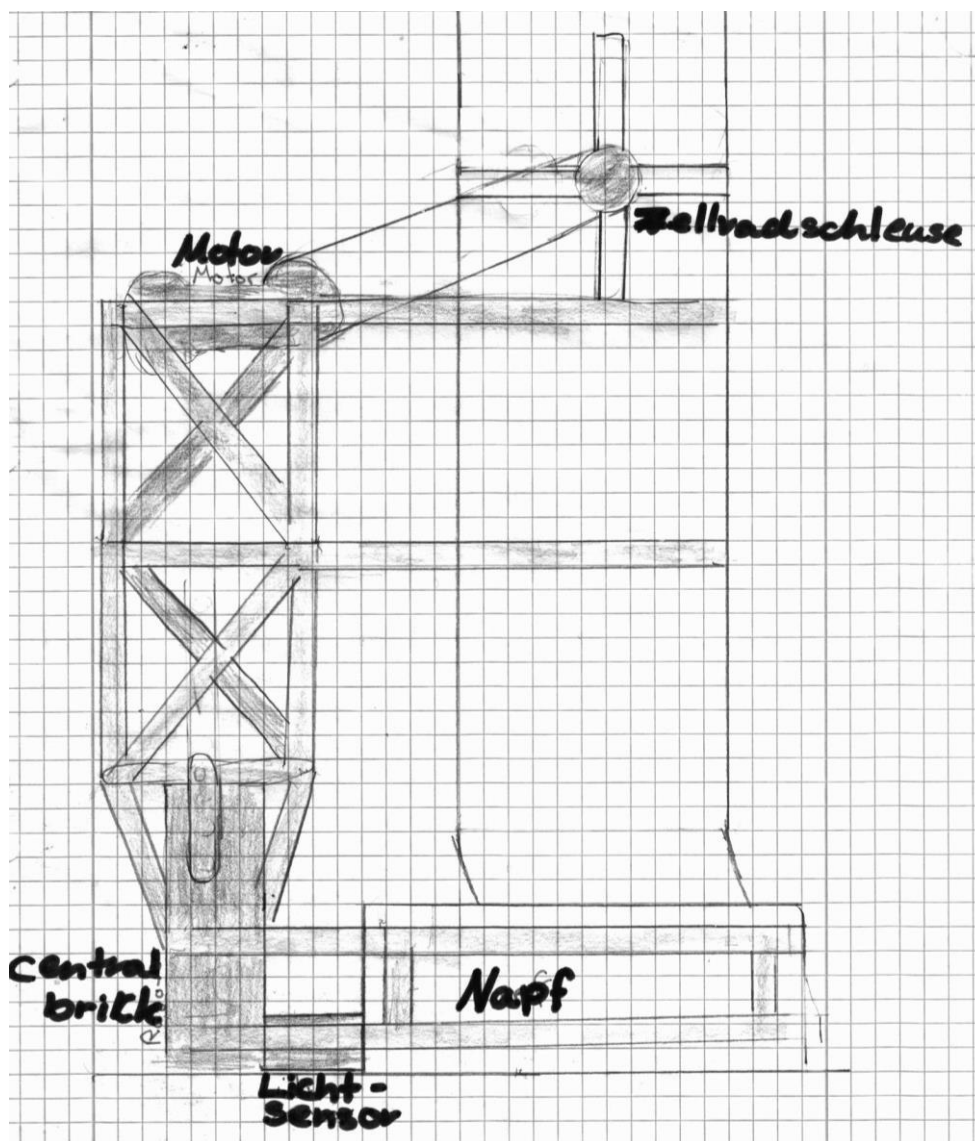
Weil sich kleine und ältere Katzen oft überfressen, haben wir in dem Automaten eine Diätfunktion vorgesehen, welche dafür sorgt, dass die Katze abhängig von ihrem Gewicht nur eine bestimmte Futtermenge pro Tag bekommt und erst neues Futter bereitgestellt wird, wenn ein Tag vorbei.

Während der Konstruktionsphase haben wir leider festgestellt, dass es bereits Futterautomaten für Katzen gibt. Da diese Automaten allerdings keine Diätfunktion anbieten, haben wir uns entschlossen, unser Projekt zu Ende zu bauen.

Ursprünglich war vorgesehen, einen Futterautomaten mit Ultraschallsensor zu konstruieren. Der Sensor sendet wenn Futter in der Nähe liegt (bzw. in großen Mengen vorhanden ist) ein Signal aus, so dass sich der Motor dann nicht dreht. Auf diese Weise kann kein neues Futter durch die vom Motor angetriebene Zellradschleuse in den Trog gelangen. Wenn das Futter entfernt wird, bewirkt das Signal, dass sich der Motor dreht und so weiteres Futter in den Trog befördern wird. Weil der Sensor unzuverlässig arbeitete, haben wir ihn im Laufe des Projektes durch einen Lichtsensor ersetzt.

## Vorgehensweise

Als erstes haben wir einen Plan gezeichnet, der das ungefähre Design und den Aufbau des Roboters festhielt.



Planungsskizze für den KaFu-Mat

Als Baumaterial benutzten wir Lego Mindstorms NXT. Zur Futterverteilung brauchten wir noch eine Zellradschleuse, welche wir allerdings nur in industriellen Größen fanden. Also bauten wir uns diese aus Holz in der passenden Größe nach.

Das Rohr, welches das Futter von dem Futterbehälter durch die Zellradschleuse in den Trog befördern sollte, erstanden wir bei einem Baumarkt. Nun bauten wir ein stabiles Gestell für den „Controller Brick“, der alle Signale an die Sensoren schickt. In unseren Bau integrierten wir ebenfalls den Ultraschallsensor, mit dessen Hilfe die Futterentfernung gemessen und so bestimmt wird, ob sich die Zellradschleuse drehen sollte; sie dreht sich, wenn der Futtertrog leer ist und ansonsten nicht.

Um die Funktionen des KAFU-Mat programmieren zu können, besuchten wir den Kurs „GET-Together“ an der Hochschule Rhein-Sieg. Dort lernten wir, wie man den Automaten mit Hilfe von NXT (graphische Benutzeroberfläche) programmieren kann. Das Programm sorgt dafür, dass der Motor nur dann Futter spendet, wenn kein altes Futter übrig ist.

Beim Testen ist uns dann aufgefallen, dass die Werte des Ultraschallsensors davon abhängig sind, wie das Futter in den Trog fällt. Deshalb haben wir stattdessen einen Lichtsensor eingebaut, welcher bewirkt, dass ein Signal an den Motor gesendet wird, wenn es kein Futter mehr gibt (bzw. wenn es hell ist). Wenn noch ausreichend Futter zur Verfügung steht, wird kein Signal gesendet. Die Diätfunktion konnten wir auch mit dem Lichtsensor realisieren (wenn die Katze zwar ihr Futter aufgegessen hat, aber der Tag noch nicht verstrichen ist, wird trotzdem kein neues Futter zur Verfügung gestellt).

Bei Tests bemerkten wir ebenfalls, dass unser Gestell nicht stabil genug war, um den Motor, der die Zellradschleuse dreht sicher zu halten. Also entwarfen wir ein neues technisches Design, welches durch Kreuzstreben stabiler wurde.

Um die Futtermenge an die Katze anzupassen, brauchten wir noch einige Angaben des Benutzers. Dazu zählten, wie lange der Ka-Fu-Mat laufen muss (wie lange die Besitzer der Katze verreisen) oder das Gewicht der Katze. Das Gewicht wurde benötigt, damit das Programm die optimale Futtermenge pro Tag berechnet (dies war eine Idee um den Funktionsumfang des Automaten zu erweitern). Da die graphische Benutzeroberfläche NXT keinen so umfassenden Programmierungsumfang besitzt, mussten wir die Programmiersprache NXC (**N**ot **E**xactly **C**) lernen, die sehr an die Programmiersprache C angelehnt ist. Damit haben wir es geschafft, das Programm so zu erstellen, dass der Benutzer mithilfe der Pfeiltasten am „Controller Brick“ die Angaben einstellen kann.

Danach rechnet das Programm die individuelle, optimale Futterbedingung für die Katze aus. Um die Formel hierzu zu entwickeln, brauchten wir weitere Informationen zum Essverhalten der Katze. Dazu besuchten wir den örtlichen Tierarzt, der uns mitteilte, dass eine Katze pro kg Körpergewicht ca. 100g Futter benötigt. Über das Gewicht der Katze ermitteln wir die Futtermenge und über die Futterdichte das Futtervolumen. Die Futterdichte haben wir mit einer Küchenwaage gemessen (430g/l, d.h. 430 kg/m<sup>3</sup>). Um zu errechnen wie oft sich die Zellschleuse drehen muss, um das Futtervolumen abzuliefern, maßen wir eine Kammer der selbst gebauten Zellschleuse aus.

Als nächstes haben wir vor, ein noch stabileres Gestell zu bauen, welches die Futterröhre und den Trog an Ort und Stelle hält, damit sie nicht verrutschen und das Futter abseits des Napfes fällt. Ein weiteres Ziel ist es, die Röhre so zu verbauen, dass das Futter eine möglichst geringe Fallhöhe hat und so nicht aus dem Napf „springen“ kann. Falls wir noch Zeit haben, würden wir das Programm verfeinern, indem wir weitere Daten abfragen und so den KaFu-Mat individueller einstellbar machen: Da kastrierte und alte Katzen einen niedrigeren Stoffwechsel als junge Katzen haben, müssen wir die Besitzer nach dem Alter und nach dem Geschlecht der Katze fragen. Auch brauchen die Tiere in den Wintermonaten mehr Futter als sonst. Deshalb wäre der Monat ebenfalls von Bedeutung. Abschließend müssen wir das ganze Verfahren an einer Katze testen und etwaige Fehler verbessern. Außerdem wollen wir eine Bedienungsanleitung entwerfen, die dem Katzenbesitzer einen Überblick der Funktionen des Automaten bietet. Ebenfalls haben wir vor, ein FMEA (**F**ailure **M**ode and **E**ffect **A**nalysis) durchzuführen und so mögliche Probleme festzustellen und zu lösen; hierüber haben wir uns bereits informiert.

Wir entwickelten und werden das Projekt noch in einem Zeitraum vom Ende August 2014 bis zum Anfang März 2015 entwickeln.

Unterstützung bekamen wir in folgenden Punkten:

- Beim Bau der Zellschleuse (Leimen der einzelnen Teile) durch einen Vater
- Beim Erlernen des Programmierens durch Kurs „GET Together“
- Informationen zum Lernen der Programmiersprache NXC aus dem Internet (siehe Quellenverzeichnis)
- Bei der Fehlersuche im KaFu-Mat-Programm durch einen Vater
- Informationen zum Essverhalten der Katze durch einen Tierarzt
- Verbinden der Zellschleuse und dem Motor (Löcher in die Röhrenwand bohren) durch einen Vater

## **Ergebnisse**

Wir haben einen KaFu-Mat mit Lego Mindstorms NXT gebaut und mit der Sprache NXC programmiert. Der Automat besteht momentan aus zwei stabilisierenden Füßen und einem seitlich fixiertem Lichtsensor. Nach oben hin ist ein Gestell mit Kreuzstreben hin ausgerichtet, das den Motor, der die Zelleradschleuse antreibt, hält. Die Maschine wird noch um einen Trockenfutter-Behälter, eine Röhre, die direkt in den Trog mündet und um ein fixierendes Gestell, um den Napf sowie um die Röhre erweitert. Der KaFu-Mat ist mithilfe einiger von dem Benutzer zur Verfügung gestellten Informationen in der Lage, eine Katze während einer Zeitperiode von 30 Tagen mit Trockenfutter zu versorgen. Dabei beachtet er folgendes: Wenn die Katze innerhalb einer Frist von 24 Stunden das Futter bereits konsumiert hat, gibt er ihr kein neues, bis die Frist beendet ist. Selber entworfen haben wir das Konzept des KaFu-Mat, den Aufbau der Maschine (den wir auch umsetzten) und das Konzept des Ultraschallsensors sowie später auch des Lichtsensors. Auch die Programmierung haben wir selber konzipiert – nur bei der Fehlersuche bekamen wir Hilfe.

## **Ergebnisdiskussion**

Der KaFu-Mat füttert die Katze mit einer auf das individuelle Körpergewicht abgestimmten optimalen Futtermenge. Er löst das Problem der Ferien-Futtermversorgung, ohne dass die Besitzer eine aufwändige Pflege organisieren müssen. Zudem sorgt der Automat dafür, dass sich die Katze nicht überfressen kann, weil sie ständig immer neues Futter bekommt. Wir sind mit der Programmierung nach anfänglichen Schwierigkeiten gut vorangekommen. Auch haben wir es geschafft ein stabiles Gestell zu entwickeln, wobei man die Fixierung des Motors noch verbessern könnte. Allerdings haben wir am Anfang des Projektes viel Zeit dabei verloren, eine Lösung mit dem Ultraschallsensor zu finden. Auch haben wir das Gestell schon dreimal gebaut, weil es entweder stabilisiert werden musste, oder weil wir noch ein anderes Bauelement hinzufügen wollten. Verbessern könnten wir noch die Programmierung, damit man sie noch besser an die jeweilige Katze anpassen kann. Ebenfalls könnte man einen zweiten Lichtsensor einbauen, sodass sich die Zelleradschleuse nur dreht, wenn die Ergebnisse beider Sensoren übereinstimmen.

## **Zusammenfassung**

Ja, wir haben unser Ziel erreicht - wir haben einen funktionstüchtigen KaFu-Mat gebaut. Wir müssen ihn allerdings noch um eine stabile Halterung für den Trog und das Rohr erweitern und auch allgemein noch etwas verbessern.

## **Quellenverzeichnis**

Zenz, Johannes: Einführung in die Programmierung des NXT Brick mit NXC, robotic.brgkepler.at, 2010/11,  
[http://www.brgkepler.at/~robotik/home/documents/BRG\\_Kepler\\_Tutorial\\_NXC.pdf](http://www.brgkepler.at/~robotik/home/documents/BRG_Kepler_Tutorial_NXC.pdf)

Benedettelli, Daniele: Programming LEGO NXT Robots using NXC, Version 2.1, 9. April 2007,  
[http://user.engineering.uiowa.edu/~cie/Labs/NXC\\_tutorial.pdf](http://user.engineering.uiowa.edu/~cie/Labs/NXC_tutorial.pdf)

Hansen, John: NXC Programmer's Guide, 18. February 2013,  
<http://bricxcc.sourceforge.net/nbc/nxcdoc/nxcapi/index.html>

## **Personenverzeichnis (Unterstützer)**

Susanne Ehrhart (Diplom-Ingenieurin, Hochschule Bonn-Rhein-Sieg)  
Art der Unterstützung: Programmierungskurs 'GET Together'

Dr. Mende (Tierarzt, Praxis in Bad Münstereifel)  
Art der Unterstützung: Angabe des Esserhaltens von Katzen

Franziska Merten (Lehrerin, Leiterin 'Jugend Forscht' am St. Michael Gymnasium)  
Art der Unterstützung: Beratung bei der Themenwahl, Hinweise zur schriftliche Ausarbeitung

Christoph Wagner-Gillen (Ingenieur bei Proctor & Gamble)  
Art der Unterstützung: Fehlersuche beim Programm des KaFu-Mat, Hilfe beim Leimen & Bohren

## Anlage: Steuerungsprogramm Ka-Fu-Mat

```
// KAFU-Mat Steuerungsprogramm
// von Lena und Hannah

//Definition von Variablen
int katzengewicht=3;
int urlaubstage=2;
int tage=0;
float jetzt=0;
float motordrehung=0;
float futtergewicht=0;
float futtervolumen=0;
float umdrehungen=0;

void benutzer_dialog()
{
  ClearScreen() ;
  TextOut(0, LCD_LINE1, "Bitte geben sie" );
  TextOut(0, LCD_LINE2, "das Gewicht der" );
  TextOut(0, LCD_LINE3, "Katze mit den" );
  TextOut(0, LCD_LINE4, "Pfeiltasten in" );
  TextOut(0, LCD_LINE5, "kg ein. Fertig? " );
  TextOut(0, LCD_LINE6, "Orange Taste." );

  while (ButtonPressed (BTNCENTER,FALSE )==FALSE )
  {
    ClearLine(LCD_LINE8 );
    TextOut (0, LCD_LINE8, "Gewicht=" );
    NumOut (60, LCD_LINE8, katzengewicht );
    TextOut (75, LCD_LINE8, "kg" );

    if (ButtonPressed (BTNLEFT, FALSE)==TRUE )//Links
    {
      katzengewicht=katzengewicht-1 ;
    }

    if (ButtonPressed (BTNRIGHT, FALSE)==TRUE )//Rechts
    {
      katzengewicht=katzengewicht+ 1 ;
    }
    Wait(350);

    if (katzengewicht>10)
```



```

{
katzengewicht= 10 ;
}

if (katzengewicht<1)
{
katzengewicht= 1 ;
}
}

Wait (500);
ClearScreen() ;
TextOut(0, LCD_LINE1, "Geben sie die" );
TextOut(0, LCD_LINE2, "Anzahl der" );
TextOut(0, LCD_LINE3, "Urlaubstage" );
TextOut(0, LCD_LINE4, "ein (mit den" );
TextOut(0, LCD_LINE5, "Pfeiltasten" );

while (ButtonPressed (BTNCENTER,FALSE )==FALSE )
{
ClearLine(LCD_LINE8 );
TextOut (0, LCD_LINE8, "Tage=" );
NumOut (48, LCD_LINE8, urlaubstage );

if (ButtonPressed (BTNLEFT, FALSE)==TRUE )//Links
{
urlaubstage=urlaubstage-1 ;
}

if (ButtonPressed (BTNRIGHT, FALSE)==TRUE )//Rechts
{
urlaubstage=urlaubstage+ 1 ;
}
Wait(250);

if (urlaubstage>30)
{
urlaubstage=30 ;
}

if (urlaubstage<1)
{
urlaubstage= 1 ;
}
}

```

```

}

void futterdetails()
{
    futtergewicht=katzengewicht*100 ;
    futtervolumen=futtergewicht/430/1000 ;
    umdrehungen=futtervolumen/0.000032 ;

    ClearScreen() ;
    Wait(500);
    TextOut(0, LCD_LINE4, "Ihre Katze" );
    TextOut(0, LCD_LINE5, "braucht" );
    NumOut(55, LCD_LINE5, futtergewicht );
    TextOut(80, LCD_LINE5, " g");
    TextOut (0, LCD_LINE7, "KA-FU-Mat" );
    TextOut (0, LCD_LINE8, "läuft jetzt!" );
}

task main()
{
    benutzer_dialog();
    futterdetails();

    while(tage<urlaubstage)
    {
        jetzt=CurrentTick();
        SetSensorLight(IN_3);

        while ((CurrentTick()-jetzt)<86400000)
        {
            if ((motordrehung<=umdrehungen)&&(Sensor(IN_3)>30))
            {
                RotateMotor(OUT_A,40,90);
                motordrehung=motordrehung+ 1;
            }
        }
        tage=tage+ 1;
        motordrehung=0;
    }
}

```