

# KALIMB0T ein Melodieroboter

Kevin Tom Robert Heine, Elektromobilität  
Otto-von-Guericke-Universität Magdeburg

**Zusammenfassung**—Bei einem 2-wöchigen Projekt, entstand der KALIMB0T. Gebaut wurde er mithilfe eines LEGO-Mindstorms- Sets, aus über 650 Teilen. Bewegung erhielt der KALIMB0T durch Matlab und ist so in der Lage einige Melodien auf der Kalimba zu spielen.

**Schlagwörter**—Kalimba, KALIMB0T, LEGO Mindstorms, LEGO NXT, Musikautomat, Roboter

## I. MOTIVATION

Die Musik ist seit vielen Jahren ein wichtiger Begleiter der Menschheit. Bereits vor 35.000 Jahren, wurden die ersten Instrumente erfunden [1]. Sogenannte Musikautomaten spiegeln hierbei den Drang der Menschen nach stetiger Weiterentwicklung wieder. Bei diesen Musikautomaten handelt es sich um selbstspielende Musikinstrumente oder Vorrichtungen und Maschinen, welche die herkömmlichen Instrumente ohne einen Spieler spielen. Die ältesten selbstspielenden Instrumente wurden schon im Mittelalter angefertigt und bereits zu Beginn des 19. Jahrhunderts existierten selbstspielende Orchester. [2] In der heutigen Zeit macht es die Technik und ihre Computer, sowie Roboter denkbar einfach eine Maschine zu entwickeln, die in der Lage ist, ein Instrument eigenständig zu spielen. Doch wie leicht ist es mit einem LEGO-Mindstorm-Set, einen Roboter zu entwickeln der ein Instrument spielen kann? Um dieses Vorhaben mit den begrenzten Möglichkeiten dieses Sets zu realisieren, musste ein einfaches Instrument gewählt werden.



Abbildung 1. Kalimba

## II. VORBETRACHTUNGEN

Die Wahl fiel auf eine Kalimba (s. Abbildung 1). Hierbei handelt es sich um ein afrikanisches Instrument mit 17 metallischen Klangzungen, die mit den Daumen angespielt werden. [3] Um den Roboter zu konstruieren muss erstmal die

Funktionsweise der Kalimba geklärt werden. Durch das Anspielen der Klangzungen mit dem Daumennagel, werden diese in Vibration versetzt, welche wiederum in dem Klangkörper in einen Ton umgewandelt werden. Dies geschieht mit einer nach unten gerichteten Wischbewegung. Die Klangzungen sind in C-Dur gestimmt, wobei die mittlere Zunge ein C ist. Die darauffolgenden Noten, sind immer abwechselnd links und rechts nach außen hin verteilt. Um die Kalimba nun mit einem Lego-NXT-Roboter spielen zu können, muss der Raum um die Klangzungen abgedeckt werden. Da es für drei NXT-Motoren zu anspruchsvoll ist, die flüssige Bewegung des Daumens beim Anspielen nachzuahmen, muss eine andere Anspielmöglichkeit gefunden werden. Die Klangzungen dürfen dabei nicht zu stark nach unten gedrückt werden, um eine Beschädigung derer zu vermeiden. Es kommen dabei zwei Möglichkeiten in Frage. Die erste wäre eine sehr dünne Holzspitze, hier würde sich zum Beispiel eine Zahnstocherspitze eignen. Diese Spitze rutscht durch eine leichte Druckausübung, an der glatten Oberfläche der Klangzungen ab und bringt diese zum Schwingen. Eine andere Möglichkeit wäre eine flexible Plastikspitze, welche sich auch wieder durch die Ausübung von Druck, so biegt, dass sie frontal abrutscht. Theoretisch könnte man nun an den 3 Motoren, jeweils eine Plastikspitze befestigen und diese vor einer Klangzunge befestigen. Jedoch würden auf diese Weise nur drei Noten anspielbar sein. Drei Noten ergeben aber noch kein Lied, um nun mehr Noten spielbar zu machen, kommen drei Roboter oder Maschinen in Frage.

### A. Stiftwalze

Hierbei handelt es sich um das Prinzip der Spieldose (s. Abbildung 2), genauer um eine Walzenspieldose. Eine mit Spitzen bestückte Walze, dreht sich entlang einer Reihe von Tonzungen. Dasselbe Prinzip könnte man bei der Kalimba anwenden, in dem man zwei solcher Walzen jeweils diagonal entlang der Klangzungenreihen anbringt. Diese Walzen würden dann von zwei Motoren zum Drehen gebracht werden. Jedoch ist diese Methode sehr umständlich, da eine Walze nur ein Lied spielen könnte.

### B. Faltarm

Inspiziert durch einen Zeichen-Roboter [5], steuern zwei Motoren zwei Arme, die durch Gelenke miteinander verbunden sind. Auf diese Weise, kann eine zweidimensionale Ebene abgefahren werden. Durch eine auf und ab Bewegung dieses Faltarmes, werden dann die Klangzungen angespielt.

### C. Schienenbasiert

Durch zwei Achsen, kann man eine 2-dimensionale Ebene waagrecht zu den Klangzungen abdecken. Eine Möglichkeit,

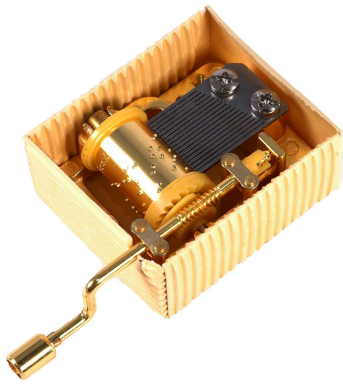


Abbildung 2. Spieldose [4]

diese Achsen zu realisieren, ist mittels Schienen. Über zwei parallele Schienen, die die x-Achse bilden, führt eine Art Brücke, welche als eine weitere Schiene fungiert und damit die Y-Achse darstellt. Die Zungen können dann mit einem Arm, an dem ein Motor befestigt ist, angespielt werden.

### III. DER ERSTE PROTOTYP

Der erste Prototyp (s. Abbildung 3) wurde nach dem Prinzip des Faltarmes konstruiert. An einem Rechteck, wird sowohl Links, als auch Rechts ein Motor befestigt. An diesen beiden Motoren, ist ein nach außen gehender Arm befestigt, welcher in einem 90° Winkel nach innen bewegt werden kann. An jedem dieser Arme, sind mit einem Gelenk weitere Arme befestigt, die am Ende wieder durch ein Gelenk miteinander verbunden sind wie in Abbildung 3 dargestellt. Die Verbindung der zwei Arme, kann so durch die Veränderung der Winkel an den Motoren, entlang der Klangzungen bewegt werden. An der Verbindungsstelle wird eine Plastikspitze befestigt, sodass die Zungen angespielt werden können. Das ganze Konstrukt wird so, als Wippe auf einen Standfuß gestellt. Durch einen dritten Motor, kann die Wippe nun auf und ab bewegt werden und die Klangzungen anspielen. Damit sie aber nicht angespielt werden, während sich der Arm bewegt, geschieht dies entweder oberhalb oder unterhalb der Klangzungen.

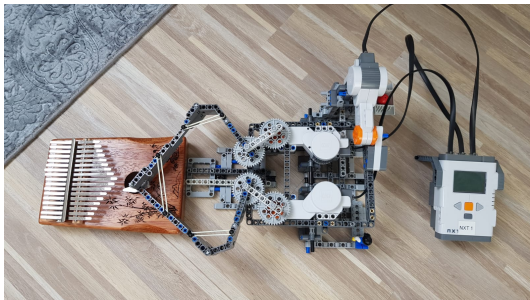


Abbildung 3. Der erste Prototyp

#### A. Probleme des Prototyps

Um verschiedene Noten anspielen zu können, muss man sehr präzise sein. Dies war auch das größte Problem des Faltarm-Prinzips. Die beiden Motoren, welche die zwei Arme bewegten,

hatten etwas Spiel beim Stillstand, sodass sie sich nach der Bewegung, noch um ein Paar weitere Millimeter bewegen konnten. Das Problem konnte durch Gummibänder, die die Gelenke zusammenziehen, minimiert werden. Jedoch war der Prototyp nach einigen Tests immer noch zu unpräzise, sodass er abgebaut werden musste und die Idee des schienenbasierten Roboter konstruiert wurde.

### IV. REALISIERUNG

#### A. Aufbau/Funktionsweise

Um die Ebene der Klangzungen abzudecken, sind zwei Achsen nötig. Die x-Achse wird durch zwei parallel zueinanderstehende Schienen realisiert. Die beiden Schienen sind durch eine Bodenfläche miteinander verbunden. Entlang der Schienen, sind Zahnstangen befestigt. Über die beiden Schienen, fährt eine Art Brücke entlang. An der einen Seite der Brücke ist ein Motor befestigt, der eine Achse dreht. Zudem ist ein Zahnrad an der Achse befestigt, welches über die Zahnstangen der Schiene dreht und damit die Brücke entlang dieser verschieben kann. Damit das Gewicht der Brücke, einfacher an den Schienen entlang verschoben werden kann, verläuft die Achse, an der der Motor befestigt ist, einmal unter der Brücke durch und ermöglicht so, dass auf beiden Schienen, je ein Zahnrad entlanglaufen kann. Die Brücke selbst fungiert ebenfalls als Schiene und stellt somit die Y-Achse da. Auf der Y-Achse fährt nun eine Art Arm entlang, der einem Kran ähnelt. Das Prinzip des Verschiebens entlang der Y-Achse, ist dasselbe wie bei der X-Achse. Entlang der Schiene ist seitlich ein Motor befestigt, welcher eine Achse mit einem Zahnrad entlang der Zahnstange dreht. Somit bewegt sich der ganze Arm entlang der Y-Achse. An dem einen Ende des Arms, ist ein weiterer Motor befestigt, welcher einen Hebel mit einer daran befestigten Plastikspitze nach oben und unten bewegen kann. Am anderen Ende des Armes, ist ein Gegengewicht angebracht. So kann nun der Bereich der Klangzungen, durch die zwei Achsen abgedeckt werden. Durch den Hebel am Arm, können die Zungen angespielt werden. Um die Probleme des ersten Prototyps zu vermeiden, war die Stabilität der Konstruktion von hoher Priorität. Durch möglichst viele Verstrebungen und eine hohe Anzahl an Teilen wird versucht mehr Stabilität zu gewährleisten.

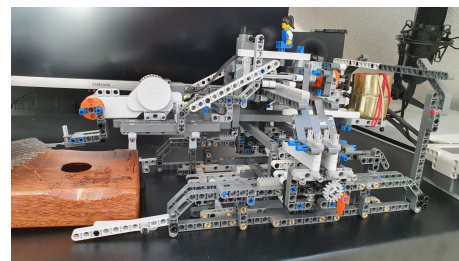


Abbildung 4. Seitenansicht des KALIMBOTs

#### B. Positionierung

Der letzte wichtige Punkt vor der Programmierung ist die Positionierung. Die Kalimba selbst wird mit den Klangzungen,

parallel zur X-Achse vor den KALIMBOT gestellt (s. Abbildung 4). Damit die Kalimba immer an dieselbe Stelle vor dem Roboter gestellt wird, existiert ein Gestell am KALIMBOT, welches die Kalimba von drei Seiten umfasst und festhält. Durch das Achsenprinzip des Roboters, ist es möglich den Bereich der Klangzungen als Koordinatensystem darzustellen. Die Position 0/0 beschreibt die äußeren Punkte der Schienen. Diese Position kann per Hand durch Fixpunkte an den Schienen exakt eingestellt werden. Dies ist wichtig, damit der Roboter beim Öffnen des Programms an dieser Position steht. Die Koordinaten der einzelnen Klangzungen, müssen nun millimetergenau abgemessen werden und dann in die Rotationszahl des Motors umgerechnet werden. Doch vorher muss der Umrechnungsfaktor bestimmt werden. Die Rotationszahl des Motors, beträgt für eine Umdrehung 360. Nun muss der Roboter eine Umdrehung ablaufen und die zurückgelegte Strecke muss millimetergenau gemessen werden. Nun kann man ausrechnen, wie groß die Rotationszahl für einen Centimeter zurückgelegte Strecke ist. Mit diesem Faktor kann man nun die Koordinaten der Klangzungen in Rotationszahlen berechnen. Die Realität zeigt jedoch, dass bei der Größendimension, die Messungen mit einem herkömmlichen Lineal zu ungenau sind. Die genauen Koordinaten müssen also mithilfe eines Testprogramms nachjustiert werden. Der KALIMBOT selbst, kann durch die Länge der Y-Achse nur 11 der 17 Töne spielen.

### C. Programm

1) *GUI des Programms:* Die GUI des KALIMBOTs ist sehr simpel gehalten (s. Abbildung 5). Durch mehrere Buttons, wird die Form der Klangzungen nachgestellt. Dabei steht jeder Button für die Note der Zunge an dieser Position. Beim Drücken eines Buttons wird die Note, die er darstellt auch angespielt. Beim anspielen einer Note färbt sich der jeweilige Button in der GUI für kurze Zeit rot, um dem Benutzer zu signalisieren, welche Note gespielt wird. Links unterhalb der Noten Button befinden sich noch zwei weitere Button. Der Button „Auf Null setzen“ fährt den Arm auf die Position 0/0 des Koordinatensystems, also den Ausgangspunkt, zurück. Und der Button „Start“ setzt die internen Koordinaten auf die Position 0/0. Links neben den beiden Button, befindet sich dann die Songauswahl, welche die vorprogrammierten Songs spielt.

2) *Programm:* Das Programm ist ebenfalls simpel gehalten, wobei jede Note ihre eigene Funktion hat, um angespielt zu werden. Diese Funktionen, werden dann in den jeweiligen Buttons benutzt um die Noten zu spielen. Um die Funktionsweise des KALIMBOTs zu erklären ist es sinnvoll, sich die Funktion einer Note anzugucken. In der Abbildung 6, ist der Programmablaufplan einer solchen Funktion dargestellt. Die Positionen der beiden Achsen, sind globale Variablen und somit in der Funktion einsehbar. Dadurch, dass mit der X und der Y Position dasselbe geschieht, kann man es nur für eine erklären. Die Funktion liest die alte Position, welche nun von der neuen Position, die von der gewünschten Note subtrahiert wird. Dabei kommt der Abstand der zwei Positionen heraus. Diesen muss der Motor nun abfahren. Dabei muss aber drauf geachtet werden, dass das Tacholimit welches man dem Motor weitergibt, positiv sein muss. Ist die Zahl des



Abbildung 5. Die GUI

Abstandes jedoch negativ, wird die Power des Motors negativ festgelegt und nicht das Tacholimit. Wenn dies nun für beide Koordinaten gemacht wurde, fährt der Arm an die gewünschte Position. Dort angekommen bewegt sich der Hebel am Ende des Arms ruckartig nach oben um die Klangzunge anzuspielen. Danach fährt der Hebel wieder nach unten. Damit die Noten aber nicht zweimal angespielt werden, fährt die X-Achse vor dem Herunterfahren des Hebels ein kleines Stück zurück und nach dem Herunterfahren wieder vor. Am Ende der Funktion werden noch die globalen Variablen der X und Y Position mit der neuen Position überschrieben. Die eben erklärte Funktion ist so für jede der Noten erstellt. Der einzige Unterschied sind die Zahlenwerte der Positionen.

3) *Ein Lied spielen:* Durch die vorgeschriebenen Funktionen für jede Note wird die Aufgabe ein Lied zu spielen stark vereinfacht. Jedoch ist es immer noch sehr kompliziert und mühsam. Das größte Problem hier, sind die richtigen zeitlichen Abstände der Noten zu treffen. Um dies zu realisieren ist es sinnvoll, die Noten für das gewünschte Lied durch die Funktionen zu schreiben. Dabei kann man nun die zeitlichen Abstände der gespielten Noten mit dem tic - toc - Befehl von Matlab zu messen. Außerdem muss man auch die zeitlichen Abstände der Noten im Lied messen. Mit diesen Werten kann man nun hochrechnen, wie man im Programm die Pausen setzen muss. Dabei schaut man sich den kürzesten Abstand der Noten im Lied an und die dazugehörige Zeit, die der Roboter braucht um von der einen Note zur anderen zu fahren. Diesen Abstand nimmt man jetzt und multipliziert ihn jeweils mit allen zeitlichen Abständen aus dem Lied. Nun hat man die zeitlichen Abstände für die Noten in der Geschwindigkeit des KALIMBOTs. Um die Pausen im Programm auszurechnen, muss man nun nur noch die Zeiten, die der Kalimbot zum Fahren zwischen den zwei Noten benötigt, von den ausgerechneten Abständen abziehen. Das Ergebnis ist nun die jeweilige Pausenzeit, die das Programm einlegen muss. Falls bei dieser Subtraktion die Pausenzeit negativ ist, müssen die Abstände neu berechnet werden.

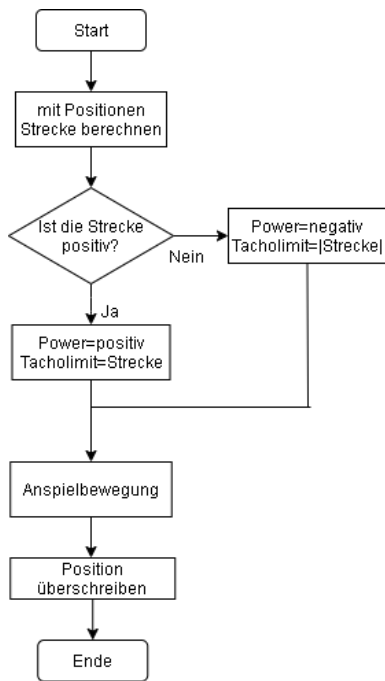


Abbildung 6. Programmablaufplan für eine Note

## V. ERGEBNISDISKUSSION

Es wurde ein Roboter konstruiert und programmiert, welcher dazu in der Lage ist eine Kalimba zu spielen. Jedoch ist die Ausführung nicht wie erwünscht. Obwohl das anspielen von einzelnen Noten perfekt funktioniert und ansehbar ist, ist das Spielen eines Liedes eher misslungen. Um die Positionen der Noten präzise zu erreichen muss dies recht langsam geschehen. Um eine zwölf Sekunden lange Melodie in den richtigen zeitlichen Abständen zu spielen braucht der KALIMBOT ganze 5 Minuten. Das größte Problem stellt dabei der Aufbau der Kalimba da. Um zum Beispiel zwei schnell aufeinander folgende Noten zu spielen Braucht der KALIMBOT manchmal lange, da sie von der Position her weit auseinander stehen. Um die Melodie nun richtig zu hören, muss man ein aufgenommenes Video in ungefähr 16-facher Geschwindigkeit abspielen. Ein Problem dabei ist, dass so der Klang der Kalimba extrem verzerrt wird. Ein weiteres Problem des KALIMBOTS ist die Genauigkeit. Um die nur wenige Millimeter von einander entfernten Klangzungen anständig anspielen zu können, ist eine hohe Präzision gefragt, welche nur schwer durch ein LEGO-Mindstorm-Set zu verwirklichen ist. Der sonstige Aufbau des Programms und der GUI ist jedoch gut gelungen und den Vorstellungen entsprechend.

## VI. FAZIT

Die Frage für dieses Projekt war, ob es möglich ist eine Kalimba mithilfe eines LEGO-Mindstorm- Sets zu spielen. Die Antwort darauf ist, dass es durchaus möglich ist wie es der funktionierende KALIMBOT zeigt. Um einzelne Noten zu spielen ist es auch ein anschauliches Projekt. Wenn man jedoch ein Lied spielen möchte, ist es durch die Trägheit des

KALIMBOTS nicht sinnvoll das LEGO-Set dazu zu benutzen eine Kalimba zu spielen.

## ANHANG

```

function playE2()
global handle MotorA MotorB MotorC posx posy
if posx==0
    posx=153
else
    posx=153-posx
end

if posy==0
    posy=60
else
    posy=60-posy
end

if posx>0
    MotorA.Power=20;
    MotorA.TachoLimit=posx;
    MotorA.ActionAtTachoLimit = 'HOLDBRAKE';
else
    MotorA.Power=-20;
    MotorA.TachoLimit=abs(posx);
    MotorA.ActionAtTachoLimit = 'HOLDBRAKE';
end

if posy>0
    MotorB.Power=20;
    MotorB.TachoLimit=posy;
    MotorB.ActionAtTachoLimit = 'HOLDBRAKE';
else
    MotorB.Power=-20;
    MotorB.TachoLimit=abs(posy);
    MotorB.ActionAtTachoLimit = 'HOLDBRAKE';
end

MotorC.Power=-20;|
MotorC.TachoLimit=50;
MotorA.SendToNXT(handle);
MotorB.SendToNXT(handle);
MotorB.WaitFor();
pause(0.5)
MotorC.SendToNXT(handle);
MotorC.WaitFor();
MotorA.Power=-10;
MotorA.TachoLimit=40;
MotorA.SendToNXT(handle);
MotorA.WaitFor();
MotorC.Power=20;
MotorC.TachoLimit=50;
MotorC.SendToNXT(handle);
MotorC.WaitFor();
MotorA.Power=10;
MotorA.TachoLimit=40;
MotorA.SendToNXT(handle);
MotorA.WaitFor();
posx=153
posy=60
    
```

Abbildung 7. Quelltext einer Notenfunktion

## LITERATURVERZEICHNIS

- [1] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Musik*. <https://de.wikipedia.org/wiki/Musik>. Version: März 2021
- [2] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Mechanischer Musikautomat*. [https://de.wikipedia.org/wiki/Mechanischer\\_Musikautomat](https://de.wikipedia.org/wiki/Mechanischer_Musikautomat). Version: März 2021
- [3] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Lamellophon (Kalimba)*. <https://de.wikipedia.org/wiki/Lamellophon>. Version: März 2021
- [4] BILD: *Spieldose*. <https://experimentis-shop.de/media/image/32/35/1e/AK01030-Spieldose-musikdose.jpg>. Version: März 2021
- [5] AVENEL, Christophe: *EV3 Print3rbot*. <https://www.youtube.com/watch?v=9pjpQoZoW6E&t=3s>. Version: März 2021