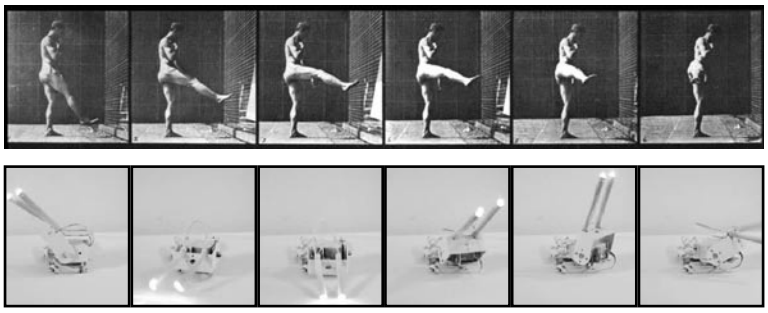
TINY TONY

EIN PROJEKT VON:
Dipl.-Ing. Jens Weber
Dipl.-Des. Andreas Wolter

BETREUUNG:
Dipl.-Ing. Jan Sieber

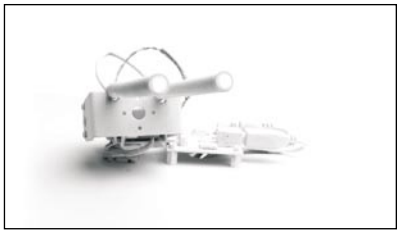
—› HTTP://WWW.MEDIAARCHITECTURE.DE/PROJEKTE/TINY-TONY



[> PROJEKTDOKUMENTATION

TINY TONY – EIN ANREGENDER AUDITIVER AUTOMAT

1 IDEE



Tiny Tony soll experimentell untersuchen, welche Gestaltmuster und mechanischen Bewegungen eines Roboters als humanoid interpretiert werden.
Der Aufbau besteht aus zwei Teilen: einer Steuerplatine und einem Servokopf, der zweiachsig schwenkbar ist. Diese Kombination wird durch ihre Anordnung und ihre Proportionen leicht als Rumpf und Kopf interpretiert. Aus dem Kopf ragen zwei Leuchtdioden an langen Federn. Ihre paarige Anordnung ist dem Gestaltmuster eines Gesichts nachempfunden. Das Leuchten der Augen zieht die Blicke auf sich und verstärkt den Eindruck des Lebendigen. Die langen Federn assoziieren insektenartige Körperteile und nehmen den Bewegungen die maschinelle Präzision.
Tiny Tony bewegt seinen »Kopf« autonom anhand der Geräusche seiner Umwelt. Ein selbstproduzierter, hörbarer Leitton bewegt den Kopf permanent suchend in der Horizontalen. Auf Lautstärkeunterschiede reagiert er mit vertikalen Bewegungen, die im Extremfall an Verbeugungen erinnern.
Besucher, die Tiny Tony ansprechen, bekommen immer ein direktes Feedback. Es ist möglich, ihn durch konstantes Pfeifen in einer Position verharren zu lassen. Aufgrund der Komplexität der Bewegungen ist das Verhalten nicht leicht nachvollziehbar und wird in Zusammenhang mit der Gestalt oft als menschlich verstanden.

2 UMSETZUNG

2.1 DIE HARDWARE

Bei der Umsetzung war zunächst zu entscheiden, wie der zweiachsig schwenkbare Kopf realisiert wird. Grundsätzlich ist die Steuerung über Schrittmotoren oder Servos möglich. Schrittmotoren erfordern den Selbstbau eines Getriebes und benötigen eine Ansteuerung, die die Position errechnet oder mechanisch erfasst. Der Vorteil besteht in der beliebig hohen Genauigkeit der Positionierung durch den Getriebe-Selbstbau.

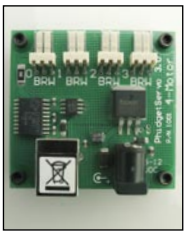
EIN PROJEKT VON:
Dipl.-Ing. Jens Weber
Dipl.-Des. Andreas Wolter

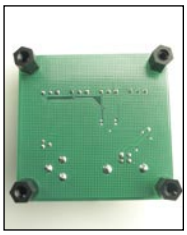
BETREUUNG:
Dipl.-Ing. Jan Sieber

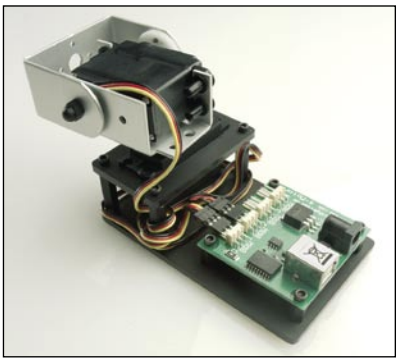
Die Wahl fiel jedoch auf Servos: Sie sind klein und enthalten Getriebe und Positionskontrolle in einem kompakten Modul. Zur Ansteuerung reicht eine einfache Pulsweitenmodulation (PWM).

Für der Rechneransteuerung ergeben sich ebenfalls zwei Möglichkeiten: Über die, gerade in der Robotik häufig benutzte, RS232-Schnittstelle, zu der viele Baupläne und Bausätze verfügbar sind oder über den USB-Port. Die USB-Port-Variante ist zwar teurer, aber langfristig die bessere Entscheidung, da Laptops zunehmend nur noch über USB-Anschlüsse verfügen und Macs gar keine RS232-Schnittstelle kennen.

Die Kombination aus Servo und USB-Ansteuerung fanden wir beim kanadischen Hersteller Phidgets (http://www.phidgets.com). Das Produkt „4-Motor PhidgetServo“ ist eine fertig bestückte Platine mit USB-Anschluss und Ausgängen für 4 Servos.
Zur Phidgets Produktfamilie gehören noch zahlreiche andere Sensoren und Aktoren wie RFID-Leser, Berührungssensoren, LCD-Panels u.a.
Alle diese Produkte sind verhältnismäßig teuer, aber in professioneller Qualität in SMD-Technik umgesetzt. Da alle Module nach dem gleichen Prinzip angesprochen werden und die Produktfamilie ständig erweitert wird, lohnt sich die Auseinandersetzung nachhaltig.
Bezogen wurde der Bausatz zum Preis von ca. 140,00 EUR über den französischen Händler Lextronik (http://www.lextronic.fr), da Phidgets (zu diesem Zeitpunkt) keinen deutschen Vertrieb hat. Zusätzlich benötigt man ein Netzteil 6-12V mit 1A, um mehr als einen Servo zu betreiben.







Nach dem erfolgreichen Testaufbau wurde Tiny Tony auf eine Grundplatte gesetzt und komplett weiss gesprüht, um die Verweise auf die technische Herkunft der Komponenten zu reduzieren und die Gesamtgestalt in den Vordergrund zu rücken. Wegen seiner Unschädlichkeit und nachträgliche Entfernbarkeit ist Kreidespray ein guter Ersatz für einen Lack.

2.2 DIE SOFTWARE

Die Ansteuerung der Phidgets-Module ist auf der Website sehr gut dokumentiert und wird kontinuierlich weiterentwickelt. Der Schwerpunkt liegt auf der Steuerung über VisualBasic.
Um willkürlichen, persönlichen Vorlieben und Abneigungen Rechnung zu tragen, sollte Tiny Tony von einen Mac über Processing gesteuert werden. Diese Schnittstelle war zu diesem Zeitpunkt jedoch nicht beschrieben.

Es existiert aber ein Mac-Framework zum Zugriff auf Phidget-Module über einen Webservice - herunterladbar auf der Seite des Herstellers.
Nach der Installation kann man per Terminal den Webservice starten:
phidgetwebservice pass -v

Der Parameter „pass“ steht für das Passwort zum Zugriff auf den Service, „-v“ schaltet den Debug-Mode ein, der sofort anzeigt, ob ein Phidget angeschlossen ist oder entfernt wurde. Standardmäßig wird der Port 5001 benutzt.

EIN PROJEKT VON:
Dipl.-Ing. Jens Weber
Dipl.-Des. Andreas Wolter

BETREUUNG:
Dipl.-Ing. Jan Sieber

Für den Zugriff über Processing müssen die ebenfalls angebotenen Java-Pakete in das Processing-"libraries"-Verzeichnis kopiert werden und können per „import Phidgets.*;" eingebunden werden.

Hier ein einfacher Quellcode zur Bewegung der Servos über die Cursortasten:

```
import Phidgets.*;
PhidgetServo phid;
float x,y;
float step;
void setup(){
   framerate(20);
   step = 1;
   phid = new PhidgetServo();
   phid.OpenRemoteIP(„localhost",5001,11033,"pass");
}
void draw(){}
void keyPressed() {
   if (phid.GetIsAttached() && key == CODED) {
     if (keyCode == UP) {
       phid.SetMotorPosition(0,phid.GetMotorPosition(0)+ step);
     } else if (keyCode == DOWN) {
        phid.SetMotorPosition(0,phid.GetMotorPosition(0)-step);
     } else if (keyCode == LEFT) {
       phid.SetMotorPosition(1,phid.GetMotorPosition(1)+ step);
     } else if (keyCode == RIGHT) {
       phid.SetMotorPosition(1,phid.GetMotorPosition(1)-step);
     }
     println(phid.GetMotorPosition(0) + „ - „ + phid.GetMotorPosi-
tion(1));
   }
}
```

Skript-Download unter:
http://mediaarchitecture.de/media/tony/processing/phidgetKeys.pde

Der Zugriff auf Phidget-Funktionen sollte immer erst erfolgen, wenn die Funktion GetIsAttached() true zurückgibt. Der Wertebereich des Übergabeparameters von SetMotorPosition() liegt bei ca. 30 bis 200.

Nach diesem Prinzip sind beide Servos positionierbar. Auf die genaue Umsetzung der weiteren Softwarekomponenten soll hier nicht näher eingegangen werden. Notwendig ist die Erzeugung des Leittones für die horizontale Bewegung und die Analyse des Mikrofon-Signals per Fast-Fourier-Transformation. Aus den gewonnenen Frequenz- und Lautstärkeparametern errechnet sich dann die Servo-Stellung.

Die vollständige Applikation inklusive Quelltext kann hier heruntergeladen werden:
http://mediaarchitecture.de/media/tony/processing/TinyTony.zip