

SmartTextiles for smart kids

# Construction K T t



# Fühlen, Kontrollieren, Interagieren



#### Ist das Licht an?

Deine Augen sehen, wenn es hell wird.
Dein Kleidungsstück kann das auch mit Lichtsensoren!
Schalter und Sensoren sind die "Sinne"
deiner intelligenten Kleidung, sie
können zum Beispiel
Helligkeit, Neigung und
Druck wahrnehmen.

Was soll passieren?

Wenn der Wecker klingelt, musst du aufstehen. Das hast du gelernt und dir gemerkt. Der Controller speichert und steuert solche Abläufe in einem Programm für deine intelligente Kleidung.



Controller

### Es bewegt sich!

Deine Hände bewegen sich, deine Füße laufen und deine Stimme macht Geräusche.

In deiner Kleidung können Motoren vibrieren, Buzzer summen und Lampen leuchten. Aktuatoren führen Aktionen wie deine Arme, Beine und Stimme aus - sie reagieren meistens auf ein bestimmtes Signal.

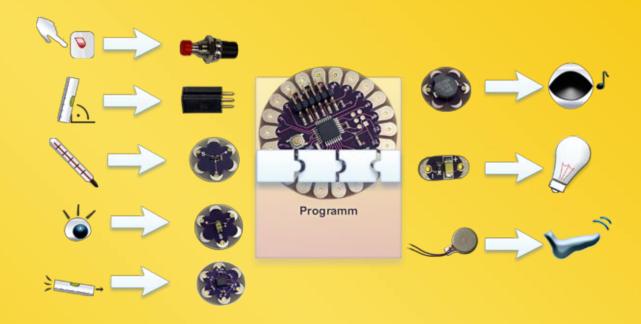
# Sensoren und Schalter, Aktuatoren, Controller



**Sensoren** sind die technische Variante der Sinne - sie können Eigenschaften der Umgebung fühlen oder wahrnehmen. Sie erkennen auch, wie viel oder wie groß eine Eigenschaft ist - der Temperatursensor zum Beispiel kann zwischen vielen Temperaturen von kalt, lauwarm, warm und heiß unterscheiden.

**Schalter** können auch Werte von außen wahrnehmen. Sie können aber nur zwei Möglichkeiten unterscheiden - An und Aus. Beim Druckschalter heißt das gedrückt oder ungedrückt, beim Neigungsschalter nach oben oder nach unten zeigend.

Die gemessenen Werte von Sensoren und Schaltern werden zum Controller geschickt.



Der **Controller** kontrolliert die Werte der Sensoren und die Aktionen der Aktuatoren. Was genau passieren soll, steht im Programm, das jemand geschrieben hat und auf dem Controller gespeichert ist. Meist soll bei einem bestimmten Wert eines Sensors ein Aktuator etwas tun.

**Aktuatoren** führen Befehle des Controllers aus - sie können summen, vibrieren oder leuchten. Durch die Aktionen des Aktuators kann man sehen, was im Programm passiert ist, zum Beispiel ob der Wert eines Sensors gelesen und daher ein Befehl für den Aktuator gegeben wurde.

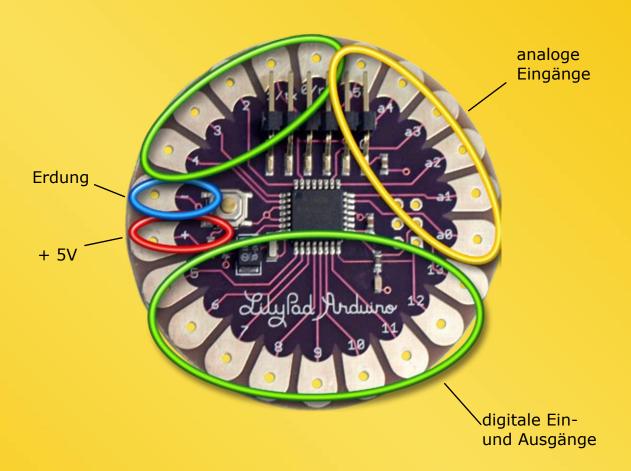
Man kann aber Aktuatoren auch unabhängig von Werten der Sensoren benutzen, zum Beispiel wenn immer eine Melodie gesummt werden soll.

# Sensoren/Schalter und Aktuatoren anschließen und programmieren



Ein LilyPad Arduino Board enthält die folgenden Anschlüsse ("Pins") für Sensoren/ Schalter und Aktuatoren:

- 14 Digitale Ein- und Ausgänge ("Digital IO") zum Anschließen von Schaltern und Aktuatoren (IO steht für "Input" und "Output" "Eingabe" und "Ausgabe")
- 6 Analoge Eingänge ("Analog in") zum Anschließen von Sensoren
- Erdung: GND oder gnd oder ("gnd" steht für "ground" "Erdung")
- +5V oder 5V



# **Digitale Pins**



- Die digitalen Pins befinden sich an den Pins 0 13. Sie können als In- oder Outputs genutzt werden, je nachdem wie es in der Software programmiert wird.
- "Digital" bedeutet, dass der Pin die Zustände "An" (1) und "Aus" (0) erkennen oder schalten kann.
- An digitale Pins können Schalter als Eingabe (Input), oder LEDs, Buzzer und kleine Motoren als Ausgabe (Output) angeschlossen werden.

Pin 0 (RX) und Pin 1 (TX)	haben auch eine besondere Funktion, weshalb es besser ist, sie nicht für Schalter oder Aktuatoren zu benutzen	
Pin 2 bis Pin 8	lassen sich wie oben beschrieben benutzen	
Pin 9 Pin 10 Pin 11	Sie funktionieren wie analoge Ausgänge, d.h. dass ein Wert zwischen 0 (keine Spannung, wie "Aus") und 255 (volle Spannung, wie "An") gewählt werden kann	
Pin 13	Der Pin hat einen eingebauten Widerstand, so dass z.B. LEDs ohne zusätzlichen Widerstand an ihn angeschlossen werden können	

## Schalter anschließen und programmieren

Zu den Schaltern zählen z.B. Neigungsschalter, textile Schalter, Reed-Schalter, Druckschalter (v.l.n.r.):

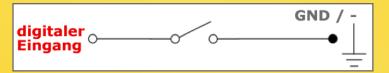






Ein Schalter muss immer an einer Seite an einen digitalen Pin und an der anderen Seite an die Erdung angeschlossen werden:





Blöcke, mit denen man Schalter prüfen kann:

Falls Sensor/ Schalter Warten auf Sensor/ Schalter Sende Sensor/ Schalter

Sensor/

Schalter

Bei den Einstellungen wählen:

- "Schalter"
- den Pin, an den der Schalter angeschlossen ist
- die Art des Schalters
- und evtl. (bei Falls und Warten auf), ob die Bedingung erfüllt ist, wenn der Schalter an oder wenn er aus ist.

Falls Rin\_2==An

Variable

# **Digitale Pins**



# Aktuatoren anschließen und programmieren

Ein Aktuator wird mit einer Seite an einen **digitalen Pin** angeschlossen, mit der anderen Seite an die **Erdung** ("-" oder "GND"):





Achte dabei auf die Polung des Aktuators:

Aktuator	Digitaler Pin	Erdung (- /GND)
LilyPad-LED	+ -Markierung	Markierung
	Langes Bein	Kurzes Bein
digitaler Pin GND / -	Zwischen LED und Pin muss : ein 220-Ohm- Widerstand oder zwei 100-Ohm Widerstände (außer bei Pin 13), damit die LED nicht durchbrennt!	zudem ( )
Vibrationsmotor	rotes Kabel	blaues Kabel
LilyPad-Buzzer	+ -Markierung	Markierung
Buzzer	rotes Kabel	schwarzes Kabel

Blöcke, mit denen ein Aktuator gesteuert werden kann:

Aktuator An

Aktuator

An für 1,000s

Aktuator Aus

Ton

Bei den Einstellungen wählen:

- Pin, an den der Aktuator angeschlossen ist
- Art des Aktuators
- ! Wenn **Pin 9, 10, oder 11** gewählt werden, muss beim Programmieren zusätzlich ein Wert von 0-255 angegeben werden, der bestimmt, mit wie viel Strom der Aktuator betrieben werden soll !

# **Analoge Pins**



- Die analogen Pins haben die Nummern 0 bis 5 (auf dem LilyPad a0 bis a5) und sind als Inputs festgelegt.
- An analoge Pins können Sensoren angeschlossen werden, um deren Werte zu messen. Der analoge Pin misst, wie viel Strom gerade zwischen 5V und Erdung fließt und wandelt dann den Volt-Messwert des Sensors in eine Zahl von 0 bis 1023 um, die du mit deinem Programm abfragen kannst.

#### Sensoren anschließen und programmieren

An analoge Pins können Sensoren wie z.B. Lichtsensoren, Temperatursensoren, textile Stretchsensoren (v.l.n.r.) u.ä. angeschlossen werden:







Ein Sensor muss immer an drei Anschlüsse angeschlossen werden:

- GND oder –
- Analoger Pin
- 5V oder +5V

(Der analoge Pin misst, wie viel Strom gerade zwischen 5V und Erdung durch fließt.)

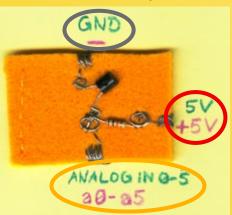
Zwischen Sensor und +5V muss ein Widerstand gesetzt werden (dieser skaliert die Sensorwerte, so dass sie in einem gut unterscheidbaren Bereich liegen).

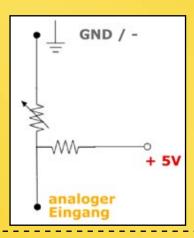
Günstig sind meist ein 1000-Ohm oder

10000-Ohm-Widerstand.



So wird ein Sensor angeschlossen:





Blöcke, mit denen Sensorwerte abgefragt werden können:

Falls Sensor/ Schalter

Warten auf Sensor/ Schalter Sende Sensor/ Schalter

Variable

Sensor/

Schalter

Bei den Einstellungen wählen:

- "Sensor"
- der Pin, an den der Sensor angeschlossen ist
- Art des Sensors
- evtl. (bei Falls und Warten auf), ob die Bedingung (<, >, = und Wert) erfüllt ist, wenn der Sensor einen bestimmten Wert (Schwellwert) über- oder unterschreitet.

# LilyPad mit dem PC verbinden und ein Programm hochladen



#### Mit dem PC verbinden

Verbinde das USB-Kabel mit dem USB-Adapter



Verbinde das USB-Kabel mit dem PC



Verbinde den Adapter mit dem LilyPad (1)



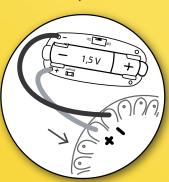
Die LED (5) auf dem LilyPad blinkt



#### **LILYPAD Arduino board**

#### **Externe Strom**versorgung

Verbinde das Batteriefach mit dem Adapterstecker des LilyPads



Schalte das Batteriefach an (10)



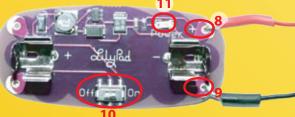
Die Strom LED (11) leuchtet



**USB-Adapter** 



- 14 Stecker
- LYPAD Batteriefach



10 - An/aus Schalter 11 - Strom LED

- 1 Stecker für Adapter

- 2 "-" Pin 3 "+ 5V" Pin 4 RESET-Knopf
- 5 Reset-LED 6 Digitale Pins Nr. 0 13 7 Analoge Pins Nr. 0 5

#### **Ein Programm** hochladen



Die blinkenden LEDs auf dem Adapter zeigen an, dass Dein Programm hochgeladen wird.



#### **Amici einstellen**

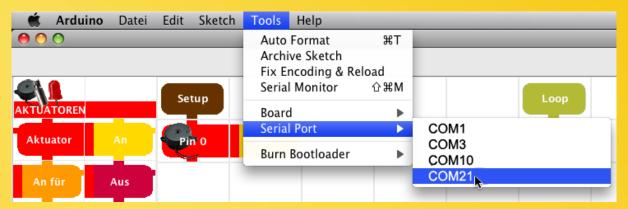


Wenn die Software Amici neu gestartet worden ist, müssen zunächst Einstellungen vorgenommen werden, damit ein Programm hochgeladen werden kann. Dafür musst du zuerst das Board mit dem Computer verbinden. Dann musst du im Menü oben folgende Einstellungen wählen:

#### **COM-Portnummer wählen**

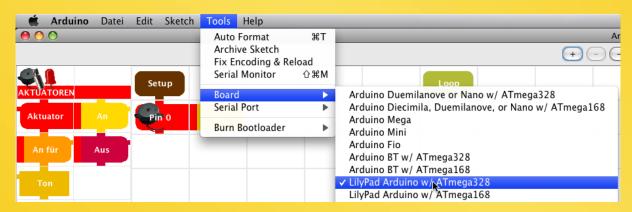
Tools -> Serial Port:

In der Regel der Port der mit der höchsten Nummer (z.B. COM21)



#### Boardtyp wählen

Tools -> Board: LilyPad Arduino w/ ATmega328



# **Programmieren mit Amici**



Der erste Block in deinem Programm muss immer der "Setup"- oder der "Loop"-Block sein.



Blöcke, die hinter dem Setup-Block stehen, werden nur einmal ausgeführt, wenn das Programm startet.



Die Blöcke, die hinter dem "Loop"-Block stehen, werden immer wieder wiederholt.

Danach kannst du beliebig viele andere Blöcke unter den Setup- oder Loop-Block reihen. Die "Haken" an den Blöcken zeigen dir, wo du sie sinnvoll ins Programm einbauen kannst.

#### Aktuatoren programmieren



Wenn du einen Aktuator programmieren willst, beginne immer mit diesem Block. Wähle aus, welche Art Aktuator an welchem digitalen Pin angeschlossen ist. Bei Pin 9-11 musst du zusätzlich festlegen, mit wie viel Strom er betrieben wird - in Zahlen von 0-255.

Hinter dem Aktuatorblock kannst du dann bestimmen, was mit dem Aktuator passieren soll.



schaltet einen Aktuator an.



schaltet einen Aktuator aus.



schaltet einen Aktuator für eine bestimmte Dauer an und danach wieder aus.

Die Zahl legt fest, wie lange der Aktuator angeschaltet wird. Hier wird er für eine Sekunde (=1000 Millisekunden) angeschaltet.



erzeugt Töne, wenn du einen Piezo-Summer an Pin 9, 10 oder 11 anschließt. (Er funktioniert dann ähnlich wie der "An"-Block.) Dieser Block kann allein unter Loop oder Setup stehen, und nicht neben dem Aktuatorblock.

#### **Beispiel**

Dieses Programm schaltet einmal bei Programmstart (Setup-Block) eine LED am digitalen Pin 2 an. Dann spielt der Piezo-Summer 3 Sekunden den Ton C. Schließlich wird die LED an Pin 2 wieder ausgeschaltet.





#### Schalter und Sensoren programmieren



Der Sensor/Schalter-Block fragt den Wert eines angeschlossenen Sensors oder Schalters ab.

Im Menu musst du einstellen, ob du einen Sensor oder Schalter programmieren willst.

Bei einem **Sensor** musst du dann einstellen, welche Art Sensor an welchen analogen Pin angeschlossen ist, und welcher Grenzwert unterschritten, genau erreicht oder überschritten sein soll, damit etwas passiert. Der Grenzwert ist eine Zahl zwischen 0 und 1023. Du kannst auch einen vorher bestimmten Variablennamen anstelle des Wertes angeben.

Bei einem **Schalter** musst du dann einstellen, welche Art Schalter an welchen digitalen Pin angeschlossen ist, und ob er an- oder ausgeschaltet sein soll, damit etwas passiert.

Wie die Sensor- und Schalterwerte genutzt werden, legst du mit anderen Blöcken fest, die du vor den Sensor/Schalter-Block tust.

Falls

prüft, ob eine Bedingung erfüllt ist, z.B. ob ein Sensor oder Schalter einen bestimmten Wert hat.

Hinter diesen Block muss immer ein Sensor- oder Schalterblock, der angibt wo der Sensor oder Schalter angeschlossen ist und bei welchem Wert etwas passieren soll.

Warten auf

wartet solange, bis ein Schalter oder Sensor den gewünschten Wert erreicht, und fährt erst dann mit dem nächsten Block fort. Hinter diesen Block muss immer ein Sensor- oder Schalterblock, der angibt wo der Sensor oder Schalter angeschlossen ist und bei welchem Wert das Warten aufhört.

Variable

speichert den Wert eines Sensors oder Schalters in einer Variable. Du musst der Variable einen Namen geben (nur ein Wort!). Später kannst du den Wert mit einem neuen Sensorwert vergleichen, z.B. mit dem Falls-Block.

Sende

zeigt den aktuellen Wert eines Sensors (oder Schalters) an. Damit ist der Block sehr nützlich um herauszufinden, was für Werte gerade gemessen werden.

Um die Werte zu sehen, musst du auf "Empfangen" oben über dem Block-Programm klicken (nachdem du dein Programm auf das Board geladen hast).

Dann werden im Fenster daneben die Werte angezeigt.



Empfangen

Um das Anzeigen der Werte wieder zu deaktivieren, klicke noch mal auf "Empfangen":





ermöglicht es, Sensor-Daten zu sammeln. Das Board schreibt die Daten in seinen Speicher (der so genannte "EEPROM"). Du kannst Daten von bis zu zwei Sensoren gleichzeitig sammeln, und musst angeben, wie oft die Daten gesammelt werden sollen. Insgesamt kannst du 512 Werte speichern. Wenn du also z.B. jede Sekunde einen Wert speicherst, dauert deine Messung 512 Sekunden, also etwa 8,5 Minuten.

Um die gesammelten Daten dann vom Board runterzuladen und anzugucken musst du oben aus dem Menü unter "Tools" auf den Eintrag "Daten hochladen" wählen:



Die gesammelten Daten werden dann in einer Excel-Datei (.xls) angezeigt. (Dazu muss MS Excel oder ein vergleichbares Programm auf deinem Computer installiert sein).

#### **Beispiel**

Dieses Programm speichert beim Programmstart (Setup-Block) den aktuellen Wert des Lichtsensors am analogen Pin a0 in der Variablen "hell". Im Loop überprüft das Programm immer wieder, ob der aktuelle Wert des Sensors an Pin a0 kleiner als der Anfangswert in Variable "hell" ist. Wenn ja, wird die LED am digitalen Pin 2 angeschaltet. Wenn der aktuelle Wert des Sensors an Pin a0 größer ist als der Anfangswert, wird die LED an Pin 2 ausgeschaltet.





#### Kontrollstrukturen verwenden



wartet eine bestimmte Zeit lang, d.h. das Programm macht eine Pause. Wie lange die Pause dauern soll, wird im Menü in Millisekunden festgelegt. Der Block eignet sich z.B. um eine blinkende LED zu programmieren.



wiederholt den Block, der rechts von "Wiederhole" steht, so oft, wie du im Menü festgelegt hast.



erhöht eine Zahl um 1, jedes Mal, wenn er aufgerufen wird. Die aktuelle Zahl kannst du im "Empfangen" Fenster lesen (wie beim Sende-Block).

Du kannst mit dem Zähler-Block z.B. auch zählen, wie oft ein Schalter gedrückt wird oder ein Sensor einen bestimmten Wert annimmt. Dazu musst du den Zähler z.B. mit einem "Falls"-Block verbinden.

#### Methoden schreiben

Wenn du mehrere aneinanderhängende Blöcke öfter verwenden willst, ist es leichter, sie in einem großen Block zusammenzufassen. So musst du nicht immer wieder alle Blöcke neu zusammensetzen.

Eine Methode ist wie ein eigenes kleines Programm und kann beliebig viele andere Blöcke enthalten. Deswegen beginnt sie, wie auch "Setup" und "Loop" mit einem runden Block, dem runden "Methode"-Block.



zeigt an, dass hier eine "Methode" beginnt, also ein Programmteil, der isoliert vom Hauptprogramm steht.

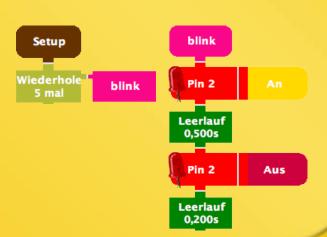


ruft solch eine "Methode" auf, die dann direkt ausgeführt wird. Erst wenn alle Blöcke in der Methode fertig sind, werden die restlichen Blöcke des Hauptprogramms durchlaufen.

Mit einer Methode kannst du zum Beispiel auch viele verschiedene Blöcke zusammenfassen und sie damit alle von einem "Wiederhole"-Block ausführen lassen.

#### **Beispiel**

Beim Start dieses Programms (Setup-Block) wird fünf mal die Methode "blink" ausgeführt. Die LED am digitalen Pin 2 wird angeschaltet. Nach einer halben Sekunde (Leerlauf 500 Millisekunden) wird sie wieder ausgeschaltet. Dann bleibt sie eine fünftel Sekunde (Leerlauf 200 Millisekunden) aus, bevor das Blinken wiederholt wird.





#### Informationen zur Benutzung des Kits

Weitere Informationen zur Programmierumgebung Amici finden Sie unter http://www.dimeb.de/eduwear.

Eigene Projekte können im Virtual Lab (http://www.techkreativ.de/virtuallab) veröffentlicht und mit anderen Nutzer(inne)n geteilt werden.

Bei sonstigen Fragen wenden Sie sich bitte an: eduwear@tzi.de

#### Aktuelle EduWear-Aktivitäten

Aktuelle EduWear- Workshops sind eingebettet in das TechKreativ-Programm der Arbeitsgruppe "Digitale Medien in der Bildung" des Technologie-Zentrum Informatik und Informationstechnik (TZI) der Universität Bremen und unter http://www.techkreativ.de einsehbar.







## Informationen zum Projekt EduWear





# **Socrates** Minerva

Das Projekt EduWear wurde von der Europäischen Union im Rahmen des Socrates (Minerva) Programms von Oktober 2006 bis September 2008 gefördert Grant Agreement Number: 2006-2062 /001-001 SO2-5-MIOD 229651-CP-1-2006-1-DE-MINERVA-M).

Es waren folgende Partner am Projekt beteiligt: The Swedish School of Textile (Boras, Schweden), Comenius University (Bratislava, Slowakei),

St Patrick's College (Dublin, Irland), X10D International IT Services Ltd. (Budapest, Ungarn), Universität Bremen (Bremen, Deutschland).















Dieses Werk steht unter einer Creative Commons Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz. Um die Lizenz anzusehen, gehen Sie bitte zu

http://creativecommons.org/licenses/by-nc-sa/3.0/de/ oder schicken Sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.