Arduino für FunkAmateure

Arduino & graphische Programmiersprachen

Workshop

- Sensoren aus ALLNET 4Arduino Set (ArdDevKIT1)
- Schaltung 1: Analoge Pins …
- Schaltung 2: Digitale Pins …
- Schaltung 3: Blinkende LED
- Schaltung 3: Blinkende LED, Mixly-Programm & Code
- Schaltung 3: Blinkende LED, Blöcke
- Schaltung 4: Taster & Serielle Schnittstelle (Monitor) …
- Schaltung 5: Taster & LED …
- Schaltung 6: LDR …
- Schaltung 7: Temperatur LM35 …
- Schaltung 8: DHT11 …
- Schaltung 9: Audio-Sensor KY-038 …
- Schaltung 10: Wasser-Sensor …
- Schaltung 11: LCD …

H39@email.de

Sensoren aus ALLNET 4Arduino Set (ArdDevKIT1)

Taster

Potentiometer 10 kΩ

LDR (Fotowiderstand)

Lageabhängiger Schalter mit Kugel

LM35, Temperatursensor im TO-92 Gehäuse,
Messbereich: -55°C bis +150°C

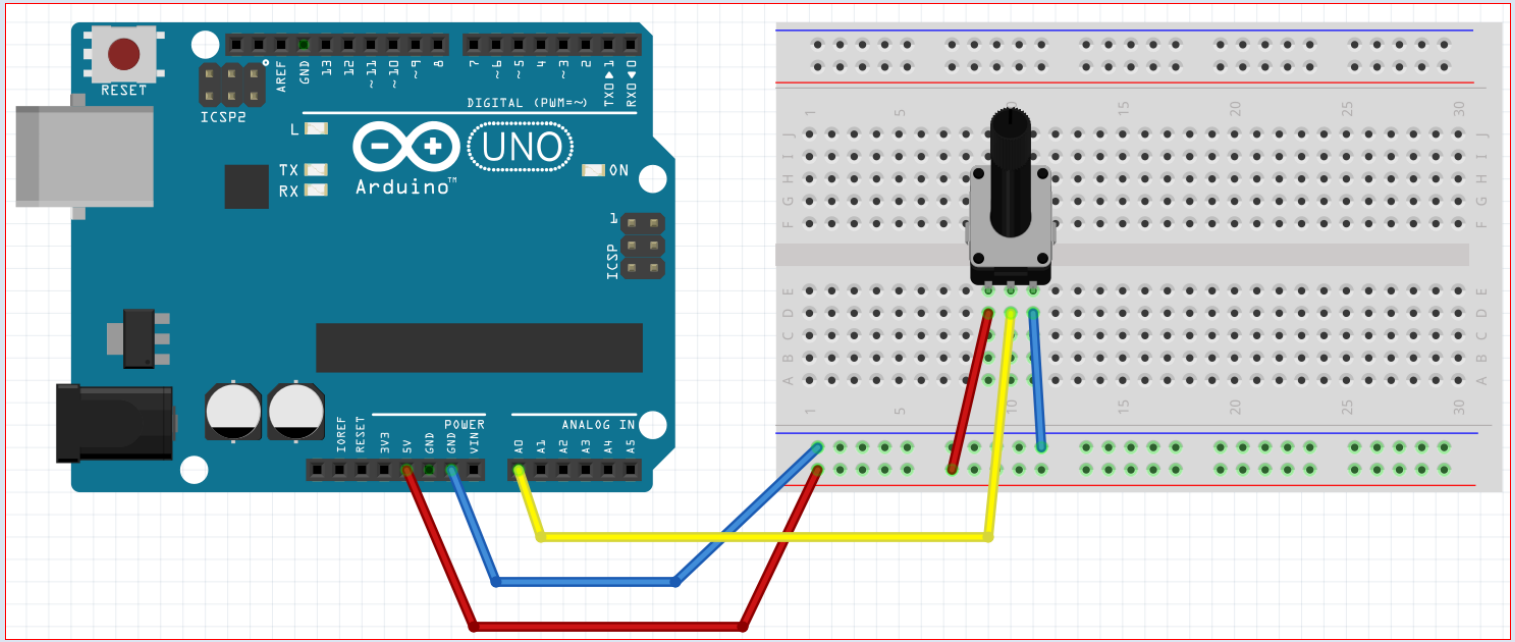
Wasser-Sensor (Platine mit Schalttransistor)

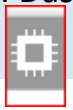
Temperatur & Luftfeuchtigkeits-Sensor Typ: DHT11

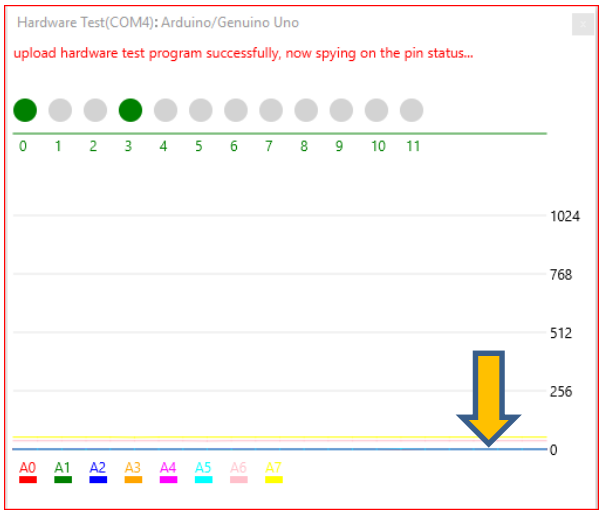
Audio-Sensor KY-038

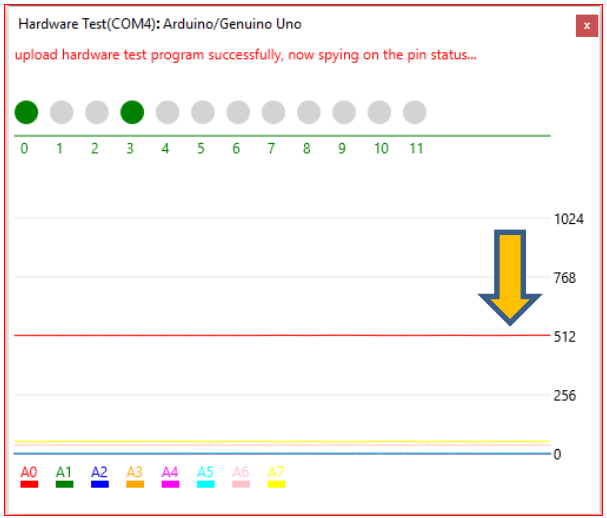
Schaltung 1: Analoge Pins

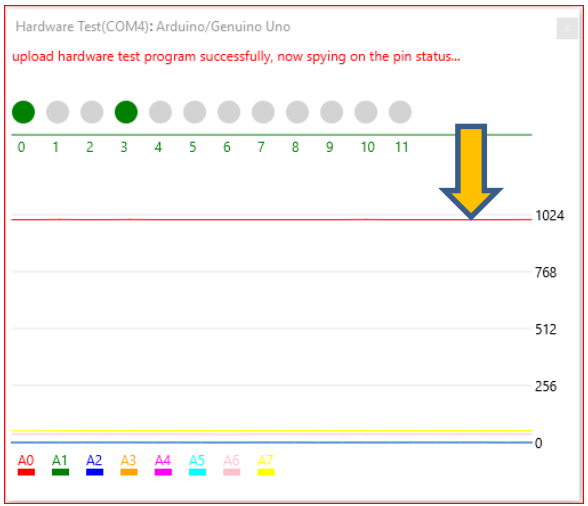
Aufgabe	Analoge Pins auf Eigenschaften untersuchen (Analogdigitalwandler)	
Fritzing Schaltung	Potentiometer 10 kΩ	Es empfiehlt sich A1 bis A5 mit GND zu verbinden!











Schaltung 1: Analoge Pins

Vorhandenes Testprogramm öffnen:

Achtung: Das in der Mixly-IDE eingebaute Testprogramm lädt sich beim Klicken auf das Symbol auf den Arduino und überschreibt ein vorhandenes Programm.

Menüzeile

Analog-Wert A0 bei 0; ca. 1,5 mV

Analog-Wert A0 bei 512; ca. 2,4 V

Analog-Wert A0 bei 1024; ca. 4,6 V

Hardware Test(COM4): Arduino/Genuino Uno

upload hardware test program successfully, now spying on the pin status...

Hardware Test(COM4): Arduino/Genuino Uno

upload hardware test program successfully, now spying on the pin status...

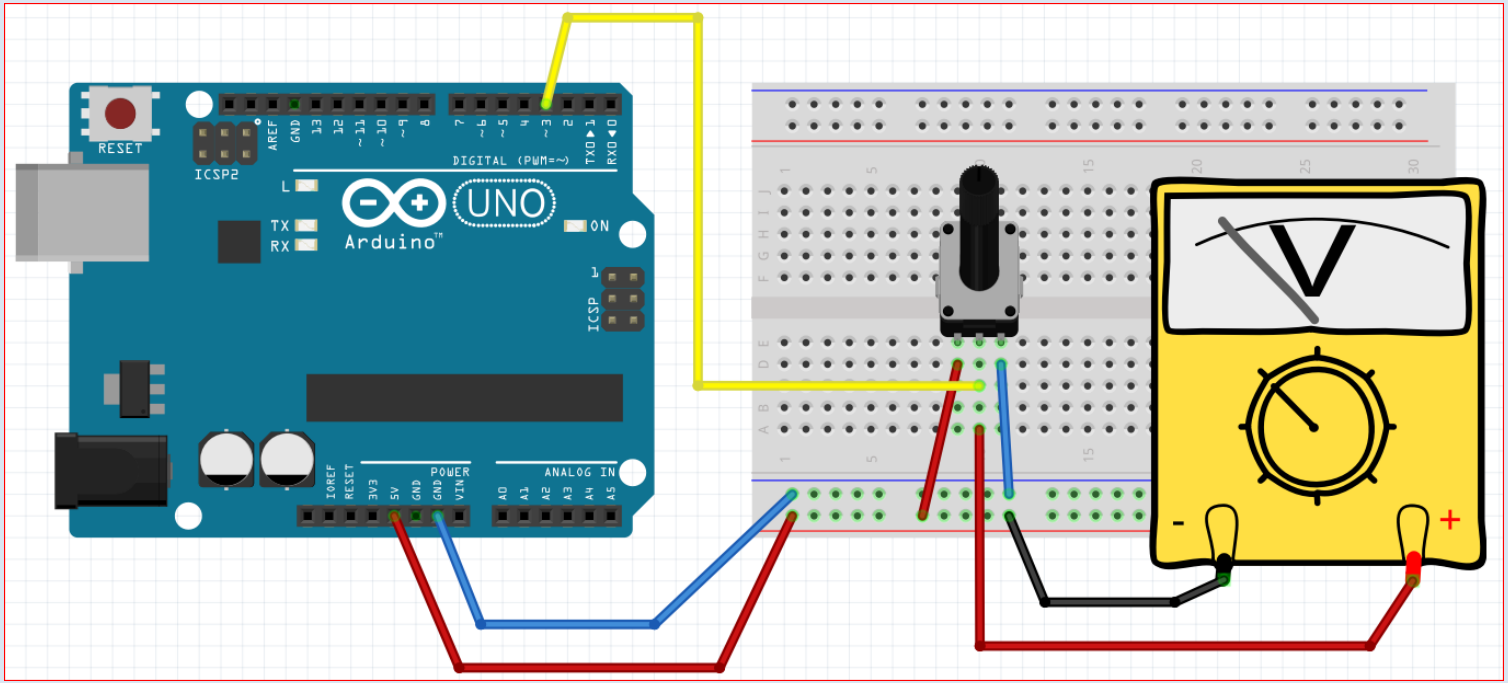
Hardware Test(COM4): Arduino/Genuino Uno

upload hardware test program successfully, now spying on the pin status...

Ergebnis: Die analogen Pins liefern Werte zwischen 0 und 1023.

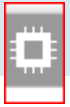
Schaltung 2: Digitale Pins

Aufgabe	Digitale Ports auf Eigenschaften untersuchen. Bei Welcher Spannung ist HIGH?
Fritzing Schaltung	Potentiometer 10 kΩ



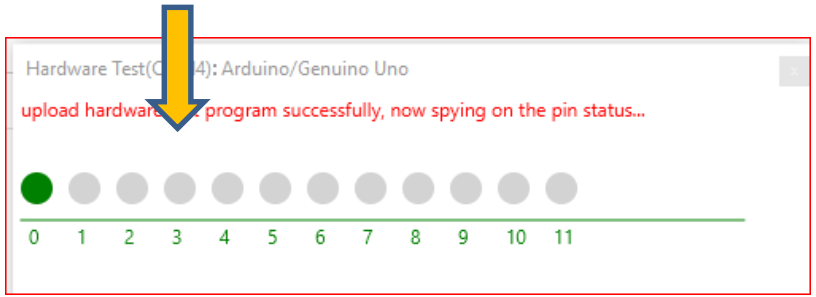
Schaltung 2: Digitale Pins

Vorhandenes Testprogramm öffnen:

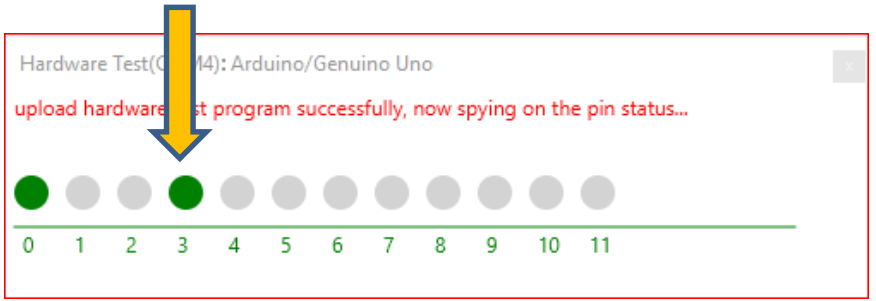
Achtung: Das in der Mixly-IDE eingebaute Testprogramm lädt sich beim Klicken auf das Symbol  auf den Arduino und überschreibt ein vorhandenes Programm.

Menüzeile

Digitaler Pin 3 kleiner 2,2 V => LOW



Digitaler Pin 3 größer 2,4 V => HIGH

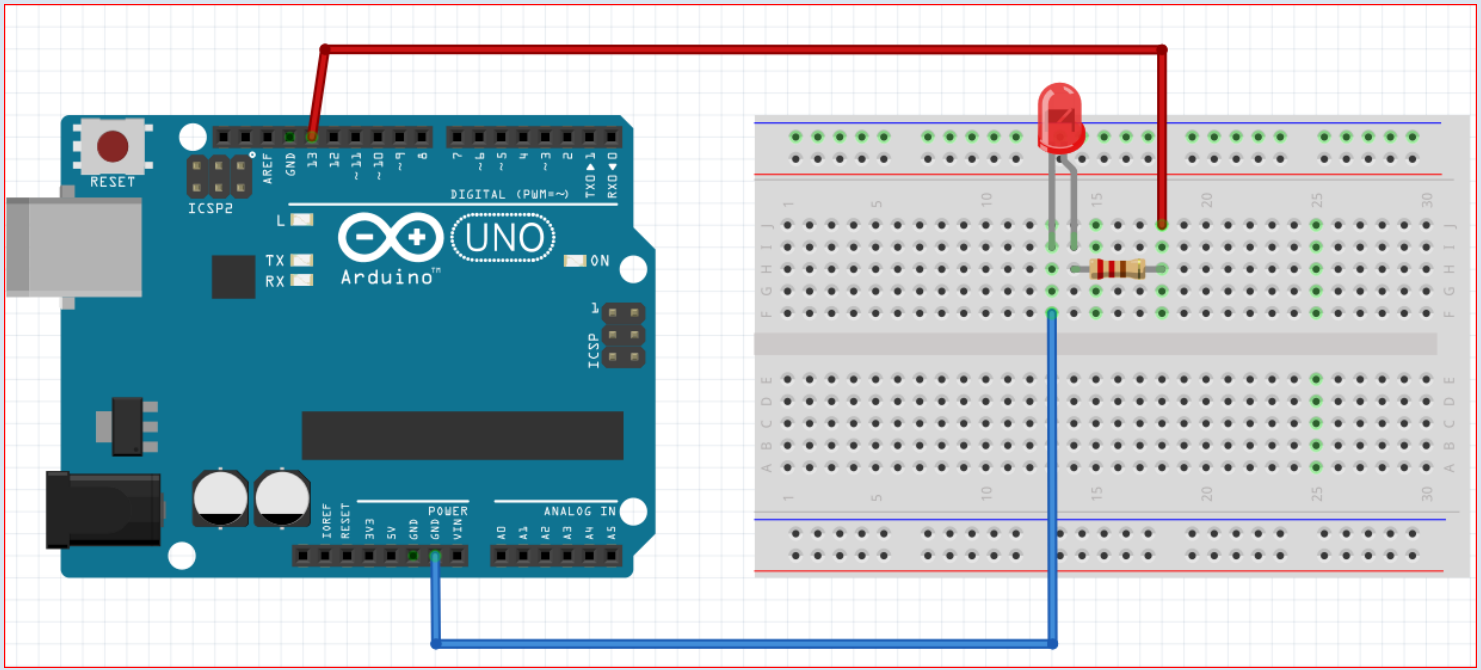


Ergebnis: Ab ca. 2,3 V wird ein digitaler Pin „HIGH“. Garantiert wird „HIGH“ erst ab 3 V und „LOW“ kleiner 2 V.

Versionen	
Empfehlung	Keyestudio-Version Mixly 0.998 mit Arduino-IDE 1.8.5
Link	https://drive.google.com/open?id=1CtP1bvZB-o4M5SfvIOOwFz-488gWsFTJ
Github-Version	Die Github-Version „Mixly 1.0.0“ läuft mit der aktuellen „Arduino-IDE 1.8.10“, ist aber nicht einfach einzurichten.
	Die folgenden Beispiele sind mit der „Keyestudio-Version“ getestet.

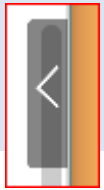
Schaltung 3: Blinkende LED

Aufgabe	Die LED soll im Sekundentakt blinken.
Fritzing Schaltung	Widerstand 220 Ω; LED

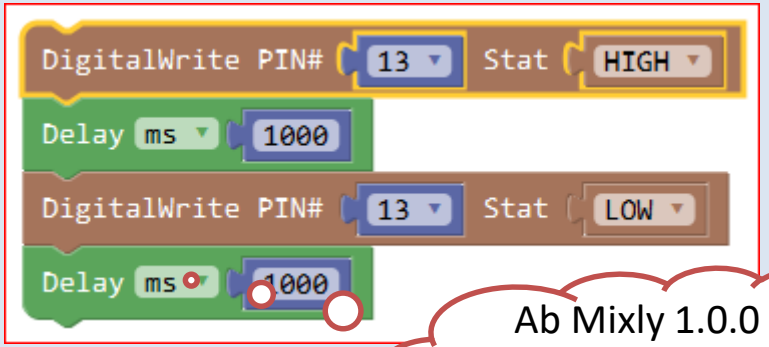


Schaltung 3: Blinkende LED, Mixly-Programm & Code

Vorhandenes Programm öffnen

- Menü „Open“.
- Aufsuchen: „C:\Users\Public\Programme\Mixly0.998_WIN(7.9)\sample“.
- Klicken auf „01闪烁LED.xml “.
- Code anzeigen durch Klicken auf  am linkem Rand.

Block-Programm



Ab Mixly 1.0.0 steht hier „millis“ statt „ms“

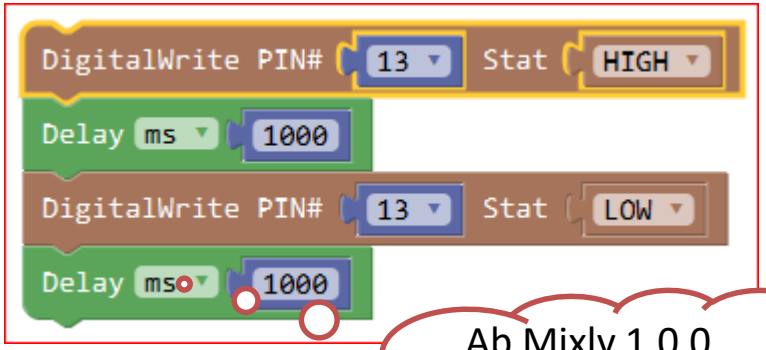
Code

```
1  void setup(){
2    pinMode(13, OUTPUT);
3  }
4
5  void loop(){
6    digitalWrite(13,HIGH);
7    delay(1000);
8    digitalWrite(13,LOW);
9    delay(1000);
10
11 }
```

Arbeitsschritte: „Open 01Blink.xml“ > „Compile “ > „Upload“
Was soll passieren: Die LED blinkt.

Schaltung 3: Blinkende LED, Blöcke

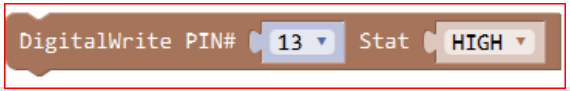
Block-Programm	Code



Ab Mixly 1.0.0
steht hier „millis“
statt „ms“

```
1  void setup(){
2    pinMode(13, OUTPUT);
3  }
4
5  void loop(){
6    digitalWrite(13,HIGH);
7    delay(1000);
8    digitalWrite(13,LOW);
9    delay(1000);
10
11 }
```

Mixly-Block Bezeichnungen	Arduino-IDE Schlüsselwörter
Digitalen PIN auf HIGH/LOW mit Mixly-Block: ***DigitalWrite PIN# Stat HIGH/LOW***	Digitalen PIN auf HIGH/LOW Arduino-Sketch: ***digitalWrite(PIN#, HIGH oder LOW)***

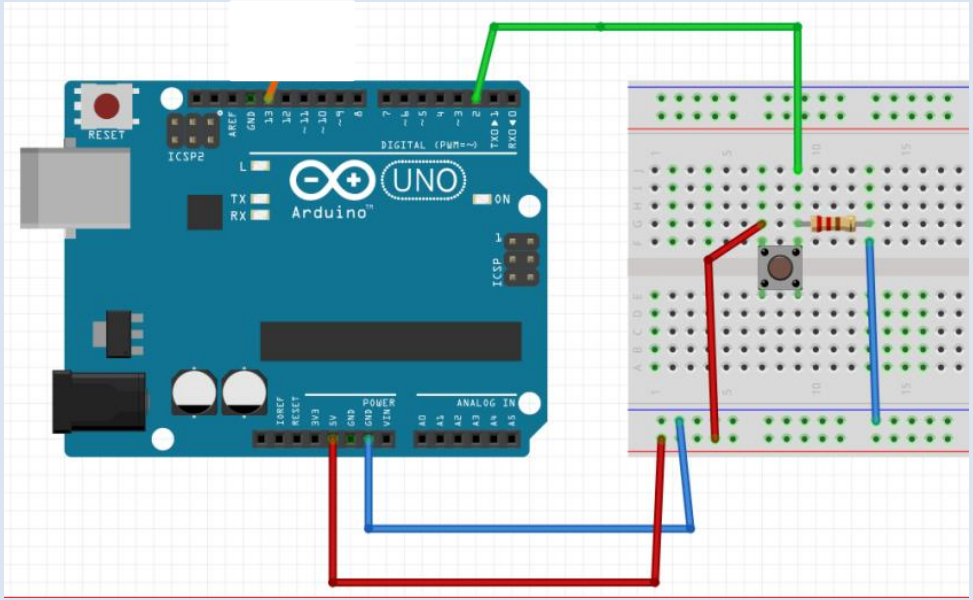


Mit Hilfe der Arduino-Referenz https://www.arduinoforum.de/arduino_referenz_down.php kann man die Arduino-Sprache nachschlagen und lernen.

Schaltung 4: Taster & Serielle Schnittstelle (Monitor)

Aufgabe Taster gedrückt, dann Nachricht ausgeben.

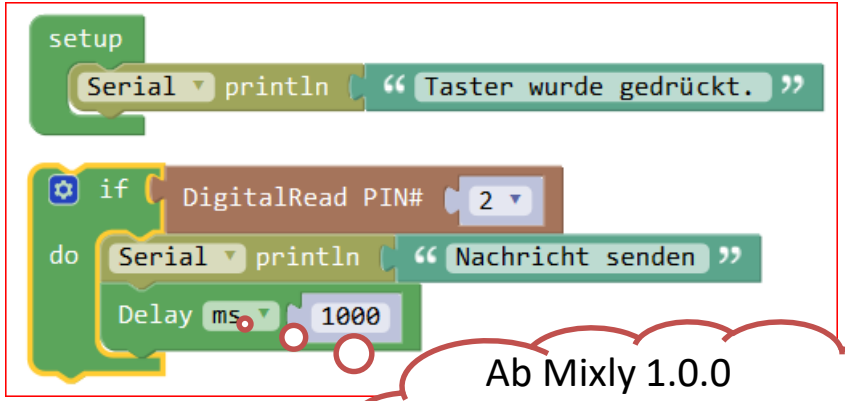
Fritzing Schaltung



Taster
Widerstand 10 kΩ

Schaltung 4: Taster & Serielle Schnittstelle (Monitor), Mixly-Programm & Code

Block-Programm

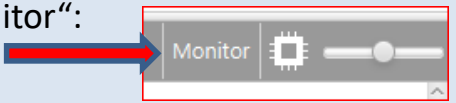


Ab Mixly 1.0.0
steht hier „millis“
statt „ms“

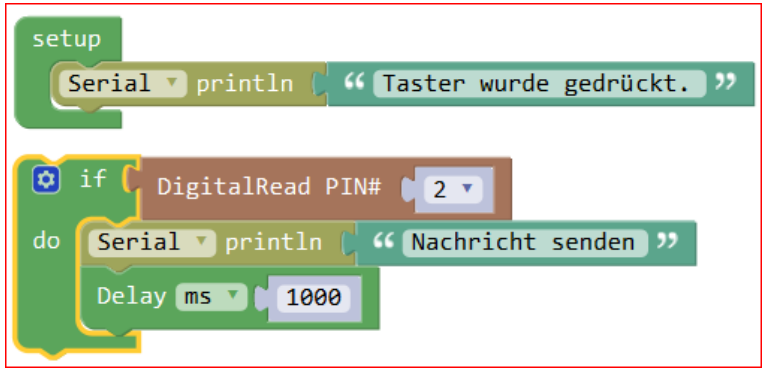
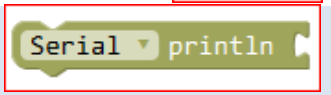
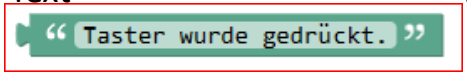
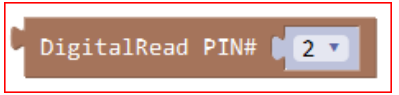
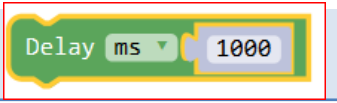
Code

```
1  void setup(){
2    Serial.begin(9600);
3    Serial.println("Taster wurde gedrückt.");
4    pinMode(2, INPUT);
5  }
6
7  void loop(){
8    if (digitalRead(2)) {
9      Serial.println("Nachricht senden");
10     delay(1000);
11
12   }
13
14 }
```

Arbeitsschritte: „Programmieren“ > „Save as“ > „Compile „ > „Upload“
Was soll passieren? Nach dem Programmstart soll bei Tastendruck eine Nachricht an den „Monitor“ geschickt werden.
Fenster Monitor öffnen durch Klicken auf „Monitor“:



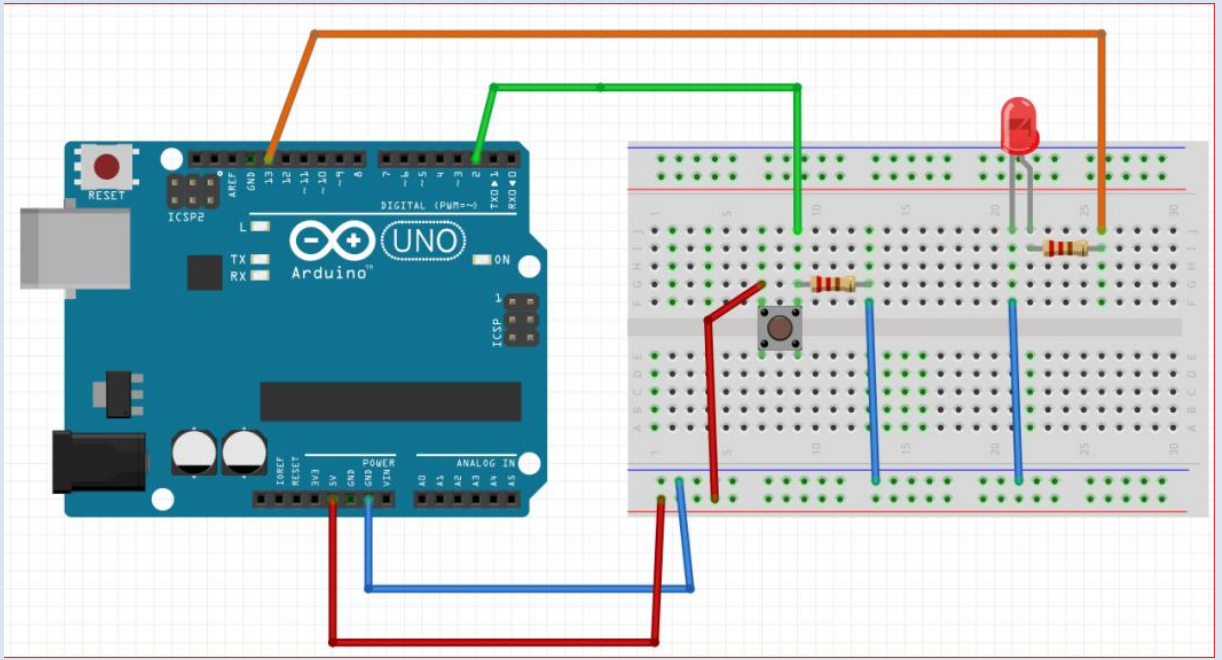
Schaltung 4: Taster & Serielle Schnittstelle (Monitor), Blöcke

Mixly-Programm	Blöcke	
	setup 	Bildet den Programmblock, der nur 1-mal ausgeführt wird.
	println 	Spricht den seriellen Ausgang an.
	Text 	Enthält den Text/Nachricht.
	if 	Bildet einen Auswahl-Block, bestehend aus Bedingung und Anweisungsteil wenn Bedingung „true“ ist
	Digitalread PIN# 	Gibt von einem digtalem PIN den Zustand HIGH / LOW zurück.
	Delay 	Programm für 1 Sekunde anhalten.

Schaltung 5: Taster & LED

Aufgabe	Taster drücken, dann LED an. Taster drücken, dann LED aus.

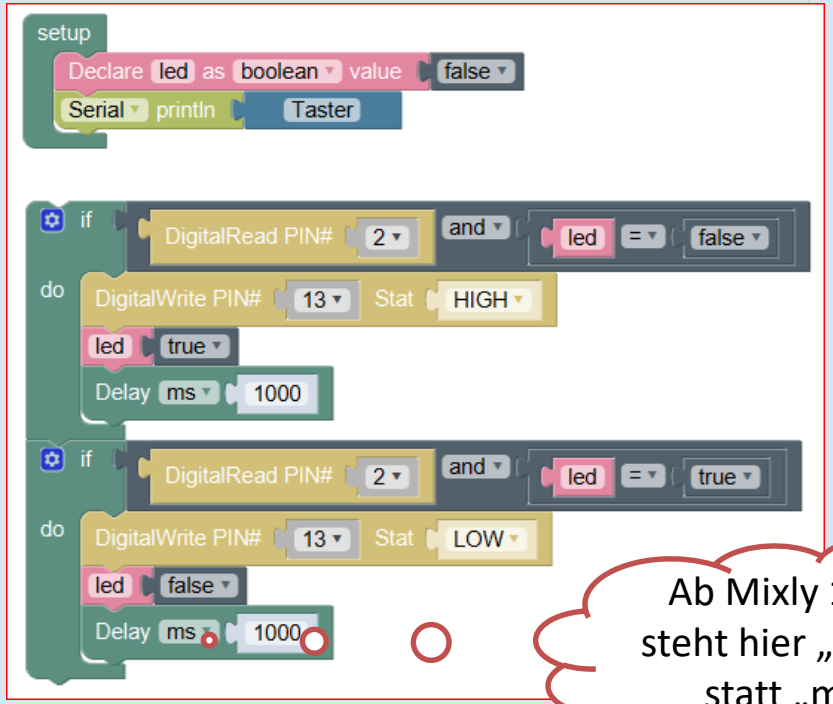
Fritzing Schaltung



Taster
Widerstand 10 kΩ

LED
Widerstand 220 Ω

Schaltung 5: Taster & LED, Mixly-Programm & Code

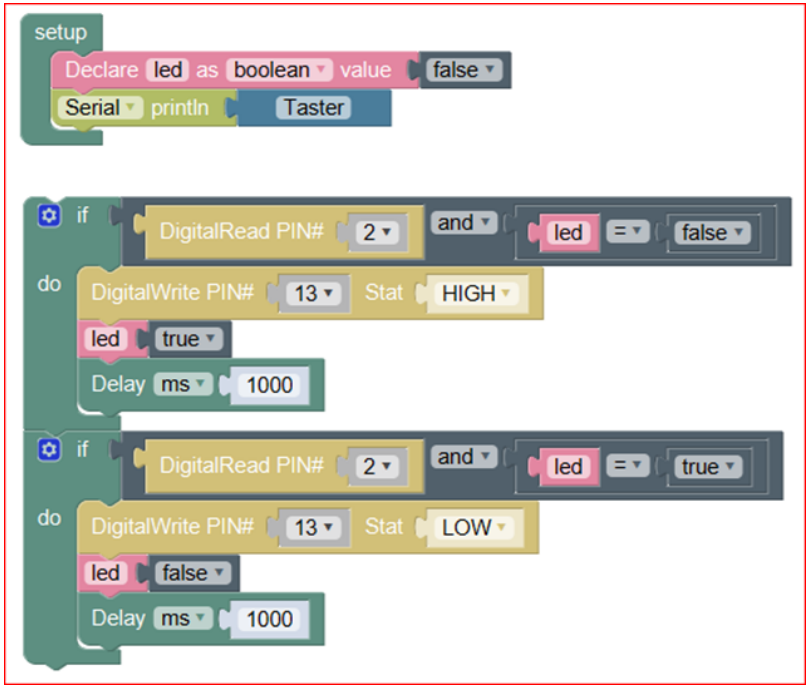


Ab Mixly 1.0.0
steht hier „millis“
statt „ms“

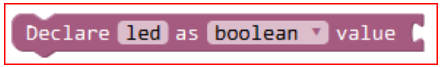
```
boolean led;

void setup()
{
  led = false;
  Serial.begin(9600);
  Serial.println("Taster");
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
}

void loop()
{
  if (digitalRead(2) && led == false) {
    digitalWrite(13,HIGH);
    led = true;
    delay(1000);

  }
  if (digitalRead(2) && led == true) {
    digitalWrite(13,LOW);
    led = false;
    delay(1000);

  }

}
```

Arbeitsschritte: „Programmieren“ > „Save as“ > „Compile „ > „Upload“
Was soll passieren? Bei Tastendruck geht die LED an bzw. aus.

Mixly-Programm	Blöcke (nur hinzugekommene)

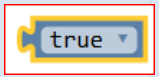


Declare



Deklariert eine Variable mit dem Bezeichner „led“ vom Typ „boolean“ und initialisiert diese mit „false“.

false



true/false-Block, stellt eine logische Wertzuweisung dar.

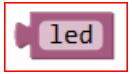
and

and-Block, enthält Platzhalter für zwei Bedingungen (logisches und).

=

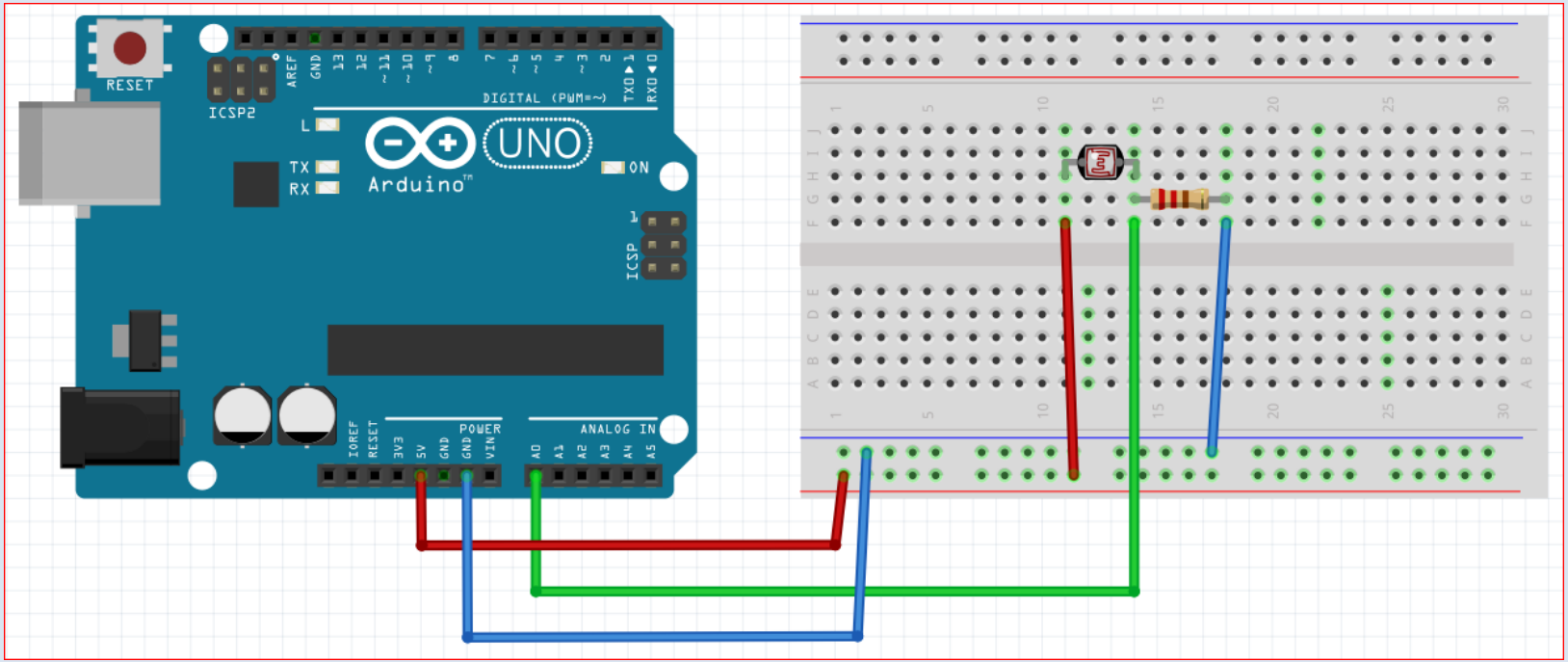
„=“-Block, enthält Platzhalter für zwei Bedingungen (prüft auf Gleichheit).

led

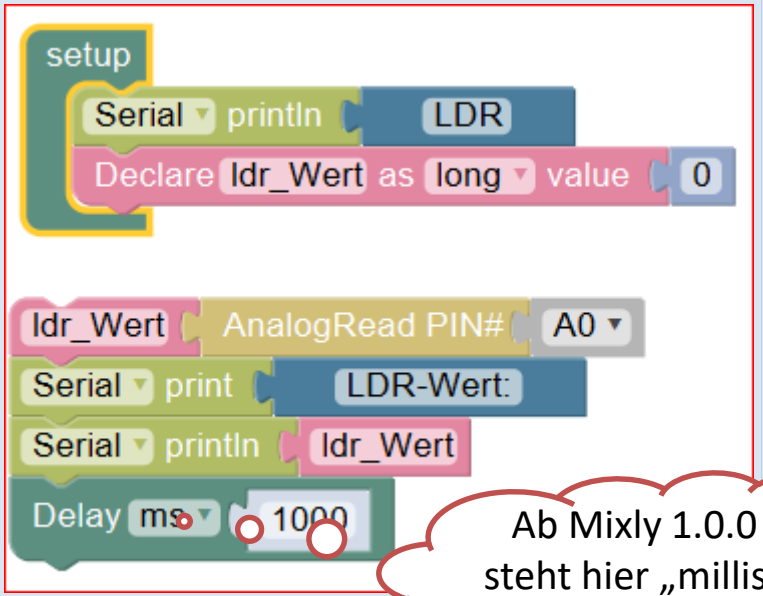


Variablen-Block

Schaltung 6: LDR	
Aufgabe	Analoge Messwerte des LDR auf dem Monitor ausgeben.
Fritzing Schaltung	LDR; Widerstand 10 kΩ



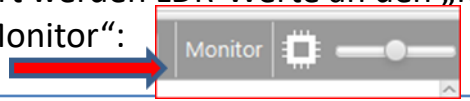
Schaltung 6: LDR, Mixly-Programm & Code



Ab Mixly 1.0.0
steht hier „millis“
statt „ms“

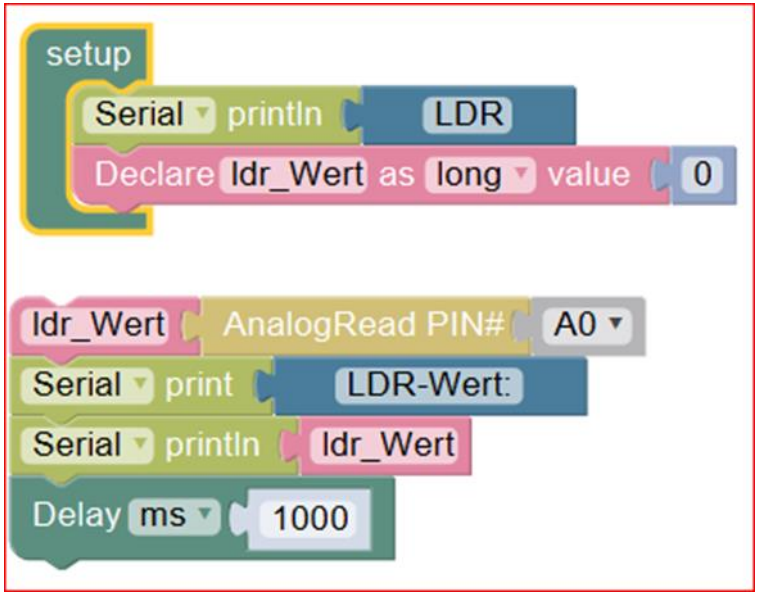
```
long ldr_Wert;

void setup()
{
  Serial.begin(9600);
  ldr_Wert = 0;
  Serial.println("LDR");
}

void loop()
{
  ldr_Wert = analogRead(A0);
  Serial.print("LDR-Wert: ");
  Serial.println(ldr_Wert);
  delay(1000);

}
```

Arbeitsschritte: „Programmieren“ > „Save as“ > „Compile „ > „Upload“
Was soll passieren? Nach dem Programmstart werden LDR-Werte an den „Monitor“ geschickt.
Fenster Monitor öffnen durch Klicken auf „Monitor“:

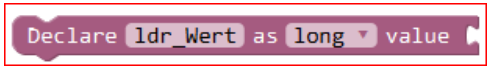


Schaltung 6: LDR, Blöcke

Mixly-Programm	Blöcke (nur hinzugekommene)

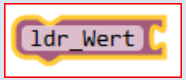


Declare



Deklariert eine Variable mit dem Bezeichner „ldr_wert “ vom Typ „long“ und initialisiert diese mit „0“.

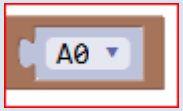
Ldr_wert



Zuweisungs-Block, dient der Zuweisung eines Wertes.

AnalogRead PIN#

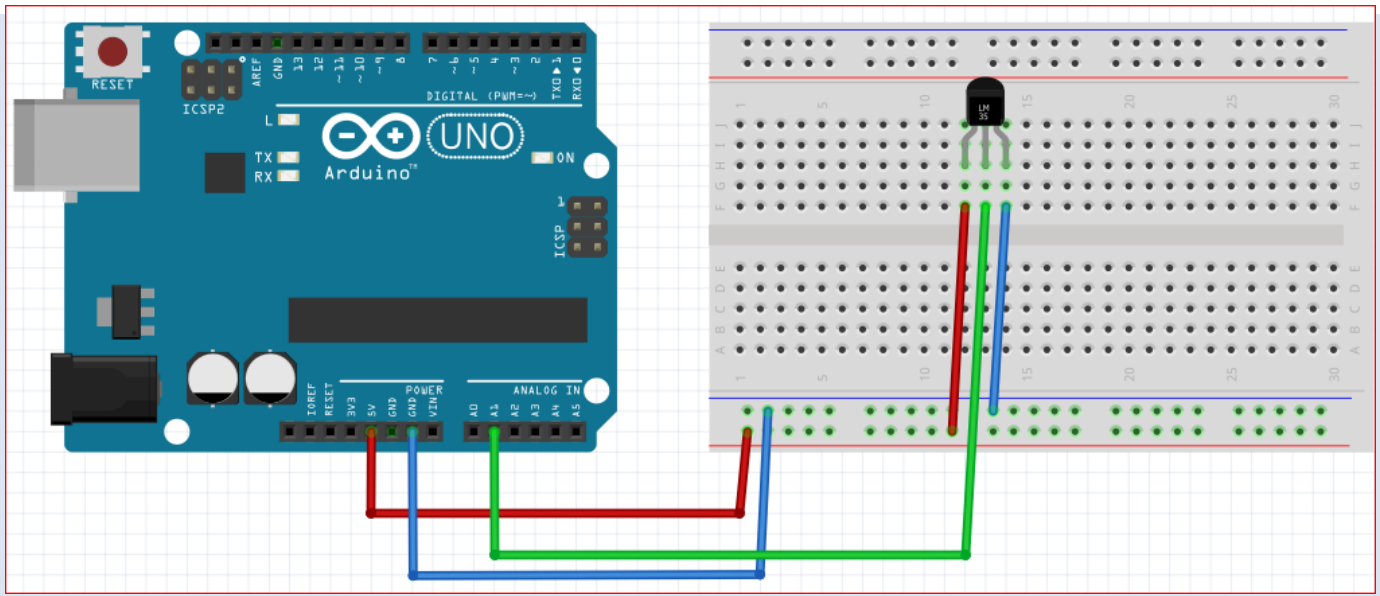
Gibt von einem analogem PIN den Zustand im Wertebereich 0 bis 1023 zurück.

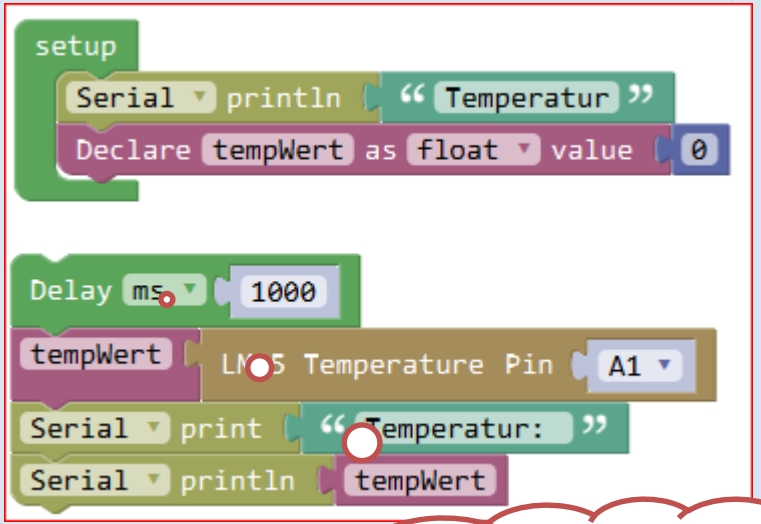


Aufklappbare Liste der analogen Pins.

Schaltung 7: Temperatur LM35

Aufgabe	Analoge Messwerte des LM35 auf dem Monitor ausgeben.
Fritzing Schaltung	LM35; -55 °C bis +150 °C; 5000 mV entsprechen 1024 Schritten (Analogdigitalwandler) milliVolt = analogRead(A1) / 1024 * 5000 Umrechnung mV in °C: tempWert = milliVolt/10 Oder: tempWert = analogRead(A1) * 0,4883

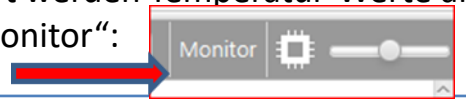




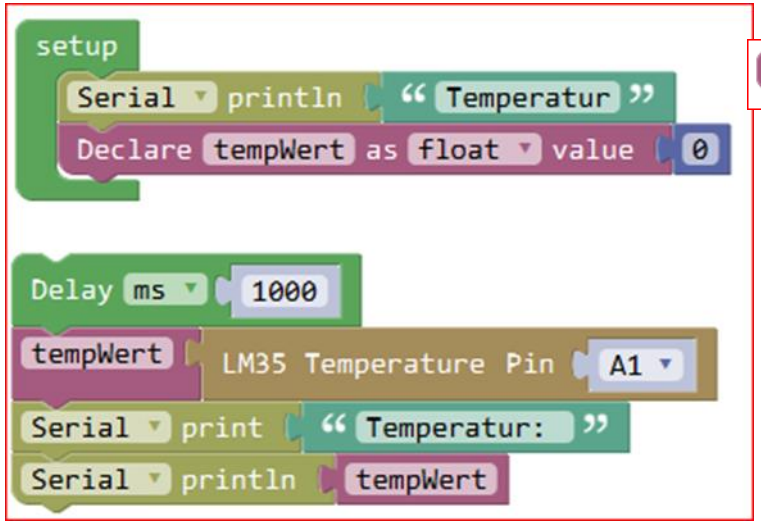
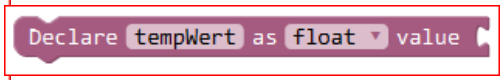
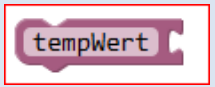
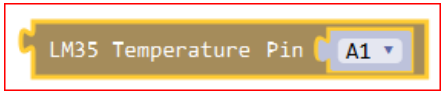
Ab Mixly 1.0.0
steht hier „millis“
statt „ms“

```
1  volatile float tempWert;
2
3  void setup(){
4    Serial.begin(9600);
5    tempWert = 0;
6    Serial.println("Temperatur");
7  }
8
9  void loop(){
10   delay(1000);
11   tempWert = analogRead(A1)*0.488;
12   Serial.print("Temperatur: ");
13   Serial.println(tempWert);
14
15 }
```

Arbeitsschritte: „Programmieren“ > „Save as“ > „Compile „ > „Upload“
Was soll passieren? Nach dem Programmstart werden Temperatur-Werte an den „Monitor“ geschickt.
Fenster Monitor öffnen durch Klicken auf „Monitor“:

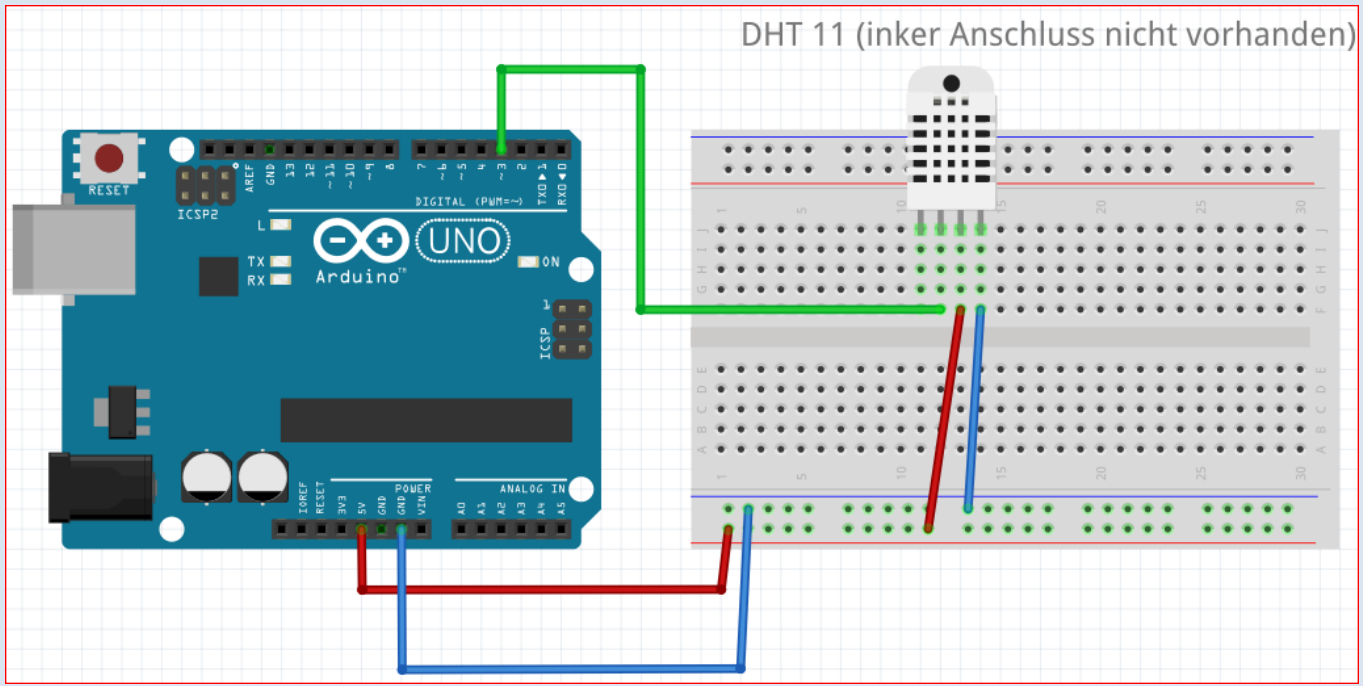


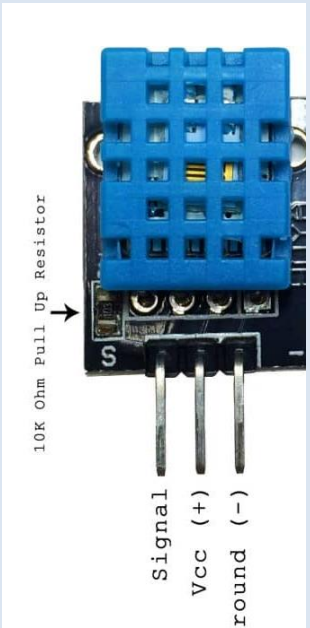
Schaltung 7: Temperatur LM35, Blöcke

Mixly-Programm	Blöcke (nur hinzugekommene)	
	Declare 	Deklariert eine Variable mit dem Bezeichner „tempWert “ vom Typ „float“ und initialisiert diese mit „0“.
	tempWert 	Zuweisungs-Block, dient der Zuweisung eines Wertes.
	LM35 Temperature Pin 	LM35-Block: Gibt von einem analogem Pin den Zustand im Wertebereich 0 bis 1023 zurück. Fügt den Code zur Umrechnung in °C automatisch ein: „tempWert= analogRead(A1)*0.488“

Schaltung 8: DHT11

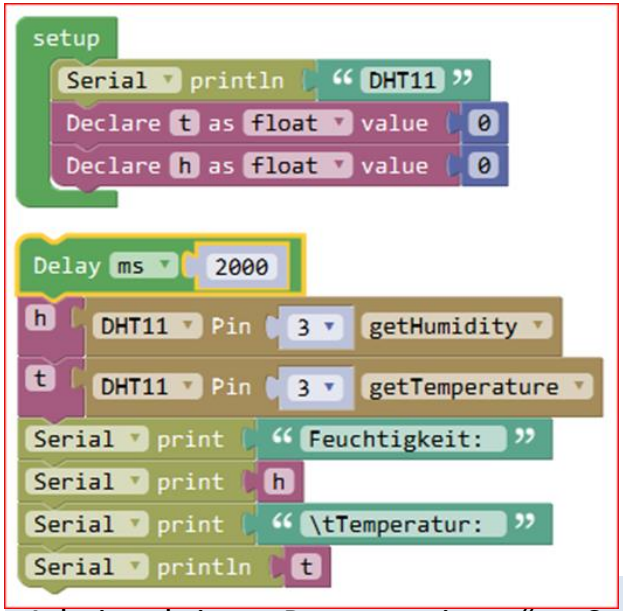
Aufgabe	Analoge Messwerte des DHT11 auf dem Monitor ausgeben.
Fritzing Schaltung	Temperatur/Luftfeuchtigkeits-Sensor Typ: DHT11





Schaltung 8: DHT11, Mixly-Programm & Code

Library „DHTlib“ ist in der „Keyestudio-Version“ enthalten.



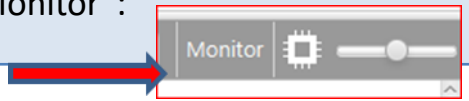

```
#include <dht.h>

volatile float t;
volatile float h;

dht myDHT_3;
int dht_3_gethumidity() {
  int chk = myDHT_3.read11(3);
  int value = myDHT_3.humidity;
  return value;
}

int dht_3_gettemperature() {
  int chk = myDHT_3.read11(3);
  int value = myDHT_3.temperature;
  return value;
}

```

```
void setup(){
  Serial.begin(9600);
  t = 0;
  h = 0;
  Serial.println("DHT11");
}

void loop(){
  delay(2000);
  h = dht_3_gethumidity();
  t = dht_3_gettemperature();
  Serial.print("Feuchtigkeit: ");
  Serial.print(h);
  Serial.print("\tTemperatur: ");
  Serial.println(t);

}
```

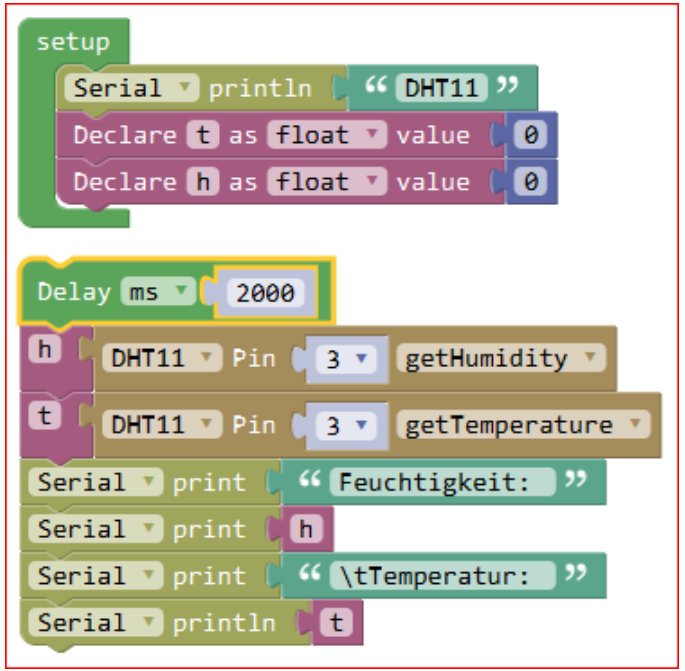
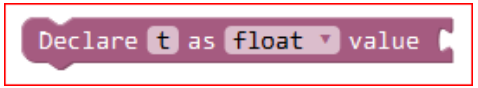
Arbeitsschritte: „Programmieren“ > „Save as“ > „Compile „ > „Upload“

Was soll passieren? Nach dem Programmstart werden Werte an den „Monitor“ geschickt.

Fenster Monitor öffnen durch Klicken auf „Monitor“:

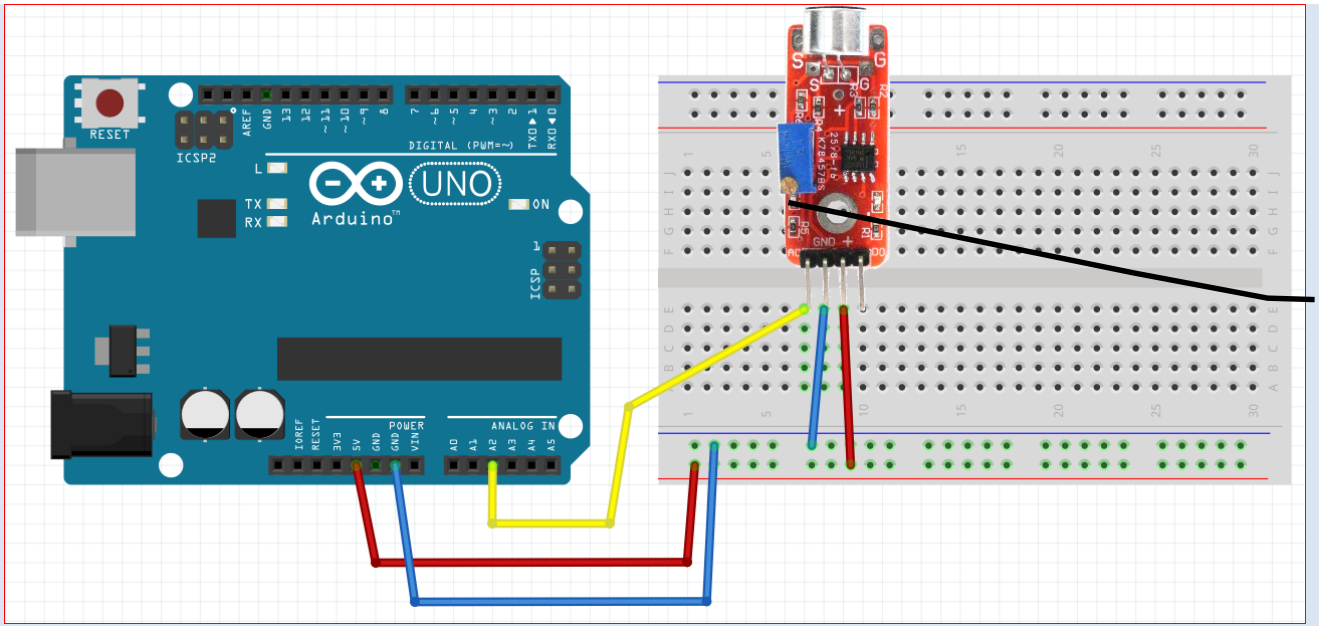


Schaltung 8: DHT11, Blöcke

Mixly-Programm	Blöcke (nur hinzugekommene)	
	Declare 	Deklariert eine Variable mit dem Bezeichner „t “ vom Typ „float“ und initialisiert diese mit „0“.
	DHT11 Sensor 	Gibt von einem digitalem Pin die Feuchtigkeit zurück. Berechnung in Library „dht.h“.
	DHT11 Sensor 	Gibt von einem digitalem Pin die Temperatur in °C zurück. Berechnung in Library „dht.h“.

Schaltung 9: Audio-Sensor KY-038

Aufgabe	Analoge Messwerte des KY-038 auf dem Monitor ausgeben. http://sensorkit.joy-it.net/index.php?title=KY-038_Mikrofon_Sound_Sensor_Modul	
Fritzing Schaltung	Audio Sensor: KY-038	AO, Analoger Output, Spannungssignal vom Sensormodul DO, Digitaler Output

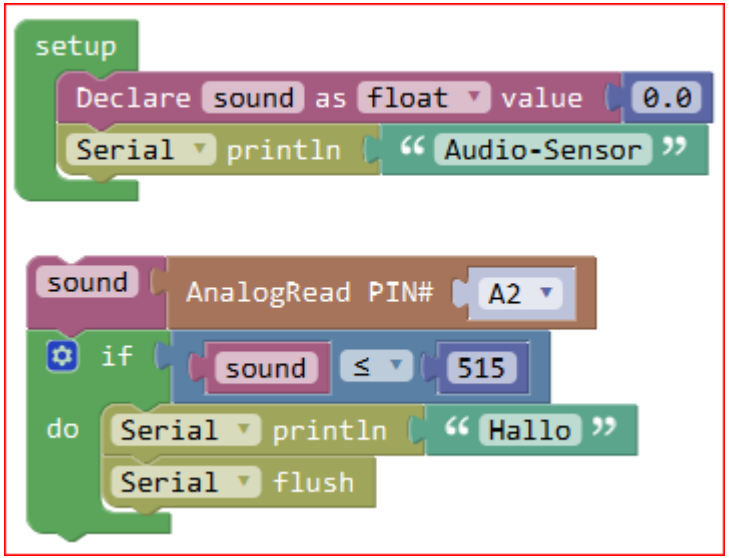


Pin + an Arduino 5+
Pin - an Arduino GND
Pin A0 an Arduino A2

Linke LED an die Schwelle aus-an justieren

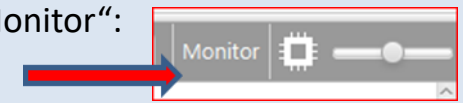
Schaltung 9: Audio-Sensor KY-038, Mixly-Programm & Code

Die Empfindlichkeit am Sensor so einstellen, dass die linke LED an der Schwelle zu an ist!



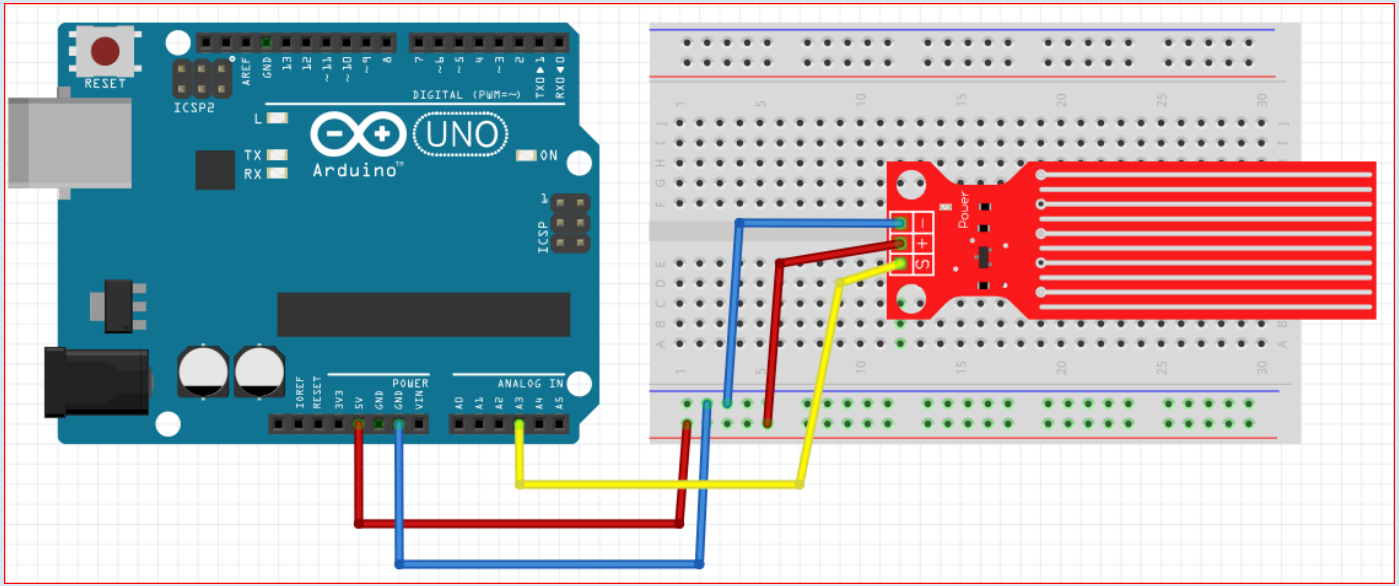

```
volatile float sound;

void setup(){
  sound = 0.0;
  Serial.begin(9600);
  Serial.println("Audio-Sensor");
}

void loop(){
  sound = analogRead(A2);
  if (sound <= 515) {
    Serial.println("Hallo");
    Serial.flush();

  }

}
```

Arbeitsschritte: „Programmieren“ > „Save as“ > „Compile „ > „Upload“
Was soll passieren? Nach dem Programmstart wird bei einer bestimmten Lautstärke „Hallo“ ausgegeben.
Fenster Monitor öffnen durch Klicken auf „Monitor“:



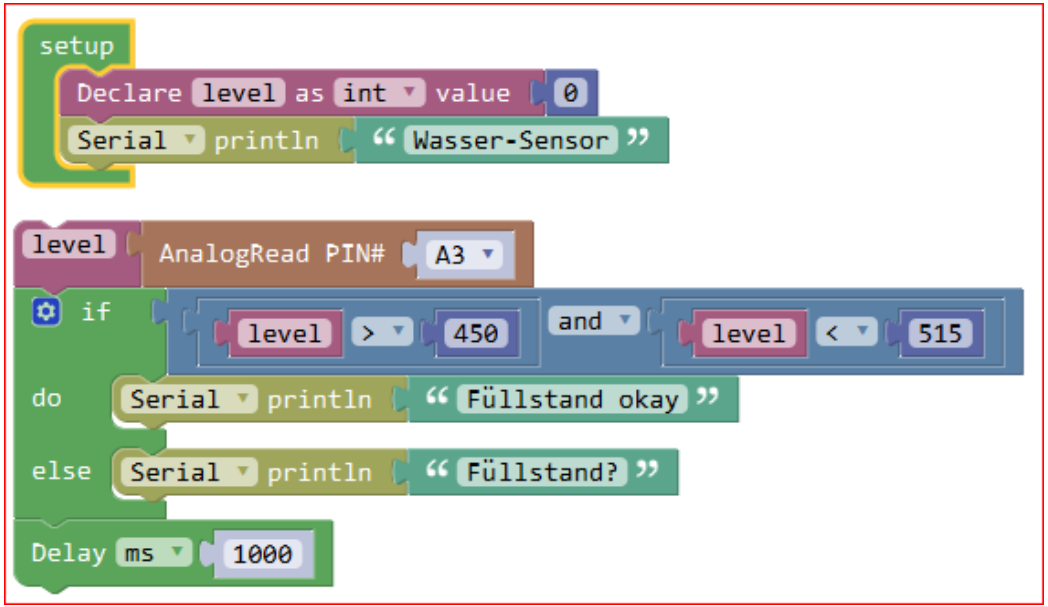
Schaltung 10: Wasser-Sensor

Aufgabe	Analoge Messwerte des Wasser-Sensors auswerten. Bei zu hohem oder zu niedrigem Wasserstand Warnmeldung auf Monitor ausgeben.
Fritzing Schaltung	Wasser-Sensor



Pin + an Arduino 5+
Pin - an Arduino GND
Pin S an Arduino A3

Den Bereich des erlaubten Wasserstandes durch Probieren herausfinden.

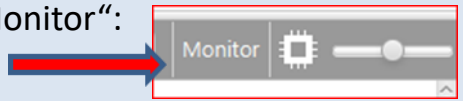



```
volatile int level;

void setup(){
  level = 0;
  Serial.begin(9600);
  Serial.println("Wasser-Sensor");
}

void loop(){
  level = analogRead(A3);
  if (level > 450 && level < 515) {
    Serial.println("Füllstand okay");

  } else {
    Serial.println("Füllstand?");

  }
  delay(1000);

}
```

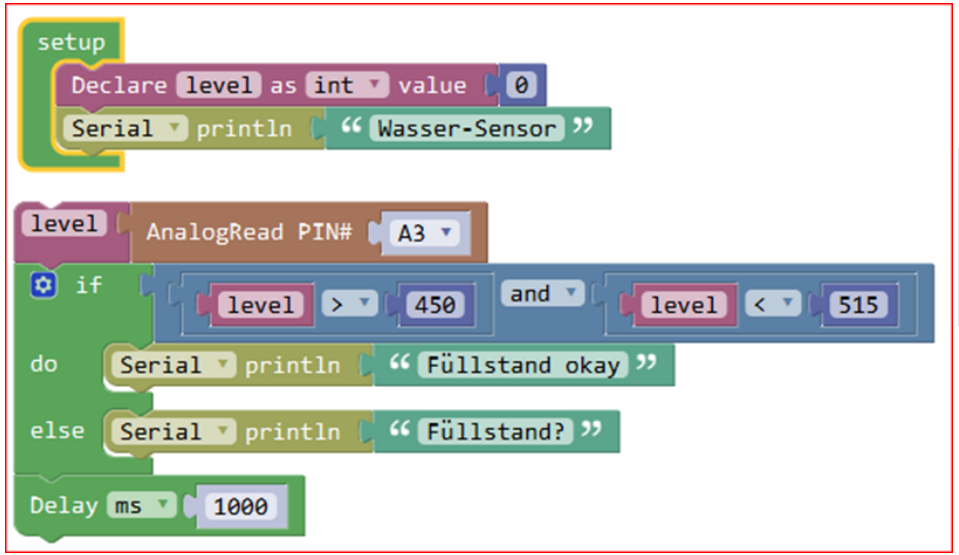
Arbeitsschritte: „Programmieren“ > „Save as“ > „Compile „ > „Upload“

Was soll passieren? Nach dem Programmstart wird bei einem bestimmten Wasserstand „Füllstand okay“ ausgegeben.

Fenster Monitor öffnen durch Klicken auf „Monitor“:

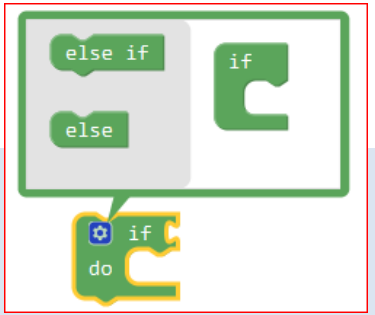


Mixly-Programm

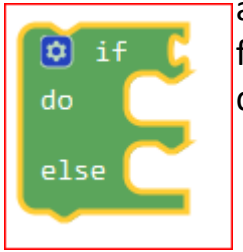


Blöcke (nur hinzugekommene)

if-else bauen



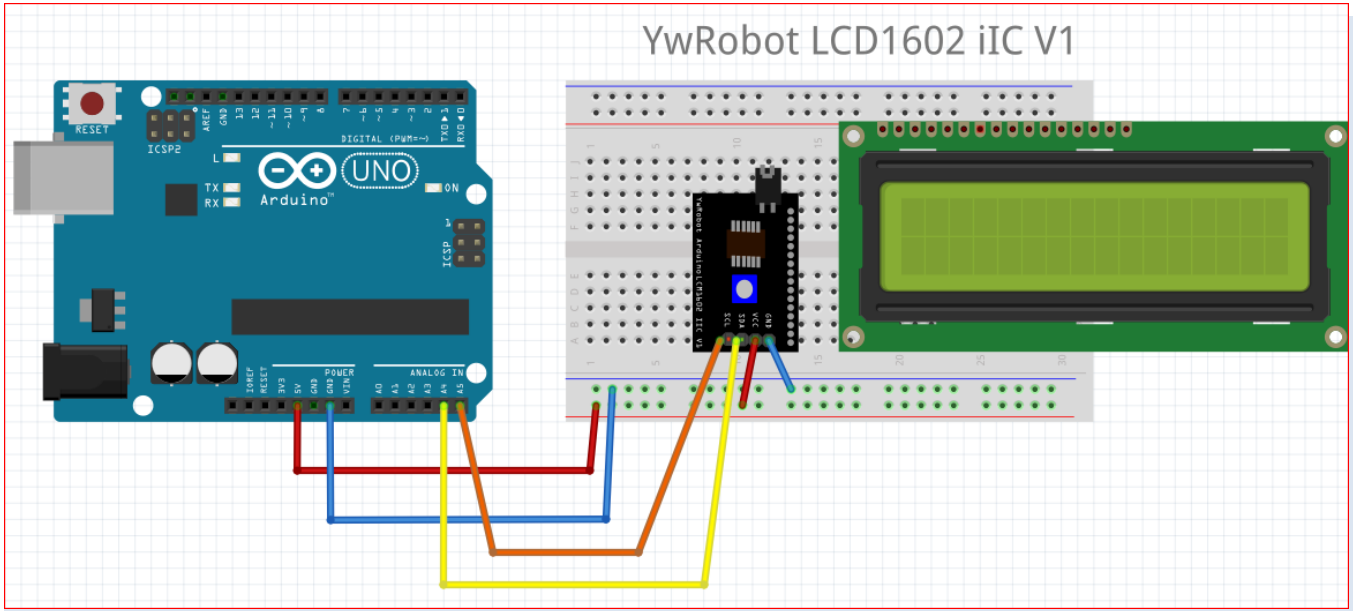
if-else



Bildet einen Auswahl-Block, bestehend aus Bedingung und Anweisungsteilen für die Bedingung „true“ (do) und für die Bedingung „false“ (else).

Schaltung 11: LCD

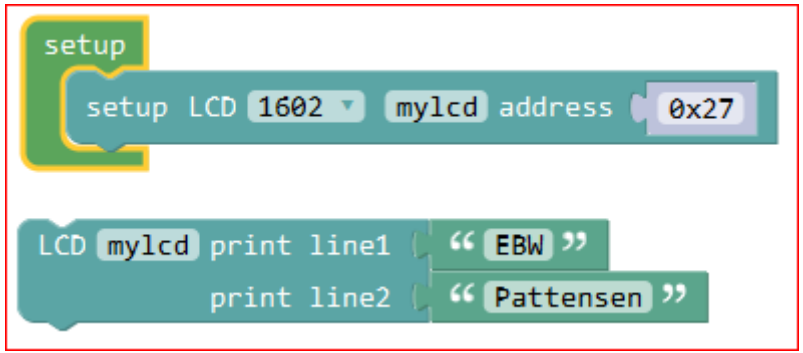
Aufgabe	Nachrichten auf einem 2-zeiligem LCD ausgeben.
Fritzing Schaltung	YwRobot LCD1602 iIC V1



I2C LCD1602	Arduino Uno
GND	GND
VCC	5V
SDA	A4
SCL	A5

Schaltung 11: LCD, Mixly-Programm & Code

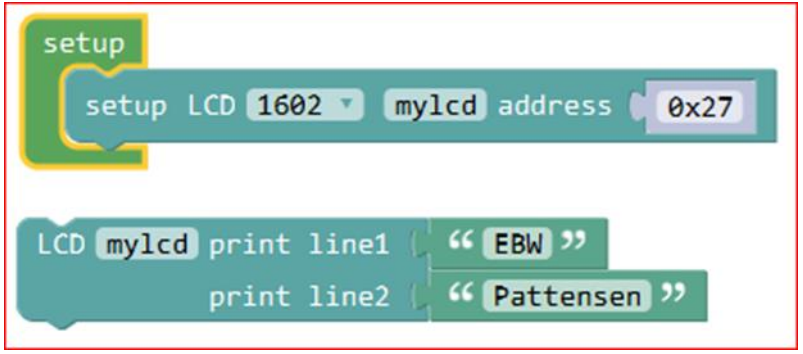
Library *„LiquidCrystal_I2C “* einbinden: Kopiere *„…\Mixly_Arduino-master\mixly_arduino\arduino-1.x.x\libraries\LiquidCrystal_I2C“*
nach: *„…\Mixly_Arduino-master\mixly_arduino\arduino-1.8.5\libraries“*




```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C mylcd(0x27,16,2);

void setup(){
  mylcd.init();
  mylcd.backlight();
}

void loop(){
  mylcd.setCursor(0, 0);
  mylcd.print("EBW");
  mylcd.setCursor(0, 1);
  mylcd.print("Pattensen");

}
```

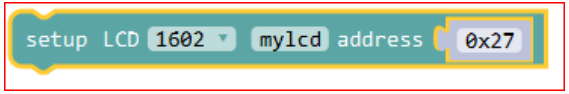
Arbeitsschritte: „Programmieren“ > „Save as“ > „Compile „ > „Upload“
Was soll passieren? Nach dem Programmstart werden Nachrichten auf dem LCD angezeigt.

Mixly-Programm



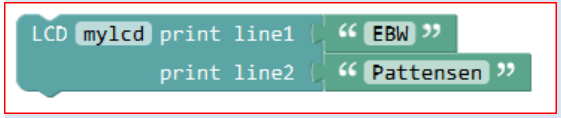
Blöcke (nur hinzugekommene)

Kategorie Monitor:
setup LCD-Block



Fügt die Library ein. Deklariert und initialisiert das Objekt „mylcd“.

Kategorie Monitor:
LCD-print line Block



Ermöglicht die Ausgabe von Text oder Werten in Zeile 1 bzw. Zeile 2.