

# Vorwort

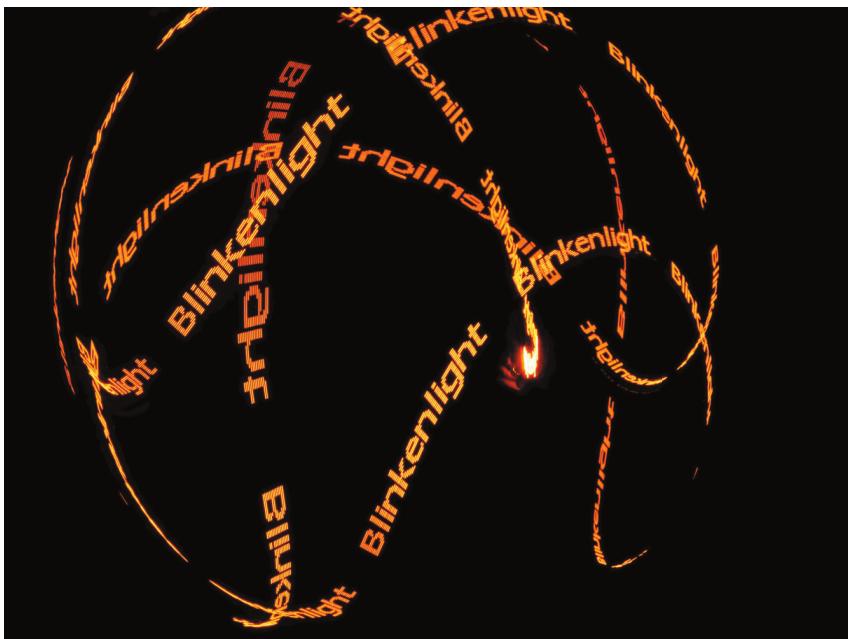
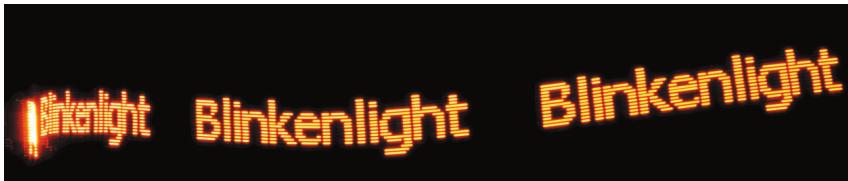
Das Lernpaket »Lichteffekte mit Arduino« wurde – genauso wie die Arduino Plattform – für Künstler, Designer und Bastler entwickelt. Es richtet sich ganz besonders an alle, die möglichst schnell erste Ideen umsetzen wollen. Gleichzeitig ist es aber auch für jene konzipiert, die nicht lange auf dem Anfängerniveau verharren, sondern möglichst bald in die Tiefen der Arduino-Programmierung vordringen wollen. Damit das gelingt, fokussieren wir auf Themen, die mit Licht zu tun haben. Das umfasst die Erstellung eigener Programme genauso wie die Bearbeitung von Bildern per Grafikprogramm.

Diesem Paket liegt eine spezielle Arduino-kompatible Platine bei. Alle benötigten Teile sind bereits fest auf der Platine integriert. Dadurch entfällt jeglicher Verdrahtungs- oder Bastelaufwand und Sie können schneller mit den ersten Schritten beginnen. Selbst Fortgeschrittene profitieren von diesem speziellen Aufbau, weil er sowohl bei der Entwicklung als auch bei der Fehlersuche nützlich ist.

Mit einfachen Programmen werden zunächst die Entwicklungsumgebung und die Grundfunktionen des Arduino untersucht. Danach beginnen wir mit grundlegenden Lichteffekten. Diese führen wir fort bis zu »Persistence of Vision«-Effekten die sich vor allem für Lightpainting (Lichtmalerei) gut eignen. Unterwegs erlernen wir den Umgang mit dem EEPROM und Interrupts. Der Lichtmalerei ist dann ein eigenes Kapitel gewidmet, das sich vor allem mit der Nachbearbeitung in Grafikprogrammen beschäftigt. Danach tauchen wir in die Tiefen der Entwicklungsumgebung ein, um zu erkunden, wie genau ein Programm in den Arduino kommt. Ein weiteres Kapitel ist dem Thema Performance-Optimierung gewidmet. Die gesammelten Erkenntnisse setzen wir ein, um die Lichteffekte deutlich zu verbessern. Abschließend folgen ausführliche Hinweise für eigene Projekte und zur geschickten Fehlersuche.

Mein besonderer Dank geht an Frau Wilma Murr, Herrn Uwe Federer und Herrn Sebastian Pinno, die die Rohfassung des Textes durchgearbeitet und dabei sehr gute Verbesserungsvorschläge eingebracht haben.

Udo Klein



# Inhaltsverzeichnis

<b>1</b>	<b>Arduino – was ist das? .....</b>	<b>11</b>
1.1	Arduino-Hardware .....	11
1.2	Arduino-Software .....	12
1.3	Arduino-Community .....	12
<b>2</b>	<b>Entwicklungsumgebung .....</b>	<b>13</b>
2.1	Installation .....	13
2.1.1	Windows .....	14
2.1.2	Linux .....	14
2.1.3	Apple .....	14
2.1.4	Fehlersuche bei der Installation .....	15
2.2	Grundfunktionen .....	15
<b>3</b>	<b>Einfache Programme .....</b>	<b>25</b>
3.1	Alle Pins high .....	25
3.2	Alle Pins low .....	29
3.3	Alle Pins hochohmig .....	31
3.4	Alle Pins als Eingabe mit Pull-up .....	33
3.5	Nichts tun .....	35
<b>4</b>	<b>Einfache Effekte .....</b>	<b>37</b>
4.1	Springender Ball .....	37
4.2	Zufallsblinker .....	39
4.3	Larson Scanner .....	41
<b>5</b>	<b>EEPROM und Resets .....</b>	<b>45</b>
5.1	Resets zählen .....	46
5.2	EEPROM schonen .....	48
5.3	Eine Bibliothek anlegen .....	51
5.4	Effekte umschalten .....	57

<b>6</b>	<b>Persistence of Vision .....</b>	<b>63</b>
6.1	MsTimer2 und Interrupts .....	65
6.2	Direkte Port-Manipulation .....	67
6.3	Arrays und Progmem .....	69
6.4	Makros .....	72
6.5	Strom sparen im Schlaf .....	74
<b>7</b>	<b>Persistence-of-Vision-Programme generieren .....</b>	<b>77</b>
7.1	Python installieren .....	77
7.2	Persistence-of-Vision-Generator .....	78
7.3	Persistence-of-Vision-Effekte umschalten .....	80
<b>8</b>	<b>Malen mit Licht .....</b>	<b>83</b>
<b>9</b>	<b>Hinter den Kulissen .....</b>	<b>105</b>
9.1	Die Entwicklungsumgebung .....	105
9.1.1	Arbeitsverzeichnis erstellen .....	108
9.1.2	C++-Programm erstellen .....	108
9.1.3	C-Programm kompilieren .....	109
9.1.4	Bibliotheken kompilieren .....	110
9.1.5	Bibliotheken einpacken .....	111
9.1.6	Programm binden .....	112
9.1.7	hex-Datei erzeugen .....	112
9.1.8	Programm in den Prozessor laden .....	113
9.2	Der C-Compiler .....	113
9.3	Der Linker .....	117
9.4	Avrdude und der Bootloader .....	119
9.5	Board und Programmieradapter-Einstellungen .....	122
<b>10</b>	<b>Performance-Optimierung .....</b>	<b>125</b>
10.1	Performance messen .....	126
10.2	Code Tuning .....	127
10.2.1	Unnötige Funktionen entfernen .....	127
10.2.2	Aktives Warten .....	128
10.2.3	Ein- und Ausgabeoperationen .....	128
10.2.4	Operationen, die implizit warten .....	128
10.2.5	digitalRead und digitalWrite .....	128
10.2.6	Delay .....	128
10.2.7	Bibliotheksfunktionen .....	129
10.2.8	Fließkommaoperationen .....	129

10.2.9	Einfachere Operationen und passende Variabtentypen .....	129
10.2.10	Näherungsverfahren .....	129
10.2.11	Algebraische Identitäten .....	130
10.2.12	Mehrfaehauswertungen vermeiden .....	130
10.2.13	Vorberechnete Werte .....	131
10.2.14	Entscheidungen möglichst früh und schnell treffen .....	131
10.2.15	Schleifen .....	131
10.2.16	Mehrdimensionale Arrays durch eindimensionale ersetzen ..	135
10.2.17	Compiler-Einstellungen .....	135
10.2.18	Assembler .....	135
<b>11</b>	<b>Flimmern entfernen .....</b>	<b>137</b>
11.1	Larson Scanner .....	138
11.2	Binäre Pulsweitenmodulation .....	146
11.3	Binäre Pulsweitenmodulation mit Interrupts .....	161
11.3.1	Larson Scanner mit binärer Pulsweitenmodulation .....	168
11.3.2	Zufallsblinker mit binärer Pulsweitenmodulation .....	170
<b>12</b>	<b>LED-Kamera .....</b>	<b>171</b>
<b>13</b>	<b>Eigene Projekte .....</b>	<b>175</b>
13.1	Vorgehensweise .....	175
13.2	Fehlersuche .....	177
13.2.1	Sicherstellen, ob überhaupt ein Fehler vorliegt .....	177
13.2.2	Fehler stabil reproduzieren .....	178
13.2.3	Fehler isolieren .....	178
13.2.4	Nachsehen statt spekulieren .....	179
13.2.5	Fehler beheben .....	180
13.2.6	Prüfen, ob die Korrektur den Fehler beseitigt .....	180
13.2.6	Nachsehen, ob weitere ähnliche Fehler vorhanden sind .....	180
13.3	Typische Fehler .....	180
13.3.1	Upload funktioniert nicht .....	181
13.3.2	Programm lässt sich nicht kompilieren .....	181
13.3.3	Kurzschluss .....	181
13.4	Arduino ohne fertig gekaufte Platine .....	181
13.4.1	Low Fuse .....	185
13.4.2	High Fuse .....	186
13.4.3	Extended Fuse .....	187
13.4.4	Wahl eines ISP .....	188

<b>14</b>	<b>Elektronik – Einführung .....</b>	<b>189</b>
14.1	Strom und Spannung .....	189
14.2	Widerstand .....	191
14.3	Reihen- und Parallelschaltung .....	192
14.4	Diode und Leuchtdiode .....	194
14.5	Kondensator .....	197
<b>15</b>	<b>Arduino-Platine .....</b>	<b>199</b>
15.1	Schaltplan .....	201
15.2	Pin-Belegung .....	202
15.3	Technische Daten .....	202
<b>16</b>	<b>Befehlsübersicht .....</b>	<b>203</b>
16.1	Grundstruktur .....	204
16.2	Kontrollstrukturen .....	204
16.3	Funktionen .....	206
16.4	Operatoren .....	206
16.4.1	Rechenoperatoren .....	206
16.4.2	Bitweise Rechenoperatoren .....	207
16.4.3	Kurzschreibweisen für Operatoren .....	207
16.4.4	Vergleichsoperatoren .....	208
16.4.5	Logische Operatoren .....	208
16.5	Werte .....	209
16.6	Datentypen .....	209
16.6.1	Qualifikatoren .....	210
16.7	Häufig benötigte Konstanten .....	210
16.8	Arrays und Strings .....	210
16.9	Pin-Zugriffe .....	211
16.10	Direkte Port-Zugriffe .....	211
16.11	Bit-Operationen .....	212
16.12	Zeit .....	213
16.13	Mathematik .....	214
16.14	Zufallszahlen .....	215
16.15	Serielle Schnittstelle .....	215
16.16	EEPROM .....	216
16.17	Progmem .....	216