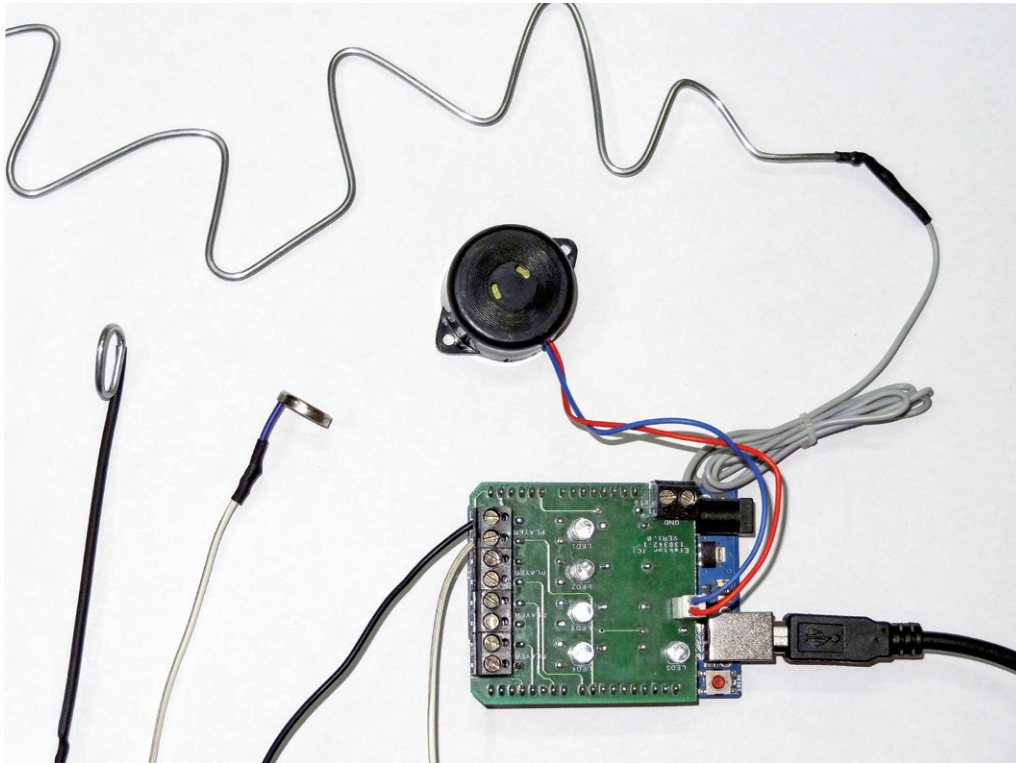


Ringspiraal

Stalen zenuwen en Arduino als scheidsrechter



Dit gezelschapsspel (voor maximaal 4 personen) is een versie van de geveesde 'ringspiraal': Hierbij moeten de deelnemers een metalen lus langs een gekronkelde staaf of ijzerdraad bewegen, zonder deze aan te raken. De eerste die daar in slaagt zonder het alarm te laten afgaan, wint. Geïnspireerd door een schakeling uit Elektor's analoge verleden besloten we een Arduino Uno toe te voegen om het spel geschikt te maken voor ons digitale tijdperk.

Sunil Malekar
(Elektor-lab India)

Het spelverloop

De ringspiraal is een behendigheidsspel, het heeft dus niets te maken met intelligentie, slimmigheden of strategische vaardigheden. Onze Arduino-versie is ontworpen voor maximaal vier spelers, elk met zijn/haar eigen draadlus om 'de proef te doorstaan'. Het is de bedoeling om de lus van het begin naar het einde langs de draad te bewegen zonder die aan te raken. Als een speler dat toch doet, klinkt er een alarmtoon en een status-LED geeft aan welke speler een fout heeft

gemaakt. Als een speler met goed gevolg het einde van zijn/haar draad bereikt en zijn/haar eindpunt aanraakt, wint hij/zij het spel. Er klinkt dan een kort muziekje en alle LED's gaan knipperen.

De schakeling: van de basis naar het schema

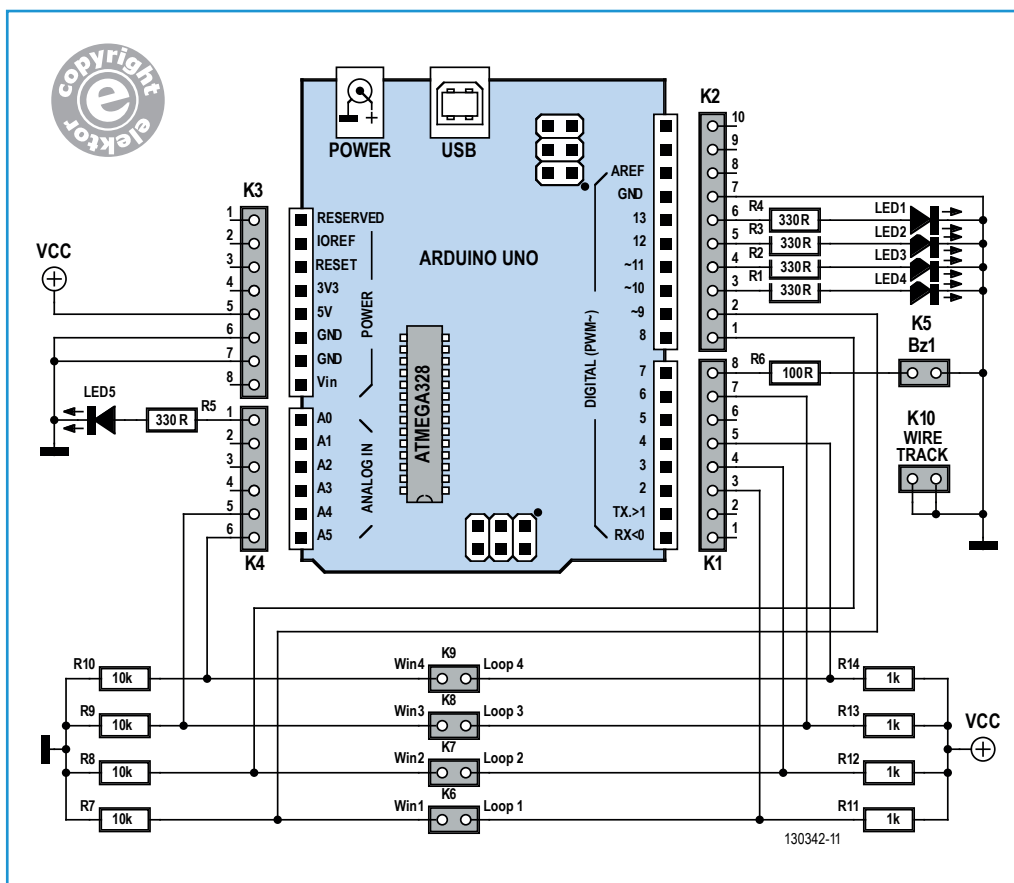
De belangrijkste delen van de schakeling zijn de draadlussen, de draadsporen, de foutindicator en de buzzer die verschillende geluiden kan voortbrengen. De grootte van

de schakeling hangt af van het aantal spelers. Onze versie is geschikt voor maximaal vier spelers, die elk een draadlus krijgen die is gekoppeld met een poortpen van de Arduino-microcontroller. De draadlussen hebben een externe pullup-weerstand naar V_{CC}. Elke deelnemer heeft een eigen Finish-contact, dat is gekoppeld met de Arduino. Deze contacten hebben een pulldown-weerstand naar GND.

De spelers houden ieder hun draadlus vast. Normaal gesproken produceert de lus een logisch '1'-signaal. Het kronkelige spoor is gemaakt uit stijf koper- of staal draad en is verbonden met de GND-aansluiting. Als het spel start, moet de speler zijn/haar draadlus van de start naar zijn persoonlijke eindpunt bewegen, zonder de draad aan te raken. Als de speler de draad aanraakt, wordt het logische '1'-signaal naar '0' getrokken. De Arduino detecteert dit en reageert meteen door de LED van die speler in te schakelen en de zoemer te laten klinken. Elke speler

heeft een eigen zoemer-geluid en een eigen LED-patroon. Als een speler zijn/haar persoonlijke Finish-contact raakt, wint hij/zij het spel en speelt de Arduino een vrolijke fanfare.

Dit project is opgebouwd rondom een Arduino Uno R3-kaart met een ATmega328P-microcontroller (figuur 1). De schakeling heeft de vorm van een 'shield' dat de GPIO-pennen van de microcontroller verbindt met de connectoren K1, K2, K3 en K4. Deze connectoren zijn verbonden met de draadlussen en de Finish-contacten van de vier spelers (Loop1, Loop2, Loop3, Loop4 respectievelijk Win1, Win2, Win3, Win4). Op de print zitten de externe pullup- en pulldown-weerstanden (R11, R12, R13, R14 respectievelijk R7, R8, R9, R10). De te volgen kronkeldraden kunnen via K10 worden verbonden met massa. De draadlussen en de eindcontacten van de vier spelers horen twee aan twee bij elkaar. Ze zijn via de aansluitingen 3, 4, 6, 7, 8, 9, A4 en A5 verbonden met de I/O-pennen van de ATmega328P volgens het schema van de



Figuur 1. Schema van de ringspiraal, een shield voor de Arduino Uno speciaal voor deze toepassing.

Arduino Uno R3. De vier indicatie-LED's zijn met de Arduino verbonden via de pennen 13, 12, 11 en 10. Die geven per speler aan wie een fout heeft gemaakt en corresponderen met respectievelijk de draadlussen loop1, loop2, loop3 en loop4. De zoemer produceert verschillende tonen voor de individuele spelers als ze met hun draadlus het kronkelspoor aanraken. LED5 (verbonden met Arduino-pen A0) geeft het begin en einde van het spel aan.

De opbouw

De dubbelzijdige, doorgemetalliseerde print die we voor dit spel hebben ontworpen, is weergegeven in **figuur 2**. De print-layout is te downloaden van [1]. Het te bouwen 'shield' is super-eenvoudig en het bevat alleen componenten met gewone aansluitdraden. De weerstanden en de vier pinheaders worden op de onderkant van de kaart geplaatst. De pennen van de pinheaders kunnen alleen aan de bovenkant van de print worden gesoldeerd. De

Onderdelenlijst

Weerstanden

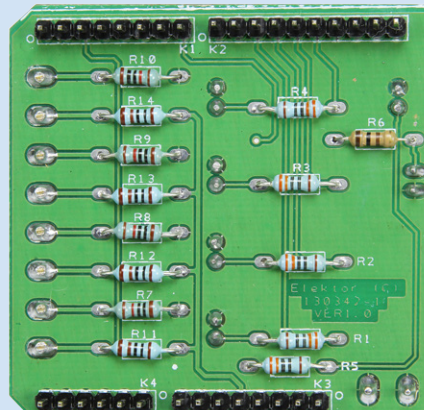
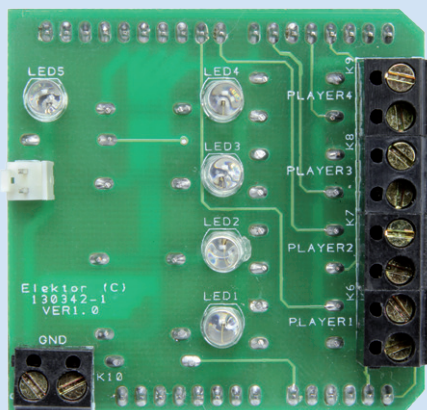
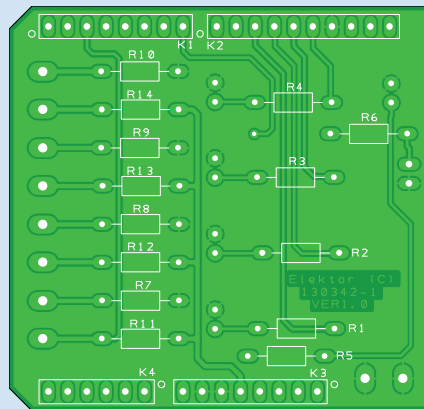
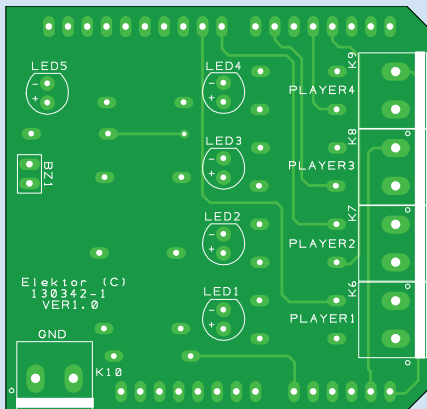
- R1...R5 = 330 Ω
- R6 = 100 Ω
- R7-R10 = 10 k
- R11...R14 = 1 k

Halfgeleiders

- LED1...LED4 = LED rood, 5 mm (bijv. Farnell-nr. 1780754)
- LED5 = LED groen, 5 mm (bijv. Farnell-nr. 2112108)

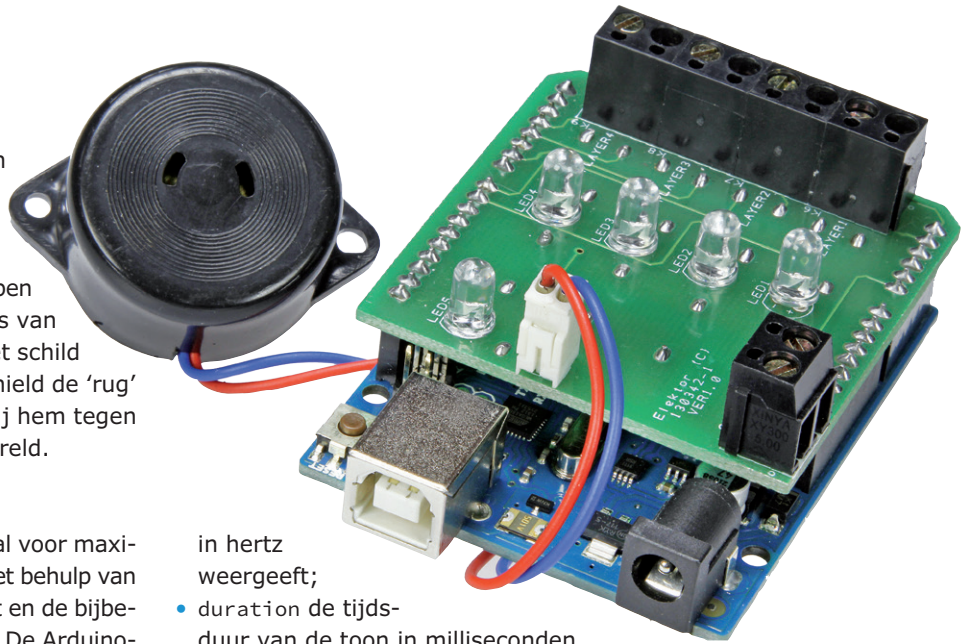
Diversen

- K1,K3 = 8-pens pinheader, steek 2,54 mm
- K2 = 10-pens pinheader, steek 2,54 mm
- K4 = 6-pens pinheader, steek 2,54 mm
- K5 (Bz1) = 2-pens pinheader, steek 2,54 mm
- K6...K10 = 2-polige printkroonsteen, steek 5,08 mm
- Buzzer (bijv. Farnell-nr. 1022400)
- Print nr. 130342



Figuur 2. De print voor de ringspiraal is ontworpen in de vorm van een Arduino-shield.

draden naar de draadlussen van de spelers en de kronkeldraden worden aangesloten via stevige schroefconnectors. Een 'shield' is trouwens een kaart die is ontworpen om op de uitbreidingsconnectors van de Arduino te steken. Net als het schild van een schildpad bedekt een shield de 'rug' van de Arduino en beschermt hij hem tegen de dingen in de harde buitenwereld.



Software

De software voor deze ringspiraal voor maximaal vier spelers is ontwikkeld met behulp van een Arduino-microcontrollerkaart en de bijbehorende Arduino-ontwikkel-IDE. De Arduino-code is gratis te downloaden van [1].

De configuratie van de firmware wordt bepaald door het aantal spelers. De individuele Finish-contacten van de spelers, de LED voor foutindicatie en de buzzer voor het aankondigen van fouten zijn allemaal geconfigureerd via de standaard I/O-pennen van de microcontroller van de Arduino.

De functie `tone ()` zorgt ervoor dat de Arduino een noot weergeeft via de buzzer. Met nog wat meer software er omheen kunnen we daar ook melodietjes mee afspelen. De functie wordt als volgt aangeroepen:

`Tone (pin, frequency, duration)`

of

`Tone (pin, frequency)`

Voorbeeld:

`Tone (3,440)`

waarin

- `pin` het nummer is van de I/O-pen waar de speaker/buzzer op is aangesloten;
- `frequency` de frequentie van de toon

in hertz weergeeft;

- `duration` de tijdsduur van de toon in milliseconden vastlegt.

Het geluid is te stoppen met de aanroep:

`noTone (pin)`

De toonfunctie genereert een blokgolf met de gespecificeerde frequentie (en een duty-cycle van 50 %) op een I/O-pen. Als de tijdsduur van de toon niet is gespecificeerd, dan gaat de golf door totdat `noTone()` wordt aangeroepen. De pen wordt verbonden met een piezo-buzzer of een ander soort speaker.

Er kan maar één toon tegelijk worden gegenereerd. Als er al een toon wordt weergegeven via een andere pen, heeft het aanroepen van `tone ()` geen effect. Als er al een toon wordt weergegeven op dezelfde pen, wordt de frequentie veranderd.

Voor het weergeven van een reeks van tonen om een simpel deuntje af te spelen, gebruiken we het bestand 'pitches.h'. In dit bestand worden macro's gedefinieerd, die de frequentie van verschillende noten in hertz weergeven. Zo kunnen we de melodie coderen in de vorm van muziknoten. Dat maakt het programmeren van 'liedjes' een stuk gemakkelijker. In

Weblinks

- [1] www.elektor-magazine.nl/post
- [2] <http://code.google.com/p/rogue-code/wiki/ToneLibraryDocumentation>
- [3] <http://en.wikipedia.org/wiki/Note>
- [4] <http://arduino.cc/en/Tutorial/Tone>

Tabel 1. Muzieknoten	
Constante	Frequentie (Hz)
NOTE_B2	123
NOTE_C3	131
NOTE_CS3	139
NOTE_D3	147

het bestand 'pitches.h' staan alle te gebruiken codes voor de noten en de afspraken voor het coderen.

Muzieknoten maken

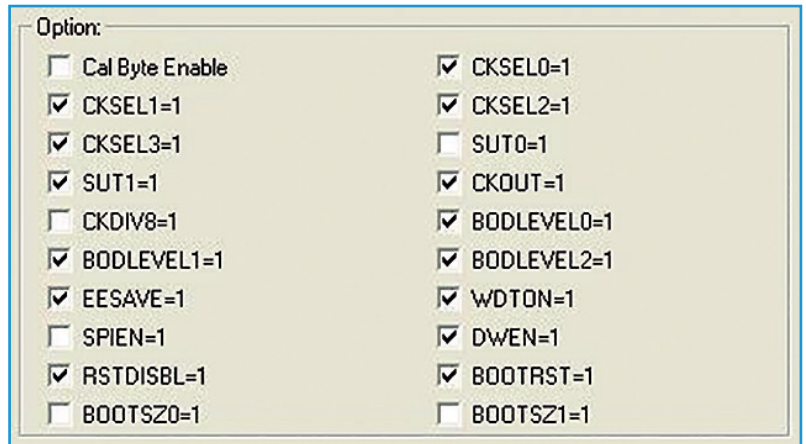
Simpele tonen werken net als muzieknoten. Toonhoogte en tijdsduur van de noten (octaven) zijn te coderen met behulp van het bestand 'pitches.h'.

De toon-library is te downloaden van [2], een inleiding in het gebruik ervan van [3] en Arduino-specifieke zaken van [4].

In de firmware zijn wat muzieknoten beschikbaar op basis van 'pitches.h', die zijn gemaakt met behulp van de bovenstaande links. Enkele veelgebruikte noten zijn weergegeven in **tabel 1**.

Het programmeren van de Arduino

De Arduino Uno-kaart kan worden geprogrammeerd vanuit de Arduino-IDE met behulp van de ingebouwde bootloader. Hier hebben we de Arduino Uno geprogrammeerd met behulp van een externe programmer. Bij gebruik van een externe programmer moeten we zelf zorgen voor de instelling van die ellendige fuse-bits.



Figuur 3. Let bij het gebruik van een externe programmer op deze instellingen voor de fuse-bits.

Ze worden geconfigureerd zoals weergegeven in **figuur 3**.

Spelen maar!

Als de schakeling is opgebouwd, sluiten we de Arduino aan op een netspanningsadapter. LED5 zal gaan knipperen, wat aangeeft dat het spel kan beginnen! De speler die het eerst het einde van zijn/haar kronkelspoor bereikt zonder het aan te raken, is de winnaar en dat wordt gevierd door het oplichten van de juiste LED en het afspelen van het bijbehorende muziekje via de buzzer. Een nieuw spel wordt gestart door op de reset-knop van de Arduino-kaart te drukken.

(130342)