

Retro Gaming met Arduino

We maken een kleine spelconsole, geschikt voor retro games.

Laatste versie code+materialen: <https://github.com/ingegno/RetroGames>

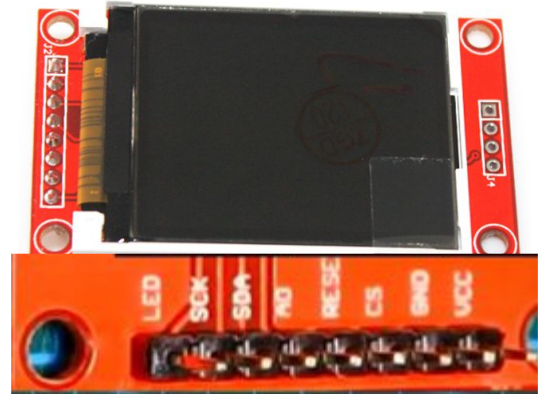
Componenten:

Het ST7735 1.8" Color TFT scherm

- 128 * 160 (resolutie), 1.8 inch TFT LCD scherm
- Grote kijkhoek, Groot contrast
- Display interface is serial, en heeft **5 draden** (CS, RS, SCL, SDA, RST) nodig om aan te sturen.
- SDcard aanwezig met **SPI** interface (CS / MOSI / MISO / SCK) indien gewenst voor beelden.

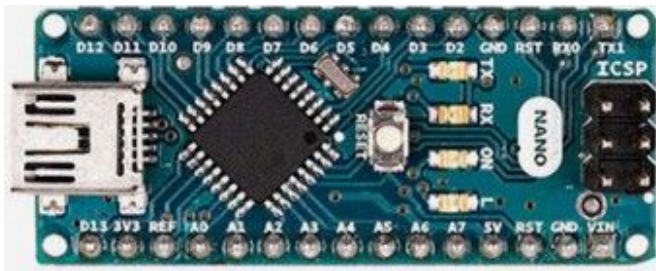
Kopen: Zoek op TFT LCD Display module ST7735S 128x160 51/AVR/STM32/ARM 8/16 bit, bv

- <https://www.adafruit.com/product/358>
- [Ebay](#) of Aliexpress zoeken op de module



Arduino Nano

We gebruiken de kleine Arduino Nano. Deze werkt met pinnen van 5 Volt. Overal te koop



Joystick

De joystick heeft 5 mogelijke bewegingen: voor, achter, links, rechts en je kan hem indrukken.

Hij heeft 5V voeding nodig. De richtingen zijn potentiometer. De knop is via output SW.

Kopen: [velleman](#)



Druknoppen

Een spelconsole heeft minsten 2 drukkknoppen. Een A en een B, met A links, B rechts.

Neem 2 verschillende kleuren!



Batterij + Schuifschakelaar

Om zonder kabel te kunnen spelen hebben we ook nog een 9V batterij nodig. Voorzie ook een schuifschakelaar.

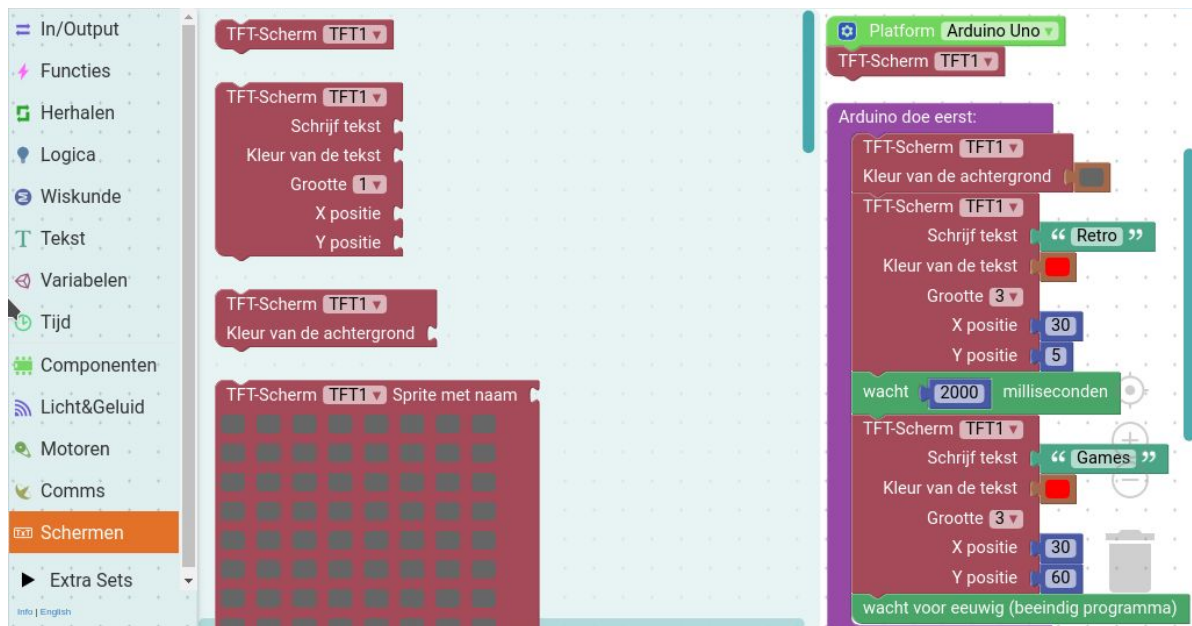
Retro Game: Speel met Tekst

We koppelen het scherm aan de Arduino Nano en testen het. De connecties moeten als volgt:

Arduino Pin	TFT Pin
5V/Vcc	Vcc
Gnd	Gnd
8	RESET
9	AO - D/C
10	CS
11	SDA (MOSI)
12 Niet Nodig	SDO (MISO) Niet nodig
13	SCK
3.3V	LED

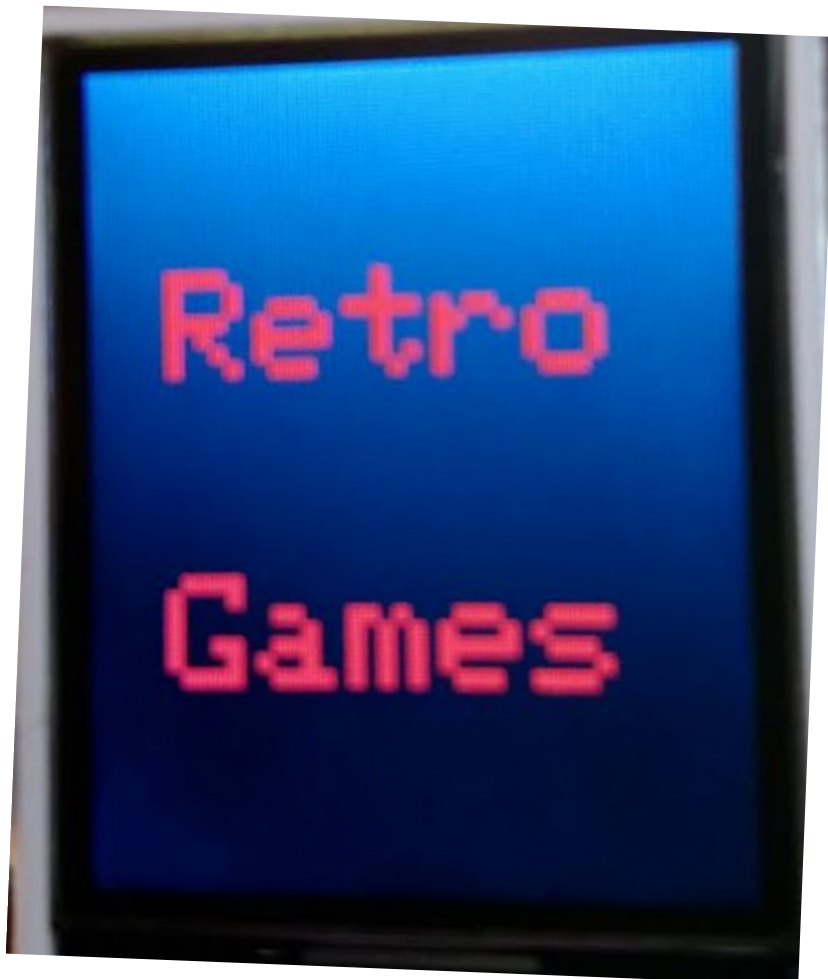
Maak ook de batterij connectie met een schuifschakelaar om de GND te onderbreken. Voorzie een **gemeenschappelijke GND** lijn. De 9V (rode draad) van de batterij moet verbonden worden met de **VIN** pin van de Nano.

Maak volgend programma met de blokken in menu Schermen en Tijd:



Eigen Tekst Screen Saver

Test of de code werkt. Het resultaat zou moeten zijn:



Opdracht:

Speel met de tekst: Schrijf wat anders. Andere kleuren

Opdracht:

Maak een screensaver: laat de tekst bewegen over het scherm

Retro Game: Wandel Sprite!

Een sprite is een klein speelfiguurtje gemaakt met pixels. We maken er een en laten hem bewegen. Om dat te kunnen hebben we de JoyStick nodig. Bekabel hem al volgt:

Arduino Pin	Joystick Pin
Vcc 5V	Vcc 5V
Gnd	Gnd
A1	VRx
A0	VRy
D4	SW

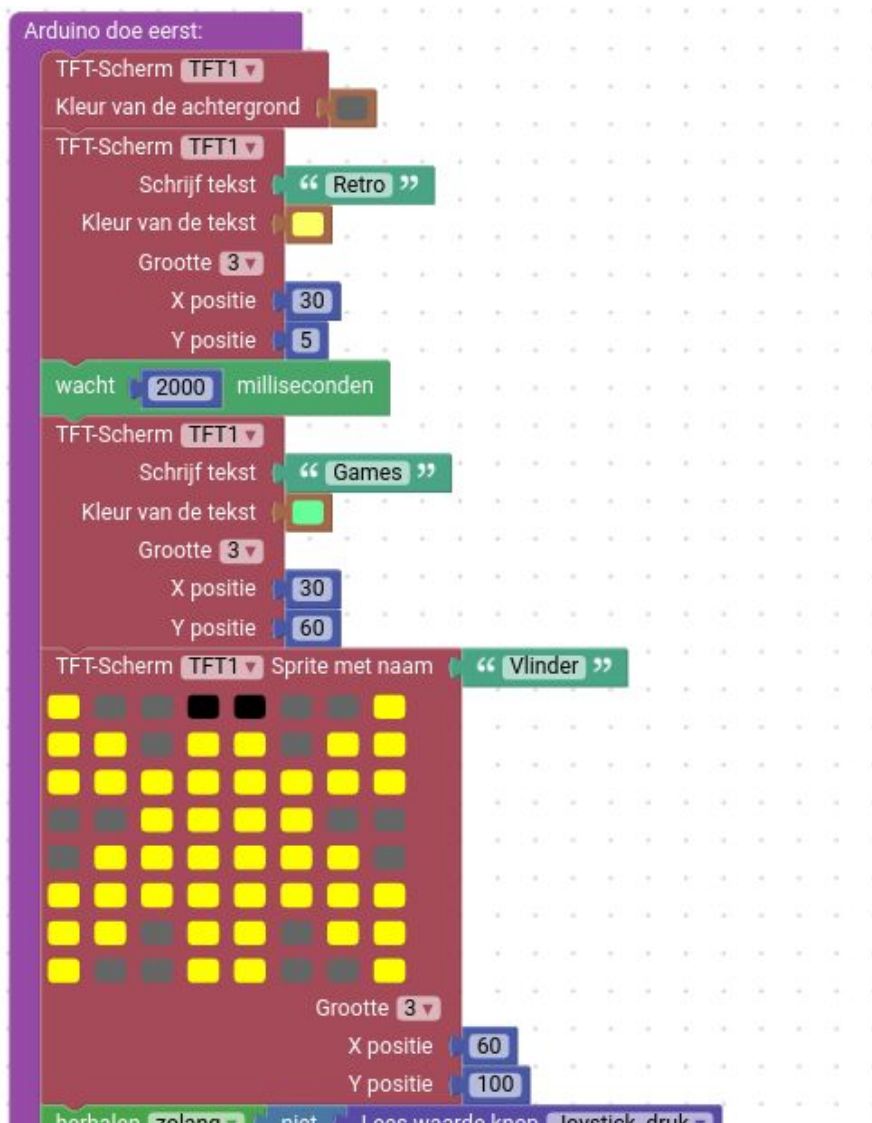
Maak een Sprite, bijvoorbeeld 'Vlinder':



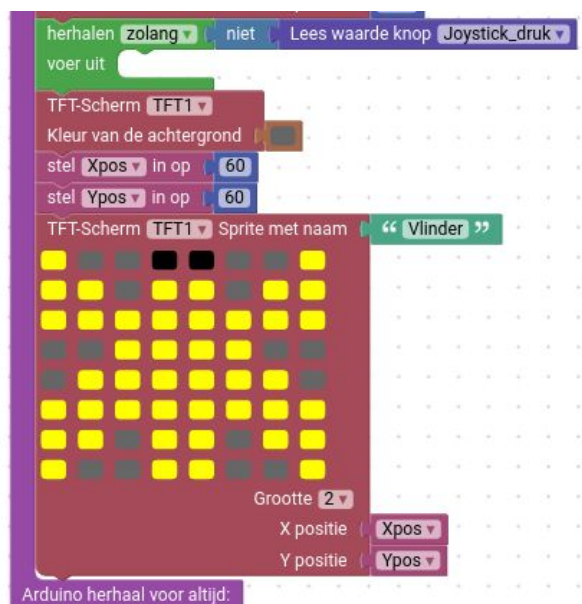
We lezen nu de toestand van de Joystick in, en laten de Vlinder bewegen! Eerst moeten we de joystick aanmaken:



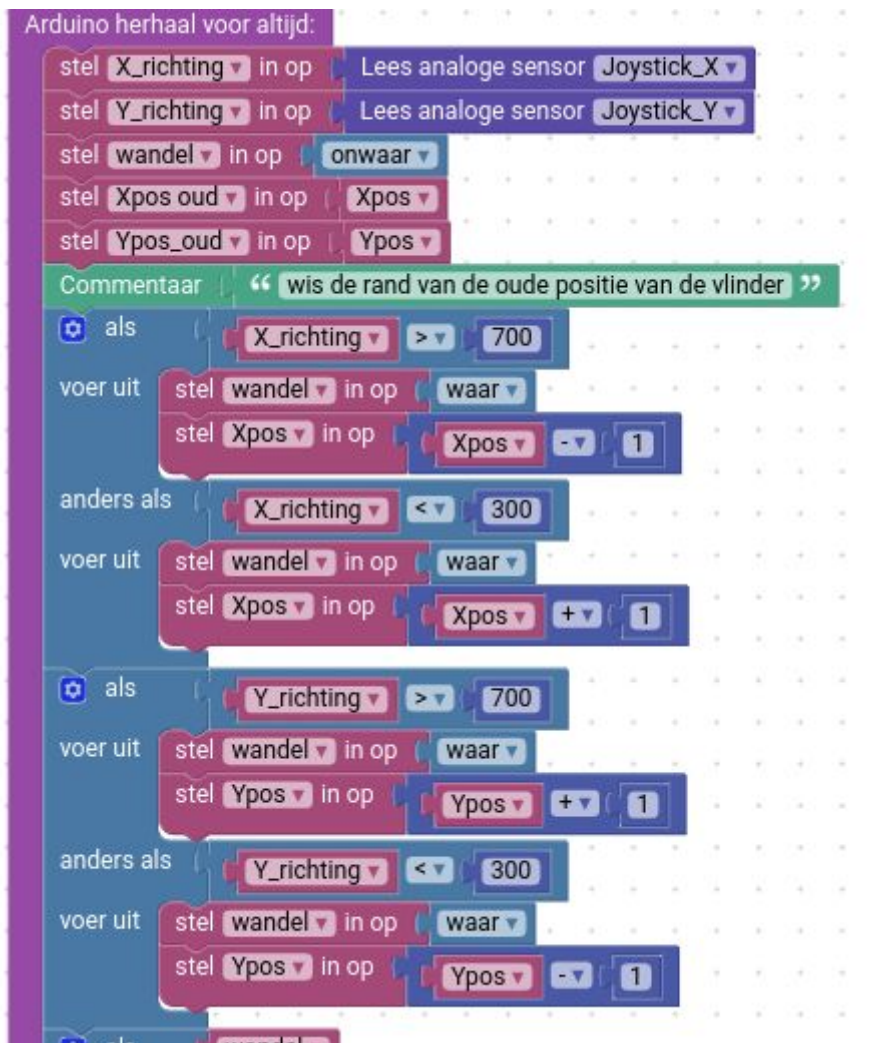
In de setup maken we het startscherm. Als we de joystick indrukken begint het spel:



Dit is het beginscherm. Daarna wachten we tot de joystick ingedrukt wordt en kan bewegen beginnen:



De sprite staat op het scherm nu, en we reageren op de joystick om rond te bewegen. We doen dit als volgt. Lees de waarde van de joystick in en laat de vlinder bewegen:



We hebben de oude positie opgeslagen, en weten wat de nieuwe positie is. We wissen de rand van de oude figuur voor we bewegen (Oefening: doe dat eens niet en kijk naar resultaat).



We mogen het scherm niet verlaten. Dus als we schermrand verlaten, corrigeren we:

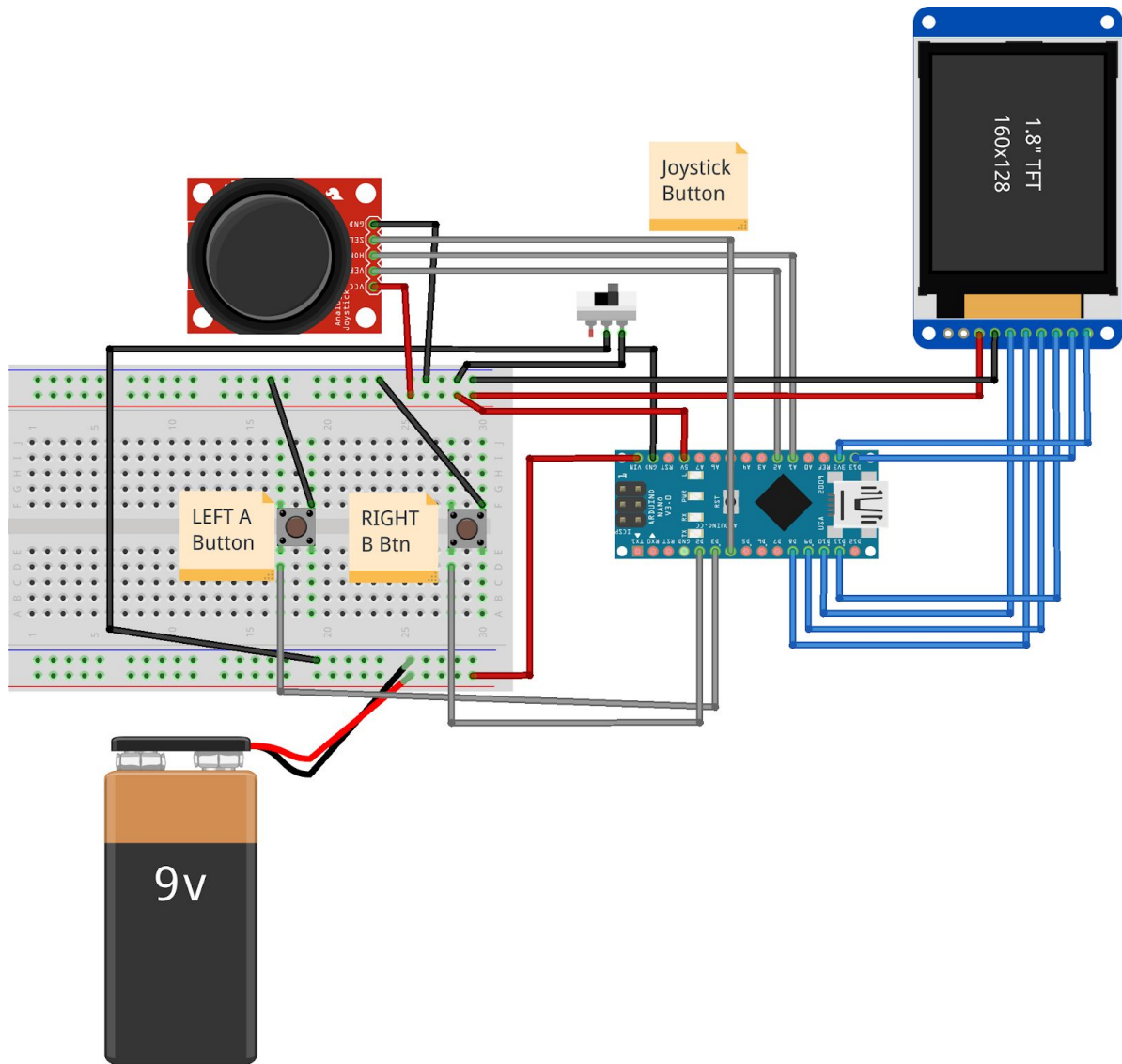


We eindigen met het tekenen van de vlinder op de nieuwe positie:



Retro Game: Spelen met Knoppen

We verbinden nu de twee drukknoppen en maken onze spelconsole met de LaserCutter. De A-knop naar GND en pin **D3**, de B-knop naar **D2**, de Joystick indrukken was naar **D4**. De volledige verbinding moet zijn:



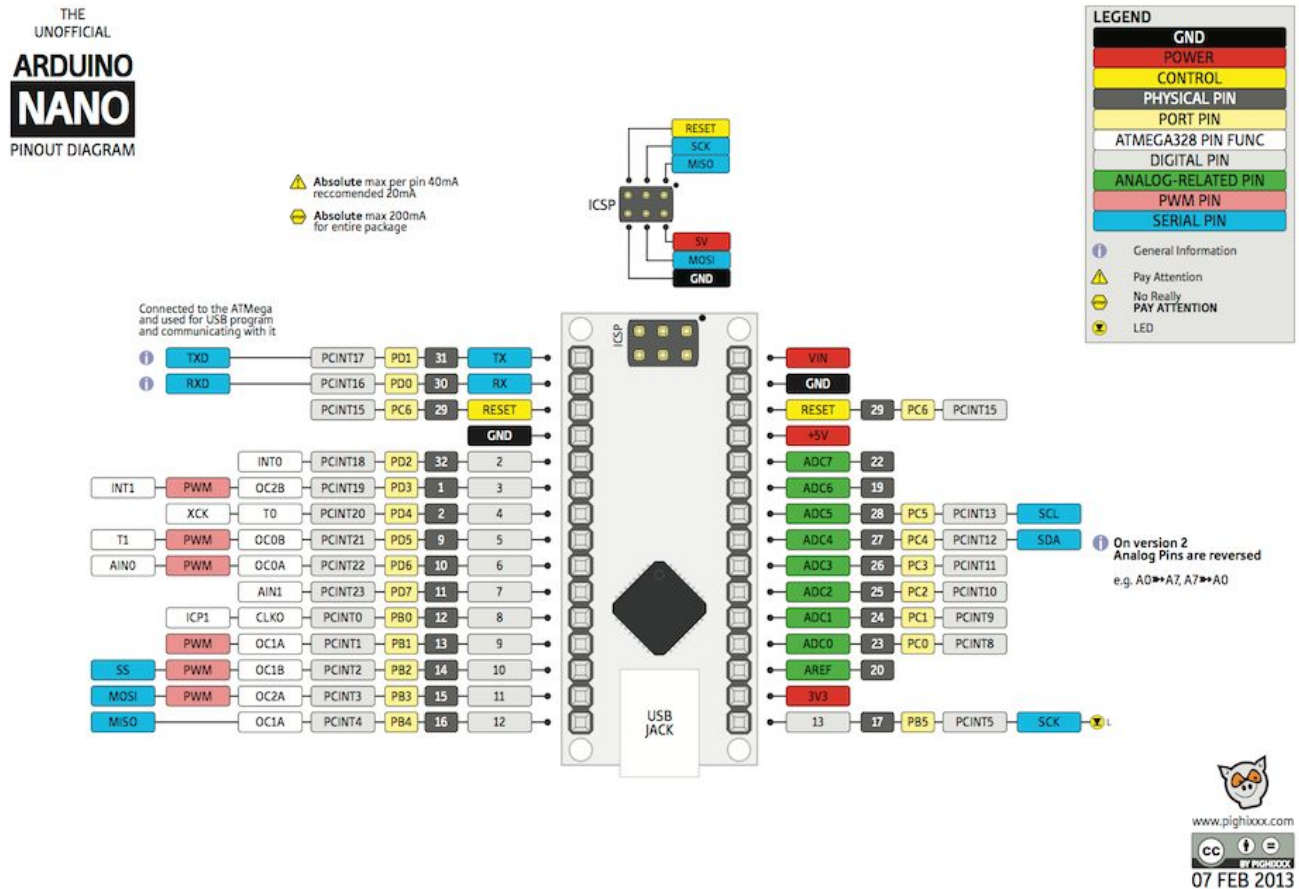
fritzing

We hebben enkele voorgemaakte spelletjes die je kunt opladen. Je vindt ze op <https://github.com/ingegno/RetroGames>

Maak het doosje om je spelconsole af te werken!

Appendix

De Arduino Nano heeft volgende pinnen beschikbaar:



Je ziet dat de MOSI D11 is (gaat naar SDA), en de SCK D13.

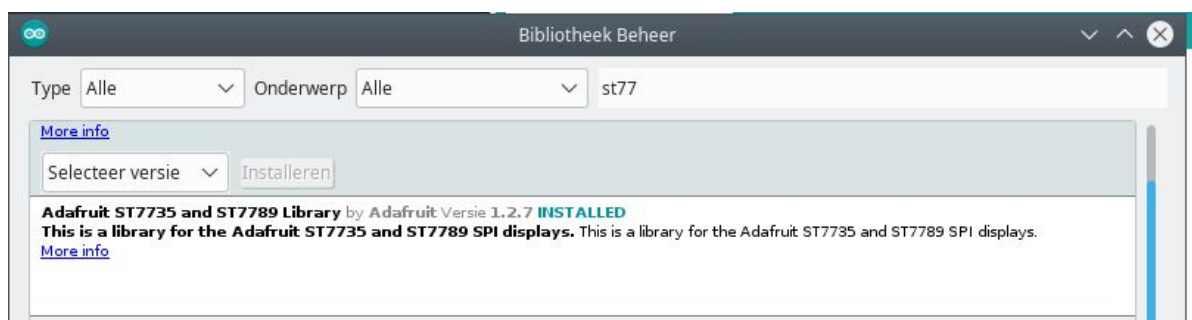
Software

Installeer de Arduino IDE vanuit <https://www.arduino.cc/en/Main/Software>

We gebruiken bibliotheken om het scherm aan te sturen:

- Adafruit TFT library
- Adafruit GFX: <https://github.com/adafruit/Adafruit-...> Leren ermee werken: <https://learn.adafruit.com/adafruit-gfx-graphics-library>
- Adafruit ST7735: <https://github.com/adafruit/Adafruit-...>
- TrueRandom bibliotheek: <https://github.com/sirleech/TrueRandom>

Installeer deze via *Schets->Bibliotheek gebruiken->Bibliotheken beheren*, en dan te zoeken op tft, gfx en st7735, en **Installeer** te drukken :



Voor Snake is de TrueRandom bibliotheek nodig, je moet de zip downloaden en als zip bibliotheek installeren, de zip vind je in <https://github.com/ingegno/RetroGames/tree/master/ArduinoSnake>.

Oefeningen en Games

De oefeningen kun je in xml vorm vinden op <https://github.com/bmcage/ardublockly/tree/blockly4arduino/blockly/blockly4Arduino/examples/TFT>

De games kun je downloaden van: <https://github.com/ingegno/RetroGames>
Klik rechtsboven op downloaden als zip, unzip, en open een van de directories in de Arduino IDE om een spel op te laden naar je Ingegno Retro Games spelconsole!

Referenties

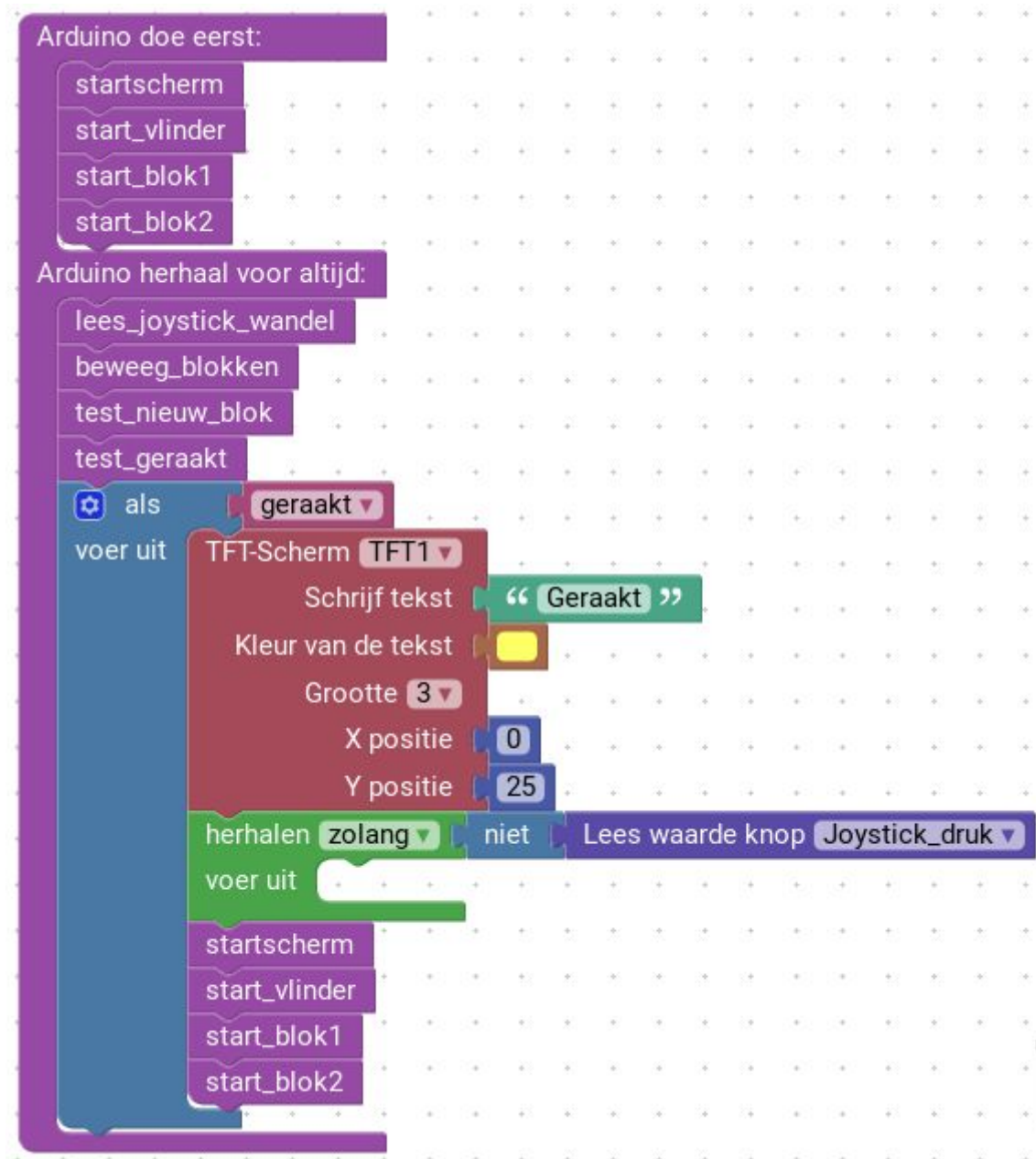
- <https://atmelcorporation.wordpress.com/...>
- <http://www.elec Freaks.com/wiki/...>
- <http://vilaca.eu/handheld-arduino-color-console/>
- <http://blog.simtronyx.de/en/...>

Retro Game: Race

We passen de bewegende sprite aan naar een racegame:

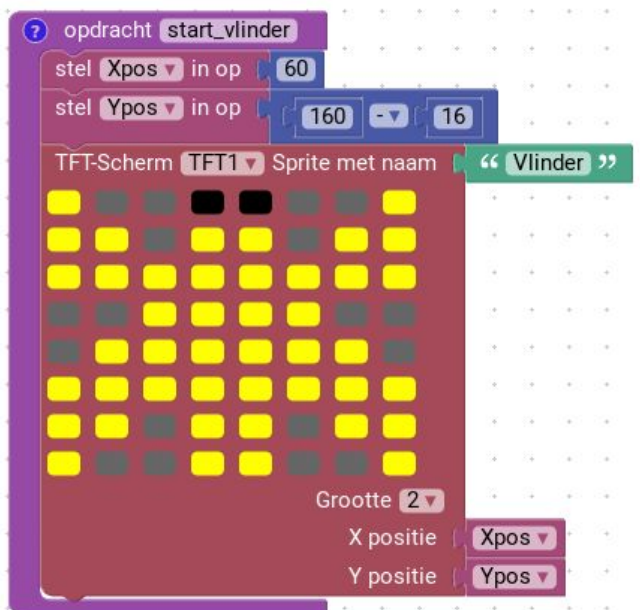
- De sprite kan enkel bewegen links en rechts onderaan scherm
- Van boven vallen blokken naar beneden die je moet ontwijken

De code heeft volgende vorm, maak functies om dit te maken:

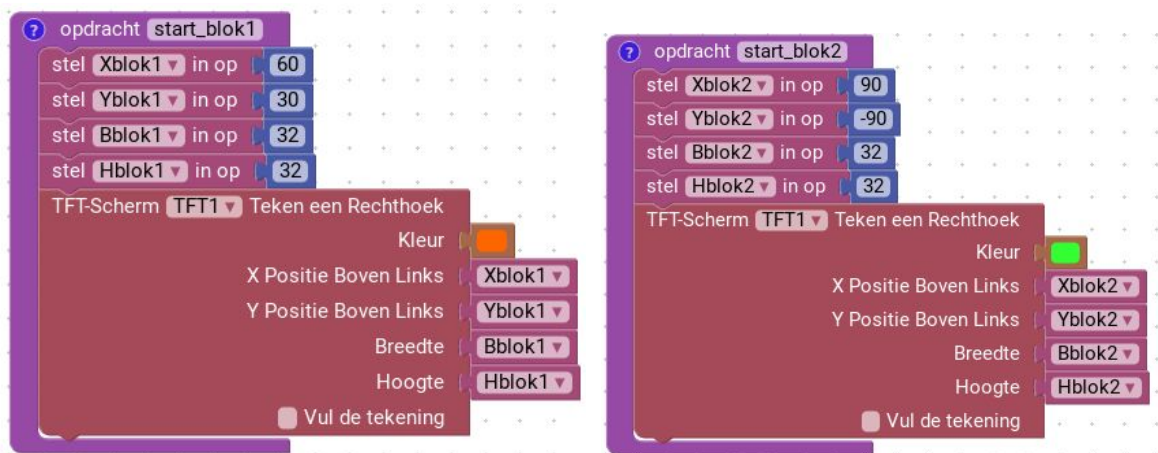


startscherm is een functie met je eerder gemaakte startscherm.

start_vlinder is een functie jouw sprite in de startpositie tekent:

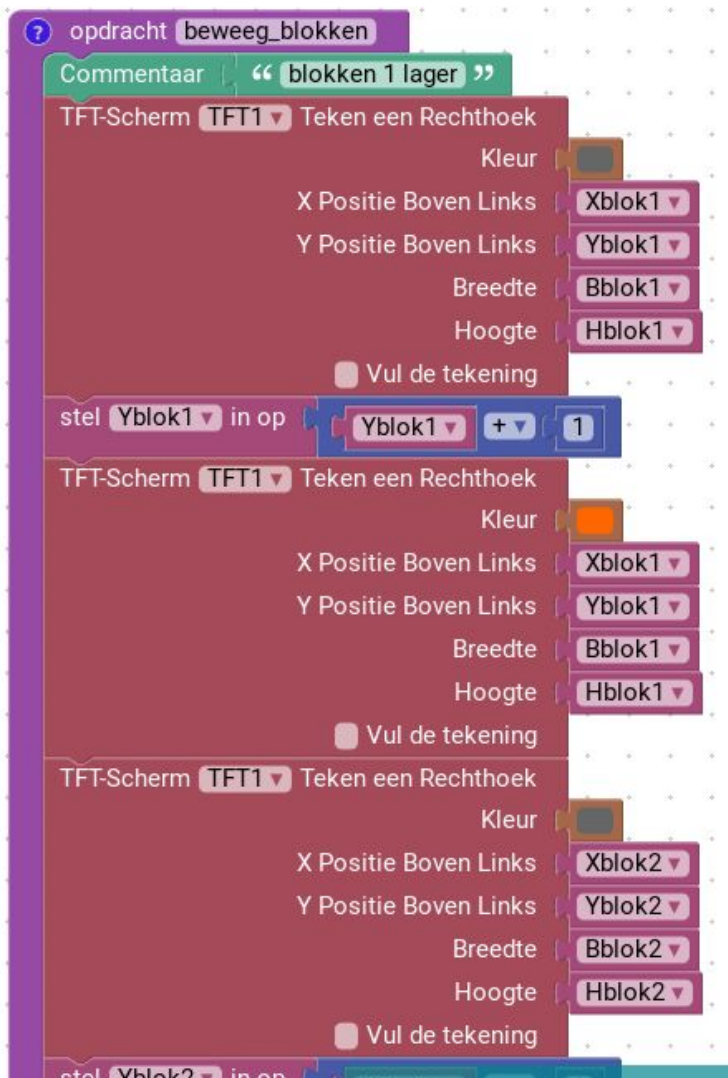


De start blokken zijn blokken die zullen vallen. We zorgen dat positie en breedte kan veranderen:

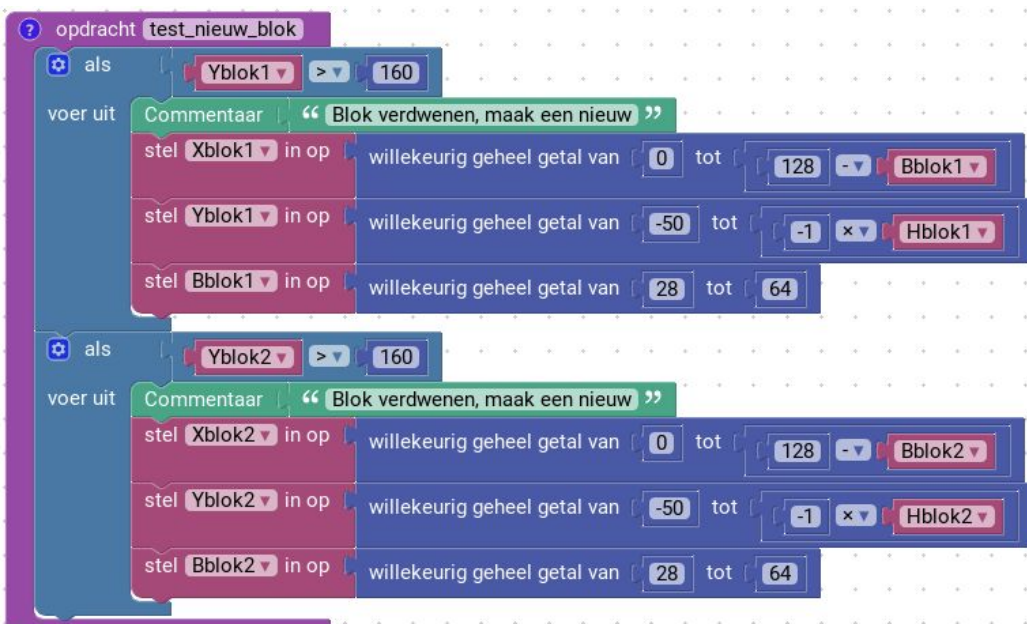


De functie **lees_joystick_wandel** zijn je blokken van eerder om de joystick te lezen. Wijzig dit zodat je sprite NIET op en neer kan bewegen. Wijzig dus Ypos **NIET**.

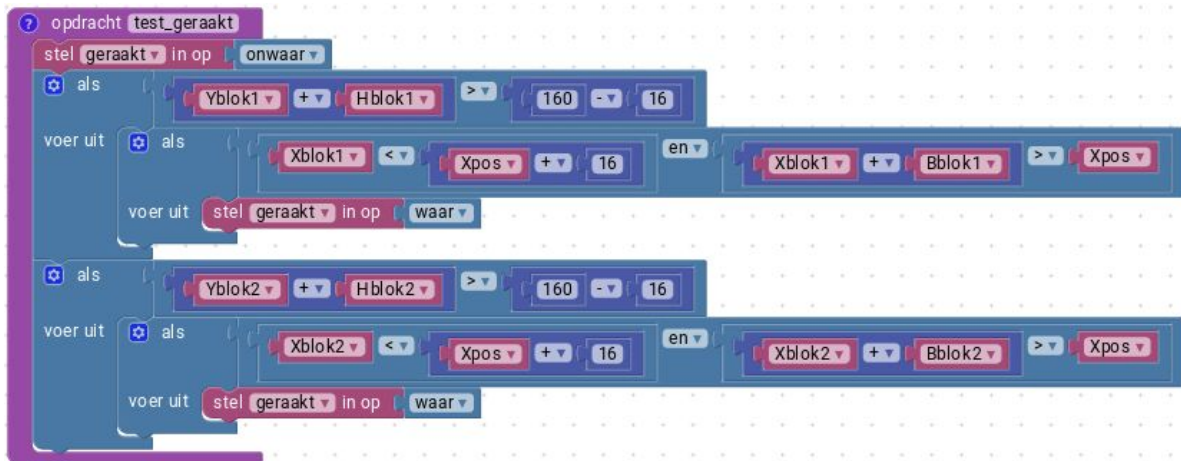
In functie **beweeg_blokken** laat je de blokken een pixel naar beneden bewegen (neem 2 pixels voor DUBBEL zo snel). Hier stuk voor eerste blok, voeg eronder nog beweging van tweede blok toe:



In **test_nieuw_blok** kijken we of een blok verdwijnt onderaan. Zodra dat het geval is maken we een nieuw blok bovenaan het scherm met een andere x-positie en hoogte. Voor variatie wijzigen we ook wat wanneer blok zal verschijnen (ypos):



Nu moeten we enkel nog weten wanneer het spel gedaan is. Dit is zo zodra blok je sprite raakt. Bekijk volgende code om dat te doen. Versta je het?



Proficiat, je hebt een spel gemaakt. Vroeg klaar? Verander misschien volgende:

- Is het spel nog speelbaar als je blokken 2 pixels per stap laat vallen? Dan gaan ze dubbel zo vlug als je sprite
- De hoogte van de blokken is vast nu. Maak ook dat variabel
- Kun je een derde blok toevoegen? Dan wordt het uitdagender.
- Laat een bol vallen in plaats van een rechthoek. Hoe weet je dan of je je vlinder raakt (moeilijke wiskunde nodig!)
- Te traag? Je kan spel versnellen door een sprite van 8x8 te nemen. Nog sneller als je je sprite maakt door horizontale lijnen te tekenen met twee kleuren (dus lijn per lijn), maar dat is moeilijker!