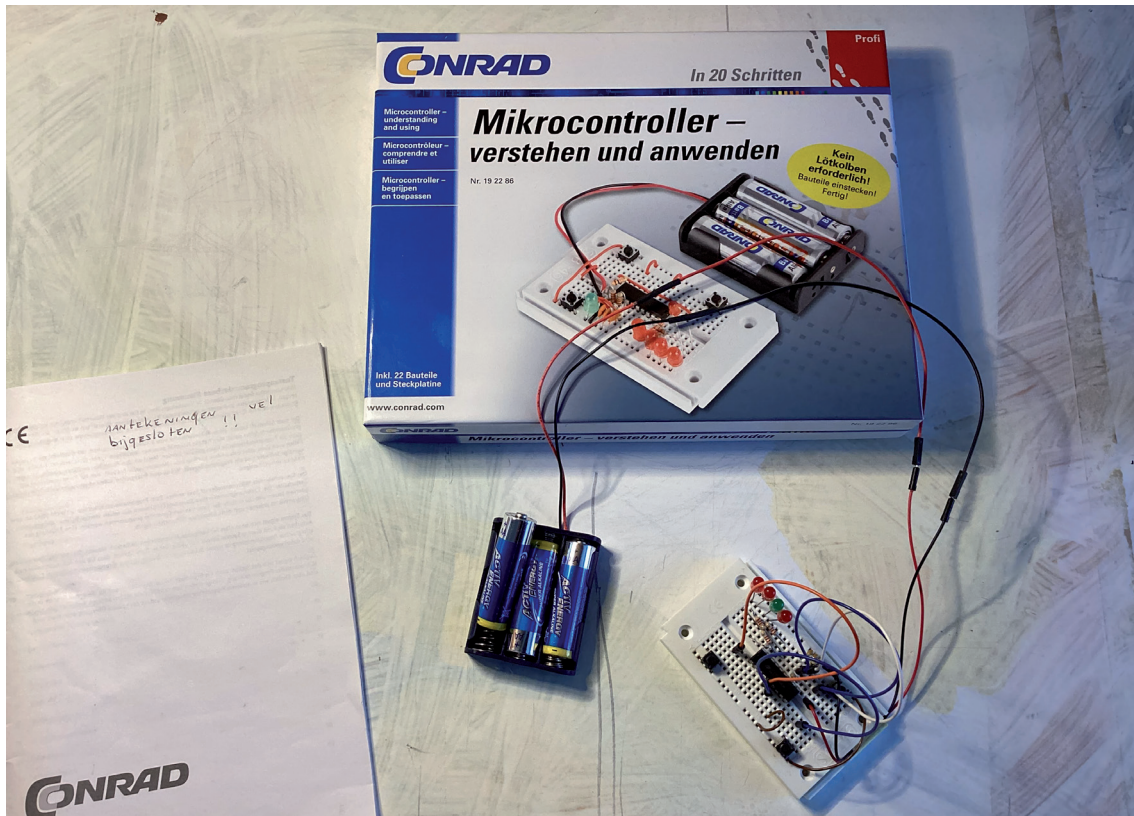


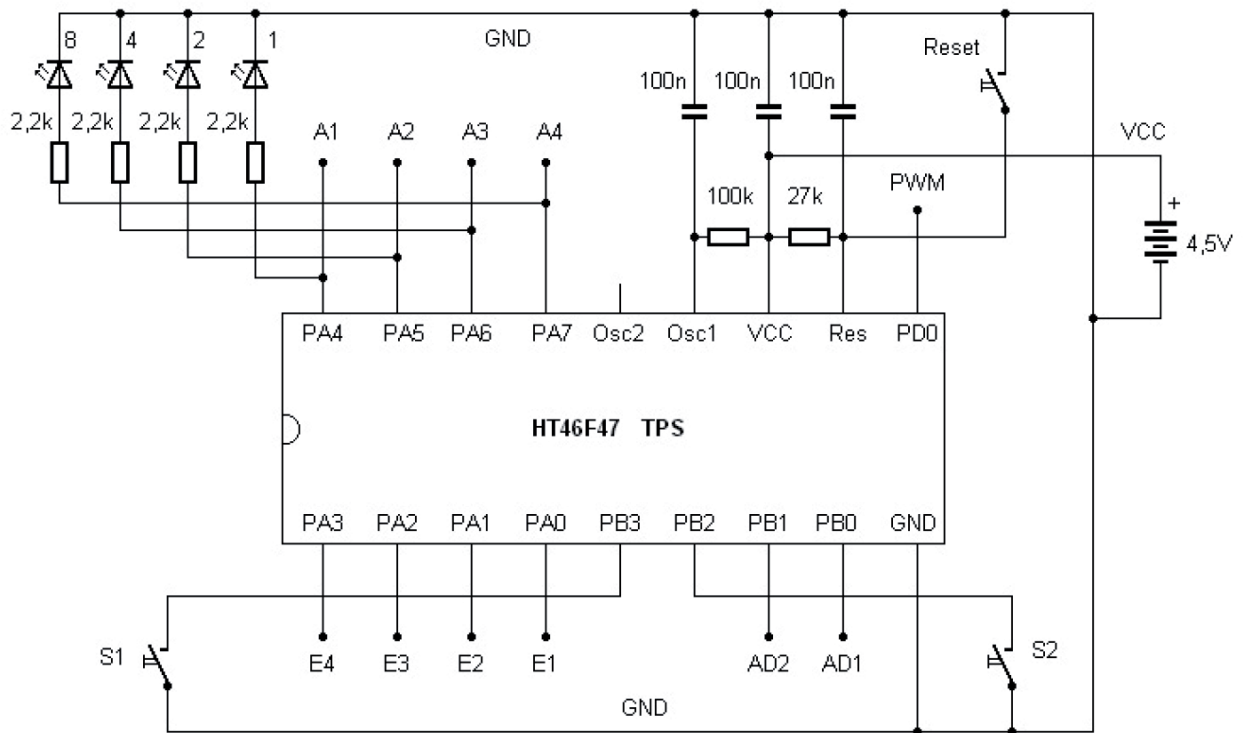
MIKROCONTROLLER verstehen und anwenden



In september 2021 heb ik deze set aangeschaft. Deze set was leerzaam en geeft inzicht in hoe een microcontroller met eeprom werkt. Hoe je manueel door de adressen kan bladeren en waarden kan opslaan. Het heeft mij geholpen om zonder IDE inzicht te krijgen in het 8 bit programmeren zonder PIC programmer software.

Als u toch nog een keer deze puzzle wilt gaan leggen:

- Breadboard SYB-46
- Batterijhouder 3*AA
- jumper kabels
- HT46F47 met TPS-firmware
- 3 drukknoppen 6x6 mm (! buig de pootjes recht voor plaatsing)
- 4 LED's 5mm, rood
- 1 LED 5mm, groen
- 1 fototransistor
- 3 keramische schijfcondensatoren 100nF
- 1 Elko 47uF
- 5 weerstanden 2,2 k Ω
- 1 weerstand 10 k Ω
- 1 weerstand 27 k Ω
- 2 weerstanden 100 k Ω



De basischakeling van het systeem.

Deze set is voor zover ik weet niet meer te koop maar omdat het inzicht geeft in de basis-principes van het programmeren blijft het een leuk projectje om eens mee te spelen. Vooral ook omdat het dwingt om op een analytische wijze en geconcentreerd de tabellen te lezen en toe te passen.

Opdracht:

Laat de LED op uitgang A1 10 seconden branden.

Begin met programmeren:-

1. druk op S2 + Reset.

Ik schrijf deze tutorial in mijn vrijetijd, eind mei 2026 staat hier de volledige tekst.

Terugschakelen naar de fabrieksinstellingen. Dit kan door 2 bytes FF in te voeren.

S2 + reset

S1 x16 (vier led's branden = Hex F)

S2

S1 x16 (vier led's branden = Hex F)

S2

S1 x16 (vier led's branden = Hex F)

S2

S1 x16 (vier led's branden = Hex F)

S2

reset

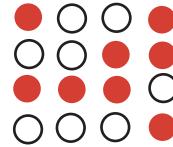
Nu start de controller weer op in de fabrieksinstellingen.

In de onderstaande schakeling bevinden zich 4 led's op een rij. Met deze led's kunnen we getallen vormen. We lezen de rij led's van rechts naar links.

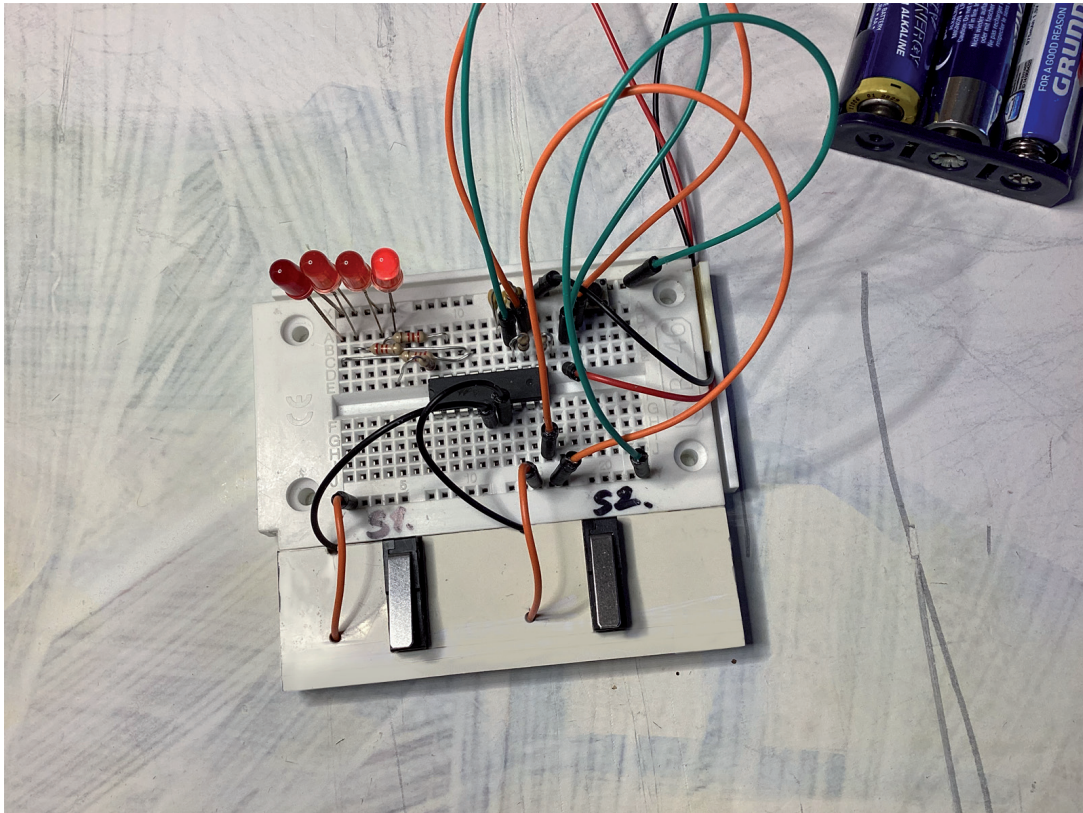
$$2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

De led's vertegenwoordigen opeenvolgend een exponentiële waarde van 2.

Branden de led's 1001 dan heeft de code de waarde 9.
 Branden de led's 0011 dan heeft de code de waarde 3.
 Branden de led's 1110 dan heeft de code de waarde 14.
 Branden de led's 0001 dan heeft de code de waarde 1.



Op deze manier kunnen we met de reeks dus tellen tot 15. Dan branden alle led's.



Aantekeningen:

- + Bij een reset (s2 + reset) moet u schakelaar s2 twee seconden langer vasthouden zodat led 1 gaat branden.
- + Als de schakeling “raar” gaat doen na een reset, sluit de schakeling dan voor een kwartier af. Dit geeft de interne condensator de tijd om volledig te ontladen.

Voorbeeld 1

Laat led 1 branden.

1. s2 + reset
2. 2 x s1 (blader naar **adres 1**)
3. s2 (vastleggen)
4. 2 x s1 (led 1 brandt)
5. s2 (vastleggen)
6. **adres 2**
7. wachten = 0010 instructie (0001= branden | 0010 = wachten | 0011 = terug)
8. s2 (vastleggen)
9. s1 waarde = 0000
10. s2 (vastleggen)
11. reset

Opmerking:

- + Hierboven een voorbeeld van een programma dat zonder **assembler** rechtstreeks in de chip wordt geprogrammeerd. Zo ging het dus omstreeks 1950 toen men nog rechtstreeks programmeerde in **machinaal**.
- + Tegenwoordig schakelen we in nano-seconden tussen **adressen**.

Voorbeeld 2

Laat led 1, 15 seconden branden.

1. s2 + reset
2. 2 x s1 (blader naar **adres 1**)
3. 0001 (instructie = branden)
4. s2
5. 0001 (waarde = led 1)
6. s2 (adres 2 licht kort op | **adres 2**)
7. 0010 (instructie = wachten)
8. s2
9. 1100 (waarde = 15 seconden)
10. s2 (adres 3 licht kort op | **adres 3**)
11. 0000 (instructie = geen)
12. s2
13. 0000 (waarde = 0)
14. reset.

Opmerking:

- + Met s1 voer je een waarde in.
- + Met s2 blader je tussen adressen, voer je de instructie in en geef je het een waarde.

Toets s2 het adres knippert **zeer** kort.

Voer instructie in + s2 | nibble 1

Voer waarde in + s2 | nibble 2, het **volgende** adres knippert kort en is **actief**.

Voer instructie in + s2 | nibble 1

Voer waarde in + s2 | nibble 2 enz.

Voorbeeld 3

Voer een waarde voor a? in, tel hier 1 bij op en toon dit binair. (a=a+1)

1. s2 + reset
2. 0100 + s2 (instructie a=?)
3. 0011 + s2 voer waarde voor a in = 3
4. 0111 + s2 instructie (a=a+1)
5. 0001 + s2
6. 0010 + s2 wachten
7. 0110 + s2 waarde = 100 ms
8. 0101 + s2 instructie (port = a)
9. 0100 + s2 toon de waarde (port = a)
10. reset (toont de waarde a+1 = 4)

HEX = 43 71 26 54

Opmerking:

- + Hierboven staat een voorbeeld van hoe je gebruik kan maken van besluitstructuren (**digital logical gates**) die ingebouwd zijn in de chip.