

Zelf PIC's programmeren

Deel-6 Analooq inlezen

Digitale signalen zijn of laag (0 Volt) of hoog (5 Volt).

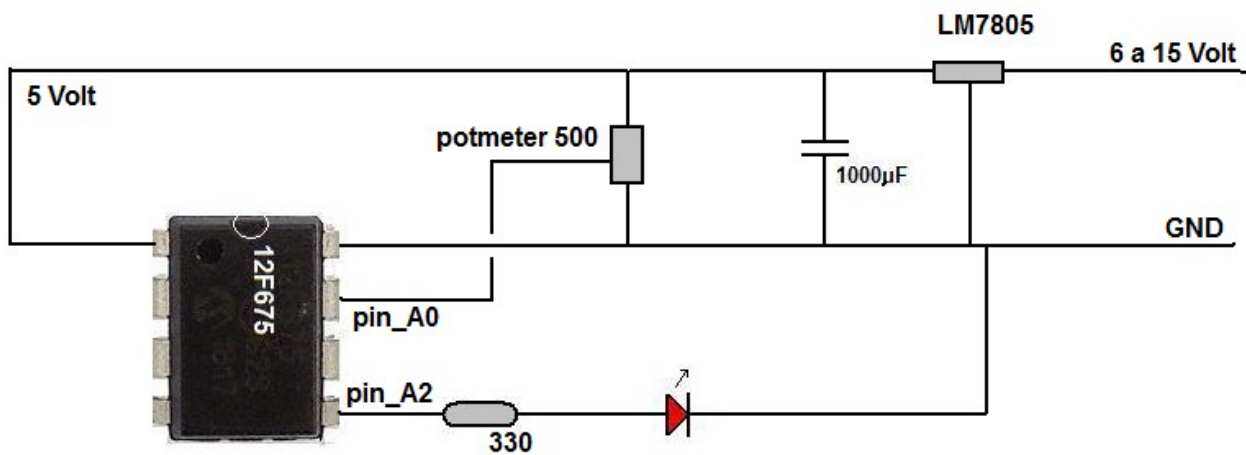
Analooqe signalen kunnen alle waarde tussen 0 en 5 Volt hebben.

Deze signalen kun je ook inlezen met b.v. De 12F675.

Het analooqe signaal wordt omgezet in een waarde tussen 0 en 255.

Zo geeft:

- 0 volt ==> 0
- 1 volt ==> 51
- 2,5 volt ==> 128
- 4 volt ==> 204
- 5 volt ==> 255



Met de potmeter van 500 Ω wordt het analooqe signaal gemaakt, deze komt op pin_A0. De rode LED wordt gebruikt als indicatie lampje.

De 12F675 kan alleen op pin-A0, pin_A1 en/of pin_A2 een analooqe signaal inlezen.

De JALEdit code

```
include 12f675

pragma target clock 4_000_000           -- oscillator frequency
pragma target OSC INTOSC_NOCLKOUT      -- Internal oscillator
pragma target WDT disabled             -- no watchdog
pragma target MCLR internal            -- make MCLR pin available I/O
pragma target BROWNOUT disabled        -- no brownout

enable_digital_io()                   -- disable analog I/O (if any)

pin_A2_direction = Output
alias LED is pin_A2

const byte ADC_NVREF = ADC_NO_EXT_VREF
const word ADC_RSOURCE = 5_000        -- maximum source resistance = 5 kOhm
include adc
adc_init()

var Byte potmeter                      -- lage resolutie 0 - 255

forever loop

potmeter = ADC_read_low_res(0)         -- inlezen tussen 0-255

If potmeter>128 then LED=high else LED=low end if -- schakelt indicatie LED

end loop
```

Met deze code krijgt de potmeter een waarde tussen 0 en 255 in JALEdit.
Komt de potmeter boven de 128 dan gaat de indicatie LED branden.

Het ingelezen signaal heeft een resolutie van 0 t/m 255, wil je een hoge resolutie hebben dan moet je de code **ADC_read_high_res(0)** gebruiken. Deze heeft een resolutie van 0 t/m 1023. Ook moet je **var Byte potmeter** veranderen in **var WORD potmeter**

De totale code wordt dan voor een hoge resolutie:

```

include 12f675                                -- target PICmicro

pragma target clock 4_000_000                 -- oscillator frequency

pragma target OSC INTOSC_NOCLKOUT            -- Internal oscillator
pragma target WDT disabled                    -- no watchdog
pragma target MCLR internal                  -- make MCLR pin available I/O
pragma target BROWNOUT disabled              -- no brownout
--
enable_digital_io()                           -- disable analog I/O (if any)

pin_A2_direction = Output
alias LED is pin_A2

const byte ADC_NVREF = ADC_NO_EXT_VREF
const word ADC_RSOURCE = 5_000                -- maximum source resistance (5 kOhm)
include adc
adc_init()

var WORD potmeter                             -- voor hoge resolutie

forever loop

potmeter = adc_read_high_res(0)               -- inlezen tussen 0- 1023

If potmeter>512 then LED=high else LED=low end if

end loop

```