

Zelf PIC's programmeren

Deel-17 Master en Slave

Je kan met meerdere PIC's werken in een systeem.

Een z'n systeem is een master & slave (meester en knecht).

De opbouw die vaak wordt gekozen werkt met 1 master en een of meerdere slaves.

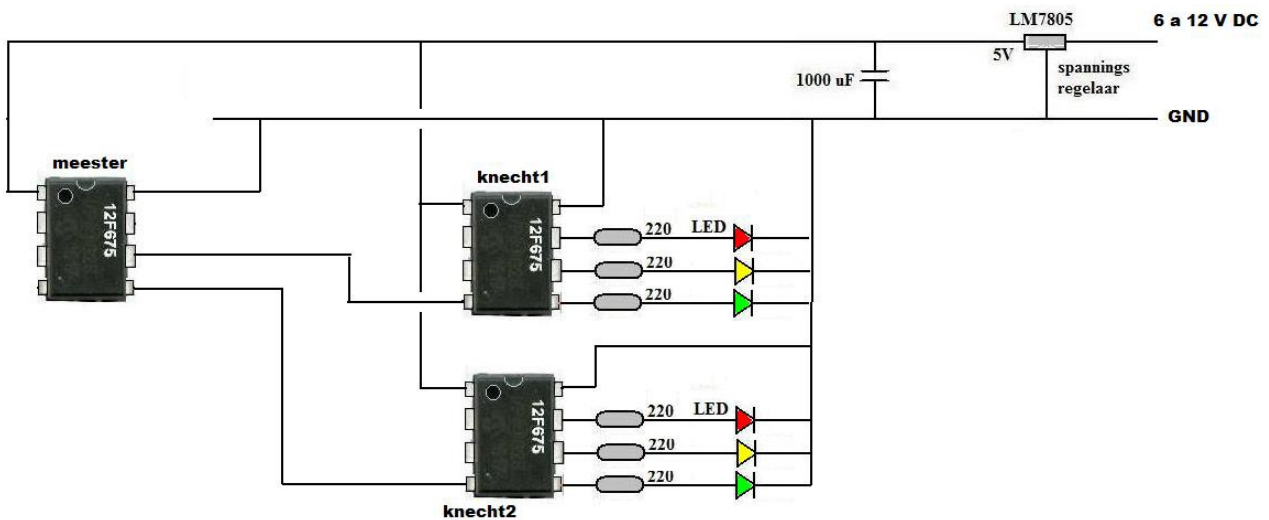
De master heeft gewoonlijk de regie in handen en de slaves voeren elk hun een eigen taak uit.

De slaves wachten op het startcommando van de master en gaan als ze deze krijgen aan het werk met hun eigen taak.

Als voorbeeld de aansturing van twee stoplichten op een kruispunt.

De master bepaalt wanneer een stoplicht (een slave) op groen mag.

De slave zal hierna zijn eigen programma aflopen door het licht op groen te zetten en daarna via geel weer naar rood. Omdat de tijd van groen via geel naar rood een vaste tijd is hoeft je hier geen terugmelding te hebben naar de master.



De meester geeft het signaal door naar de knecht door heel even (1 seconde) zijn uitgang naar die knecht hoog te zetten.

Je hebt hier te maken met twee JALedit programma's, een voor de master en een voor elke slave.



De code voor de meester.

```
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)

Include delay                                 -- het laden van de delay's

pin_A1_direction = Output    alias Mast1 is pin_A1    -- Mast1 (slave 1)
pin_A2_directet = Output    alias Mast2 is pin_A2    -- Mast2 (slave 2)

Forever loop

Mast1=high      Delay_1s(1)
Mast1=low       Delay_1s(12)

Mast2=high      Delay_1s(1)
Mast2=low       Delay_1s(12)

End loop
```

De code voor de slave.

```
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)

include delay                                  -- het laden van de delay's

pin_A0_direction = Output    alias Rood is pin_A0    -- Rood
pin_A1_direction = Output    alias Geel is pin_A1    -- Geel
pin_A2_direction = Output    alias Groen is pin_A2   -- Groen
pin_A3_direction = Input     alias Start is pin_A3   -- Start

Forever loop
Repeat Until Start==high

Rood=low  Groen=high  Delay_1s(10)
Groen=low Geel=high  Delay_1s(2)
Geel=low  Rood=high

End loop
```