



Udvidelse af pin-antal på Arduino Uno

Der findes et hav af IC'er, der kan bruges i forbindelse med Arduino, - og jo også i forbindelse med andre processorer.

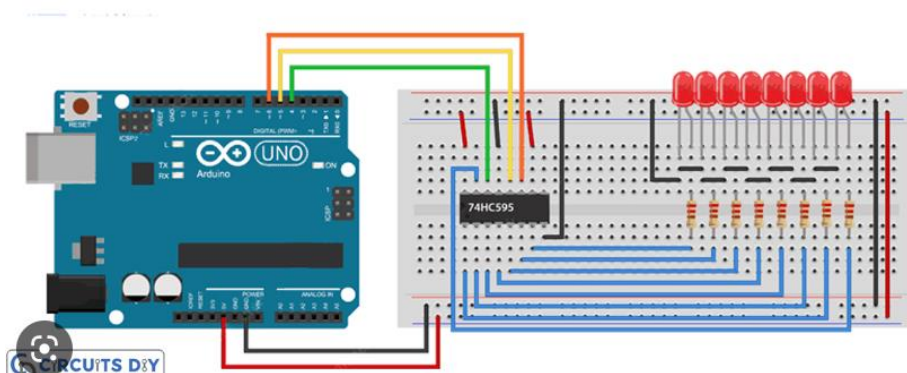
Her ses på 2 IC'er, der kan udvide antallet af I/O.

Først ses på en 74HC595, der kan bruges til at skabe ekstra outputs.

Den skal kontrolleres, og det stjæler 3 pins. Men vi får 8 nye outputs, dvs. i alt 5 nye.

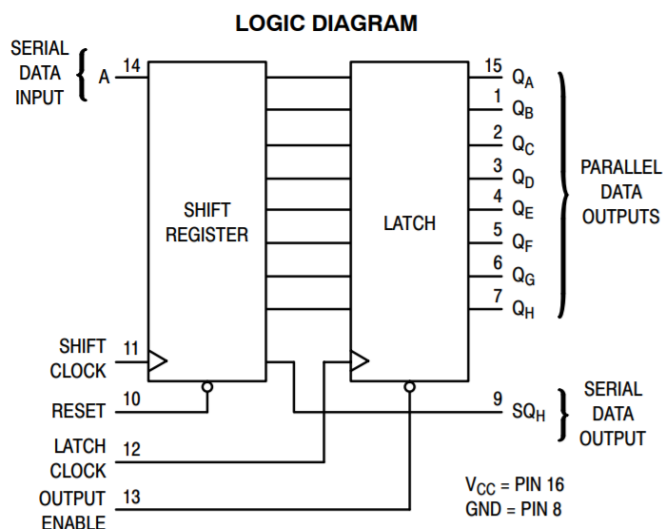
Men sættes 2 stk. 74HC595 i serie, fås i alt $5 + 8 = 13$ ekstra Outputs.

Først ses på 1 stk.



PINS 1-7, 15	Q0 " Q7	Output Pins
PIN 8	GND	Ground, Vss
PIN 9	Q7"	Serial Out
PIN 10	MR	Master Reclear, active low
PIN 11	SH_CP	Shift register clock pin
PIN 12	ST_CP	Storage register clock pin (latch pin)
PIN 13	OE	Output enable, active low
PIN 14	DS	Serial data input
PIN 16	Vcc	Positive supply voltage

PIN ASSIGNMENT		
Q _B	1 •	16 V _{CC}
Q _C	2	15 Q _A
Q _D	3	14 A
Q _E	4	13 OUTPUT ENABLE
Q _F	5	12 LATCH CLOCK
Q _G	6	11 SHIFT CLOCK
Q _H	7	10 RESET
GND	8	9 SQ _H



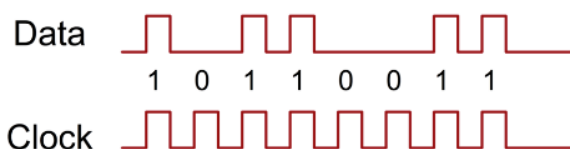
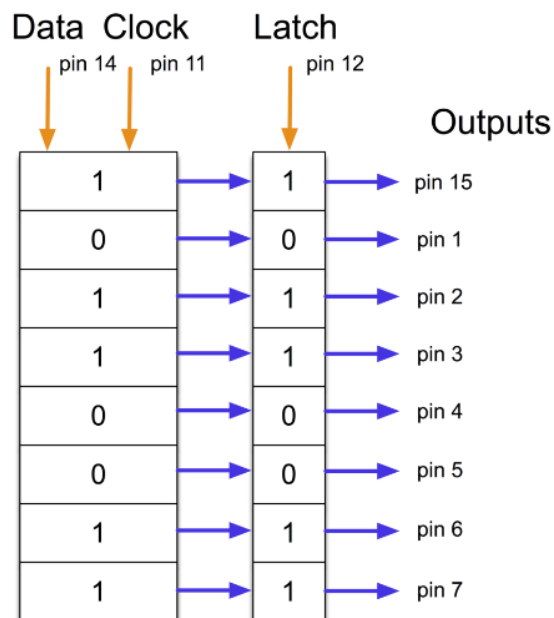
I IC-en er der et skifteregister. Derfor skal de styres med en clock-signal, der skal fortælle den, hvornår den skal tage data-signalet ind.

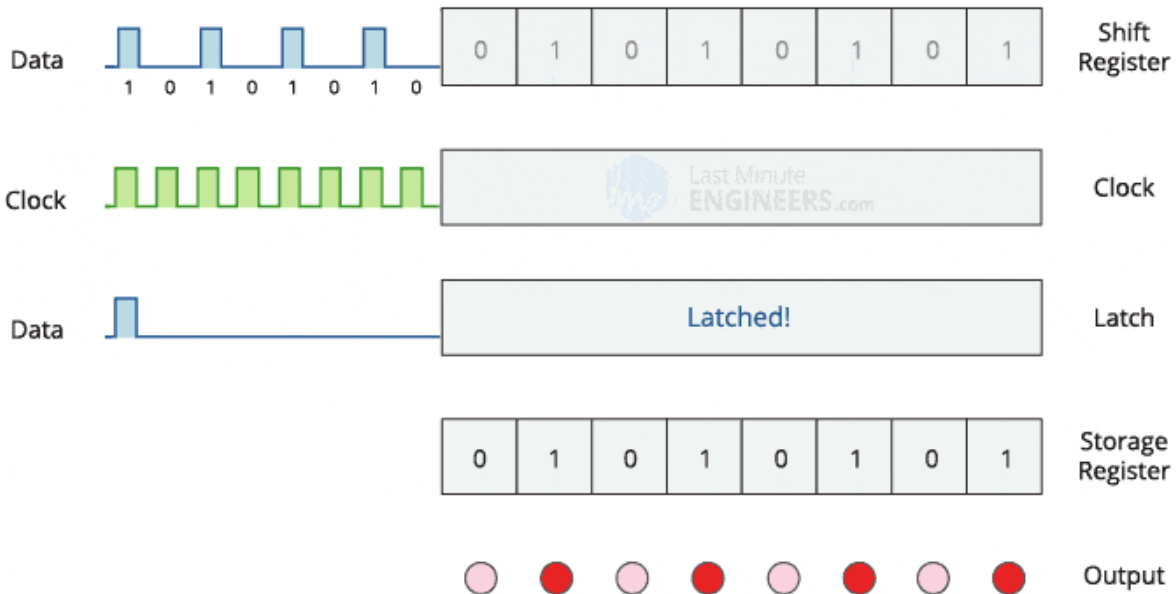
Dvs. processoren skal sætte dataledningen høj eller lav, og så skabe en kort clock-puls.

Dette gentages 8 gange.

Til slut skal dataene overføres fra skifteregistret til output-benene. Det sker gennem en latch, der blot skal ha en clock-puls.

Dette er animeret herunder !

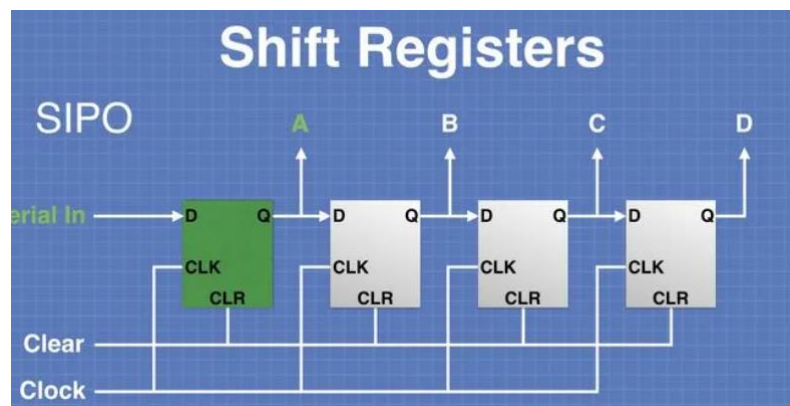




Fra: https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/?utm_content=cmp-true

Der findes forskellige former for skifteregistre. Kan være opbygget af D-FF.

SiPo, (74HC595)
PiSo, (74HC165)
PiPo, (= Latch)
SiSo ?



I første øvelse sættes lysdioder på de 8 udgange !!

Kodeeksempel:

```
/*  
Udvidelse af Pin-antal med 74HC595  
Valle, 25/02-2023  
*/
```

```
int latchPin = 5;    // Latch pin of 74HC595 is connected to Digital pin 5  
int clockPin = 6;    // Clock pin of 74HC595 is connected to Digital pin 6  
int dataPin = 4;      // Data pin of 74HC595 is connected to Digital pin 4
```



```
byte leds = 0;           // Variable to hold the pattern of which LEDs are currently turned on or off

void setup() {
  // Set all the pins of 74HC595 as OUTPUT
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
} // EndSetup

void loop() {
  leds = 0;              // Initially turns all the LEDs off, by giving the variable 'leds' the value 0
  delay(500);

  for (int i = 0; i < 8; i++){          // Turn all the LEDs ON one by one.
    bitSet(leds, i);                    // Set the bit that controls that LED in the variable 'leds'
    send_to_595();
    delay(500);
  }
} // EndLoop

// Min Shift-out-routine !!
void send_to_595(){
  for (byte i = 0; i < 8; i++)
  {
    digitalWrite(dataPin, bitRead(leds,i)); // omvendt rækkefølge: bitRead(x, 7-i);
    digitalWrite(clockPin, 1);
    digitalWrite(clockPin, 0);
  }
  digitalWrite(latchPin, 1); // Latch serielle data til output-pins
  digitalWrite(latchPin, 0);
} // End_Send
```

Fra <https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/?utm_content=cmp-true>

Nu ændres koden, så der sendes 8 bit binær kode !!

Manuel shiftout: <https://forum.arduino.cc/t/using-forloop-and-byte-arrays-for-cycling-through-595-shift-register-outputs/537339/7>

Sourcekode for shiftOut:

```
// hvordan ser funktionen shiftOut ud ???

void shiftOut_(uint8_t dataPin, uint8_t clockPin, uint8_t bitOrder, uint8_t val)
{
  uint8_t i;

  for (i = 0; i < 8; i++) {
    if (bitOrder == LSBFIRST)
      digitalWrite(dataPin, !(val & (1 << i)));
    else
      digitalWrite(dataPin, !(val & (1 << (7 - i))));

    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
  }
}
```

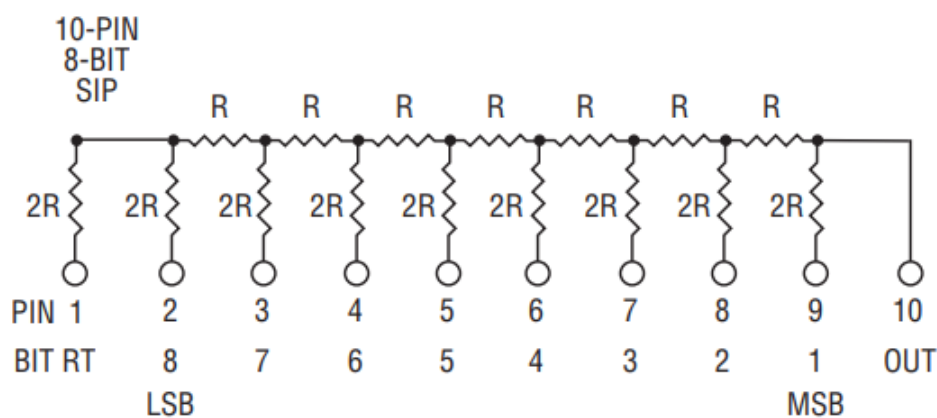
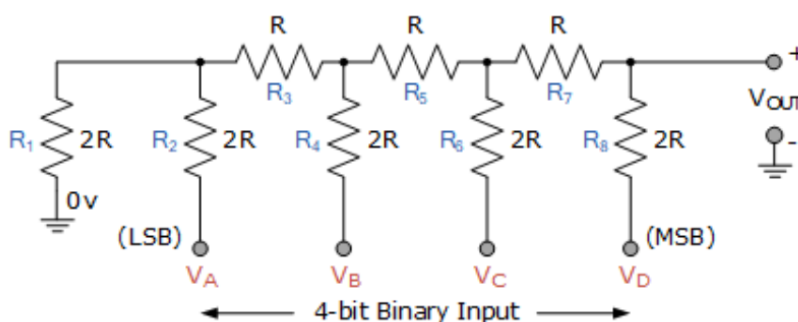


LSBFIRST og MSBFIRST er definerede værdier. Værdien 0 eller 1

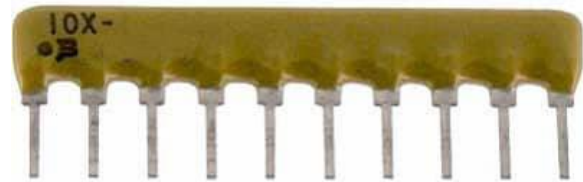
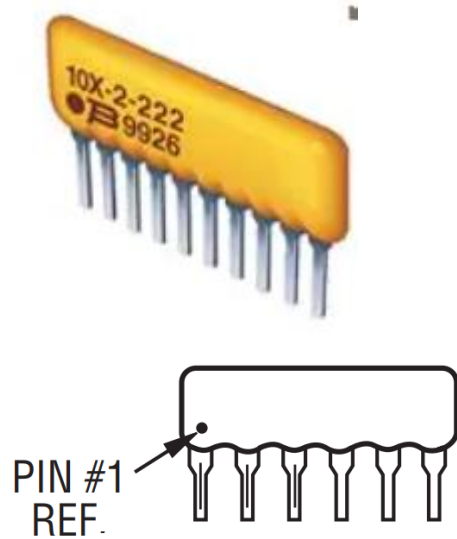
R2R-netværk, 8 bit:

Endelig monteres et R2R-netværk på de 8 udgange.

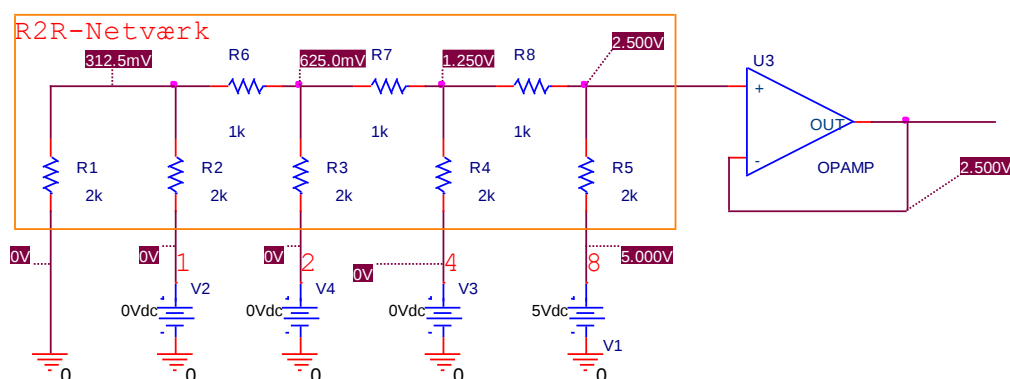
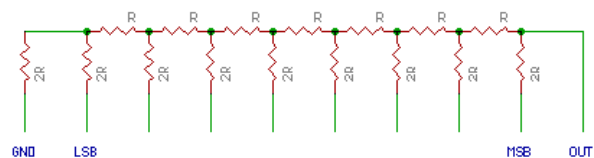
Vi skal bruge et 8-bit R2R-netværk !



Pin 1 er mærket med en sort markering.



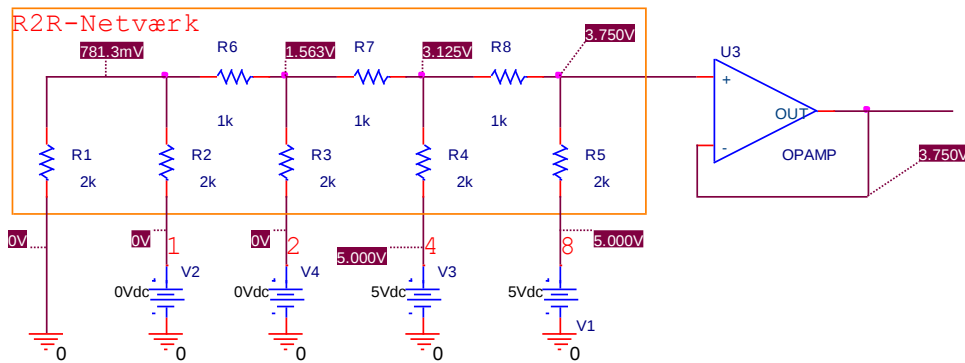
Kan bruges til D-til A-konvertering.



Et par eksempler på, at der er sat 5 Volt på et R2R-netværk !!

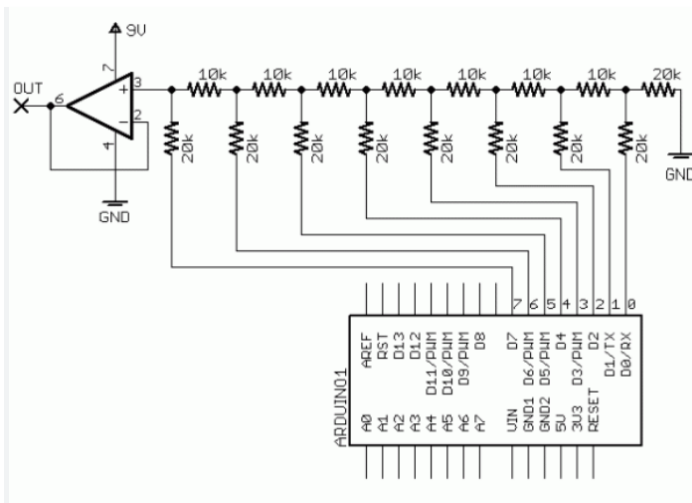
Simuleret med ORCAD.

Værdien er 1000



Dette eksempel viser værdien 1100

Se: <https://www.electronics-tutorials.ws/combination/r-2r-dac.html>



Hvad er en OP-AMP: ?? LM358

Hvis der ikke trækkes strøm ud af R2R-netværket, behøves ikke at sætte en buffer på. (En forstærker, der forstærker 1 gang)

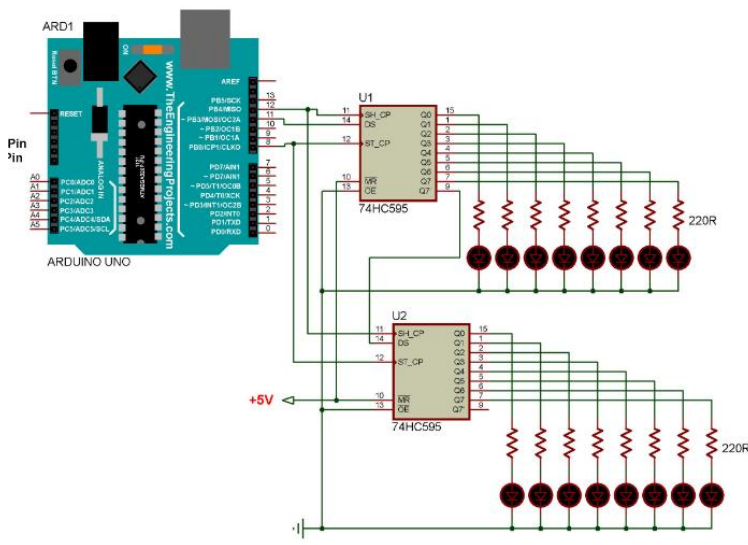
Vis udgangsspændingen med et scoop !!

Kædeforbindelse, "Daisy Chain"



Ordet ” Daisy Chain ” kommer en kæde, - eller krans af tusindfryd.

Serieforbindelse:

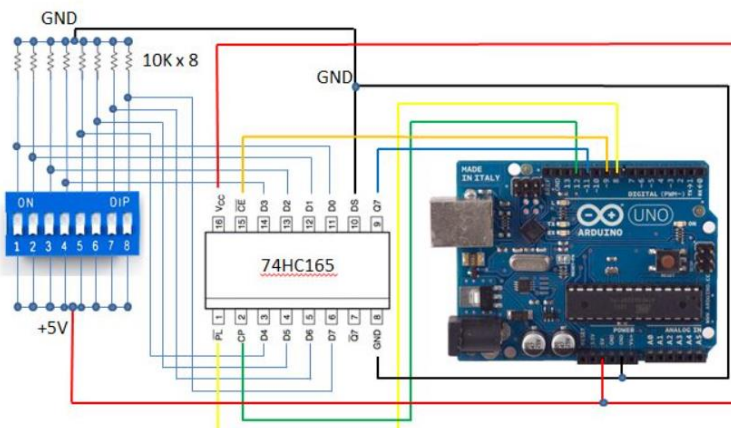


Samme antal pins på Arduino bruges, men det giver nu 8 ekstra outputs. – Og der kan forlænges i det uendelige !

Udvidelse af Arduino med flere input

Skal der laves flere indput til Unoen, kan man bruge IC-en 74HC165. Det er også et skifteregister, men en PiSo. Dvs. Parallel ind / Seriel ud.

Data loades ind fra 165's inputs, D0 til D7,



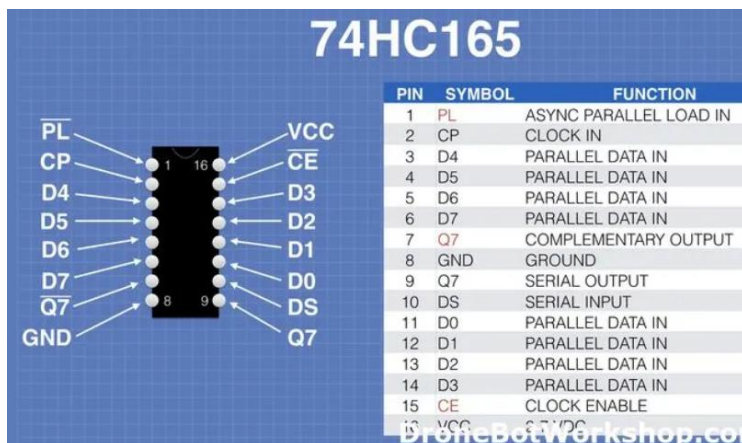
74HC165 skal ha at vide, hvornår den skal latche / låse de 8 input (og enable shifting).

Dernæst skal der hentes 8 bit ud af dens skifteregister. (Data og Clk.)

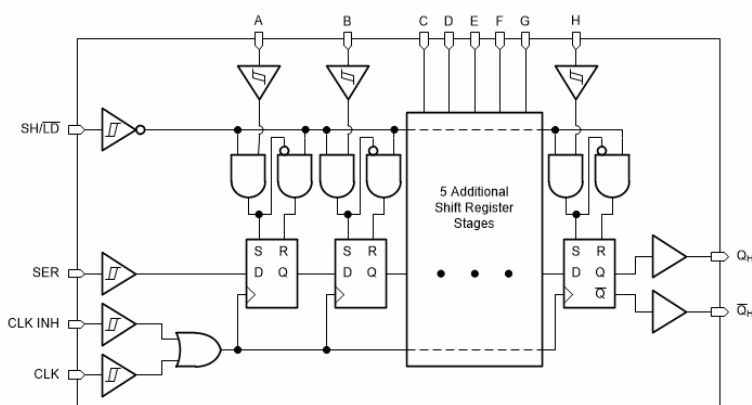
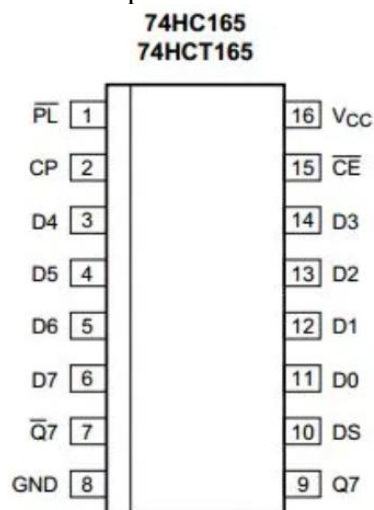
Kan også kædes (daisy chain)!!

Pin 15 er en Chip Enable input. Den kan bare stelles !!

```
digitalWrite(ISRLatchPin, HIGH); // disable input  
latching and enable shifting
```



Her er et pin-skema !!



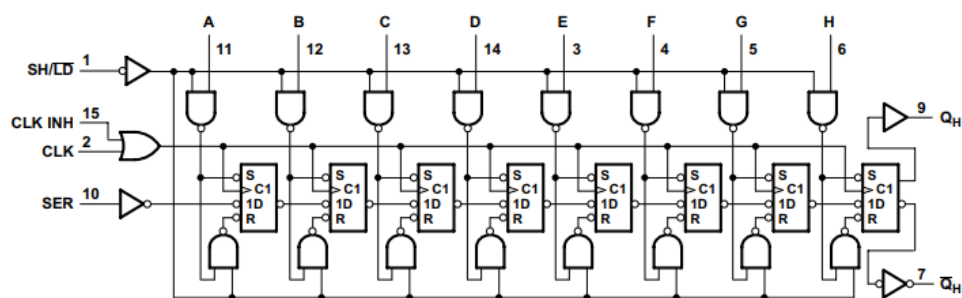
I venstre side:

SH/LD. Parallel Load, aktiv lav.
SER: Serial ind ved Daisy Chain.
ellers groundes den

Clock Inhibit: Kan bare stilles
CLK: Clockpulser

Data hentes fra Qh. Eller inverteret
fra /Qh

Her er indmaden
vist, med
pinnumre.



Og nogle andre eksempler.

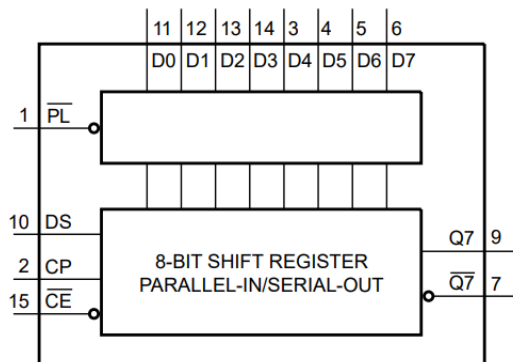


Table 2. Pin description

Symbol	Pin	Description
$\overline{PL}$	1	asynchronous parallel load input (active LOW)
CP	2	clock input (LOW-to-HIGH edge-triggered)
$\overline{Q7}$	7	complementary output from the last stage
GND	8	ground (0 V)
Q7	9	serial output from the last stage
DS	10	serial data input
D0 to D7	11, 12, 13, 14, 3, 4, 5, 6	parallel data inputs (also referred to as Dn)
$\overline{CE}$	15	clock enable input (active LOW)
V _{CC}	16	positive supply voltage

Kodeeksempel:

```
/*
  Demo af 74HC165 Skifte Register
  Læs state fra 8 switche
  / Valle, 25/02-2023

*/

/*
  Def Connections to 74HC165
  Pin 10 ( seriel in) permanent lav. Bruges ved Daisy Chain
  Pin 15 ( clock Enable ) kan bare være permanent lav
  Data læses på pin 9. Inverteret data kan læses på pin 7
  fx ved aktiv lav switche.
  Pin 15 kan blot være permanent lav
*/

uint8_t dataPin = 5; // til 165-pin 9
uint8_t clockPin = 6; // til 165-pin 2
uint8_t latchPin = 7; // til 165-pin 1, Latchpin, load latch på lav, shift på high !

uint8_t incoming = 0;

void setup() {

  Serial.begin(9600); // Setup Serial Monitor

  // Setup 74HC165 connections
  pinMode(latchPin, OUTPUT);
  digitalWrite(latchPin, 1); // er aktiv lav
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, INPUT);
}
/////
void loop() {

  modtag_fra_165(); // Læs data fra 74HC165

  // Print to serial monitor
  Serial.print("Pin States: ");
  Serial.println(incoming, BIN);
  delay(200);
} // End Loop

//*****
```



```
// min Shiftin funktion
uint8_t modtag_fra_165() {

    digitalWrite(latchPin, HIGH); // Lås Latch, og enable Shifting
    for (uint8_t i = 0; i < 8; i++) {
        digitalWrite(clockPin, 1);
        bitWrite(incoming, i, digitalRead(dataPin)); // MSB first: 7-i
        digitalWrite(clockPin, 0);
    }
    digitalWrite(latchPin, LOW); // Latch transparent !
} // end Modtag funktion
```

// Ide Fra: <https://dronebotworkshop.com/shift-registers/>