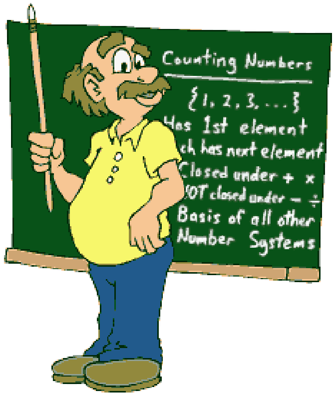
AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

# MSYS

## Microcontroller Systems

## Lektion 19: A/D konvertering




Version: 21-11-2017, Henning Hargaard

# Hvorfor ADC?

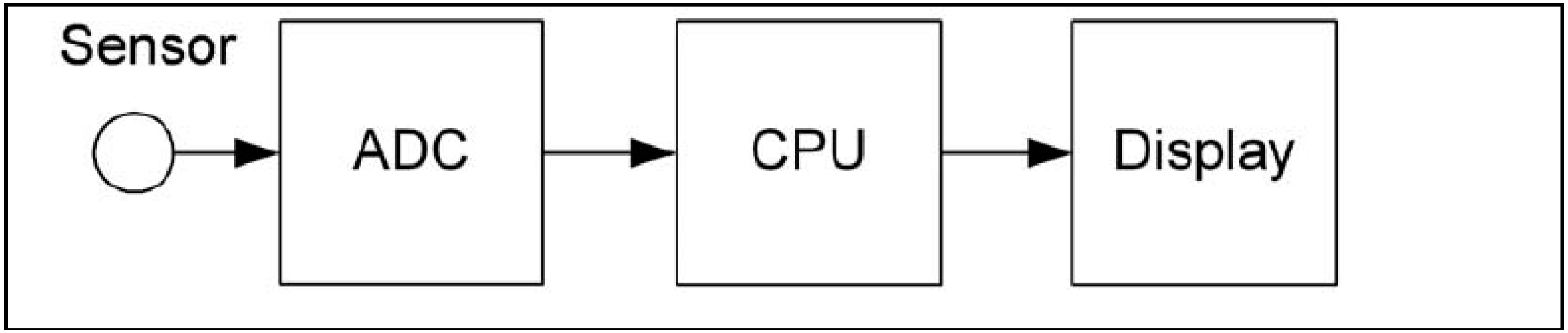
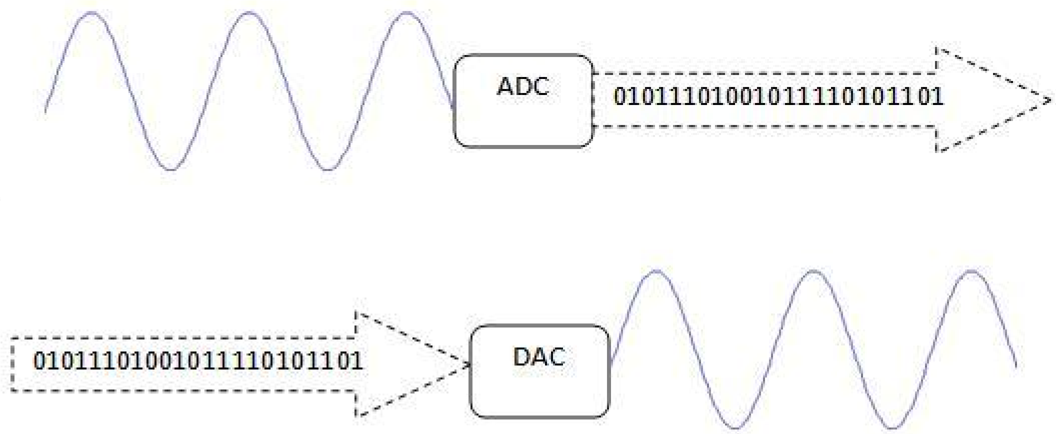
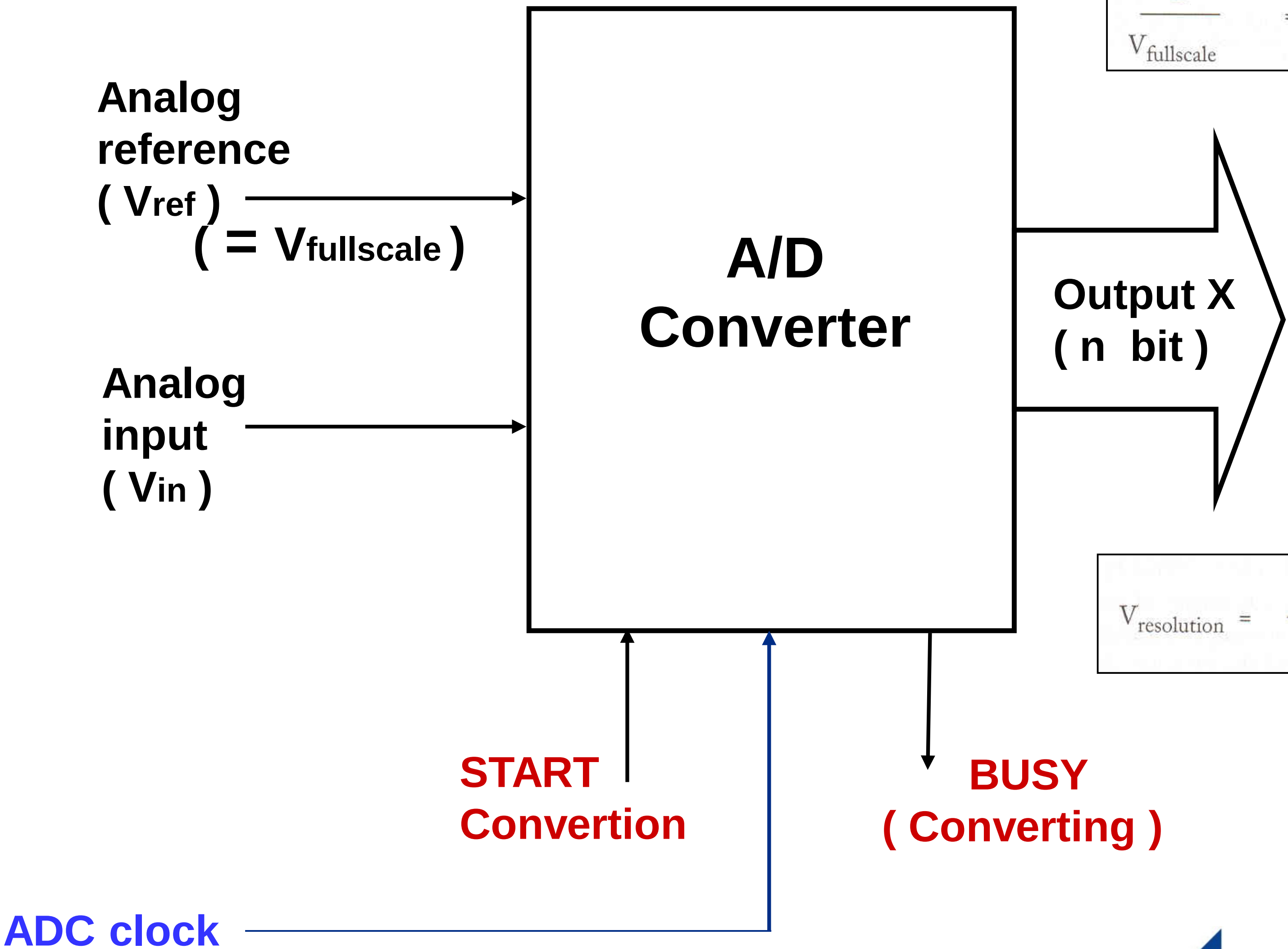


**Figure 13-1. Microcontroller Connection to Sensor via ADC**



# ADC generelt




$$\frac{V_{in}}{V_{fullscale}} = \frac{X}{2^n-1}$$

$$V_{resolution} = \frac{V_{fullscale}}{2^n-1}$$

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Test ("socrative.com": Room = MSYS)

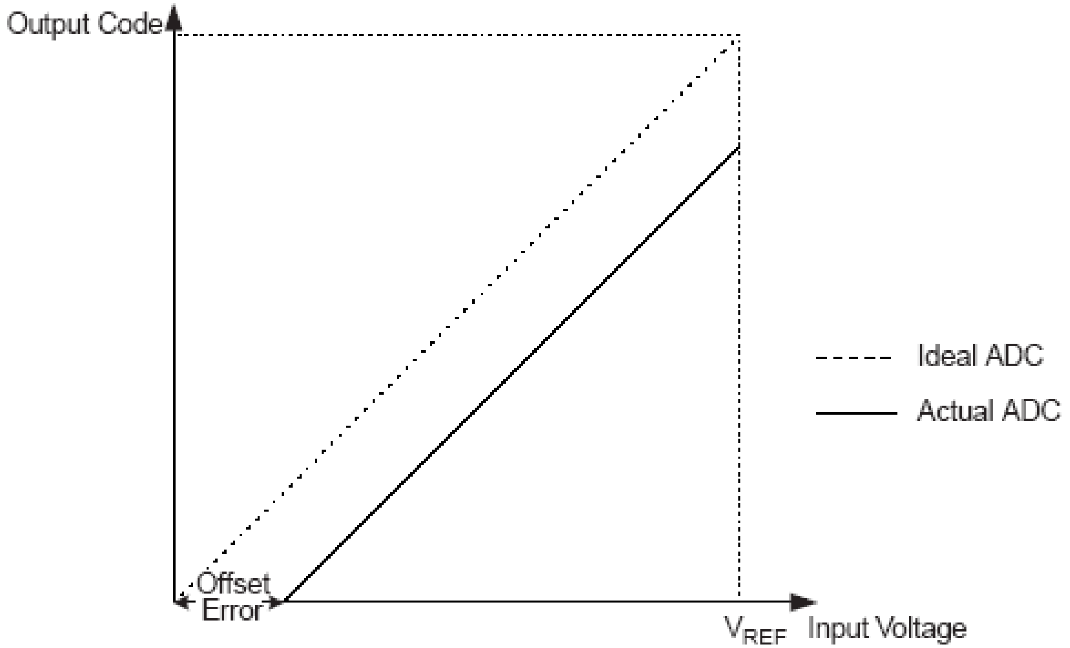
En A/D konverters referencespænding er 5 volt, og måleresultatet repræsenteres med 12 bit.

Hvad er A/D konverterens måleopløsning ?

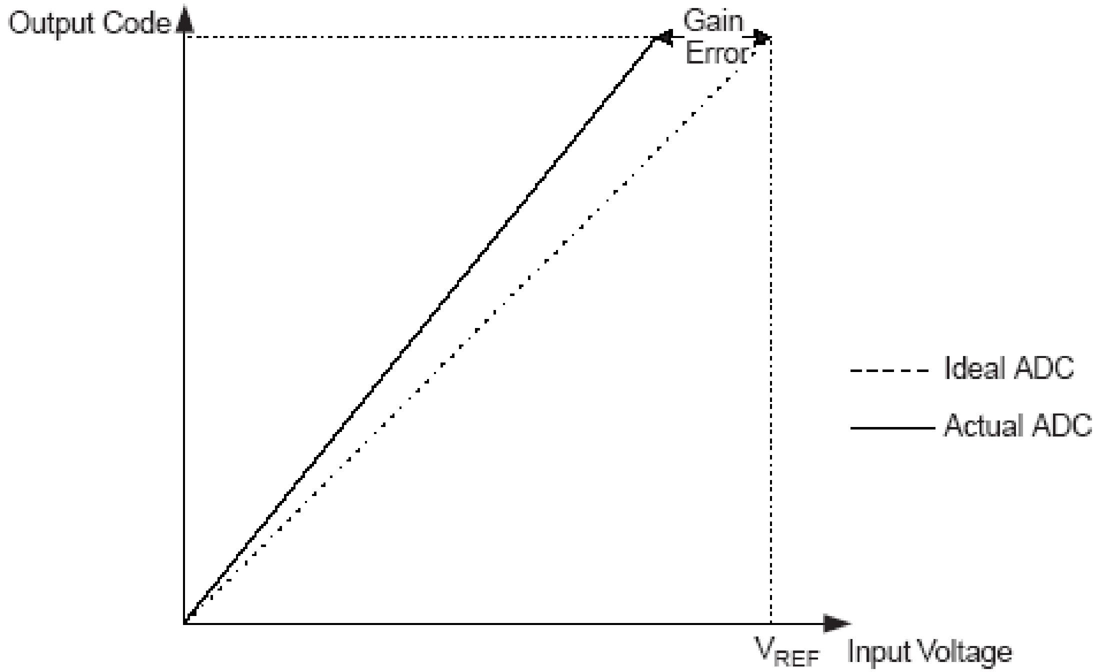
- A: 1,22 mV
- B: 12 mV
- C: 4,88 mV
- D: 5,12 mV

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

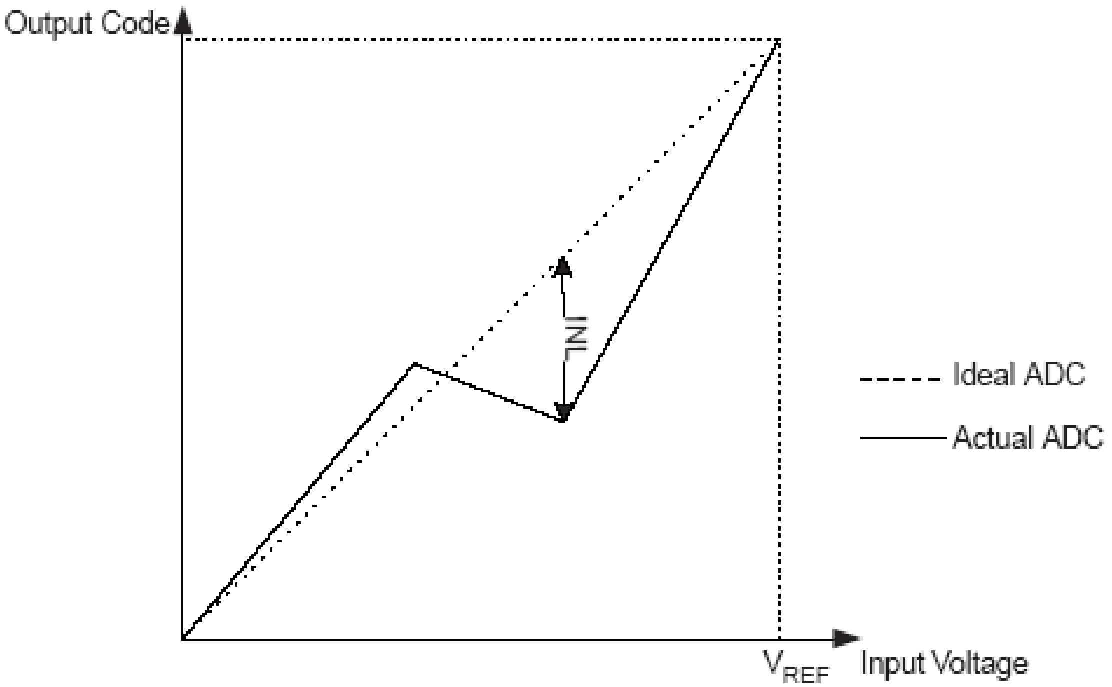
# Offset error



# Gain error


Output Code
Gain
Error
Ideal ADC
Actual ADC
$V_{REF}$
Input Voltage

# Integral non-linearitet




AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

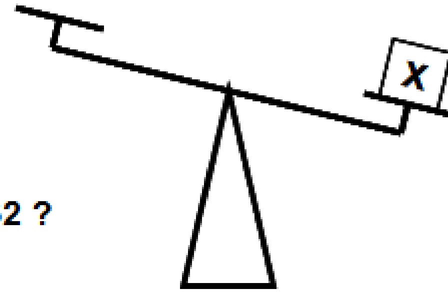
# Mega32/Mega2560 ADC

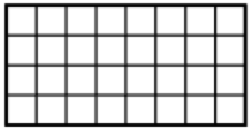
**AVR’s ADC = ”Successiv Approximation ADC”**

# Successive approximation

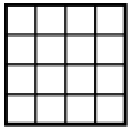
ASSUME X = 45

TEST

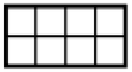


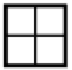
IS $X \geq 32$ ? YES → RETAIN 32 → 1



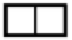
IS $X \geq (32 + 16)$ ? NO → REJECT 16 → 0



IS $X \geq (32 + 8)$ ? YES → RETAIN 8 → 1



IS $X \geq (32 + 8 + 4)$ ? YES → RETAIN 4 → 1



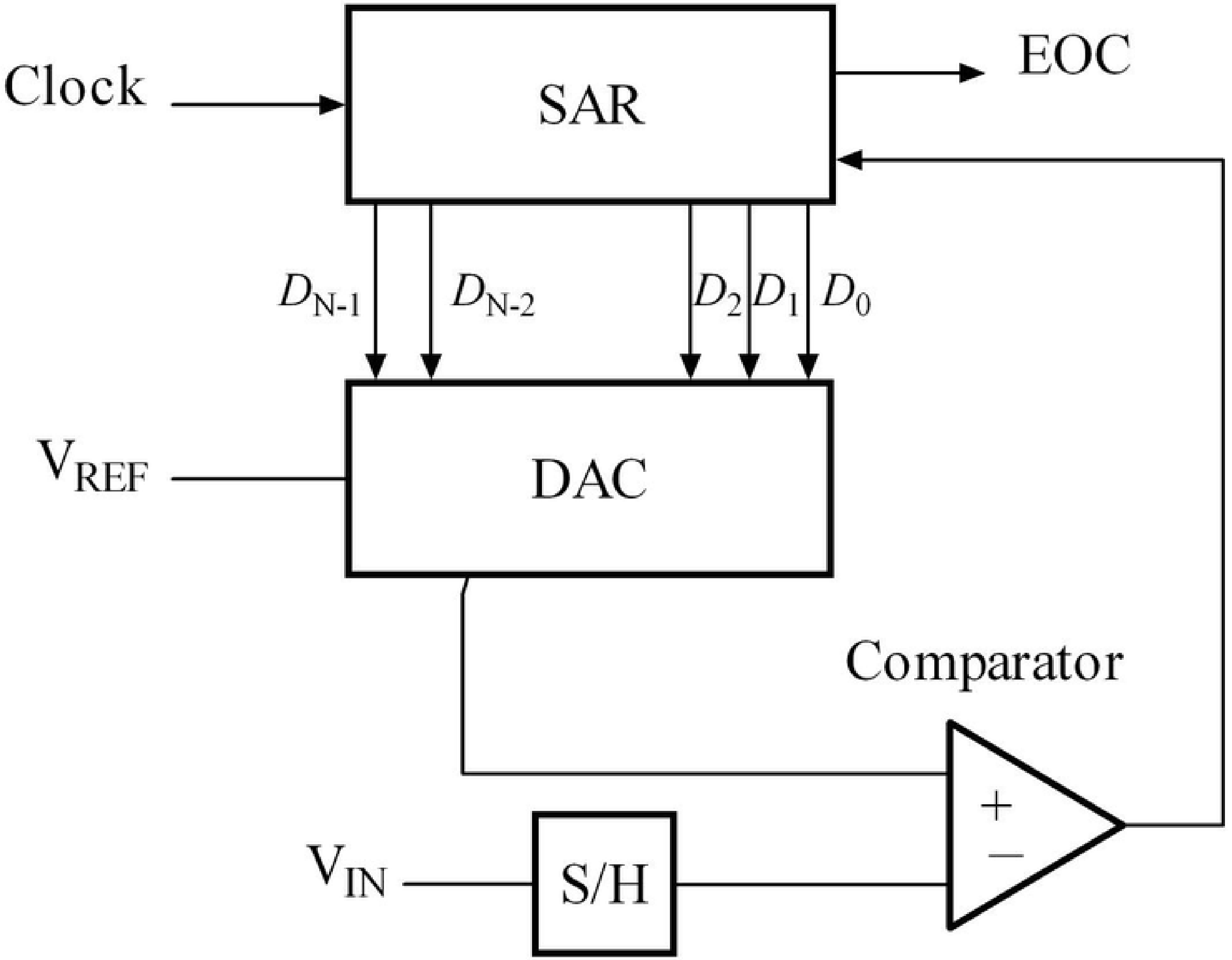
IS $X \geq (32 + 8 + 4 + 2)$ ? NO → REJECT 2 → 0

IS $X \geq (32 + 8 + 4 + 2 + 1)$ ? YES → RETAIN 1 → 1

TOTALS: $X = 32 + 8 + 4 + 1 = 45_{10} = 101101_2$

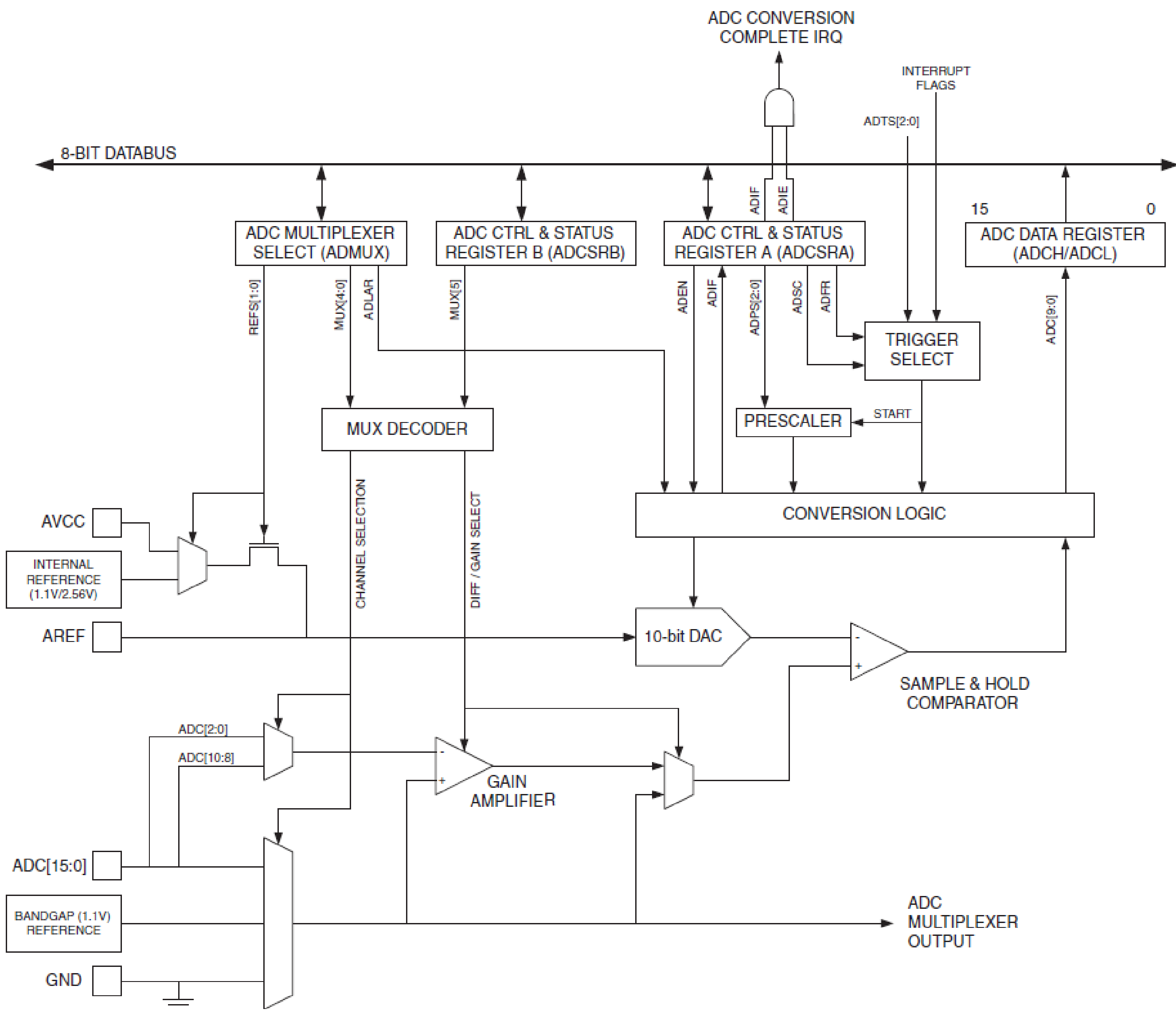
AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Successive approximation ADC




AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Blokdiagram, Mega2560 ADC



# ADC enable

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

- **Før ADC'en kan anvendes, skal den enables (skriv 1 til ADCSRA bit 7).**
- **Herved "tændes" for ADC'ens hardware.**



AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

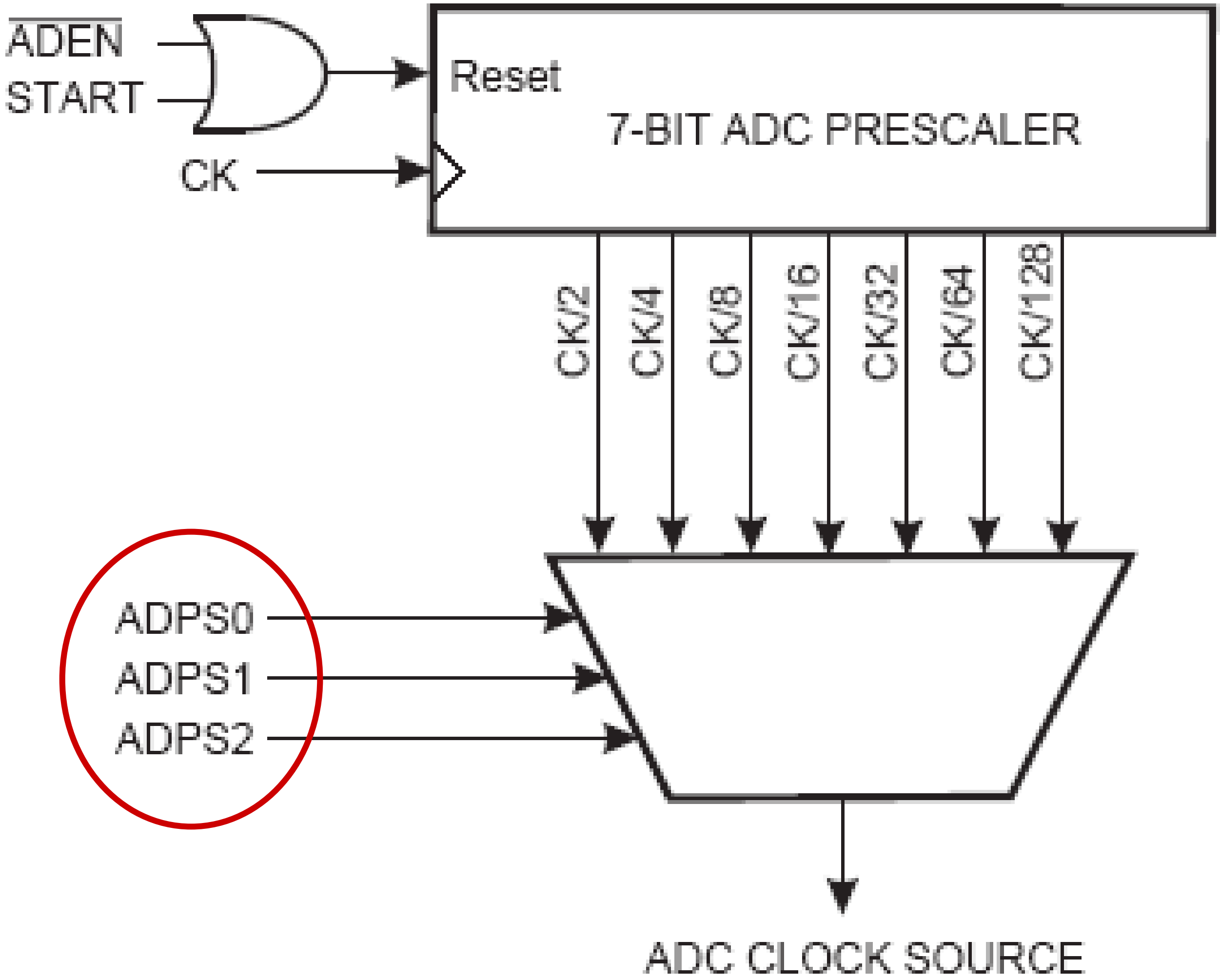
# ”ADCW” = ADCH→ADCL

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | – | – | ADC9 | ADC8 | ADCH |
| | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 | ADCL |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |

- **A/D konverterens resultat (efter konvertering) vil være i registrene ADCH og ADCL.**
- **Kan kun aflæses (ikke skrives til).**
- **Vores compiler kan aflæse til en variabel på denne måde:**
  **unsigned int x = ADCW.**

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# ADC prescaler / clock




AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# ADC prescaler

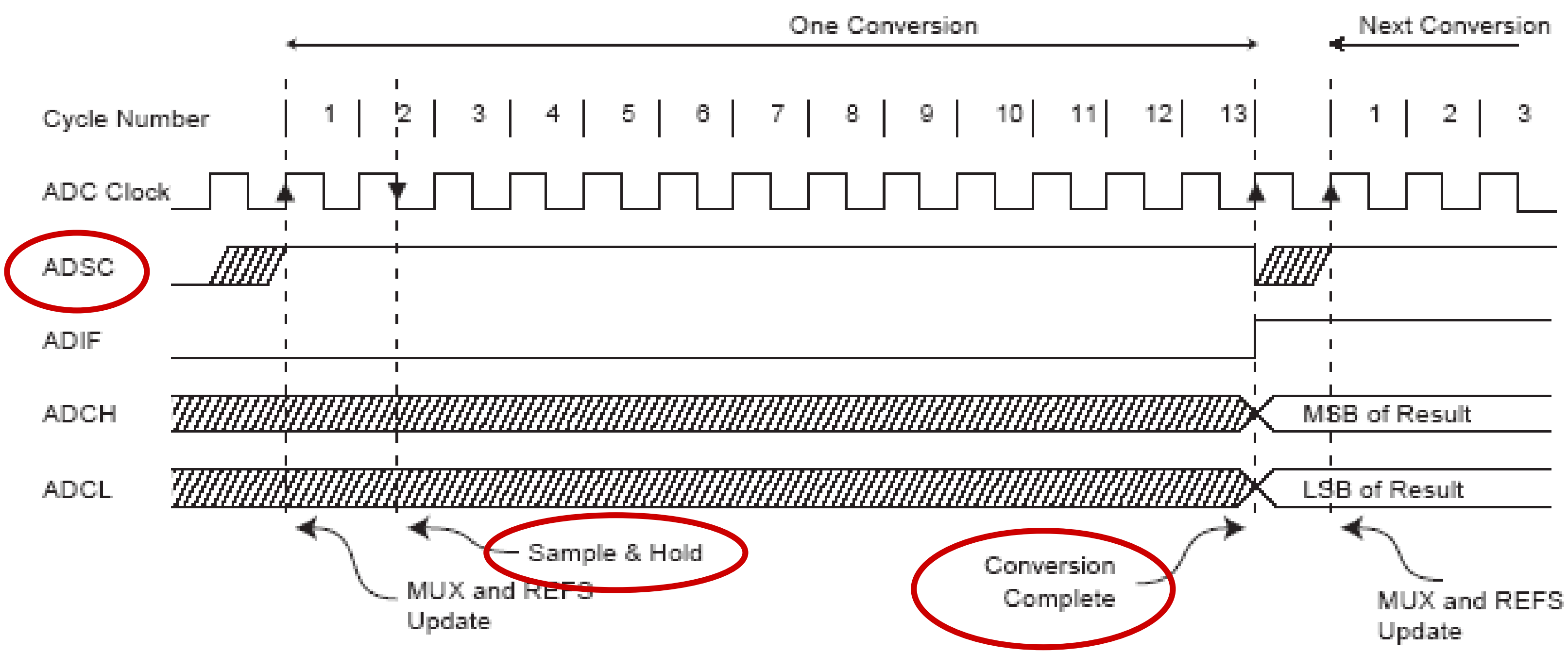
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | ADCSRA |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| ADPS2 | ADPS1 | ADPS0 | Division Factor |
|---|---|---|---|
| 0 | 0 | 0 | 2 |
| 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 4 |
| 0 | 1 | 1 | 8 |
| 1 | 0 | 0 | 16 |
| 1 | 0 | 1 | 32 |
| 1 | 1 | 0 | 64 |
| 1 | 1 | 1 | 128 |

**OBS : ADC clock skal være mellem 50 kHz og 200 kHz !**
**( For at kunne opnå 10 bit præcision ).**

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# ADC timing


One Conversion
Next Conversion
Cycle Number
1
2
3
4
5
6
7
8
9
10
11
12
13
1
2
3
ADC Clock
ADSC
ADIF
ADCH
ADCL
MSB of Result
LSB of Result
Sample & Hold
MUX and REFS Update
Conversion Complete
MUX and REFS Update

# Konverterings-tid

| Condition | Sample & Hold (Cycles from Start of Conversion) | Conversion Time (Cycles) |
|---|---|---|
| First conversion | 14.5 | 25 |
| Normal conversions, single ended | 1.5 | 13 |
| Auto Triggered conversions | 2 | 13.5 |
| Normal conversions, differential | 1.5/2.5 | 13/14 |

**Eksempel :**

**$f_{ADC}$ = 200 kHz =>**
**13 cykles = 65 uS =>**

**15380 konverteringer / sekund**

# Test ("socrative.com": Room = MSYS)

- CPU clock frekvens = 16 MHz
  ADC prescaler = 128
  Antag 13 ADC clockperioder per konvertering.
  Hvor lang tid varer en A/D-konvertering ?

A:

8 mikrosekunder

B:

104 mikrosekunder

C:

81 nanosekunder

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Mega2560: Valg af reference

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | ADMUX |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| REFS1 | REFS0 | Voltage Reference Selection[1] |
|---|---|---|
| 0 | 0 | AREF, Internal $V_{REF}$ turned off |
| 0 | 1 | AVCC with external capacitor at AREF pin |
| 1 | 0 | Internal 1.1V Voltage Reference with external capacito |
| 1 | 1 | Internal 2.56V Voltage Reference with external capacit |

**AREF = Ekstern pin**

**AVCC = 5 volt**

**Intern 1,1 volt**

**Intern 2,56 volt**

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Test ("socrative.com": Room = MSYS)

- CPU clockfrekvensen er <u>12 MHz</u>.
  Vi ønsker, at ADC clock frekvensen er <u>mellem 50 kHz og 200 kHz</u> (for bedste performance).
  Hvilken ADCSRA værdi er FORKERT ?

A:

ADCSRA = 0b10000111;

B:

ADCSRA = 0b10000110;

C:

ADCSRA = 0b10000101;

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Test (”socrative.com”: Room = MSYS)

- Vi bruger ADC’en til at måle en fast DC spænding. ADC’ens output er 400, når spændingen er 2 volt. Hvad er ADC’ens output, når spændingen ændres til 1 volt ?

A:
800
B:
400
C:
200
D:
100

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

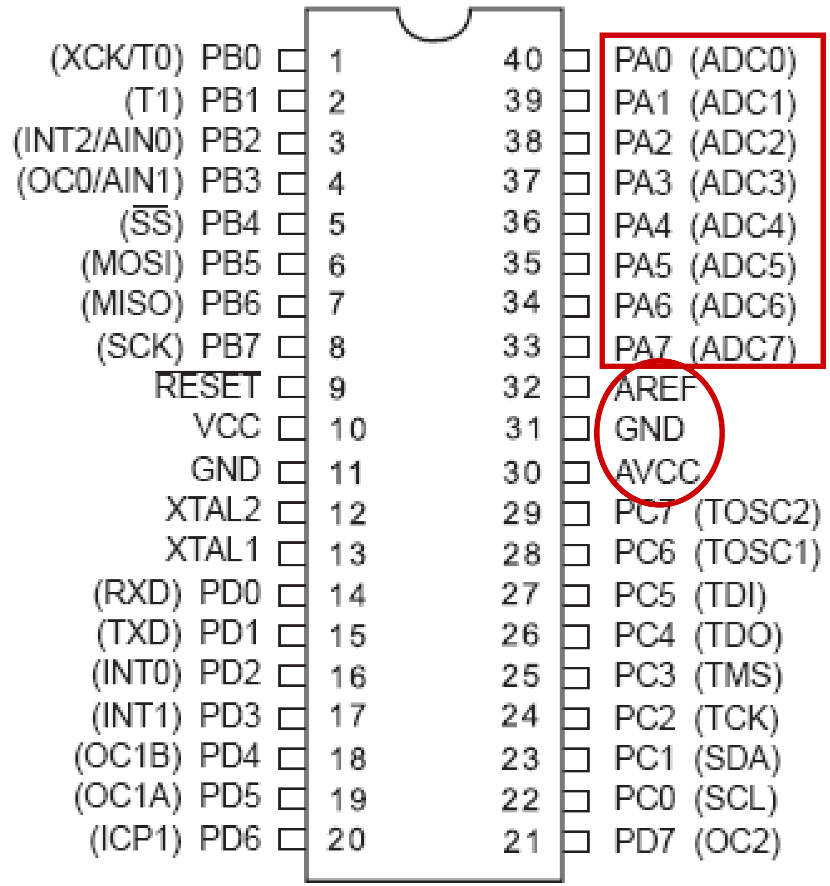
# Test (”socrative.com”: Room = MSYS)

- Vi bruger ADC’en til at måle en fast DC spænding. ADC’en output er 400, når referencen er 4 volt. Hvad er ADC’ens output, når referencen ændres til 2 volt ?

A:
200
B:
300
C:
400
D:
800

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

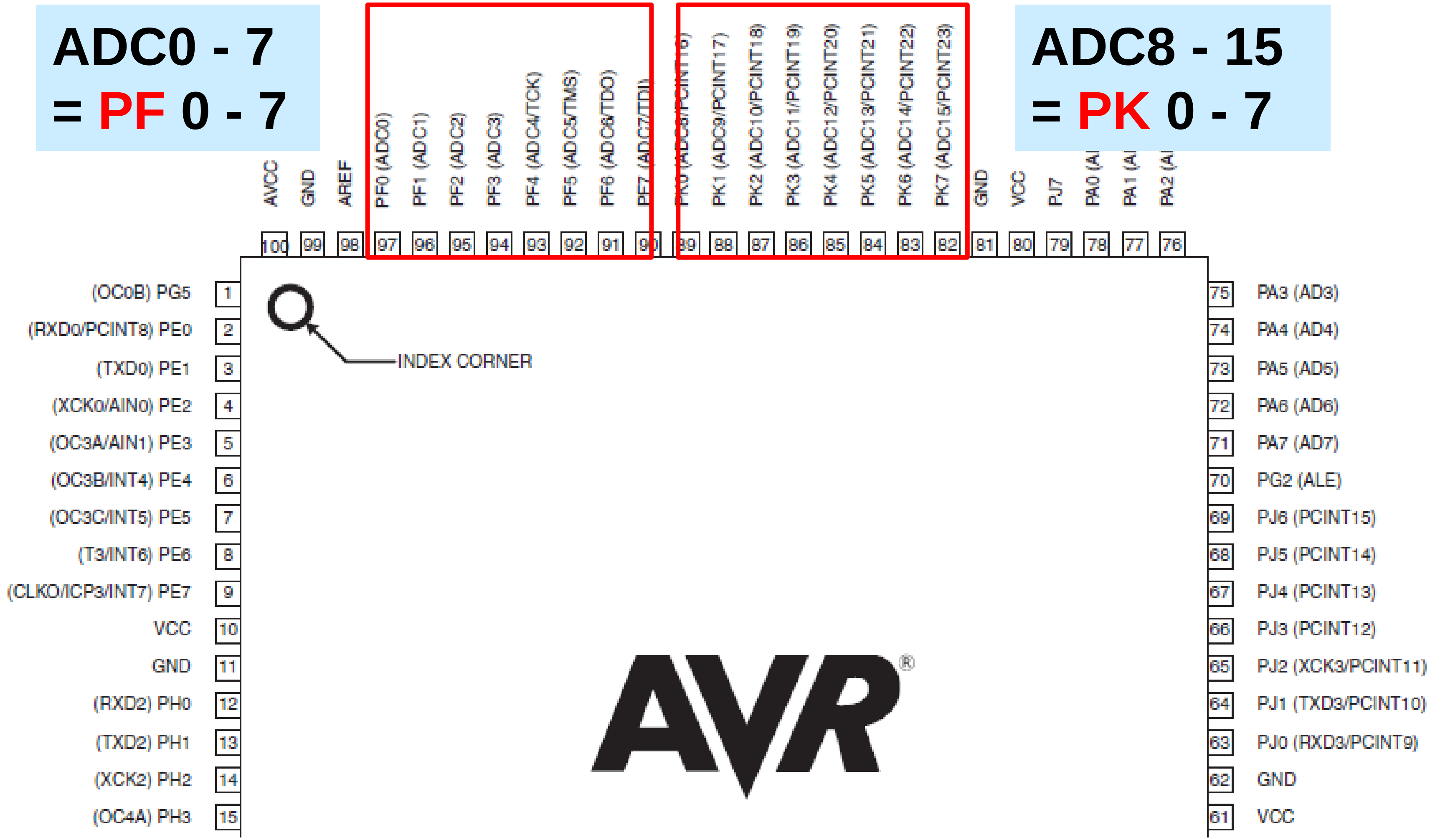
# Mega32: ADC pins (= PA ben)



# Mega2560: ADC pins (= PF og PK ben)

**ADC0 - 7**
**= PF 0 - 7**

**ADC8 - 15**
**= PK 0 - 7**




AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Mega2560: Valg af ADC-indgang (1)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|
| REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | ADMUX |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|
| – | ACME | – | – | MUX5 | ADTS2 | ADTS1 | ADTS0 | ADCSRB |
| R | R/W | R | R | R/W | R/W | R/W | R/W | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| MUX5:0 | Single Ended Input | Positive Differential Input | Negative Differential Input | Gain |
|---|---|---|---|---|
| 000000 | ADC0 | | | |
| 000001 | ADC1 | | | |
| 000010 | ADC2 | | | |
| 000011 | ADC3 | | | |
| 000100 | ADC4 | | | |
| 000101 | ADC5 | | | |
| 000110 | ADC6 | | | |
| 000111 | ADC7 | | | |

**”Single ended” = En spænding refereret til stel.**

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Mega2560: Valg af ADC-indgange (2)

| MUX5:0 | Single Ended Input | Positive Differential Input | Negative Differential Input | Gain |
|---|---|---|---|---|
| 001000[1] | N/A | ADC0 | ADC0 | 10x |
| 001001[1] | | ADC1 | ADC0 | 10x |
| 001010[1] | | ADC0 | ADC0 | 200x |
| 001011[1] | | ADC1 | ADC0 | 200x |
| 001100[1] | | ADC2 | ADC2 | 10x |
| 001101[1] | | ADC3 | ADC2 | 10x |
| 001110[1] | | ADC2 | ADC2 | 200x |
| 001111[1] | | ADC3 | ADC2 | 200x |
| 010000 | | ADC0 | ADC1 | 1x |
| 010001 | | ADC1 | ADC1 | 1x |
| 010010 | | ADC2 | ADC1 | 1x |
| 010011 | | ADC3 | ADC1 | 1x |
| 010100 | | ADC4 | ADC1 | 1x |
| 010101 | | ADC5 | ADC1 | 1x |
| 010110 | | ADC6 | ADC1 | 1x |
| 010111 | | ADC7 | ADC1 | 1x |
| 011000 | | ADC0 | ADC2 | 1x |
| 011001 | | ADC1 | ADC2 | 1x |
| 011010 | N/A | ADC2 | ADC2 | 1x |
| 011011 | | ADC3 | ADC2 | 1x |
| 011100 | | ADC4 | ADC2 | 1x |
| 011101 | | ADC5 | ADC2 | 1x |

# Mega2560: Valg af ADC-indgange (3)

| MUX5:0 | Single Ended Input | Positive Differential Input | Negative Differential Input | Gain |
|---|---|---|---|---|
| 011110 | 1.1V ($V_{BG}$) | N/A | | |
| 011111 | 0V (GND) | | | |
| 100000 | ADC8 | N/A | | |
| 100001 | ADC9 | | | |
| 100010 | ADC10 | | | |
| 100011 | ADC11 | | | |
| 100100 | ADC12 | | | |
| 100101 | ADC13 | | | |
| 100110 | ADC14 | | | |
| 100111 | ADC15 | | | |

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Mega2560: Valg af ADC-indgange (4)

| MUX5:0 | Single Ended Input | Positive Differential Input | Negative Differential Input | Gain |
|---|---|---|---|---|
| 101000[1] | N/A | ADC8 | ADC8 | 10x |
| 101001[1] | | ADC9 | ADC8 | 10x |
| 101010[1] | | ADC8 | ADC8 | 200x |
| 101011[1] | | ADC9 | ADC8 | 200x |
| 101100[1] | | ADC10 | ADC10 | 10x |
| 101101[1] | | ADC11 | ADC10 | 10x |
| 101110[1] | | ADC10 | ADC10 | 200x |
| 101111[1] | | ADC11 | ADC10 | 200x |
| 110000 | | ADC8 | ADC9 | 1x |
| 110001 | | ADC9 | ADC9 | 1x |
| 110010 | | ADC10 | ADC9 | 1x |
| 110011 | | ADC11 | ADC9 | 1x |
| 110100 | | ADC12 | ADC9 | 1x |
| 110101 | | ADC13 | ADC9 | 1x |
| 110110 | | ADC14 | ADC9 | 1x |
| 110111 | | ADC15 | ADC9 | 1x |
| 111000 | | ADC8 | ADC10 | 1x |
| 111001 | | ADC9 | ADC10 | 1x |
| 111010 | | ADC10 | ADC10 | 1x |
| 111011 | | ADC11 | ADC10 | 1x |
| 111100 | | ADC12 | ADC10 | 1x |
| 111101 | N/A | ADC13 | ADC10 | 1x |
| 111110 | Reserved | N/A | | |
| 111111 | Reserved | N/A | | |

# Normal / ”Left Adjust Result”

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | ADMUX |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

*ADLAR = 0*

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | – | – | ADC9 | ADC8 | ADCH |
| | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 | ADCL |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R | R | R | R | R | R | R | R | |
| | R | R | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

*ADLAR = 1*

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| | ADC9 | ADC8 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADCH |
| | ADC1 | ADC0 | – | – | – | – | – | – | ADCL |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R | R | R | R | R | R | R | R | |
| | R | R | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

# ADCH and ADCL Data registers

- ADCH:ADCL store the results of conversion.
- The 10 bit result can be right or left justified:

**Vores compiler:**
**ADCW er ADCH—ADCL:**
**x = ADCW;**

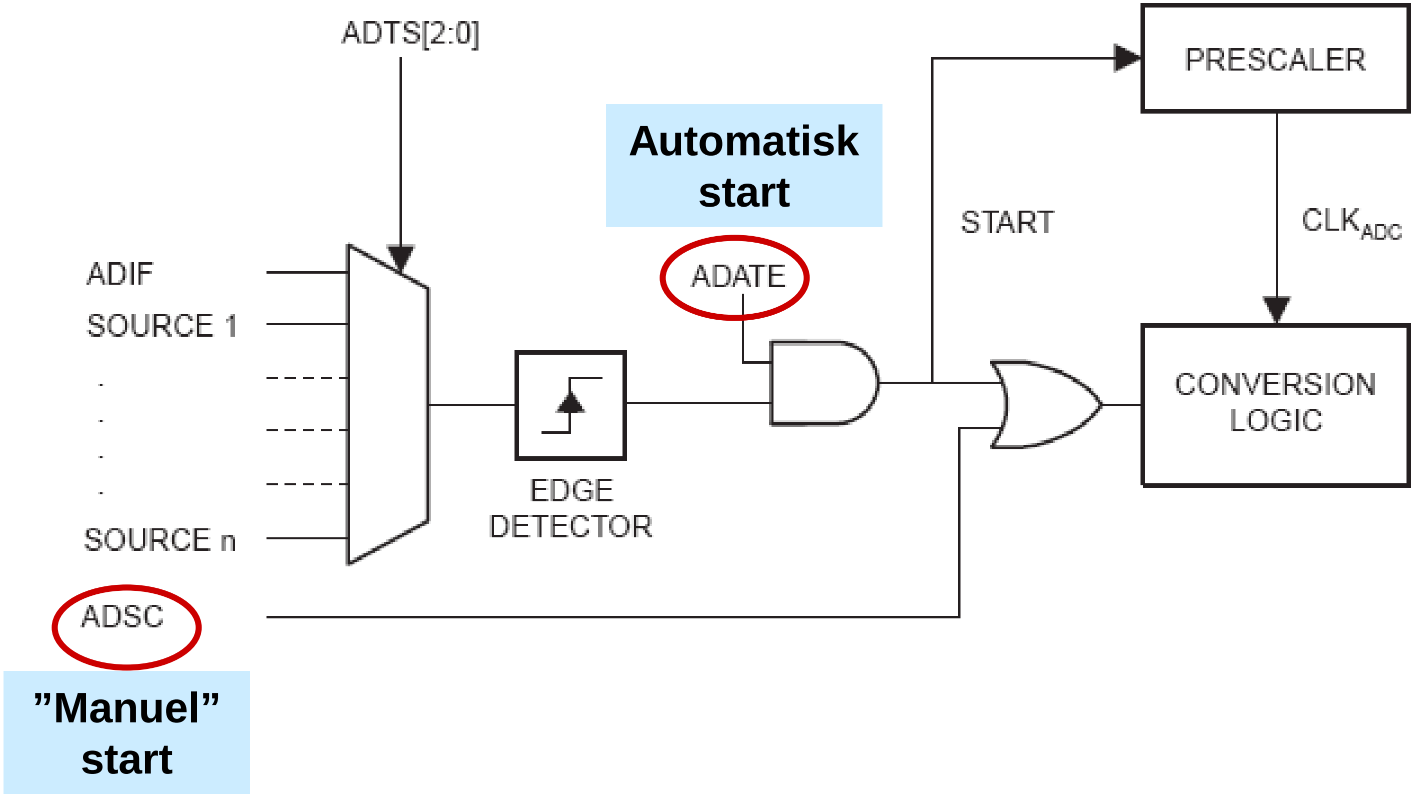
ADLAR = 0 **← Anbefales !**

| ADCH | | | | | | | | ADCL | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | ADC9 | ADC8 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 |

ADLAR =1

| ADCH | | | | | | | | ADCL | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADC9 | ADC8 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 | - | - | - | - | - | - |

# ADC start ("trigger")




AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# ”Manuel” START

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | ADCSRA |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- Bit ADSC = 1 : Konvertering starter.
  ADCSRA |= 0b01000000;

- Bit ADSC == 0 : Konvertering slut.
  while (ADCSRA & 0b01000000)
  { }
  // Herefter kan ADC aflæses
  x = ADCW;



AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# Mega2560: Automatisk START (”trigger”)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | ADCSRA |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

+

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | – | ACME | – | – | MUX5 | ADTS2 | ADTS1 | ADTS0 | ADCSRB |
| | R | R/W | R | R | R/W | R/W | R/W | R/W | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| ADTS2 | ADTS1 | ADTS0 | Trigger Source ← ”Interrupt fra…” |
|---|---|---|---|
| 0 | 0 | 0 | Free Running mode |
| 0 | 0 | 1 | Analog Comparator |
| 0 | 1 | 0 | External Interrupt Request 0 |
| 0 | 1 | 1 | Timer/Counter0 Compare Match A |
| 1 | 0 | 0 | Timer/Counter0 Overflow |
| 1 | 0 | 1 | Timer/Counter1 Compare Match B |
| 1 | 1 | 0 | Timer/Counter1 Overflow |
| 1 | 1 | 1 | Timer/Counter1 Capture Event |

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# ADC interrupt

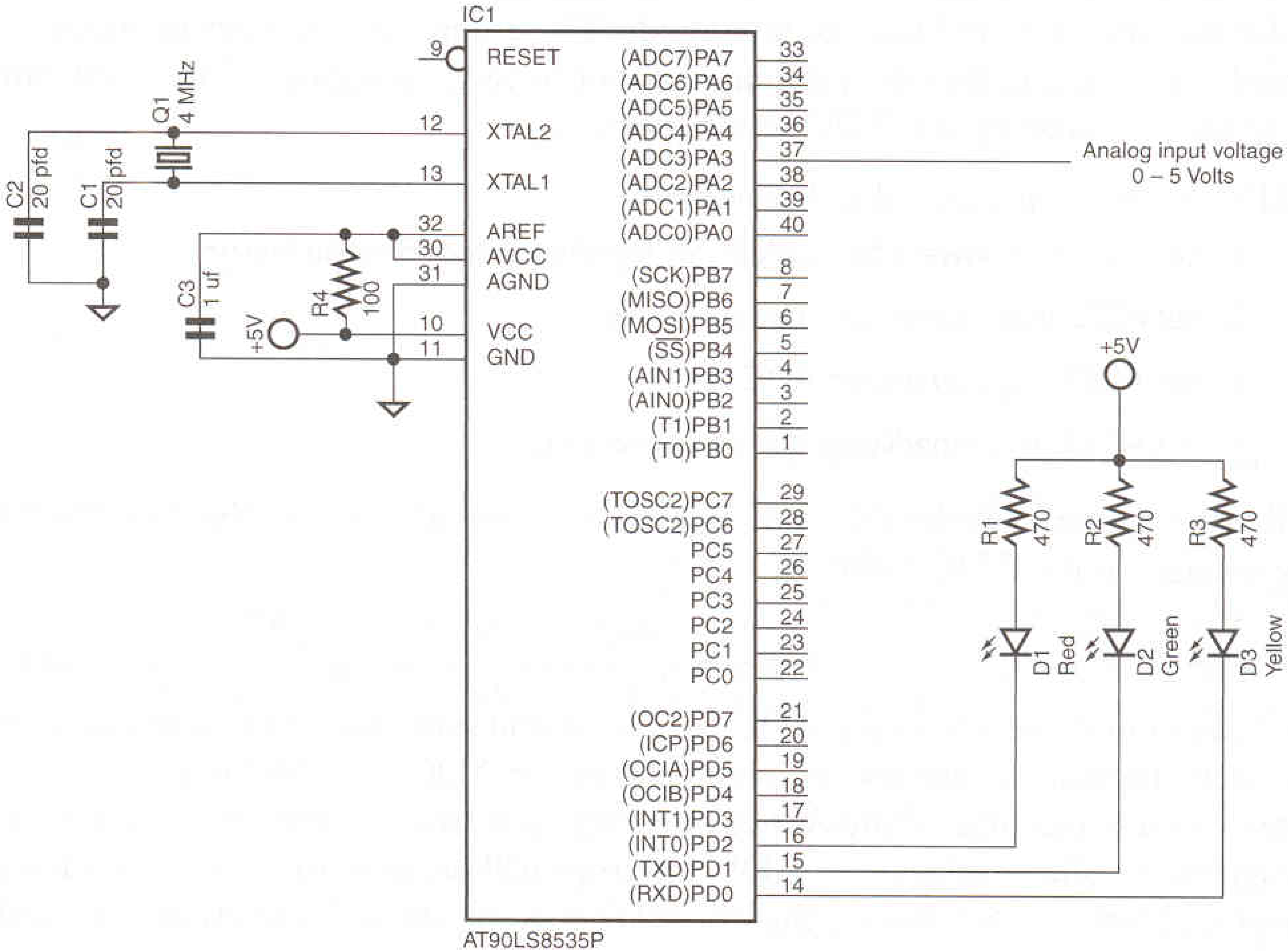
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | ADCSRA |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- ADIE: ”ADC interrupt enable”.
  1 => Der genereres ADC-interrupt, hvis global interrupt enable også er sat.

- (ADIF: ”ADC interrupt flag”)
  Sættes høj efter hver konvertering.
  Nulstilles automatisk i interruptrutinen ELLER ved at skrive et 1-tal til bit ADIF.

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

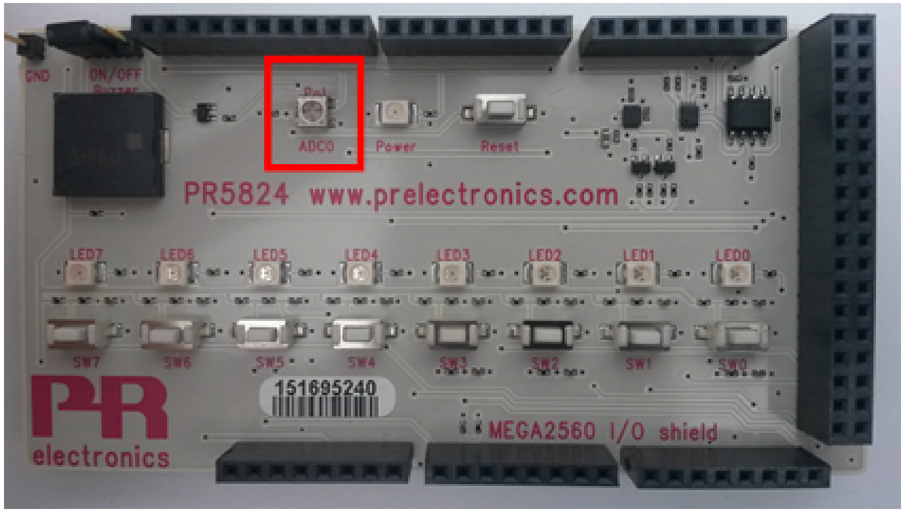
# Eksempel: Niveau-tester






AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

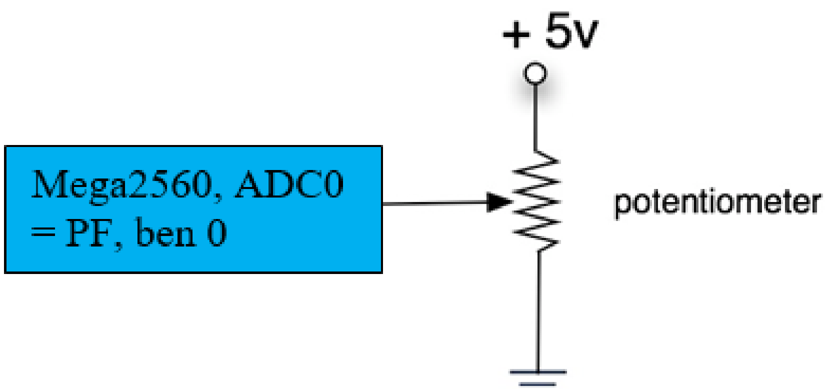
# LAB13




Der skal anvendes en skruetrækker med en speciel lille kærv for at justere på potentiometeret.

OBS: Brug ikke for mange kræfter!
Du kan komme til at ødelægge potentiometeret.

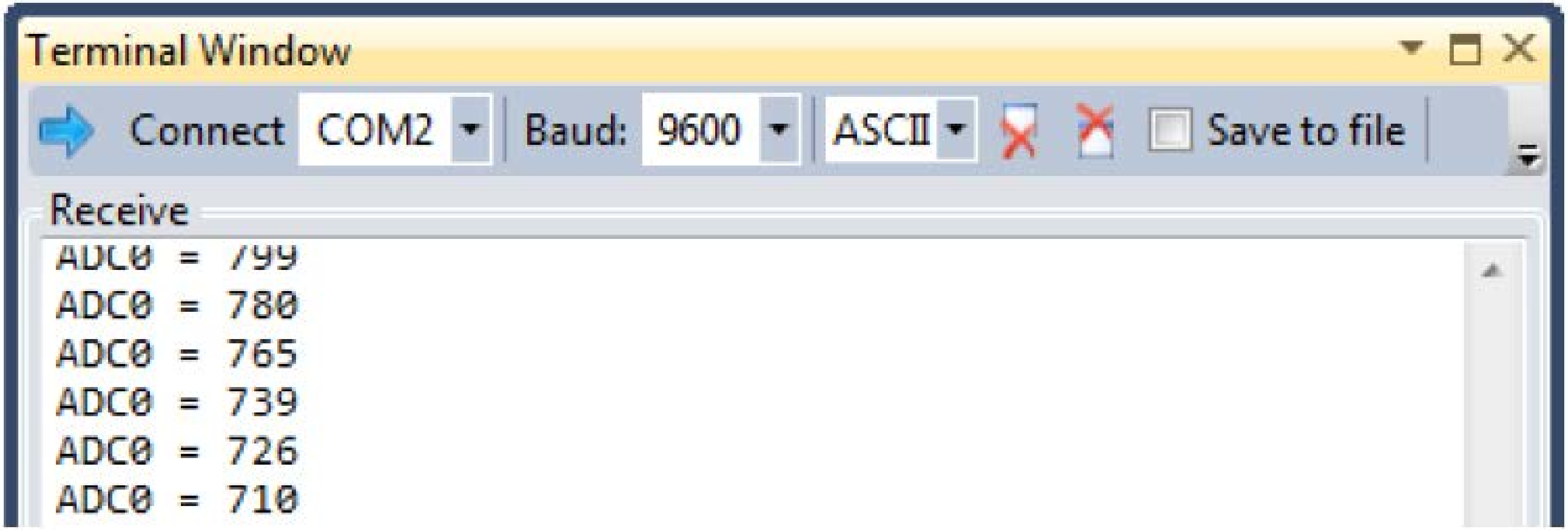
Skruetrækkeren vil kunne lånes til øvelsen.




Ved at skrue på potentiometeret kan vi justere spændingen på ADC0 benet mellem 0 og 5 volt.

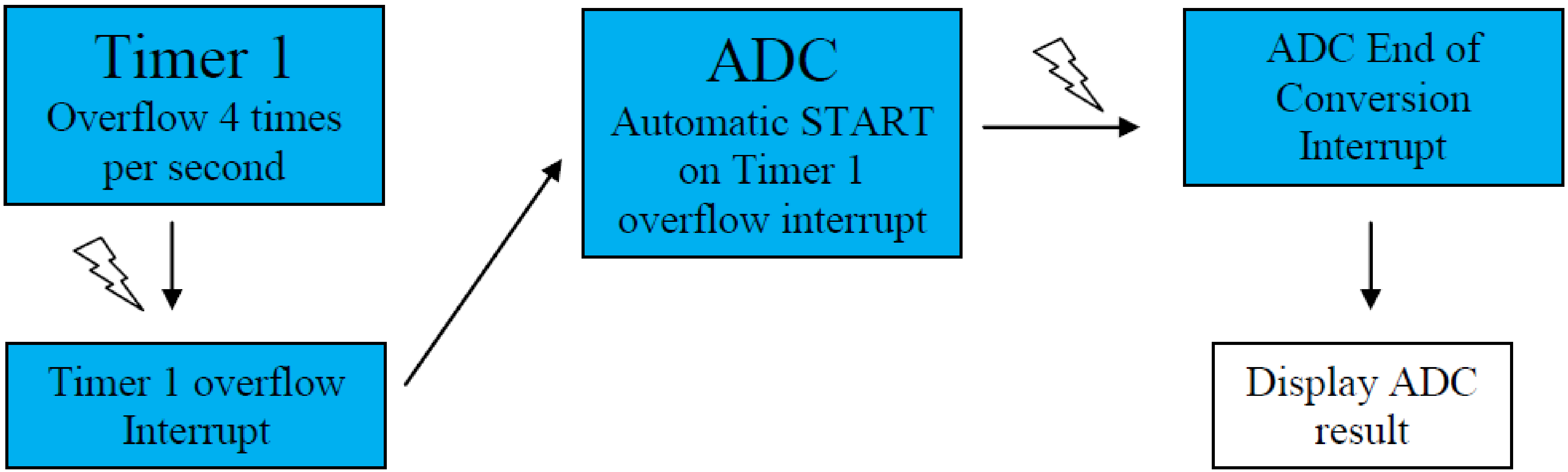
AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# LAB13, del 1



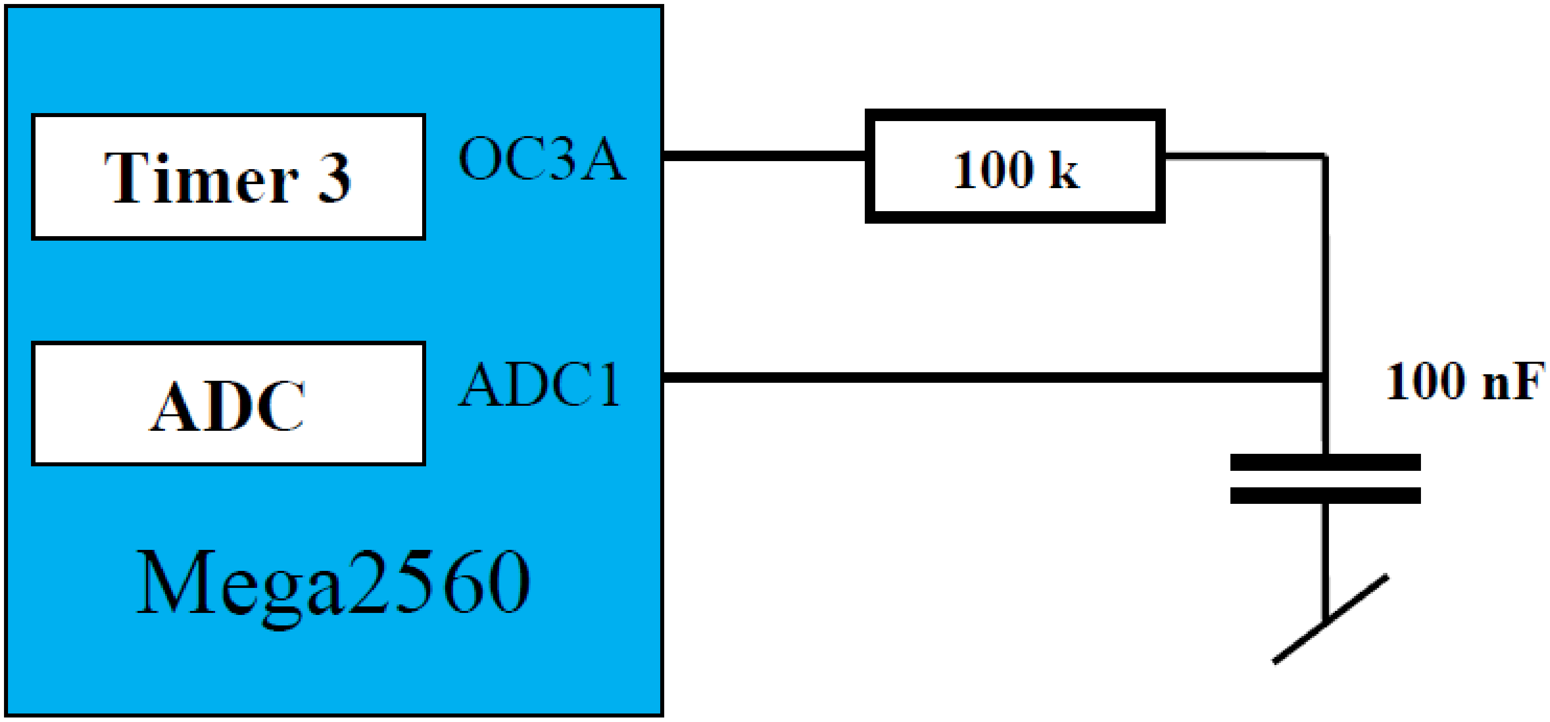

- **”Manuel” start**
- **Aflæs og display potentiometer-spændingen**

# LAB13, del 2




- **Automatisk ”Start on Timer 1 overflow”**
- **ADC interrupt enabled**
- **Aflæs og display potentiometer-spændingen**

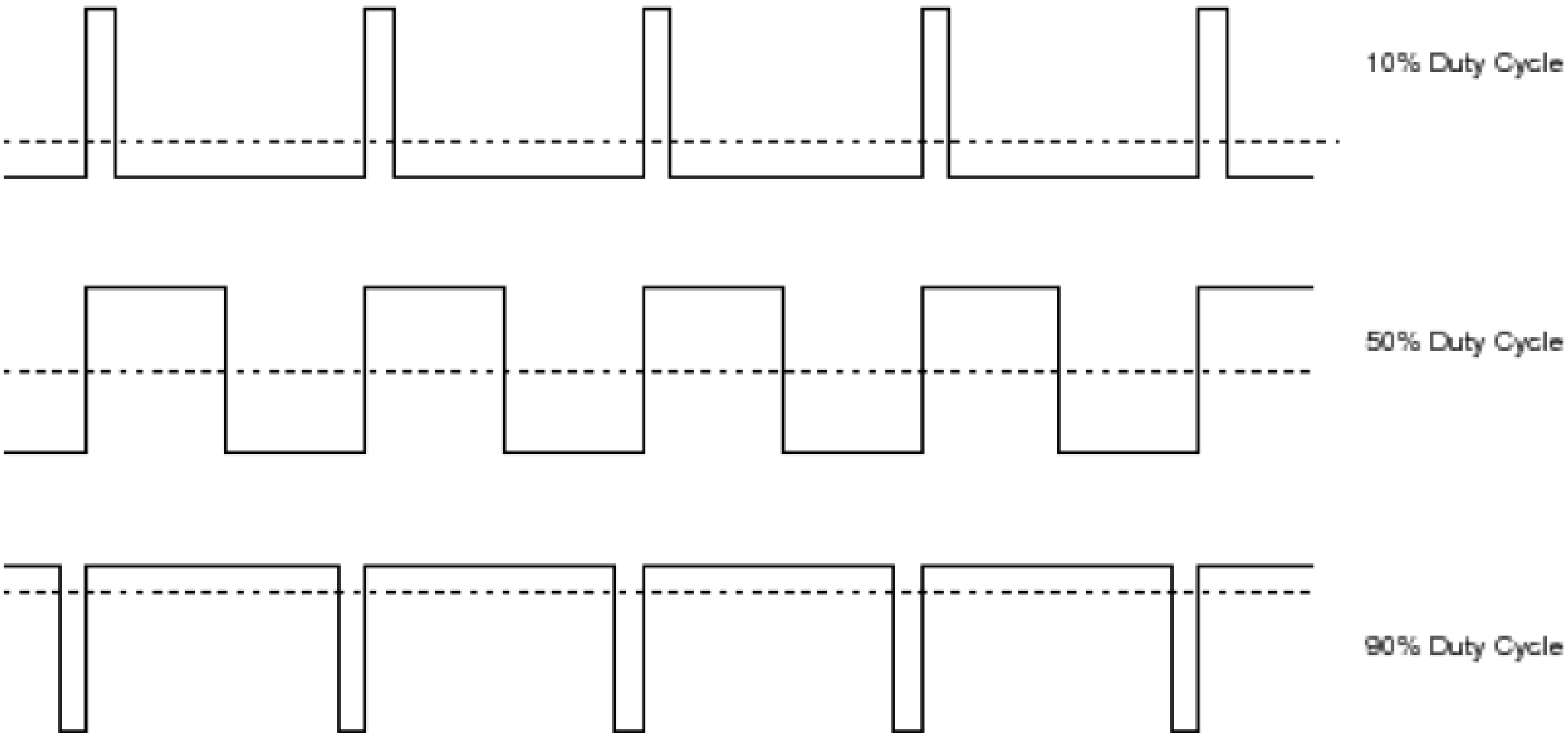
# LAB13, del 3



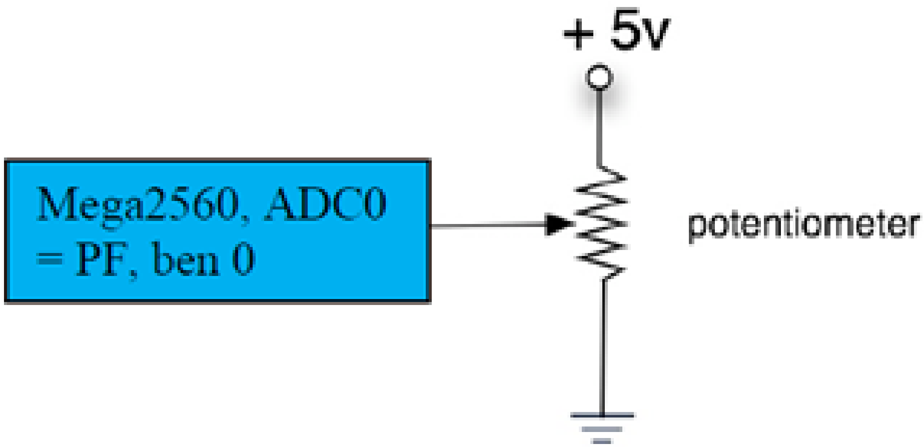


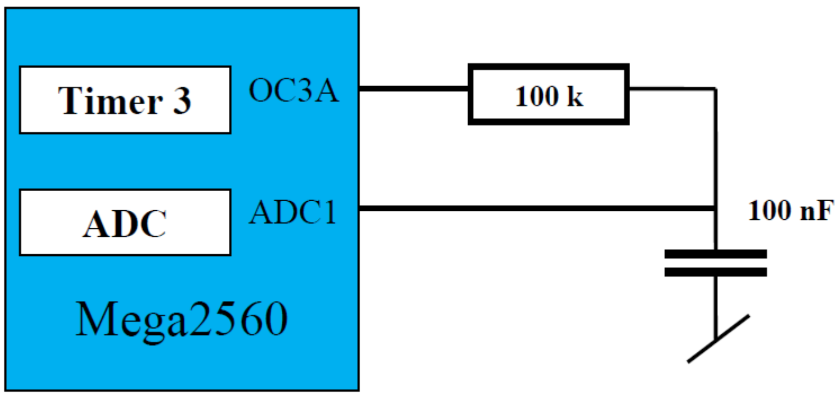

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

# LAB13, del 3



# LAB13, del 3

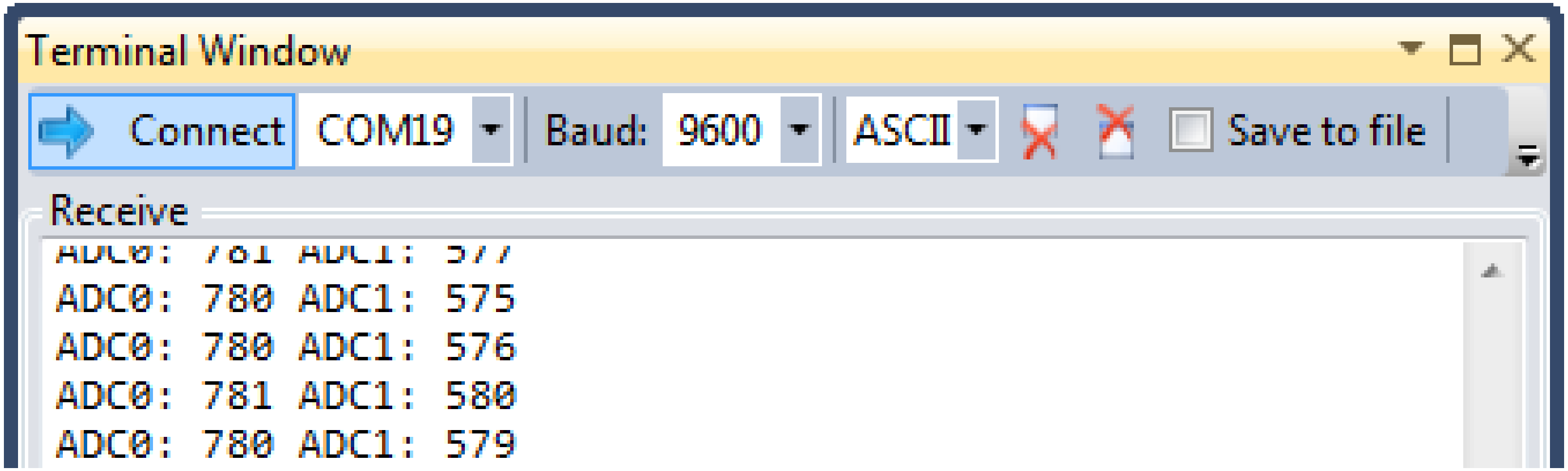



**Simpel regulering :**

**Juster spændingen over kondensatoren, så den bliver den samme som spændingen fra potentiometeret.**

# LAB13, del 3




- **ADC0 er måling fra potentiometeret**
- **ADC1 er måling fra kondensatoren**

# Forslag til yderligere studie

http://www.youtube.com/watch?v=cjmcAE1L6OQ&list=PLgO01FhQgwvuPxEFDRkYilPJiZfsSrQax

http://www.youtube.com/watch?v=hRIVGx-fTKs

# Slut på lektion 19

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING