

Exercise 2

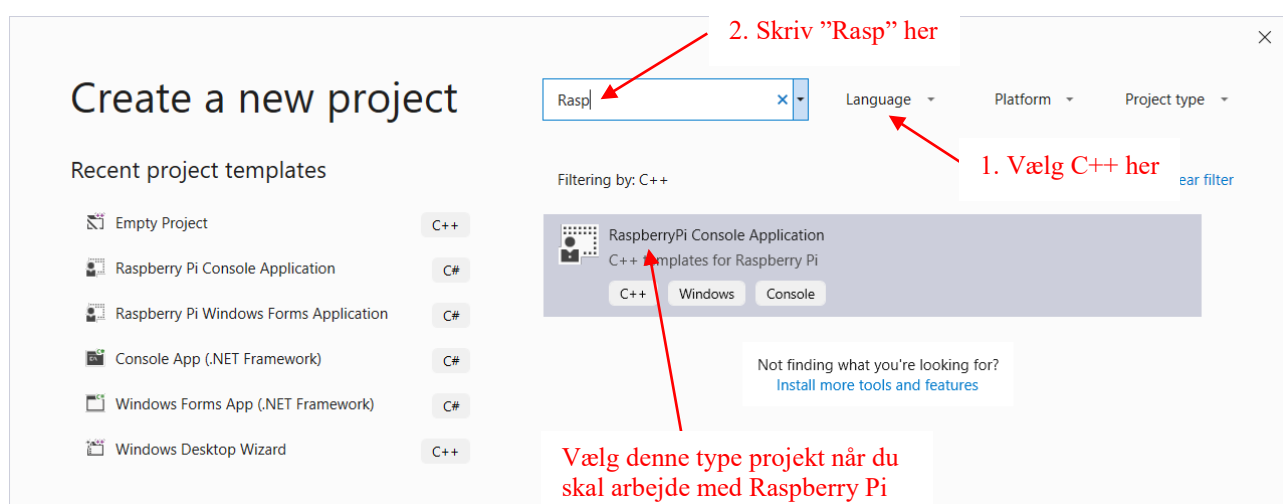
Exercise 2.0 Installation af Raspberry Pi biblioteket på PC'en

For at kunne kommunikere med din Raspberry Pi skal du have et Raspberry Pi bibliotek til Visual Studio installeret på din PC. Proceduren for dette er beskrevet i dokumentationen til Raspberry Pi:

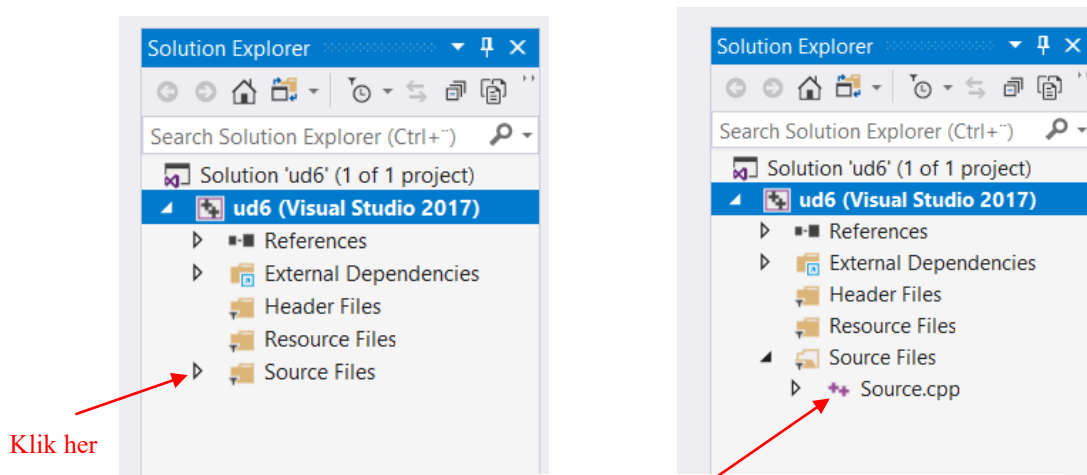
Dokumentation til Raspberry Pi Zero med shield.pdf

som ligger på BS under "Raspberry Pi files". Følg installationsvejledningen i ovennævnte dokument inden du går videre her.

Efter at du har gennemført installationen kan du vælge "RaspberryPi Console Application", når du vil oprette et nyt projekt i Visual Studio (du skal muligvis genstarte Visual Studio for at denne mulighed bliver synlig). **Denne projekttype skal du fremover vælge, når du skal arbejde med Raspberry Pi.**

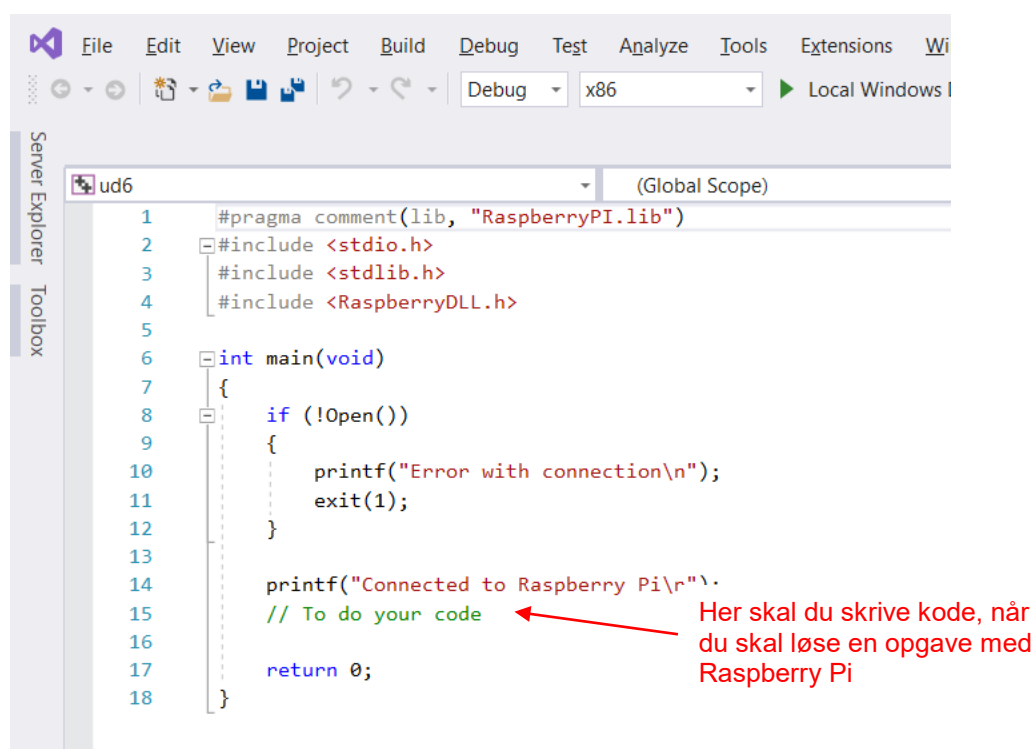


Når du åbner denne type projekt vil der allerede være tilføjet en cpp.fil:



Opgave fortsættes på næste side

Åben et RaspberryPi projekt i Visual Studio, som beskrevet ovenfor. Filen i projektet vil se således ud:



```
1  #pragma comment(lib, "RaspberryPi.lib")
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <RaspberryDLL.h>
5
6  int main(void)
7  {
8      if (!Open())
9      {
10         printf("Error with connection\n");
11         exit(1);
12     }
13
14     printf("Connected to Raspberry Pi\n");
15     // To do your code
16
17     return 0;
18 }
```

Her skal du skrive kode, når du skal løse en opgave med Raspberry Pi

NB! Det kode, som allerede står i filen, skal **altid** være til stede, når du arbejder med Raspberry Pi.

Exercise 2.1 Styring af lysdioder (Raspberry Pi + shield)

Til denne øvelse skal du bruge din Raspberry Pi med påsat shield.

I dokumentationen til Raspberry Pi:

Dokumentation til Raspberry Pi Zero med shield.pdf

kan du se, hvordan du monterer dit shield korrekt på Raspberry-printet og hvilket stik du skal anvende til at forbinde printet til din PC.

På sidste side i dokumentationen kan du se, hvilke funktioner du kan anvende i dine programmer til at kommunikere med din Raspberry Pi.

Brug nu nogle af disse funktioner til at skrive noget kode (i `main()`) i det projekt du oprettede i Exercise 2.0), der tænder og slukker en eller flere lysdioder. Du bestemmer selv hvordan den/de skal tænde og slukke (blinke, løbelys eller hvad du nu finder på - have fun 😊) (hint 2A).

Fortsættes næste side

Du skal nu teste dit program. Start med at forbinde din Raspberry Pi til din PC (hvis du ikke allerede har gjort det).

NB! Efter at du har forbundet din Raspberry Pi til din PC, skal du vente ca. 30 sekunder, inden du starter dit program med Ctrl-F5. Dette gælder hver gang USB-kablet har været frakoblet.

Exercise 2.2 Andengradsligning

Opret et **nyt** projekt (ikke til NXT). Skriv et program hvor brugeren skal indtaste koefficienterne A, B og C fra en 2. gradsligning (NB! **Kun** koefficienterne A, B og C fra ligningen $Ax^2 + Bx + C = 0$ skal indtastes) (hint 2B).

Herefter skal programmet beregne diskriminanten, derefter beregne rødderne og til sidst udskrive disse på skærmen. Husk, at værdien af diskriminanten bestemmer, om der er ingen, om der er en eller om der er to rødder. Disse tre tilfælde skal behandles **hver for sig** vha. en if/else-struktur (hint 2C).

Skriv programmet i pseudokode (gerne sammen med dine medstuderende) **inden** du begynder at skrive C kode.

Exercise 2.3 Erstatningsmodstand i elektrisk kredsløb - parallel

Opret et nyt projekt (gerne i samme Solution som opgave 2.2). Skriv her i et program (main) som kan beregne værdien af en erstatningsmodstand for to parallelt forbundne modstande og udskrive denne værdi på skærmen. Brugeren skal kunne indtaste værdierne af de to parallelt forbundne modstande.

Hvilke værdier (resistans) kan en modstand (resistor) have? Diskutér med dine medstuderende hvilken type du skal vælge til dine variable.

Skriv programmet i pseudokode (gerne sammen med dine medstuderende) **inden** du begynder at skrive C kode.

Exercise 2.4

Tilføj en while-loop i dit program fra Exercise 2.3 således, at indtastning af de to modstandsværdier og beregning af erstatningsmodstand og udskrivning af denne gentages for altid (programmet stopper aldrig, men kan selvfølgelig stoppes ved at lukke konsolvinduet – hint 2D).

Begynd med at tilpasse din pseudokode.

Fortsættes næste side

Exercise 2.5

Lav følgende ændring til dit program fra Exercise 2.4. Brugeren skal selv vælge om han/hun vil have foretaget en ny beregning. Dvs. at while-loopen ikke skal køre for altid, men derimod kun så længe brugeren indtaster et bestemt værdi – f.eks. 'y'. Dette kræver en ekstra indtastning fra brugeren, som skal foretages **efter** den første beregning og udskrift af erstatningsmodstanden (hint 2E).

Begynd med at tilpasse din pseudokode.

Diskutér med dine medstuderende hvilken type du skal vælge til den variabel som skal indeholde brugerens ekstra indtastning. Denne variabel skal initialiseres (dvs. du skal give den en værdi når du erklærer den) – diskutér med dine medstuderende hvorfor (måske kan I først se hvorfor, når I har skrevet ændringen i jeres program – men husk at diskutere det bagefter).

Ekstraopgaver:

Exercise 2.6 Erstatningsmodstand i elektrisk kredsløb - serie

Udvid dit program fra Exercise 2.5 således at det også beregner erstatningsmodstanden, hvis de to modstande er serieforbundet og også udskriver denne værdi på skærmen.

Begynd med at tilpasse din pseudokode.

Exercise 2.7

Kan du ændre dit program fra Exercise 2.6 således, at brugeren selv kan vælge om han/hun vil have beregnet erstatningsmodstanden for **enten** en parallel forbindelse **eller** en serieforbindelse? (hint 2F)

Begynd med at tilpasse din pseudokode.

Hint 2A: Prøv at indsætte disse 3 linjer kode i `main()` (på det sted hvor der står `///
To do your code`):

```
ledOn( 1 );  
wait(5000);  
ledOff( 1 );
```

Kør programmet og se hvad der sker.

Prøv så at skrive noget kode der tænder og slukker nogle af de andre lysdioder.

Prøv så at skrive noget kode som får en af lysdioderne til at blinke eller noget kode der skaber et løbelys eller noget helt tredje. Have fun 😊

Hint 2B: Først erklærer du variable til hver af de tre koefficienter. Derefter skriver du for **hver** indtastning noget kode lignende dette:

```
printf( "%s", "Indtast koefficienten A: " );  
scanf( "%f", &A );
```

Hint 2C: For at skelne mellem de tre tilfælde skriver du noget kode lignende dette:

```
if( D<0 )  
{  
    printf( "%s", "Der er ingen roedder\n" );  
}  
else if( D==0 )  
{  
    // her skriver du beregningen af x  
    // her udskriver du værdien af x sammen passende tekst  
}  
else  
{  
    // her skriver du beregningen af x1  
    // her skriver du beregningen af x2  
    // her udskriver du værdierne af x1 og x2 sammen med tekst  
}
```

Hint 2D: Din while-loop skal se således ud:

```
while( 1 )  
{  
    // her står den kode som du vil have gentaget  
}
```

Hint 2E: Din while-loop skal nu se således ud:

```
while( svar == 'y' )  
{  
    // her står den samme kode som i opgave 2.4  
  
    // her tilføjes en ekstra udskrift (F.eks. "Vil du foretage  
    // en ny beregning y/n") og en indtastning fra brugeren  
}
```

Hint 2F: Dette kræver en ekstra indtastning fra brugeren og at du tilføjer en if/else på et passende sted.