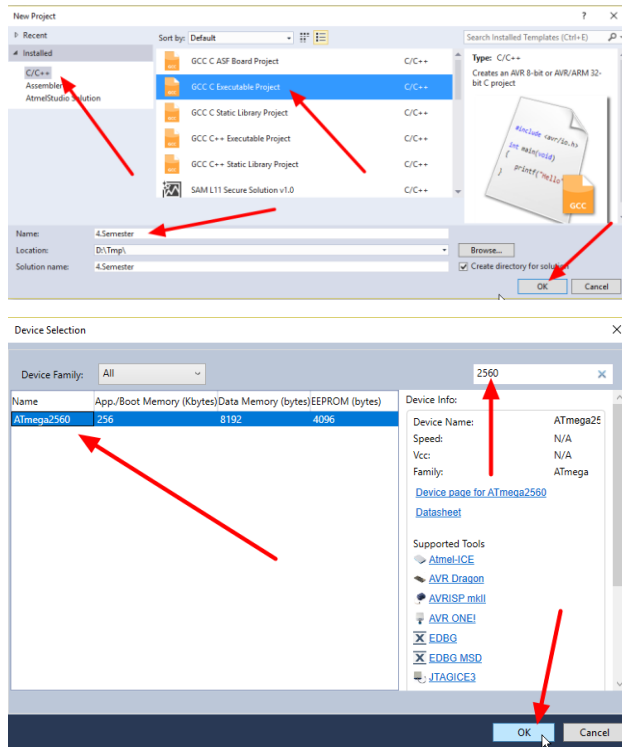


## Setup of Project

### Create Project

Open Microchip Studio.

Create a new project (here I give it the name *4.Semester*):

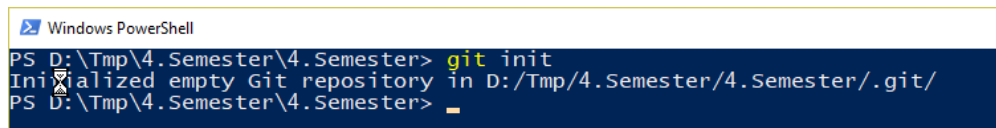


### Put Project under Git Control

Open a command prompt in the project folder (here */Tmp/4.Semester/4.Semester*).

Init the git repository with this command:

```
git init
```

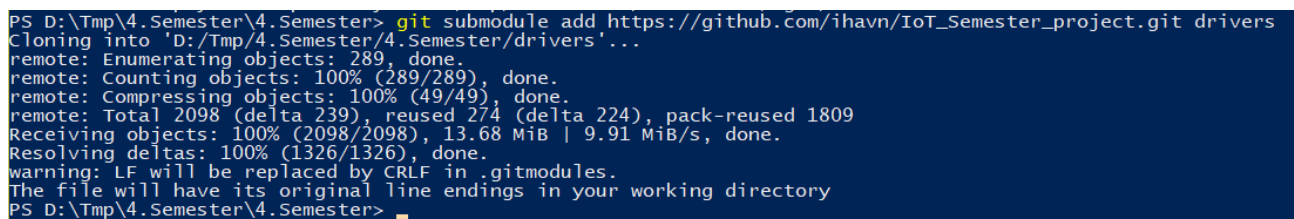


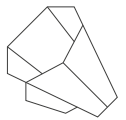
### Add Drivers as Git Submodule

Open a command prompt in the project folder (here */Tmp/4.Semester/4.Semester*).

Add a git submodule with the drivers into your project using this command:

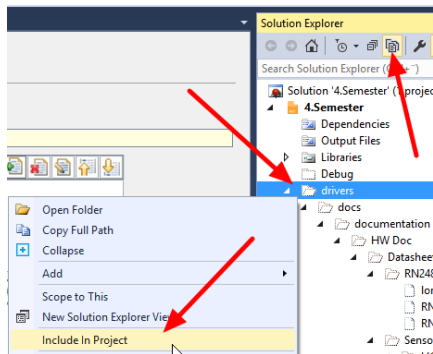
```
git submodule add https://github.com/ihavn/IoT_Semester_project.git drivers
```





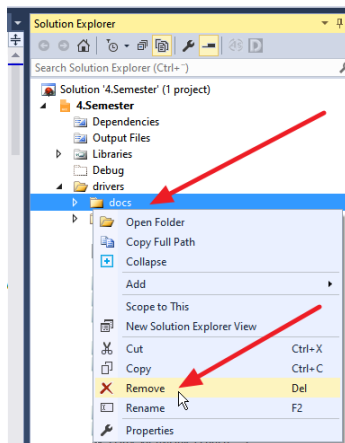
## Include the drivers in the Project

Click on the *Show All Files* icon in the *Solution Explorer* and right click on the *drivers* folder and then *Include In Project*

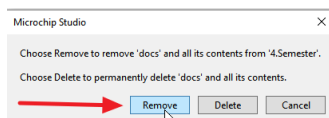


Now exclude the *drivers/docs* folder because it makes Microchip Studio slow.

Right click on the *docs*' folder and then *Remove*



And then click *Remove* not *Delete*

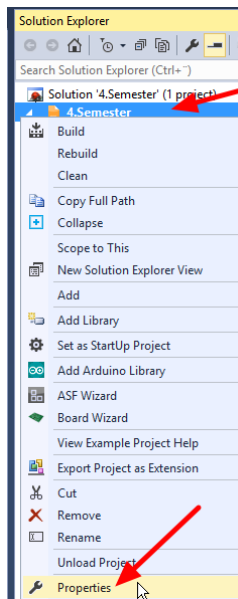
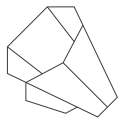


Then click again on *Show All Files* to deselect it.

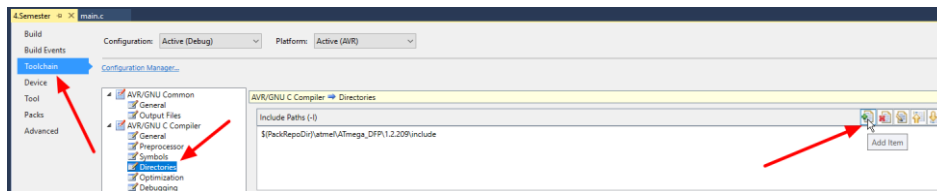
## Setup Project references to the drivers

### Include Path

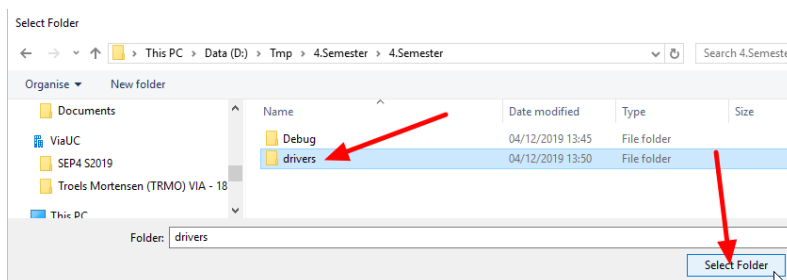
First right click on the project and select *Properties*



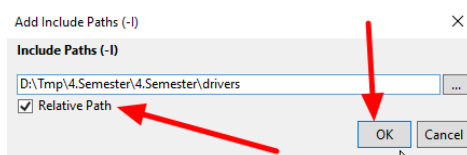
Then select *Toolchain*, *Directories* and *Add Item*



Browse to the *drivers* directory and *Select Folder*

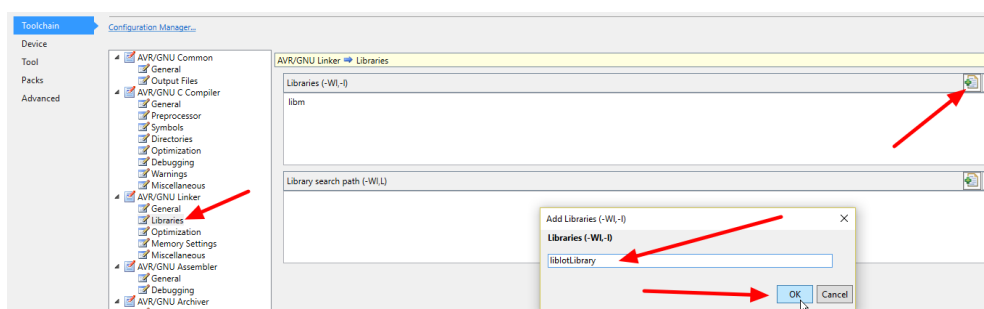


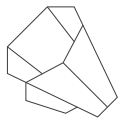
Make it a *Relative Path*:



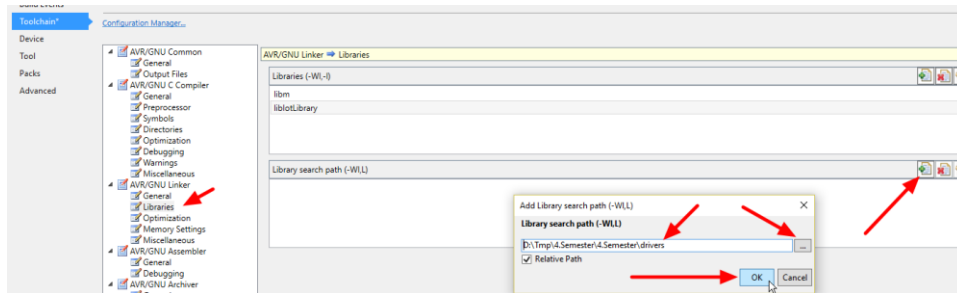
## Library Path and File

Click on *Libraries* and the add icon and put in the library name **libIotLibrary** be sure to spell it correct!!





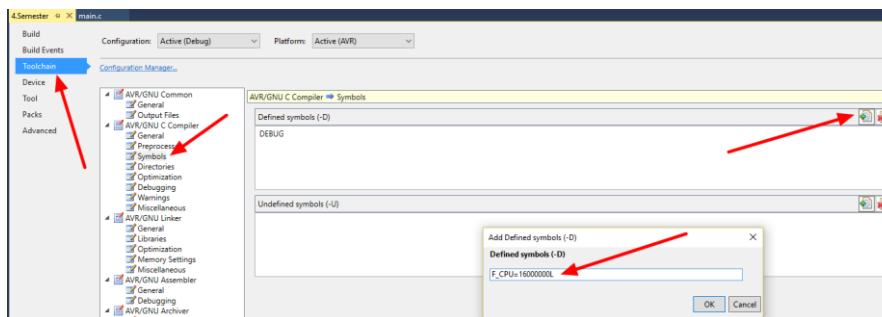
Now add the path to the folder where the library is located by clicking on the add icon and browse to the driver's folder



Then save the settings by saving the property file.

## Setup MCU Clock Frequency

Click on *Symbols* and the add icon and set  $F\_CPU=16000000L$



## Add FreeRTOS as Git Submodule

The last thing is to add the ATMEGA2560 port for FreeRTOS as a submodule. This is found and described here:

[https://github.com/ihavn/VIA\\_FreeRTOS\\_AVRMEGA](https://github.com/ihavn/VIA_FreeRTOS_AVRMEGA)

Here is how I have added the submodule

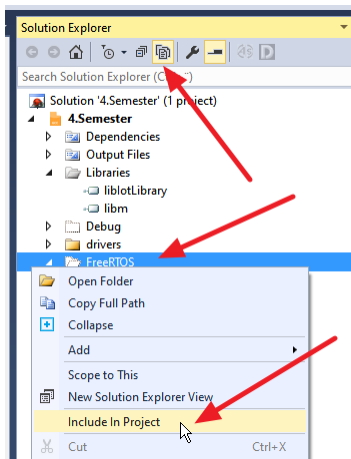
```
git submodule add https://github.com/ihavn/VIA_FreeRTOS_AVRMEGA FreeRTOS
```

```
PS D:\Tmp\4.Semester\4.Semester> git submodule add https://github.com/ihavn/VIA_FreeRTOS_AVRMEGA FreeRTOS
Cloning into 'D:\Tmp\4.Semester\4.Semester\FreeRTOS'...
remote: Enumerating objects: 77, done.
remote: Total 77 (delta 0), reused 0 (delta 0), pack-reused 77
Unpacking objects: 100% (77/77), done.
warning: LF will be replaced by CRLF in .gitmodules.
The file will have its original line endings in your working directory
PS D:\Tmp\4.Semester\4.Semester>
```

Finally include the FreeRTOS source in the project the same way as you did for the drivers (see Include the drivers in the Project).

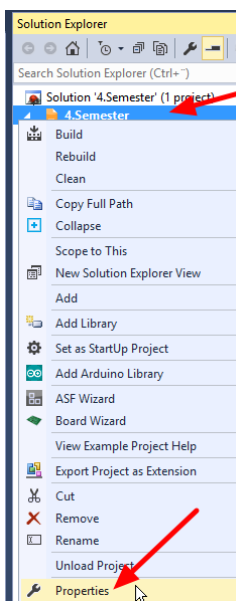
## Include FreeRTOS in the project

Click on the *Show All Files* icon in the *Solution Explorer* and right click on the *FreeRTOS* folder and then *Include In Project*

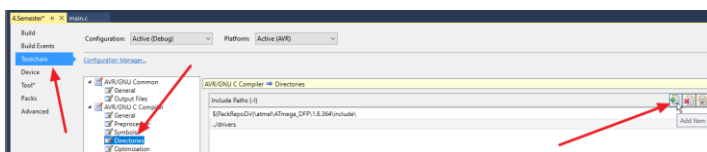


## Include Path

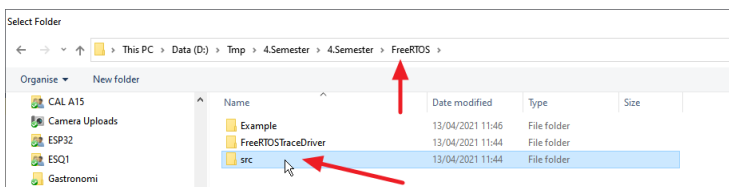
First right click on the project and select *Properties*



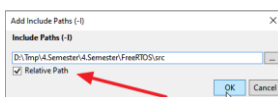
Then select *Toolchain, Directories* and *Add Item*

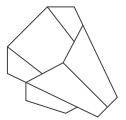


Browse to the *FreeRTOS\src* directory and *Select Folder*



Make it a *Relative Path*





## Example files

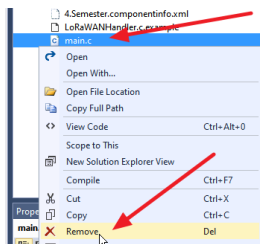
In both the FreeRTOS and drivers' submodules there are some example files that can be used as inspiration.

For this semester project you are only interested in the files found under the drivers *Example* folder.

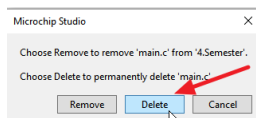
In the this C project you are interested in:

- LoRaWANHandler.c.example
- main.c.example

First delete the *main.c* file that is in the root folder of the project

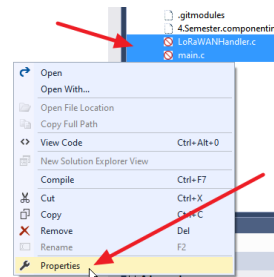


This time you must delete *main.c* completely!

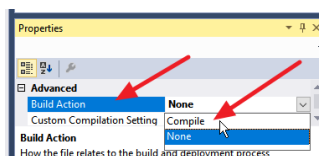


You can then copy the two example files to your projects root folder and then remove the *.example* extension from them.

Include them in the compilation by selecting both files and right click and the *Properties*:



Select *Compile* under *Build Action*



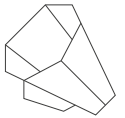
## First Run

Now it should be possible to build your project without any errors.

Connect the hardware you have been given.

Setup a serial terminal on your laptop and configure it to 57600,N,8,1. As serial terminal H-Term is recommended (<https://www.der-hammer.info/pages/terminal.html>).

When you run your program for the first time the out put to the terminal will contain a *HWEUI* hex string that you must email to Ib Havn ([ihav@via.dk](mailto:ihav@via.dk)).



```
Task2
Task1
Task1
Task2
Task1
FactoryReset >OK<
Configure to SPI >OK<
Set DevEUI >OK< 000430802552DE
Set DevEUI: 000430802552DE >OK<
Set OTAA Identity appEUI:XXXXXXXXXXXX appKEY:YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY devEUI:000430802552DE >3NVALID_PARAM<
Task1
Task1
Task1
```

I will then setup the LoRaWAN Network Server to make it possible for your device to connect to the LoRaWAN. You will receive and email with the *appEUI* and *appKEY* hex strings that you will need for your application.