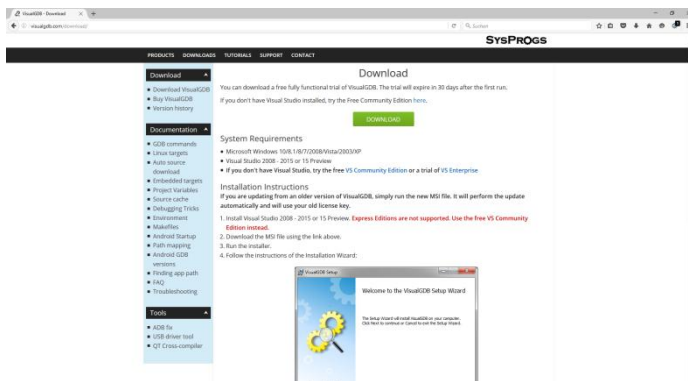


# VisualGDB Installation



1.

Visual Studio sollte installiert sein

→ Download starten

<http://visualgdb.com/download/>

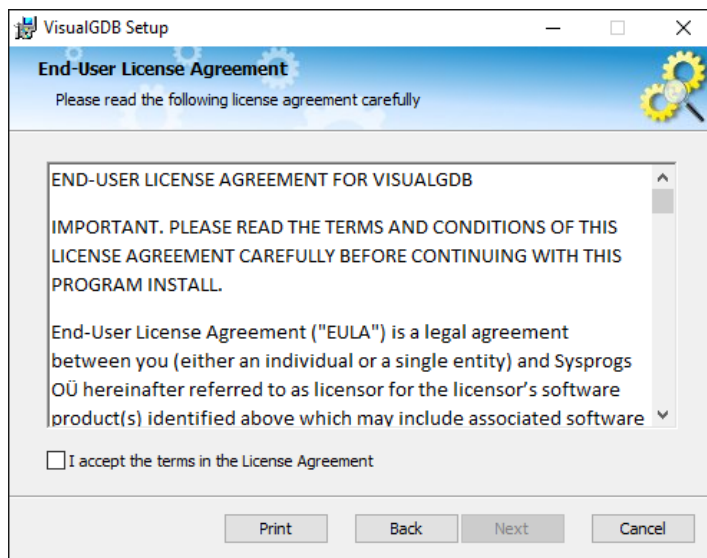
Download



2.

Setup starten, Fenster öffnet sich

„Next“ klicken

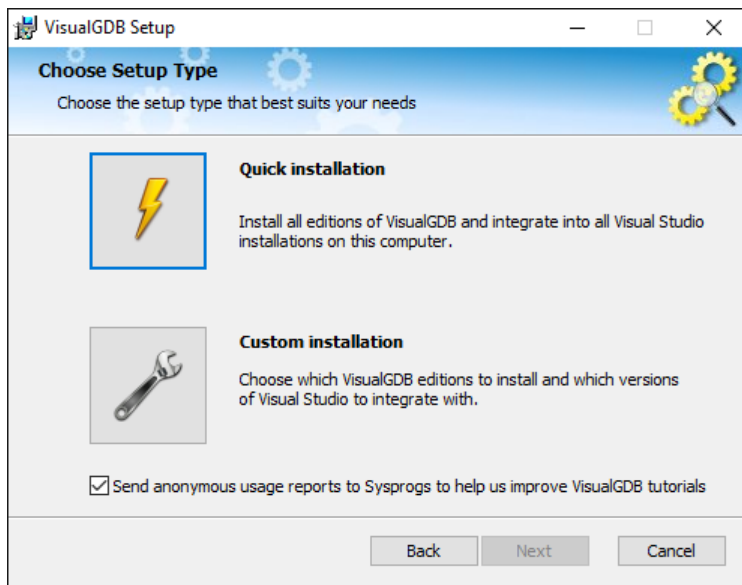


3.

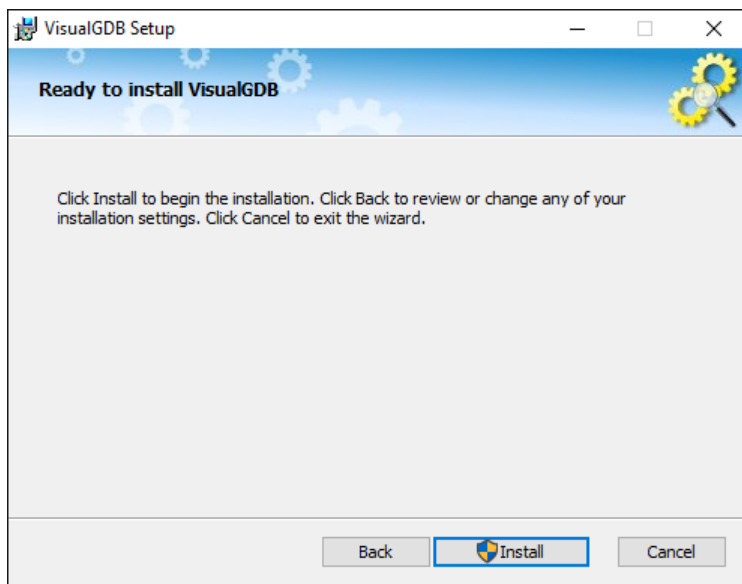
End-User License Agreement lesen

Häkchen bei „I accept the terms in the License Agreement“

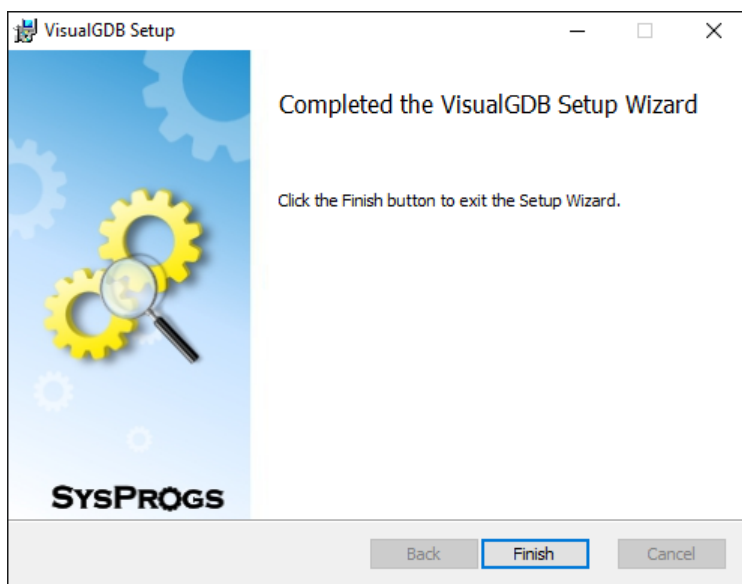
„Next“ klicken



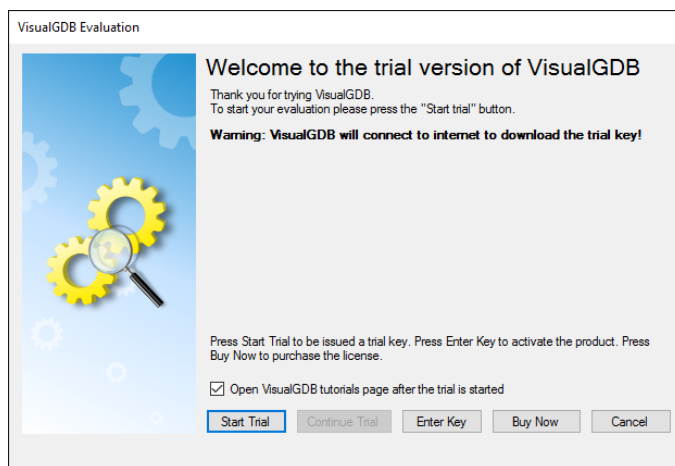
4.  
„Quick installation“ klicken



5.  
„Install“ klicken



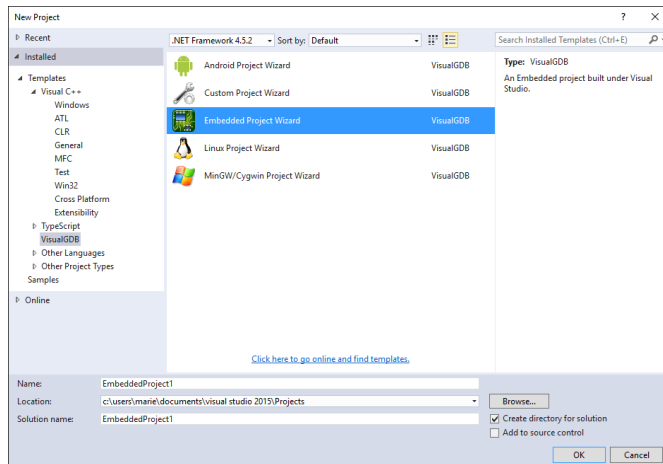
6.  
„Finish“ klicken  
Visual Studio öffnen



7.

„Start Trial“ klicken

# Erste Projekt Einrichtung



1.

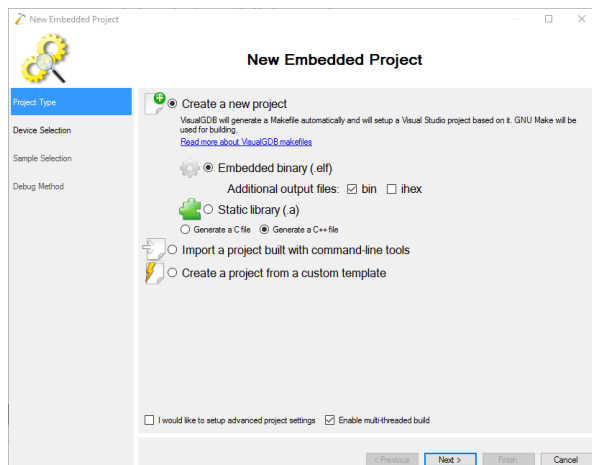
In Visual Studio (oben links): File → New → Projekt

Links „VisualGDB“

„Embedded Project Wizard“

Man kann Name, Speicherort, Solution name ändern – WICHTIG: Speicherort darf kein Leerzeichen enthalten!!!

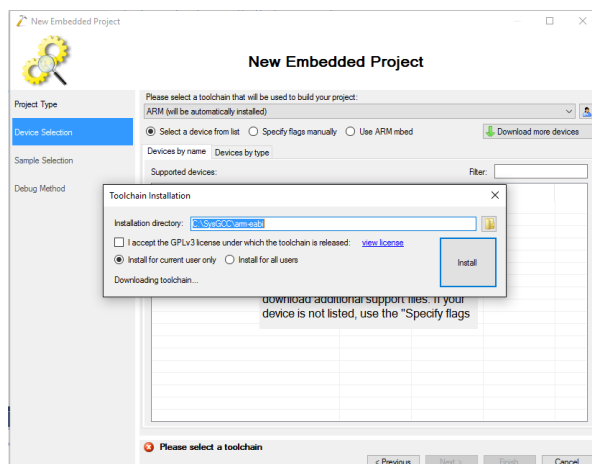
„OK“ klicken



2.

Haken bei Enable multi-threaded build → schneller kompilieren

„Next“ klicken

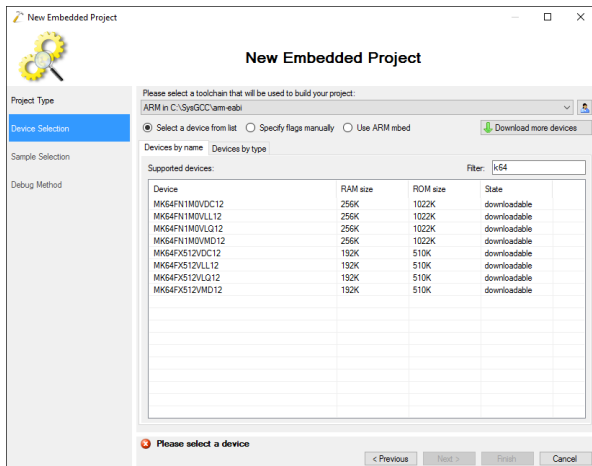


3.

Es öffnet sich ein kleines Fenster

„I accept the...“ Haken machen

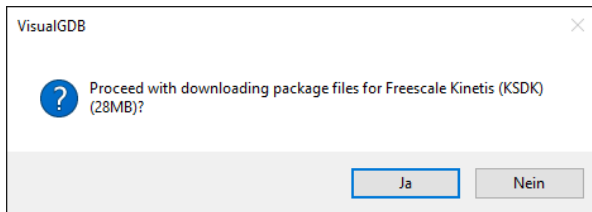
„Install“ klicken



4.

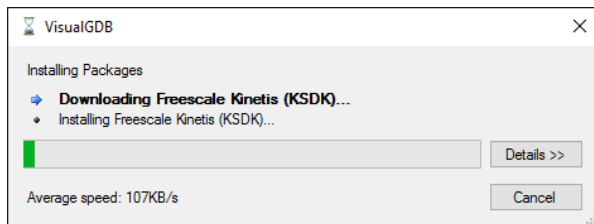
In der Liste muss man den richtigen Chip auswählen – im Filter kann man danach Suchen

In unserem Fall brauchten wir den MK64FN1M0VLL12



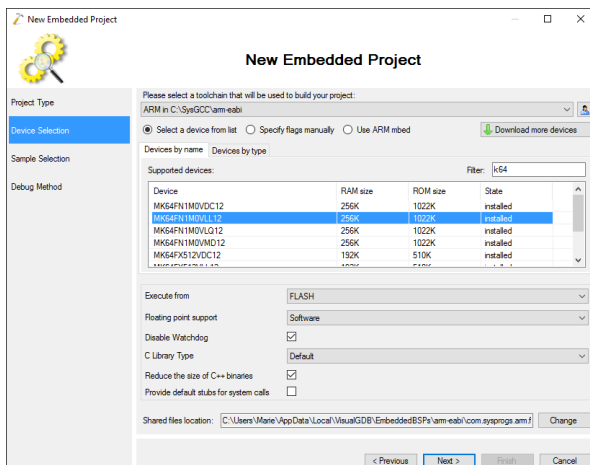
5.

„Ja“ – Download starten



6.

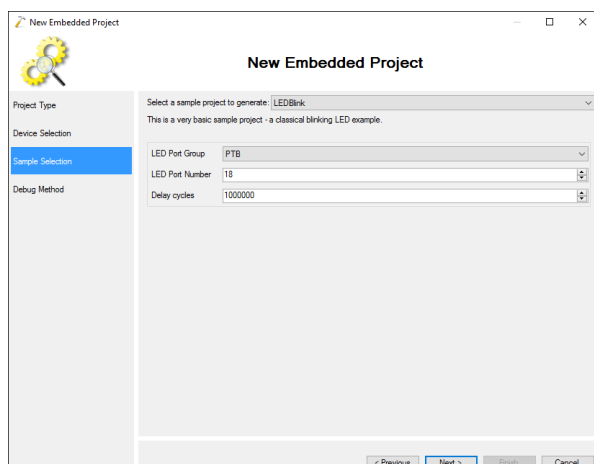
Fenster öffnet sich – Download startet (nur beim ersten Mal)



7.

Einstellungen lassen

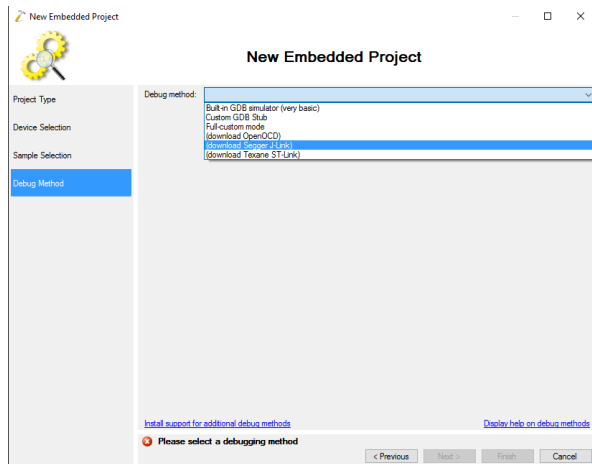
Execute from „FLASH“ lassen! Bei RAM gibt es Probleme mit Interrupts „Next“ klicken



8.

Ports stimmen nicht immer → richtig Einstellen

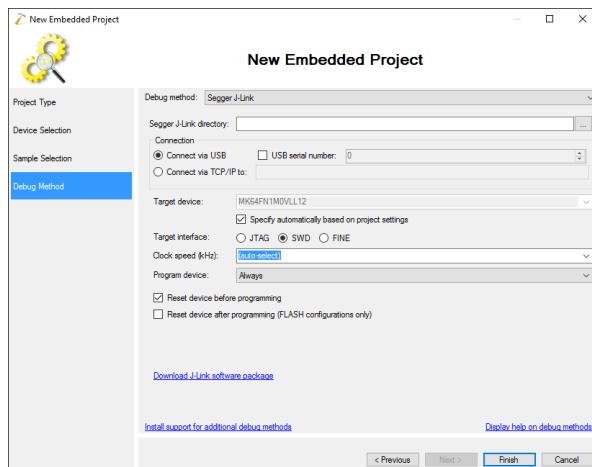
Delay cycles – Verzögerung beim Blinken



9.

Segger J-Link auswählen

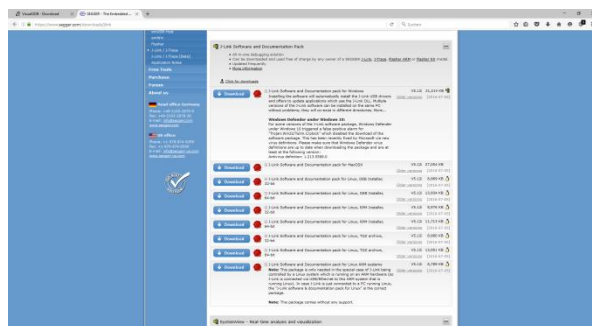
Fenster → „Ja“



10.

Haken bei Reset device after programming!!!

„Download J-Link software package“ anklicken



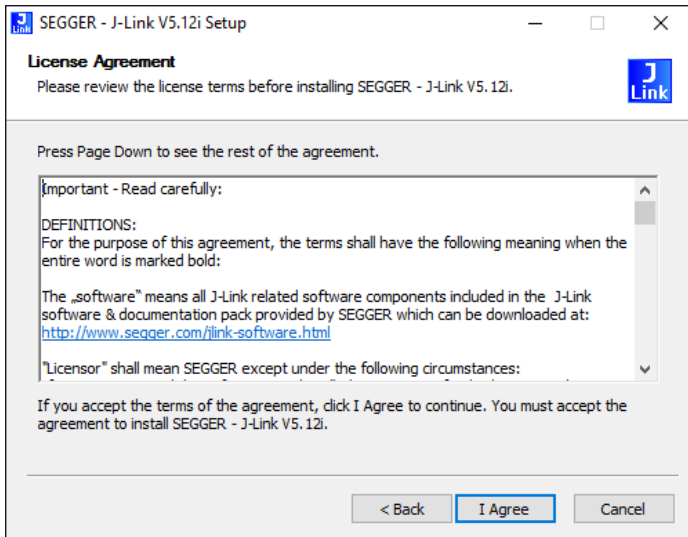
11.

„Click for Downloads“ anklicken → 1. Download starten → Lizenzbestimmungen akzeptieren → Setup starten



12.

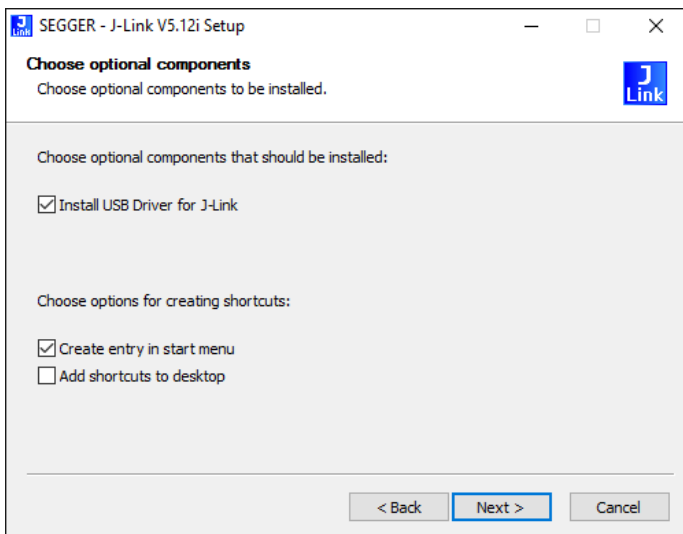
„Next“ klicken



13.

Lesen

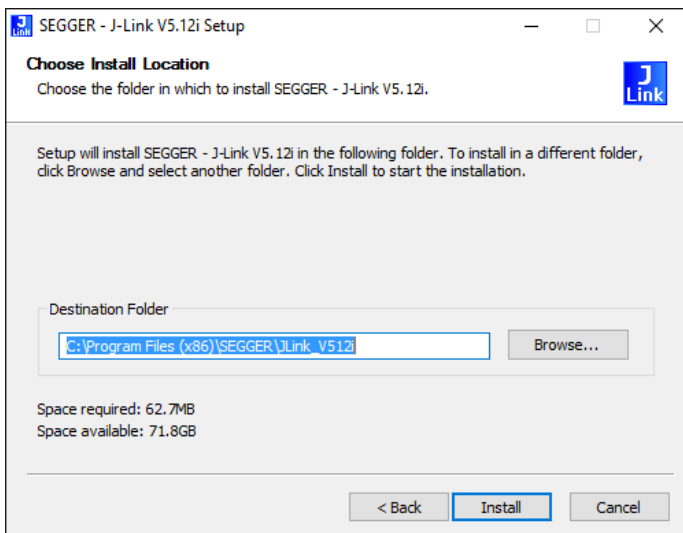
„I Agree“ klicken



14.

Haken lassen

„Next“ klicken



15.

Zielordner eingeben (kann auch beibehalten werden)

„Install“ klicken



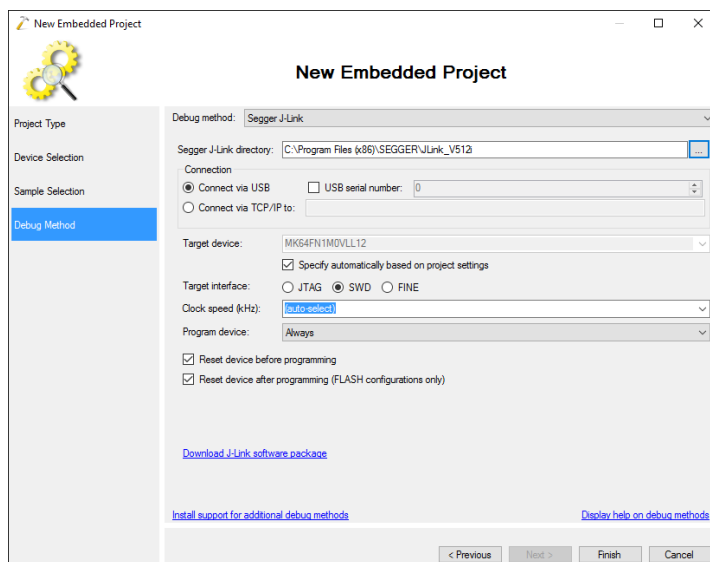
16.

„OK“ klicken



17.

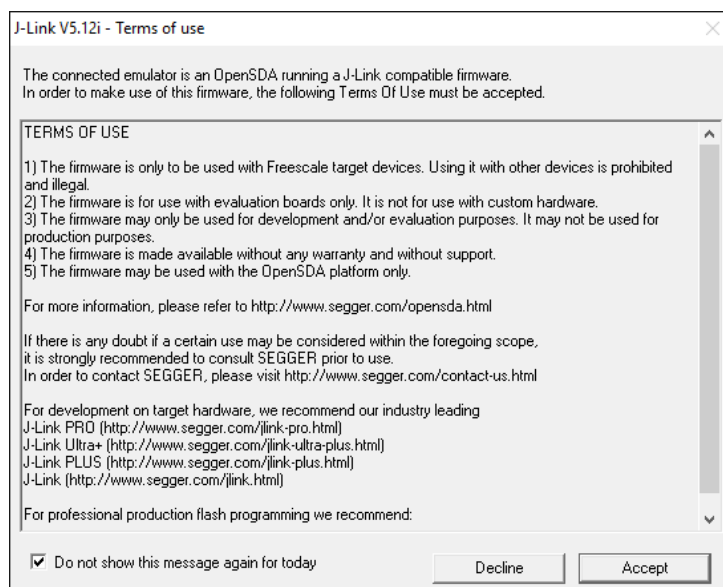
„Finish“ klicken



18.

Richtige Einstellungen wählen (siehe Bild)

„Finish“ klicken



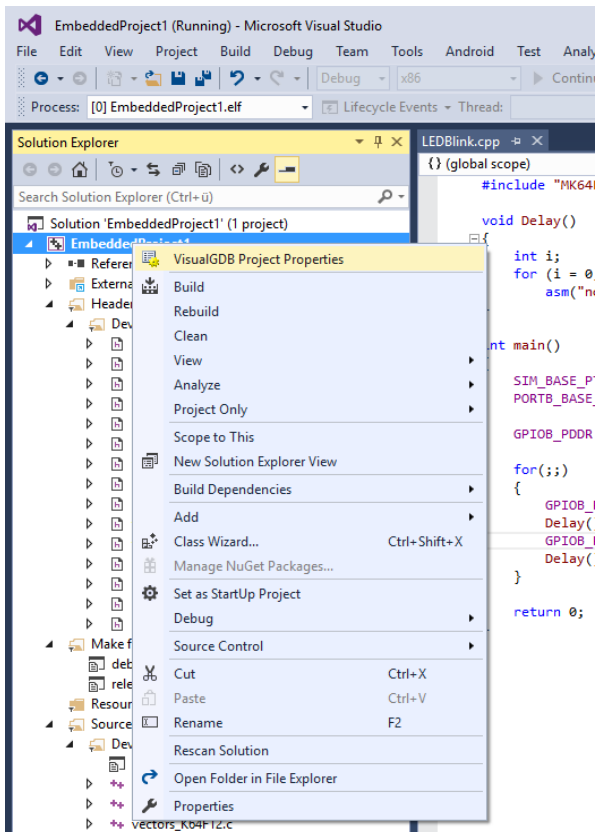
19.

Fenster erscheint bei jedem Start →  
Haken setzen, dann nur noch einmal am  
Tag

Wenn es Probleme gibt, Computer  
neustarten

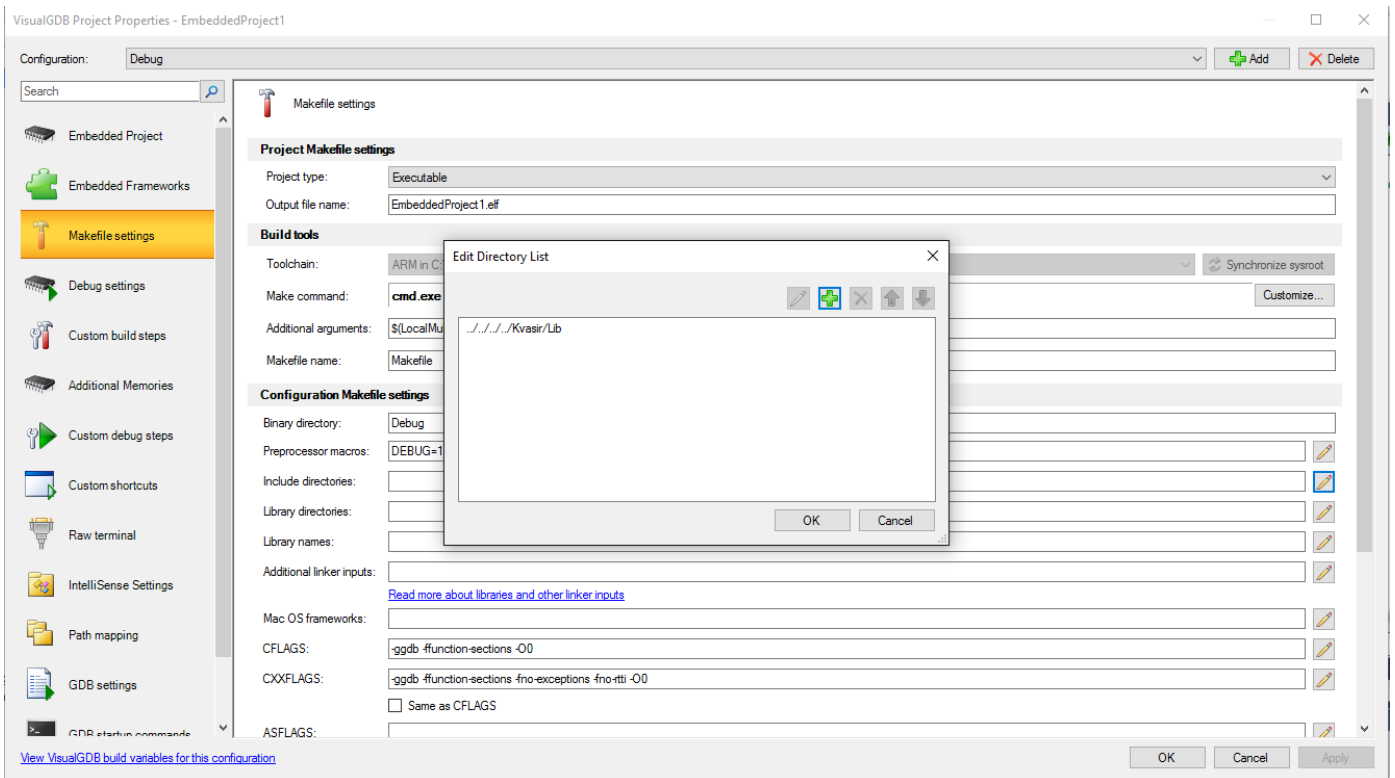


# Make it Kvasir

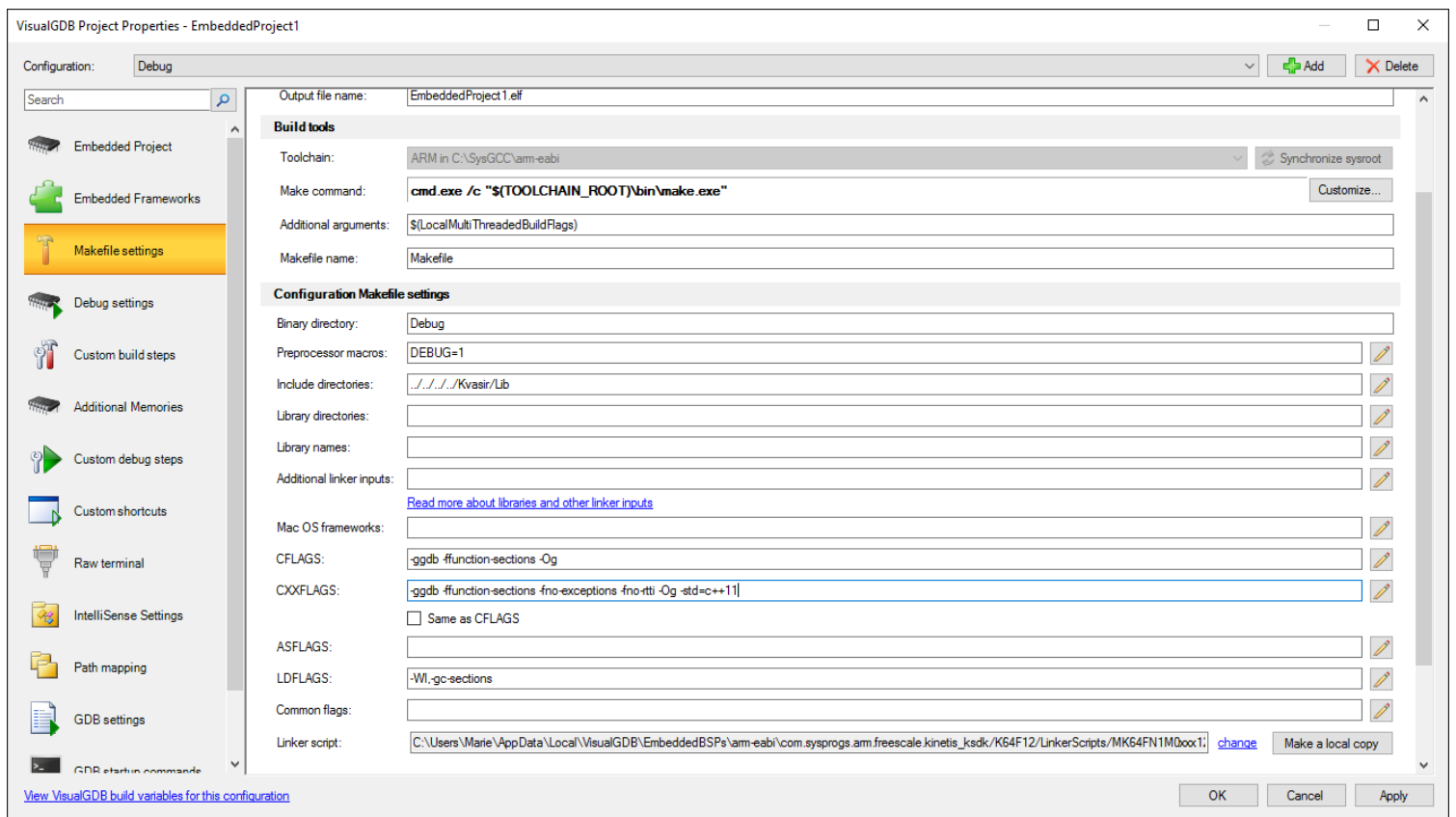


1.  
Kvasir Adresse Klonen in Git  
(<https://github.com/kvasir-io/Kvasir.git>)

2.  
In Visual Studio: Solution Explorer Rechtsklick auf Projekt  
→ VisualGDB Project Properties anklicken



3.  
Makefile Settings → Include directories – Lib in Kvasir Ordner



## CFLAGS

➔ -O0 ändern zu -Og

## CXXFLAGS

➔ -O0 ändern zu -Og

➔ -std=c++11 hinzufügen

```

// #include "MK64F12.h"
#include "Chip/MK64F12.hpp"

void Delay()
{
    int i;
    for (i = 0; i < 1000000; i++)
        asm("nop");
}

int main()
{
    // SIM_BASE_PTR->SCGC5 |= SIM_SCGC5_PORTB_MASK;
    apply(write(Kvasir::SimScgc5::PortbValC::v1));
    // PORTB_BASE_PTR->PCR[22] = PORT_PCR_MUX(1);
    apply(write(Kvasir::PortbPcr22::MuxValC::v001)); //v001=gpio
    // GPIOB_PDDR = 1 << 22;
    constexpr auto ledr = makePinLocation(Kvasir::Io::portB, Kvasir::Io::pin22);
    apply(makeOutput(ledr));
    for(;;)
    {
        // GPIOB_PSOR = 1 << 22;
        apply(set(ledr));
        Delay();
        // GPIOB_PCOR = 1 << 22;
        apply(clear(ledr));
        Delay();
    }

    return 0;
}

```

An Hand dieses Beispiels wird gezeigt wie das Blinki Programm für das frdm-k64f nach Kvasir portiert aussieht. Die ursprünglichen Befehle sind als Kommentare beibehalten.