**Schritte zur Installation von CodeBlocks in Verwendung mit Libelium Waspmote**

1. Code::Blocks herunterladen und installieren
2. Ordnerstruktur sollte schlussendlich folgendermassen aussehen (wir nun Schritt für Schritt erstellt):

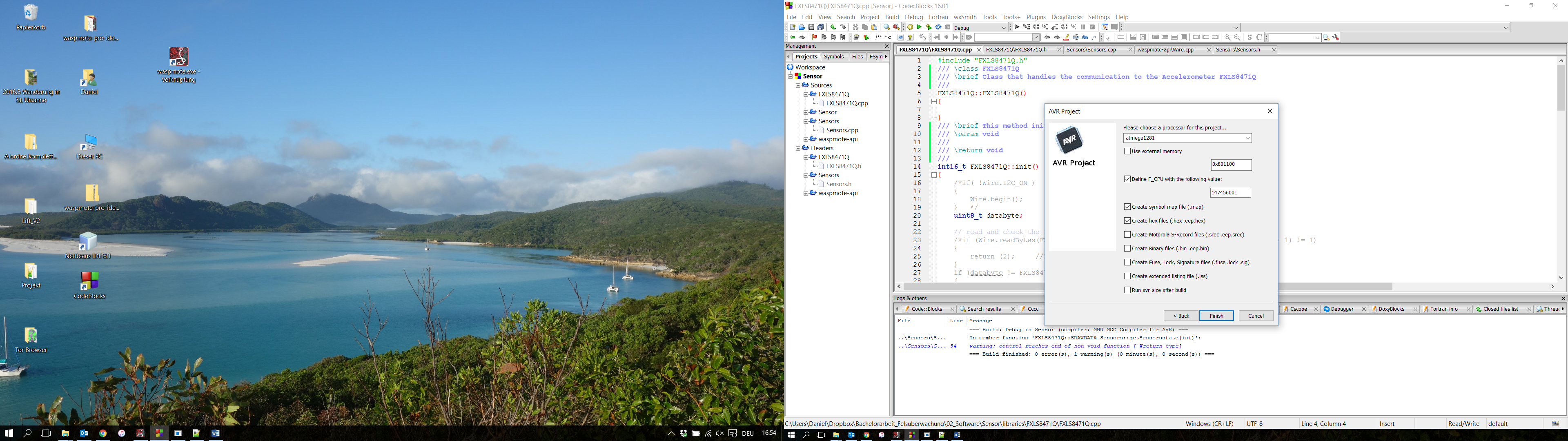
--Hauptordner

--Sketchordner  
 --Sketch von Waspmote IDE -> muss gleich heissen wie Ordner  
-- libraries ->beinhaltet alle Cpp- und h- Files, muss libraries heissen   
 **erstellen**  
 --avr ->AVR- Library kopieren von Waspmote IDE .  
 ../hardware/tools/avr/**avr**  
 --libraries ->Waspmote Library kopieren von Waspmote DIE.  
 …/**libraries**  
 --waspmote-api -> Waspmote API kopieren von Waspmote DIE.  
 ../hardware/waspmote/cores/**waspmote-api**

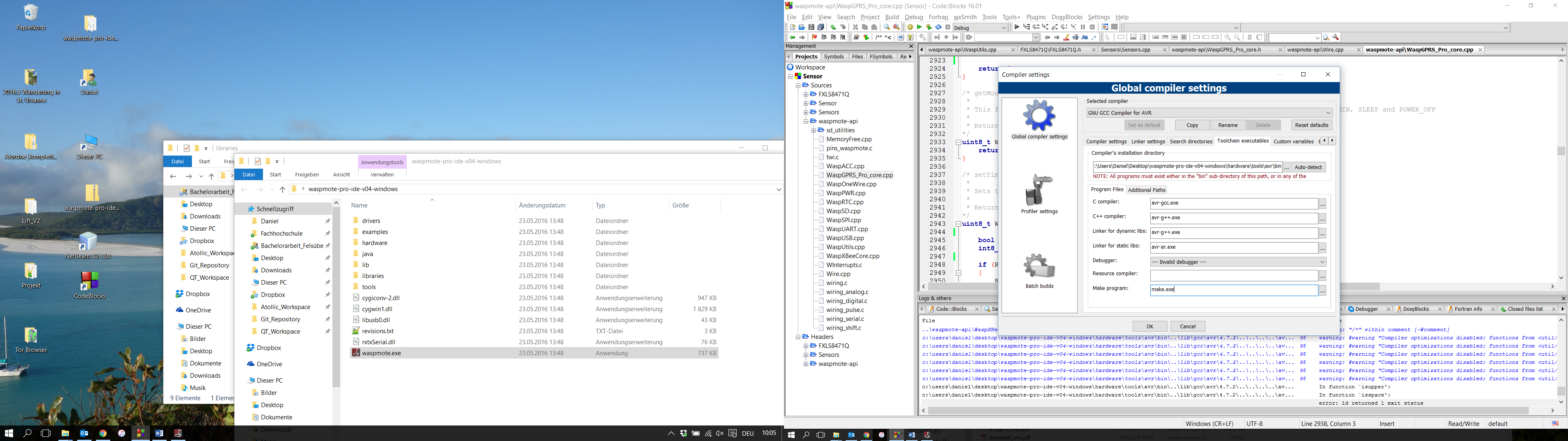
-- Projektordner ->Ordner mit AVR- Projekt von CodeBlocks  
 --Projekt.cbp  
 --main.c -> wird nicht auf Waspmote geladen, wird aber für Projekt   
 benötigt  
 --Ordner für Klasse -> für jede Klasse einen Ordner, beinhaltet cpp und h- Files  
 --Klasse.cpp  
 --Klasse.h  
 -- Ordner mit 2ter Klasse  
 …

1. Waspmote IDE herunterladen und unter Datei->Einstellungen den Pfad auf Hauptordner setzen
2. Waspmote Sketch erstellen in Hauptordner, die Waspmote IDE erstellt automatisch einen übergeordneten Ordner gleichen Namens (siehe Struktur oben)
3. In CodeBlocks AVR Projekt erstellen

Settings:



1. In CodeBlocks unter Settings->Compiler, den Compiler auf GNU GCC Compiler for AVR setzen und unter Toolchain executables den Pfad auf den AVR/bin Ordner setzen, der in der Waspmote IDE enthalten ist (siehe Bild unten). Ausserdem sollten die restlichen Einstellungen mit denen im Bild unten übereinstimmen.



1. Bei den Compilereinstellungen unter Search Directories alle Ordner unter libraries hinzufügen.
2. Bei Compilereinstellungen in den Compiler Flags das Flag Optimize generated code (for speed) -O setzen
3. Alle Files im Ordner waspmote-api zum Projekt hinzufügen (rechtsklick auf Projekt->Add files.. ), auch jene im Unterordner sd\_utilities (ausser m ain.cpp und WaspGPRS\_Pro\_core.cpp)
4. Das Main.c File, das beim erstellen des Projektes erzeugt wurde, sollte so aussehen:

#include <avr/io.h>

//define global variable for Waspmote serial id

volatile unsigned long \_serial\_id;  
  
int main(void)  
{  
 // Insert code  
 while(1)  
 ;  
 return 0;  
}

1. Projekt sollte nun ohne Errors kompilieren
2. In dem Waspmote IDE Sketch können nun die Klassen verwendet werden. Davon müssen die Header- Files eingebunden werden und über Sketch-> Library importieren die entsprechenden Ordner hinzugefügt werden.
3. In Waspmote IDE Sketch kompilieren und auf Waspmote laden