## Getting Started with ADuCM360 MiniKit and IAR Embedded Workbench

It is assumed the IAR Embedded Workbench 6.2x for ARM cores is installed and basic knowledge on usage. Also the general Getting Started Guide for the ADuCM360 has been read and the AduCM360 MiniKit is connected to the USB-SWD/UART-CONVZ with the J-Link-Lite to your PC as shown below.



## Start a simple 1<sup>st</sup> Blink Project

1.) Start IAR Embedded Workbench



2.) Create a new "Empty Project"

ile	Edit	View	Project	Tools	Window	Help
A	dd File	5				
A	dd Gro	ирла				
Ir	nport F	File List				
E	dit Cor	figurat	ions			
R	emove	:				
С	reate l	New Pr	oject			
A	dd Exi	sting Pr	oject			
10	ptions			41	T+-E7	

Create New Project	×
Tool chain:	
Project templates:	
Empty project	able
Description: Creates an empty project.	
	OK Cancel

- 3.) In target folder assumed the simple project files already are present there
  - main.c the simple software project to flash the LED on the ADuCM360 MiniKIt
    - ioaducm360.h the ADuCM360 MMR definitions for IAR C-Compiler
      - startupm360.c the simplified ADuCM360 C-startup-file
      - ADUCM360.icf the XLINK config file for ADuCM360

(attached in separate ZIP-file)



- 4.) Add the two required source-files to your project
  - main.c

\_

\_

\_

startupm360.c



Name *	Date modified	Туре	Size	
Ja settings	6/5/2012 10:54	File folder		
ioaducm360.h	11/29/2011 12:14	H File	18 KB	
main.c	6/5/2012 10:51	C File	3 KB	
startupm360.c	2/7/2012 12:16	C File	11 KB	
Date modified: 2/7/2012 12:16 - 6/5/201	Date created: 6/5/2012 10:44			
Size: 12.8 KB				
e: "startupm360.c" "main.c"			▼ Sou	rce Files (*.c;*.cpp;*.cc;*.h;*. 🕶
				Open Cancel
			4	

5.) Open main.c in IAR Embedded Workbench – it uses the local header-file – ioaducm360.h



6.) Setting the OPTIONs right - right click on the project name



Seneral Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-monitor J-Link/J-Trace TI Stellaris FTDI Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS100	Target Output Library	Configuration   Library Options   MISRA-C.2 DMI	00 <u>∢(</u> )
---	-----------------------	--	----------------

7.) Select the ADuCM360 as Target Device



8.) Select the ADuCM360 XLINK config file - ADUCM360.icf

Category:	Factory	Settings
Seneral Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-monitor J-Link/J-Trace TI Stellaris FTDI Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS 100	Config Library Input Optimizations Output List #define Linker configuration file Coveride default SPROJ_DIRstADUCM360.icf Edit Configuration file symbol definitions: (one per line)	

9.) Select J-Link/J-Trace as Debugger & Flash Programming

Category:		Factory Settings
General Options C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator Angel GDB Server IAR ROM-monitor J-Link/J-Trace TI Stellaris FTDI Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS100	Setup Download Images Extra Options Plugins Driver V Run to V-Ink/J-Trace V Run to V-Ink/J-Trace V Run to Device description file Override default STOOLKIT_DIRS\CONFIG\debugger\AnalogDevices\	ioaducM
TI XDS100	OK	Cancel

General Options     C/C++ Compiler       Assembler     Output Converter       Custom Build     Images       Build Actions     Images       Linker     Verify download       Debugger     Suppress download       Simulator     Suppress download       Angel     Verify download       GDB Server     Images       LAR ROM-monitor     STOOLKIT_DIRS\config Vlashloader\AnalogDevices	
TI Stellaris FTDI Macraigor PE micro RDI ST-LINK Third-Party Driver TI XDS 100	

10.) Set the options for the J-Link/J-Trace

neral Options //C++ Compiler					ļ	Factory Setting
Output Converter	Setup Conne	ction Brea	akpoints			
ustom Build	Reset					-
nker	Halt after b	ootloader			0	
ebugger						
Simulator	JTAG/SWD	speed		Clock setup		
Angel	Auto	22	_		10	-
GDB Server	Initial	32	kHz	CPU clock:	10	MHz
14 ink/1-Trace	C Fixed	32	kHz	SWO clock:	☐ Auto	
TI Stellaris FTDI					2000	kHz
Macraigor	C Adaptive	B				
PE micro						
RDI						
ST-LINK						
TT XDS 100						
11 X00100						

ategory.		Factory Settings
General Options		
C/C++ Compiler		
Assembler	Setup Connection Breakpoints	
Custom Build		
Build Actions		
Linker	Jenaro Jenaro.	
Debugger	C TCP/IP: IP address	
Simulator	IP address: aaa.bbb.ccc.ddd Serial no:	
Angel		
GDB Server	Interface JIAU scan chain	- Loose Lo
14 ink/1-Trace	O JTAG	e laigels
TI Stellaris FTDI	• SWD	
Macraigor	Scan chain contains nor	n-ARM devices
PE micro	Preceeding bits: 0	
RDI		
ST-LINK	Log communication	
Third-Party Driver	<pre>\$PROJ_DIR\$\cspycomm.log</pre>	
11 XDS100		

11.) Compile the project and start Debugger

He Eat Wew Project Debug Deasemby Junk Tools Window Help D ほ 夏 夏 香 冬 略 色 シ マ マイット 知道 D ゆ 命 俗 か 能 雅 咲 多 かか フ レ ジ ひ び び ご X	
5 🖌 SD6833 🗙	
(1 SWU)	
okspace × main e start um 360 c	• × Disassembly
ebug v	Go to 👻
THIS SOFTWARE IS PROVIDED BY ANALOG DEVICES INC. "AS IS!" AND ANY EXPRESS OR	Disasamhlu
Ness C.	Disassembly
ADJUCM360 V // MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE	UX9C: UXU-
Dimmain.c     // DISCLAIMED. IN NO EVENT SHALL ANALOG DEVICES INC. BE LIABLE FOR ANY DIRECT.	UxaU: UxU
He Startupm3b // INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES	Uxa4: UxU
Output     // ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE	0xa8: 0x0
// POSSIBILITY OF SUCH DAMAGE.	0xac: 0x0
	0x0: 0x0
// YOU ASSUME ANY AND ALL RISK FROM THE USE OF THIS CODE OR SUPPORT FILE.	0xb4: 0x0
	0x0: 8dx0
// IT IS THE RESPONSIBILITY OF THE PERSON INTEGRATING THIS CODE INTO AN	0xbc: 0x0
// APPLICATION TO ENSURE THAT THE RESULTING APPLICATION PERFORMS AS REQUIRED	0xc0: 0x0
// AND IS SAFE.	0xc4: 0x0
	0xc8: 0x0
// Module : main.c	0xcc: 0x0
// Description :	0xd0: 0x0
// Date : June 2012	0xd4 · 0x0
// Version : v1.00	0xd8 0x0
// Changelog : v1.00 Initial	0xdc: 0x0
	int min (mid)
// Flash LED GPIO P1.3 on ADuCM360 MinikIt	inc wain (void)
	main.
<pre>#include "ioaducm360.h"</pre>	Uxeu: Uxb
	T3CON = U;
	0xe2: 0x4
	0xe4: 0x2
void Delay(void);	0xe6: 0x8
	GP1CON = 0;
//=====================================	0xe8: 0x4
// main demo function	0xea: 0x2
	0xec: 0x8
Spint main (void)	GP10EN = (1 << 3
	0xee: 0x4
	0xf0: 0x2
// Initialise	0xf2: 0x7

12.) Run the project – the red LED on the MiniKit should flash



13.) Further-on you can remove the comment in line 40 of the main.c function This changes the default 1MHz core clock of the ADuCM360 to 16MHz and you will see after compiling and downloading the project again, the LED is flashing much faster

```
main.c * startupm360.c
      10 // YOU ASSUME ANY AND ALL RISK FROM THE USE OF THIS CODE OR SUPPORT FILE.
      11 //
      12 // IT IS THE RESPONSIBILITY OF THE PERSON INTEGRATING THIS CODE INTO AN
      13 // APPLICATION TO ENSURE THAT THE RESULTING APPLICATION PERFORMS AS REQUIRED
      14 // AND IS SAFE.
      15 //
      16 //
               Module
                               : main.c
      16// Module : main.c
17// Description :
18// Date : June 2012
19// Version : v1.00
20// Changelog : v1.00 Initial
      21 //
               Flash LED GPIO P1.3 on ADuCM360 MiniKIt
      22 //
      23 //
      24 //=
      25 #include "ioaducm360.h"
      26
      27 //=
      28
      29 void Delay(void);
      30
      31 //=
      32 // main demo function
      33 //
      34 int main (void)
      35 {
      36
      37 // Initialise
                                        // Turn off watchdog - is enabled by default after RESET
      38 T3CON = 0;
       39
      40 // CLKCONO = 0x0000; // 16 MHz
      41

      42
      GP1CON = 0;
      // All Pin GPIO

      43
      GP10EN = (1 << 3);</td>
      // LED on MiniKit on GPIO P1.3 output enabled

      44
      45 // Main Loop
      46 while(1)
      47
           {
              GP1TGL = (1 << 3); // toggle LED on GPIO P1.3
      48
      49
             Delay();
      50
           }
      51 }
      52
```