# Introduction to the IDE

# 1 Introduction

The IDE that we are using is called SourceBoost, and is available on all of the machines in the ELC. There is also a free version which has some limitations that you can download and run on your personal computer.

The steps in programming a microcontroller using the IDE are as follows:

# 1.1 Start the Software

Starting the software is done like any standard windows software.

# 1.2 Create or Open a Project

SourceBoost uses a "container" called a project to keep all of the details of your program together. These details include the file(s) that the program uses, specifics about the target processor, and other information.

When creating a new project, I would avoid using the wizard until you are more familiar with the IDE and C programming.

# 1.3 Add files to the project

All files and libraries have to be added to the project. Do not add header files to the project. They are included via the #include statements in your code.

SourceBoost includes an editor that can be used to edit and create programs, but you can use any code editor of you like.

# 1.4 Compile your project

With all of the files added to the project, it is time to use the  $\mathbb{C}$  compiler to compile your project. The bottom half of the screen will show the results of this process.

#### 1.5 Link your project

Once the project has been successfully compiled, the next step is to link the project using  $\mathfrak{I}$ . This program, run from inside the IDE, will take the output of the compiler and translate it into a HEX file.

# 1.6 Program your microcontroller chip

The next step is to program the hex file that was generated by compiler and linker into the microcontroller. The name of the hex file will be the name of your project with the file extension .hex. When the programmer software comes up, you may have to load the

appropriate hex file before downloading. This is a separate piece of software, although it is launched from the IDE. This program sends the hex file to the hardware programmer mentioned above, which writes the program into the memory on the microcontroller.

### 1.7 See if the program worked

Once the hex file is downloaded, press the reset button on the development board. The program will start from the beginning and run to the end. Note that the program usually will start running without a reset, but that may not always be the case.

# **2 Detailed Instructions**

The following is a cookbook approach to the steps necessary to run the IDE.

- 1. Determine where you are going to keep your code, libraries, etc. This needs to be a directory that you have permissions on, and can be in your AFS space, on a memory stick, etc. This document is an edited version of one used in EE 20222 and it assumes there is a directory called workfolder extant for this purpose.
- 2. Run the SourceBoost IDE. The program can be found in Start  $\rightarrow$  All Programs  $\rightarrow$  SourceBoost  $\rightarrow$  SourceBoost IDE.
- 3. Create a new project in your working directory. To do this, go to the Project menu of the IDE, and select new.

🐄 SourceBoost II	DE
File Edit View	Project Build Settings Plugins Help
🗋 🖻 🖉 🗐	Quick 🎁 😰 💥 📲
Workspace	New
🖃 💼 C:\lab224-2	Open
prog.c	Wizard
⊡ ⊡ ⊡ Dutput	Add File(s)
prog.	1 C:\lab224-2005\prog2\progc
E prog.	2 C:\lab224-2005\prog\progc
	3 C:\\abdev\I2C test\I2Cc
	4 C:\\ab224-2005\\I2Cc
	5 C:\lab224-2005\\countbc
	6 C:\lab224-2005\\oscc
	7 C:\lab224-2005\\spitestc
	8 C:\\ab224-2005\\SPIc

This will open up the dialog box shown below. Change the directory to point to some folder and choose a name for your project. (I chose in this example to call mine lab2rms.)

Open			? 🛛
Look jn: 📔	workfolder	• • •	• 📰 📩
File <u>n</u> ame:	lab1-RMSc		<u>O</u> pen
Files of type:	Prj Files(*c)	•	Cancel

A new project dialog box will appear. You should choose to create an empty project.

New Project	
Create new project:	ОК
empty with empty source file with empty source and header files	Cancel
Toolsuit: BoostC 🗨	

4. Add your file to your project.



After adding the file, it should show up in the left column.



Notice the workspace display. The project name is listed in bold on the first line. Below it you can see the file that I added to the project.

5. Add the EESDlib.c to the project.



6. Double click on the name of your C file, and it will show up on the right hand part of the screen, where it can be viewed and edited. Note that the editor uses colors to indicate various different types of information in a C program.

🖇 SourceBoost IDE - [C:\Pic soft	tware\boardtest\testprog.c]
File Edit View Project Build	
□ ☞ 🛛 🖉 🐰 🖻 🖻 🎒	) <b>E A 36 ( 1 1) A 1 A </b>
Workspace 🔺 🗙	testprog.c
C:Vic software\boardtest	<pre>/************************************</pre>

7. Note that there are a number of settings under options that should be correctly pre-set for you but with multiple students using each setup, there are no guarantees. . The correct setting are as follows:

Settings -> Toolsuit		_
	Active Toolsuit Selection	$\mathbf{X}$
	Active Toolsuit: OK BoostC Cancel	
Settings -> Target	Target	



# Settings -> Targe

BoostC compiler options	$\mathbf{X}$
Compile Options Tools Debugger	
Compiler&Linker directory:	
C:\Program Files\SourceBoost\	<b>.</b>
Programmer: C:\diypgmrs\MicroPro.exe	
	<u>G</u> et Default
	<u>S</u> et Default
	OK Cancel

Note that the specific location of the programmer software may be different than what is shown because of the installation setup in the IDE.

#### Settings -> Build

Set the options you want to happen when you click on build  $\mathfrak{B}$ . If you include program, it will automatically launch the programmer software.

×
OK
Cancel
Set Default

- 8. Compile your project. Simply click on the gothic ℂ in the top toolbar. In the bottom window, you will see the output from the compile, and if you entered the program correctly you should see no errors. (Note that there will be a number of warnings! These can be safely ignored.) If there are errors, you can double click on the error and it should take you to the offending line. (Note that it is not always possible for the compiler to figure out exactly where the error is, but it should be close to the line given.)
- 9. Link your project. Click on the gothic  $\mathfrak{X}$ , and the linker will link the project. (Note that if you had the build options as shown above, clicking on the gothic  $\mathfrak{B}$  would have done both these steps.)
- 10. Compiling and linking produces three output files. An assembly file (.asm), a list file (.lst) and a file that can be downloaded to the microcontroller (.hex).

Download the hex file to the microcontroller board. Click on the gothic **P** and the programmer will start and will appear as below. Note the window that says target. This has to be set as shown. Once it is set, it should remain so.

ile	Program	mer <u>O</u> pti	ions <u>H</u> e	зiр								
) ∈		ROM DA								•	$\odot$	
	0000:	EF62 F		209 24	408 88	07 800	0 0012	0889	b			
	0010:	OFFF A			000 2E			0012	2		1	
	0020:	A89E D		034 6H		12 A2A		98AB	4			
	0030:	88AB 6				FE 50A	E 6EOC	0012				
	0040:	6A16 6	5A17 6A	A19 6 <i>8</i>	AlA 6A	18 B81	8 0012	90D8				
	0050:	3619 3	61A 36	512 36	613 36	16 361	7 5014	5C16				
	0060:	5015 6	5217 50	C17 E3	305 50	14 5E1	6 5015	5A17				
	0070:	8019 2	A18 D7	7EA OC	012 50	09 6EA	F 500A	6 KBO				
	0080:	OE24 6	SEAC OF	890 6B	EAB OE	EB 169	6 848D	OEOS	\$		$\ominus$	
	0090:	6 <b>E</b> B8 0	012 6A	A33 50	DOA 6E	EA 500	9 2433	6EE9	33.			
	00A0:	52EF B	4D8 00	012 50	DOA 6E	EA 500	9 2433	6EE9				
	00B0:	2A33 5	OEF 6B	834 BC	C10 F0	00 D7E	D 0012	3809	3.4			
	0000:	OBOF 6	SEOB OF	EOF 14	40B 6E	OC OE3	0 240C	6EOA	0			
	00D0:	OE39 6	540A DC	003 OB	EO7 24	OA 6EO	A 500A	6 <b>E</b> 34	94			
	00E0:	EC10 F	7000 OB	EOF 14	409 6E	OB OE3	0 240B	6EOA	0	-		· · · · · · · · · · · · · · · · · · ·
∍ [	сом х						Ready				÷	
́. Эг												D
2			_						1		w i	Chip Selector
	🕒 🕒 L	oad	🛯 📳 Me	erge		🕌 <u>P</u> rogi	am	?{]⊻erify		CALIB		18F4620 💌
												1011020
	🐴 Be			- 1		B Bea		Blank	1 .	Fuses		X Cancel

Depending on the USB port that you connect the programmer to, the port setting in the programmer may change. You may have to do some searching through port numbers. Setting the port is under the file command.

					•				
🖬 DIY K150 v1	41204	<pwn< th=""><th>4620.hex&gt;</th><th></th><th></th><th></th><th></th><th></th><th></th></pwn<>	4620.hex>						
File Programmer	Options	Help							
🗁 Load	Ctrl+L								
🚽 Save	Ctrl+S	209	A4D8 EF07	<b>F</b> 000	0012	0889	b		
🕒 Merge	Ctrl+M	FOS	F000 2E09		FOOO				1 🕨 🧱 🕶 🔛 🎬
Refresh	Ctrl+R	034	6EAD 0012	A2AB	D004	98AB	4		
Clear	Carrie		AA9E D7FE						
_ Clear		A19	6A1A 6A18						90 III III III III III III III III III I
Programmer	,	612 C17			5014 5015				
Port		7EA		6EAF	5015 500A				
Turn Serial Off	Ctrl+O				848D		\$		
Turn Senai On	Cm+0	A33	500A 6EEA	5009	2433	6EE9	33.		
🗿 Recent Files	,	012	500A 6EEA				3.		
MPLAB®	,				0012		3.4		
INTERDO		EOF			240C 500A		0 94		
Exit		EOF			240B		94	-	
		BOL	1405 0205	0200	2400	OBOA			and the second second and the second s
COM 4			K15	) boam	d cor	nected		$ \rightarrow $	
) r									0
			1	_			1 0		<sup>™</sup> Chip Selector
🕞 Load		Merge		Program	n _	?{ ⊻erify	CALIE		18F4620 🔽
									_
🔁 Refresh		<u>S</u> ave		<u>R</u> ead		Blank	: 🕞 Fuse:		X Cancel
		<u>Dave</u>		<u>) Tieau</u>					· · · · · · · · · · · · · · · · · · ·
€								- e	⊖ 1:37:22 PM

There are a number of options for the programmer. I usually have my set as shown below which tells the device to ignore the "blank check" (don't bother to check if there is a program already in the microcontroller.

2007

₿÷i D	IY K15	0 v1	1204 <pwm4620.hex></pwm4620.hex>	
Eile	Program	mer	Options Help	
Ð		RO	Edit HEX Code Ctrl+H	
	0000:	EF	Edit Chip List F000 0012 OKFS	р ь
	0010:	OF	Fuse Value EF07 F000 0012	
	0020:	A8:	Calibration Word(s) A2AB D004 98AE	
	0030:	88,	SOAK 6KOC 0012	
	0040:	6A.	C Read Chip Info B818 0012 90D8	
	0050:	36.	Ignore Blank Check 3617 5014 5016	
	0060:	50. 80.	Fly Window Ctrl+F 5816 5015 5817	
۰D	0070: 0080:		Auto Update Files 1696 8480 0808	
	0080:		E ICSP Mode 5009 2433 6889	
	0090:	521	EICSP Mode 5009 2433 6KKS	
	00B0:	24	Auto Program 5009 2433 6KKS D7KD 0012 3809	
	00C0:		Cal Program Options 🔸 0830 2400 680A	
	00D0:	OE	EEPROM Override 680A 500A 6834	4 94
	00E0:	EC.	0E30 240B 6E0A	A0 💽 📲 🔜 🖉 📲 🗌
ə	COM x		Ready	е 📃
Ð				← Chip Selector ←
	ا ڪ	oad	Merge 🛛 🤽 Program 🛛 ?{] 🛽	/erifu
		-		18F4620
	<b>•</b> ••• ••			Nuclear Street
	🐴 Re	erresh	Bave Bead	
$\ominus$				⊖ 8:44:47 AM ⊖

If you haven't told the programmer to ignore the blank check, is will ask you if you really want to program over what is in the chip.

9	0000: 0008: 0010: 0018: 0020: 0028:	0000 ( 1D03 2	3FFF 0000 0000 2807 0000	0000	08B9 1903 0008 30F9 0000 v 0000 0000 0000 0000 0000 0000 0000 0000 0000 3EFF 0000 0000 0000 0000 3EFF		0	
Ð	0028: 0030: 0038: 0040: 0048: 0050: 0058: 0060: 0068: 0070:	30FA ( 2003 ( 0508 ( 00A7 : 1009 : 00B9 : 0508 ( 2003 ( 0088 :	00B9 0BA0 0427 30F0 1489 2003 0420 0008	2003 2821 0088 052 1089 2851 0088 30F0	ROM = Not Erased EEPROM = Erased ID = Erased Fuse = Not Erased Continue to erase and program this chip?	•	Φ	perioperioper periope
ə	сом з						÷	
e	e	and		Merge	Program ?{] Verify 🕝 CALI		0	Chip Selector

The programmer will download the hex file to the microcontroller board and it will usually begin executing.

The programmer will show progress of programming

jle	Program	mer <u>O</u>	ptions	Help								
) (		ROM	рата									
	0000:			5226	A4D8	REO7	7000	0012	OEF9	6		
	0010:	0000	0000	0000	0000	0000	0000	OFFF	A4D8			1 🕨 🧱 🖳 🔛 🌉 .
	0020:	EFOS	FOOD	0000	0000	0000	0000	0000	0000			
	0030:	0000	2826	EF07	F000	0012	OEFA	6826	EC02	. 6 6.		🗾 🦉 📃 🔟
	0040:	F000	OEFA	6 <b>E</b> 26	EC02	F000	OEFA	6 <b>E</b> 26	EC02			
	0050:	F000	OEFA	6 <b>E</b> 26	EC02	F000	2 <b>E</b> 02	D7EE	0012	6		
	0060:	5083	6203	OEOF	148C	1002	6 <b>E</b> 8C	908D	828D			
	0070:	928D	OEO1	6 <b>E</b> 26	EC02	FOOO	5003	6283	0012			
D	0080:	5083	6205	OEFO	1404	6E06	OEOF	148C	1006		$\ominus$	
	0090:	6E8C	908D	828D	928D	3804	OBFO	6 <b>E</b> 06	OEFO			
	00A0:	1406	6807	OEOF	148C	1007	6 <b>E</b> 8C	908D	828D			
	00B0:	928D	OEO3	6004	D005	0E14	6826	EC02	F000	6		
	0000:	D004	OEO1	6826	EC02	F000	5005	6283	0012	6		
	00D0:	508C	6E25	OEFO	1424	6 <b>E</b> 26	OEOF	148C	1026	.*.\$66		
	00E0:	6 <b>E</b> 8C	808D	828D	928D	3824	OBFO	6826	OEFO	\$.6.	-	
эĺ	COM 4					Pro	gram	ning H	ROM		<b>—</b> ə	
Эг											- D	
	·				1		-	1.1				<sup>™</sup> Chip Selector
	<u> 🖂 I</u>	oad		Merge			<u>P</u> rogram	n	<b>?{</b> ]⊻erify	🗸 🖉 🙆 🖸	5	18F4620 🔽
	h n	6		I.C		-		1				Course 1
	🛛 🔁 Re	erresh		<u>S</u> ave			<u>) R</u> ead		💾 <u>B</u> lank	🕻 🌔 😰 Fuse	S	🗙 <u>C</u> ancel

and then will show that it is verifying as shown below.

jle	Program	mer <u>O</u>	ptions	<u>H</u> elp								
Ð		ROM										
	0000-			5226	A4D8	8807	8000	0012	0779	6		
	0010:		0000				0000				-	1 🕨 🧾 🔍 🛄
	0020:						0000					
	0030:	0000						6E26		. 6 6.		
	0040:	FOOO			EC02							
	0050:	FOOO	OEFA	6826	EC02	F000	2802	D7EE	0012	6		
	0060:	5083	6203	OEOF	148C	1002	6ESC	908D	828D			
	0070:	928D	OEO1	6 <b>E</b> 26	EC02	FOOO	5003	6883	0012			
Ð	0080:	5083	6205	OEFO	1404	6206	OEOF	148C	1006		$\ominus$	
	0090:	6 <b>E</b> 8C	908D	828D	928D	3804	OBFO	6206	OEFO			
	00A0:	1406	6E07	OEOF	148C	1007	6ESC	908D	828D			
	00B0:	928D	OEO3	6004	D005	0E14	6826	EC02	F000			
	0000:	D004	OEO1	6 <b>E</b> 26	EC02	F000	5005	6283	0012			
	00D0:	508C	6E25	OEFO	1424	6226	OEOF	148C	1026	.*.\$66		
	00E0:	6 <b>8</b> 8C	808D	828D	928D	3824	OBFO	6826	OEFO	\$.6.	-	
эl	COM 4	COM 4 Verifying ROM								∃ ə		
ЭI												•
					1		-	1.1		1	1	Chip Selector
	🕞 <u>L</u>	oad		) Merge		and and a	<u>P</u> rogra	n	?{ ⊻erify	, 🛛 🕝 <u>C</u> alib		18F4620
	-				1			1	<b>W</b>	1	1	
	🛉 🔁 Re	etresh		<u>S</u> ave			) <u>R</u> ead		🖰 <u>B</u> lank	🤇 😰 🕞 Fuses		💢 <u>C</u> ancel
5											Ξəl	⇒ 2:15:25 PM

Verifying takes a long time, but you can cancel once it gets to this point, rather than wait for verification to complete. I also make a habit of closing the programmer at this point.

11. If the program doesn't start running, press the reset button on the microcontroller board.