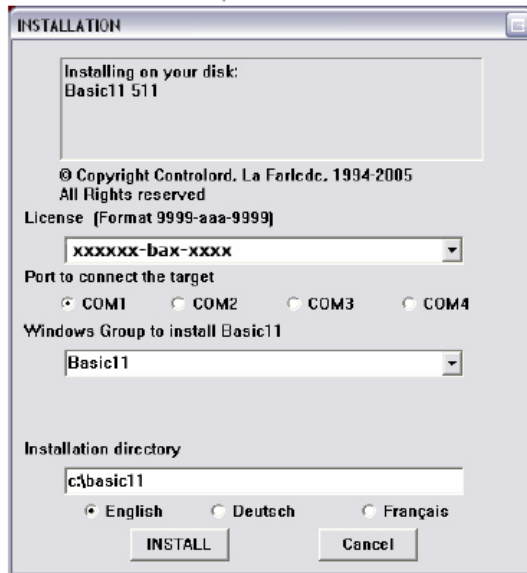
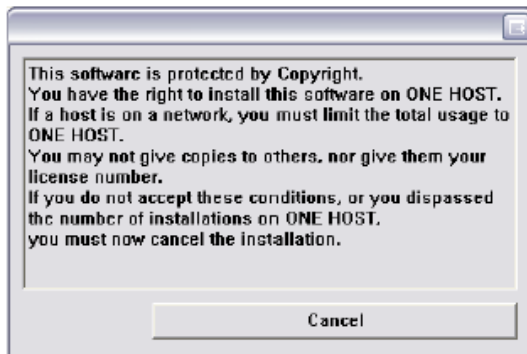


Getting Started with Basic11 and Adapt11C24DX

- 1) Download Basic11 from http://support.technologicalarts.ca/docs/BASIC11/setup11_511.exe
- 2) Double-click on the "setup11_511.exe" file and the screen shown below will be displayed
- 3) Enter the free license key **4711-bax-1459** directly into the box. (**Important!** Do not use the drop-down menu to select "Basic11 511 Freeware"; otherwise it will ask for your license key every time you start the program)



- 4) Select your COM port and language preference
- 5) Click INSTALL
- 6) You will see this screen:



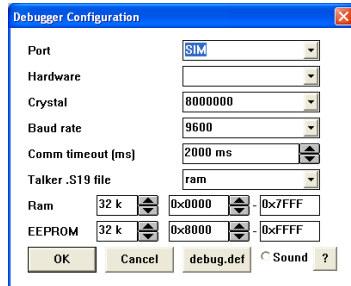
- 7) Please be patient and read this screen. Wait until the "I agree" button appears.



- 8) Select "I agree" and you are done.

Once the installer has finished, you will find an icon for Basic11 in your start menu. The next section of this document covers setting up and using BASIC11.

Setting up Basic11. Launch Basic11 via its icon, and a setup wizard will appear, as shown in the figure below. The only setting you need to change is **Port**. Set it to *SIM*. Ignore everything else, because we won't be using the loading and debugging features of the IDE. Click *OK* and that's all there is to it!



Now you can get started with a simple example. Open the example file called *C24DXflashD2.bas* (<http://support.technologicalarts.ca/docs/BASIC11/AD11C24DX/C24DXflashD2.bas>) or create it in the IDE's editor window. The contents are as follows:

```
'Target = Adapt11C24DX or MicroCore11
'This program flashes an LED connected to PA6 port pin

ProgramPointer $E000 'target is 8K (will also work for 32K board)
DataPointer $0002
StackPointer $01EF

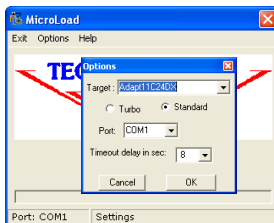
byte PORTA at $1000

int i

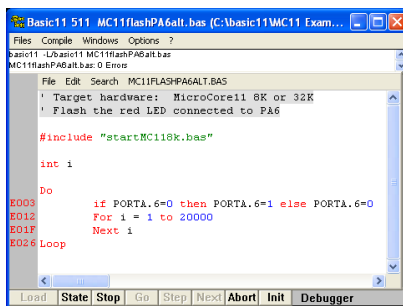
do ' forever
  if PORTA.6=0 then PORTA.6=1 else PORTA.6=0 'toggle the pin
  for i = 0 to 20000 ' delay
  next i
loop

asm org $fffe ;define reset vector
asm fdb $e000 ;to point to beginning of code
```

Compile the project and then download it to Adapt11C24DX using the Windows-based downloader MicroLoad from Technological Arts. MicroLoad can be found here: <http://support.technologicalarts.ca/docs/MicroLoad/ml131.zip>.



Using INCLUDE Files. Below, we have re-written the example program using the *include* directive. Since the memory map definitions, register names, and reset vector location are always required for every program, they can all be combined into a single file for any given hardware target. So we have created startup files containing these items. You can use the *include* directive to add them to your program. This will save you having to explicitly define them in each program you write.



```
' Target hardware: Adapt11C24DX 8K or 32K
' Flash the red LED connected to PA6

#include "startC24DX8k.bas"

int i

Do
  if PORTA.6=0 then PORTA.6=1 else PORTA.6=0
  For i = 1 to 20000
  Next i
Loop
```

We have created two such include files for Adapt11C24DX. Use *#include startc24DX8k.bas* for 8K boards and *#include startc24DX32k.bas* for 32K boards.