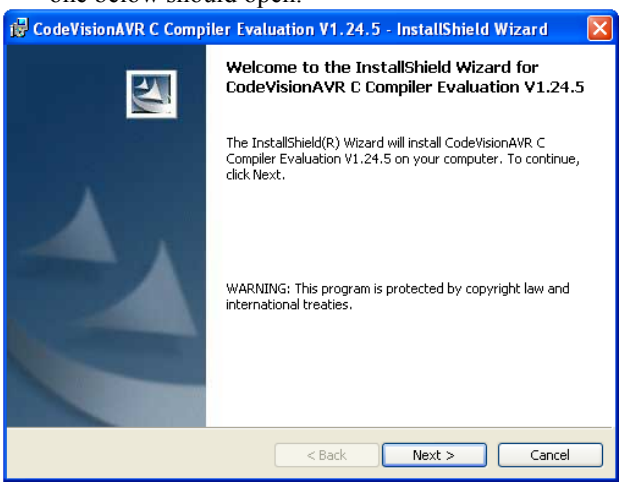
Getting Started With the Micro64

1.0 Software Installation

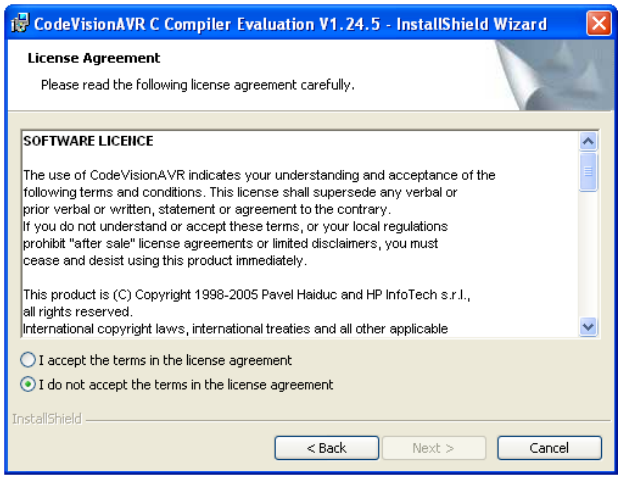
1.1 Installing the CodeVisionAVR C Compiler

1. Open the CodeVisionAVR Demo folder on the CD.

2. Click on setup icon and the window similar to the one below should open.

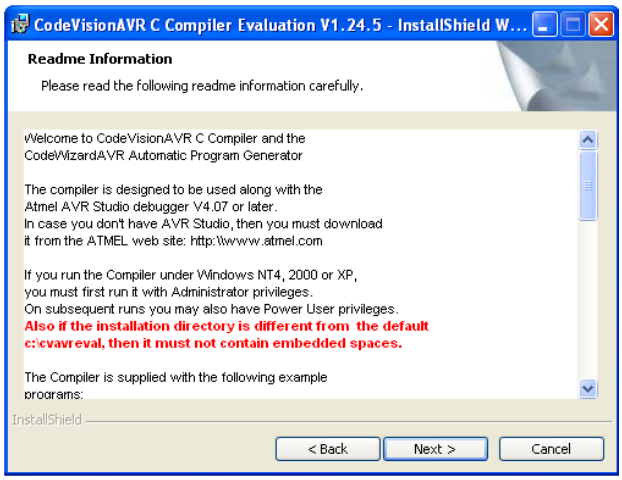


3. Click the Next button and the following window should open.

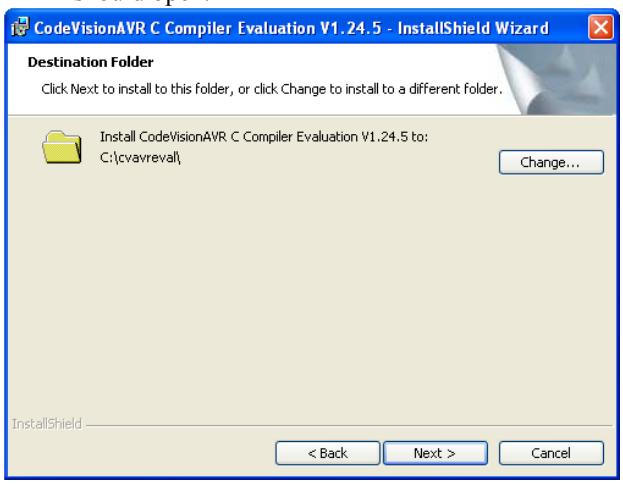


4. Accept the terms in the license agreement.

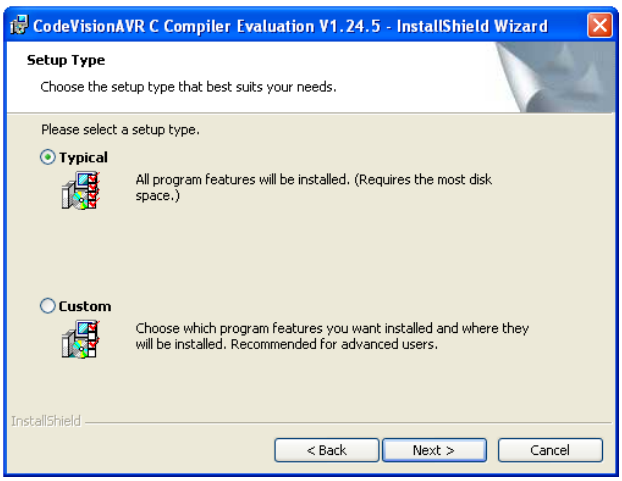
5. Click the Next button and the following window should open.



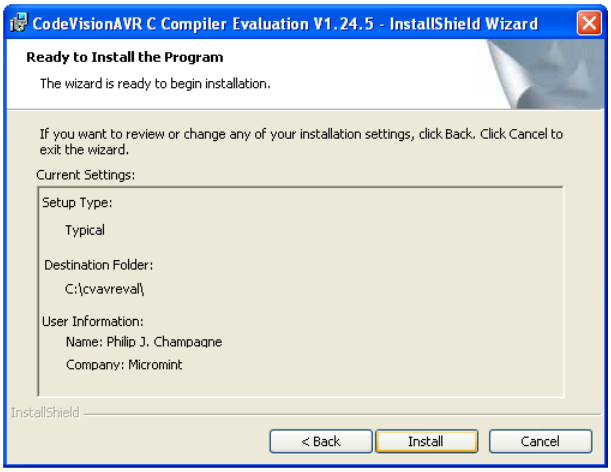
6. Click the Next button and the following window should open.



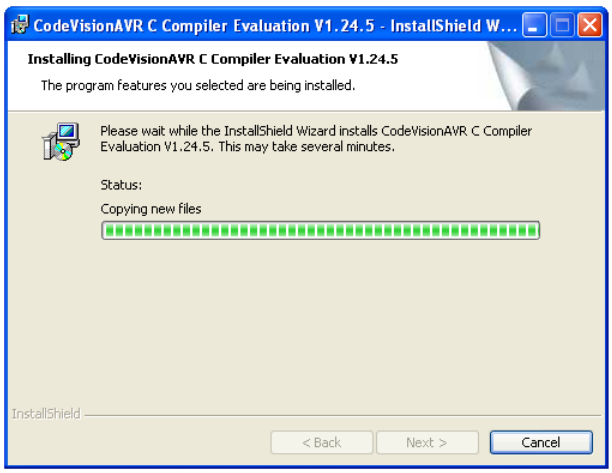
7. Click the Next button and the following window should open.



8. Click the Next button and the following window should open.



9. Click the Install button and the following window should open.

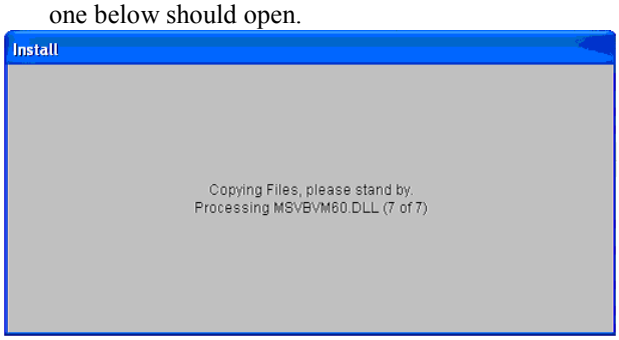


10. Wait for the program to install then press the Finish button.

1.2 Installing the Micro64's Boot Loader Software

1. Open the Boot Loader folder on the CD.

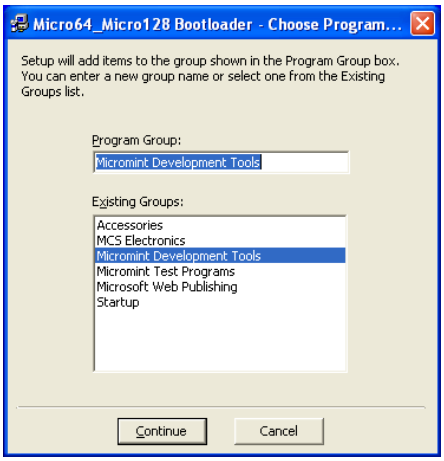
2. Click on setup icon and the window similar to the one below should open.



3. Wait for the files to finish loading and the following window should appear.

4. Click on the OK button to continue with the install and the following window should open.

5. Click on the  button and the following window should open.

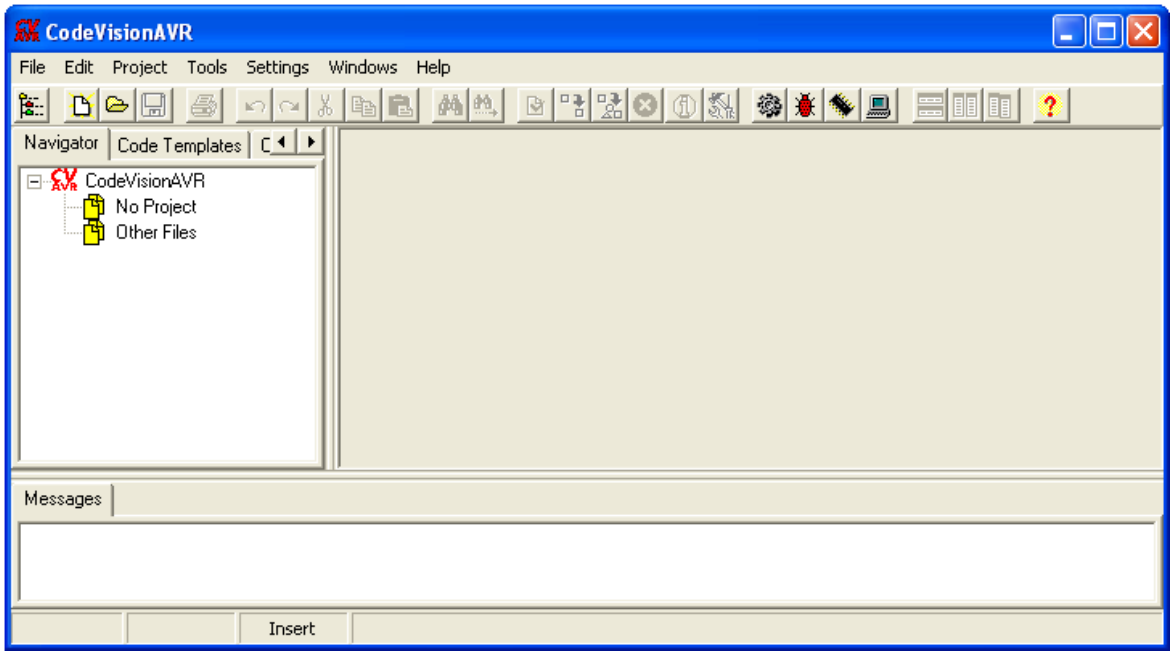


6. Click on the Continue button and files should start loading into your computer. After the files are done loading the following window should open.

7. Click OK to finish the installation.

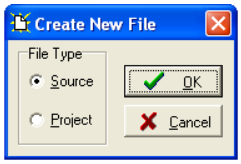
2.0 Compiling and running HelloWorld.c

When the compiler is opened for the first time the GUI (Graphical User Interface) should look like the following image.

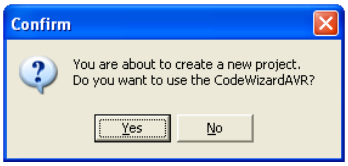


Please follow the following steps to setup the compiler for a Micro64/128.

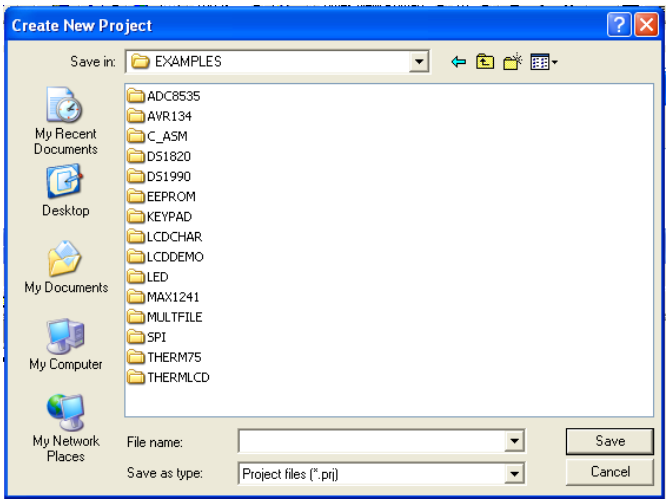
1. Click on “File|New” menu option or click the toolbar button and the following window will be displayed.



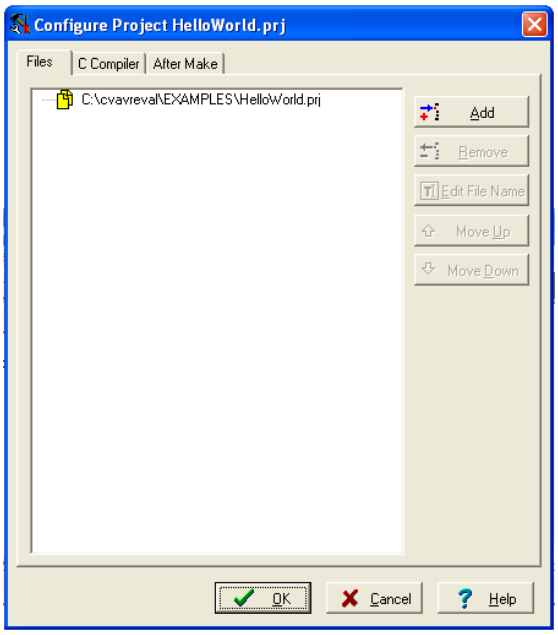
2. Select “Project” and press “OK” and the following window will be displayed.



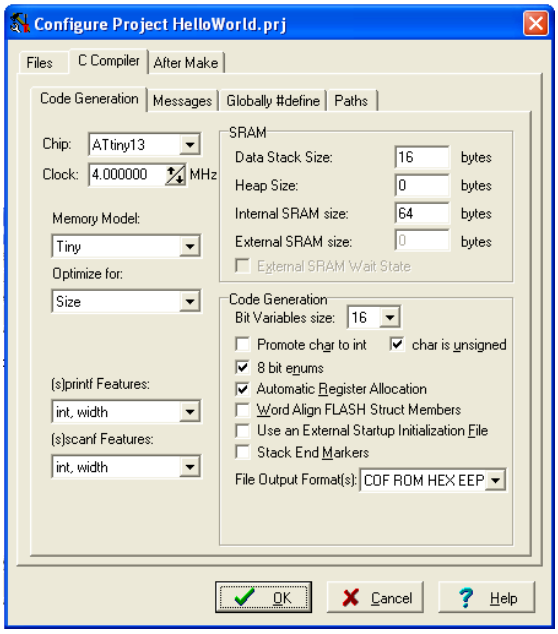
3. For simplicity we will not use the CodeWizardAVR so click “No” and the following window should open.



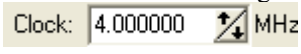
4. Type in a file name of your choice and click “Save”. For demonstration purposes HelloWorld was chosen for the File name. The following screen should appear.



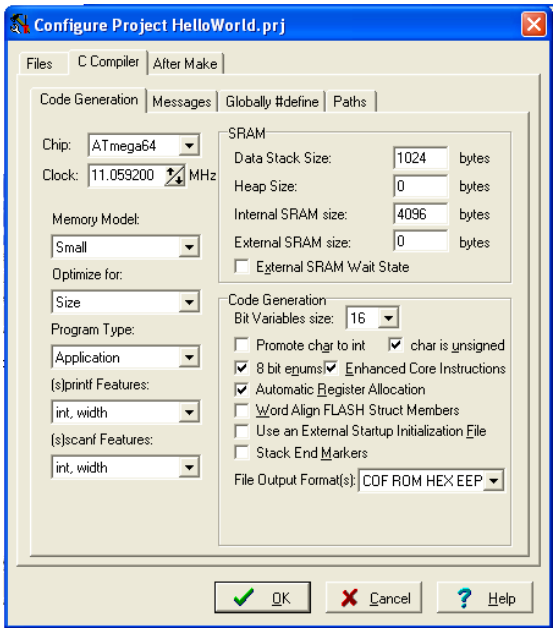
5. Click on the “C Compiler” tab and the window will look similar to the following.

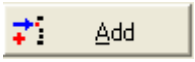


6. Click the arrow on 

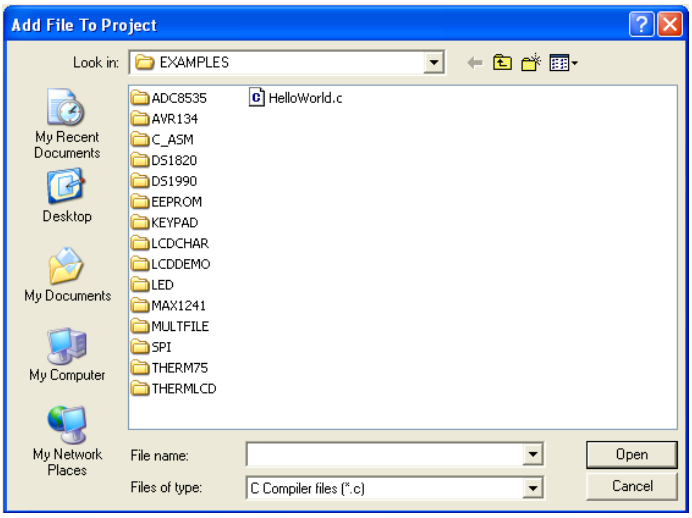
and select Atmega64 for Micro64 or Atmega128 for Micro128.
7. Click next to the first digit in the following 

and change it to 11.0592.
8. After you complete the changes the window should look like the following for the Micro64/128.

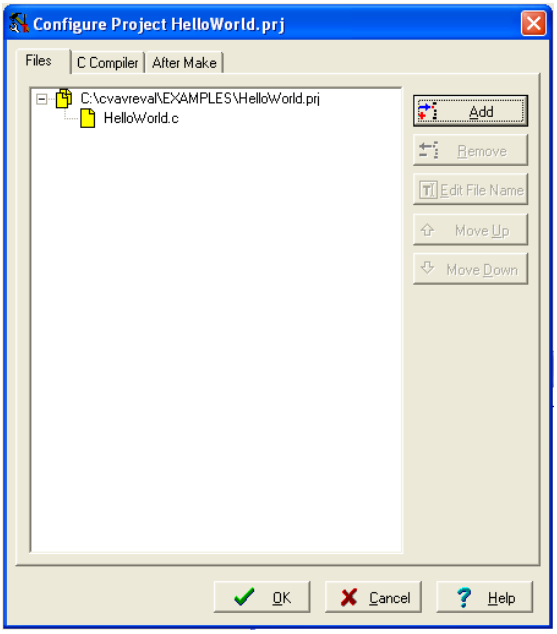


9. All other parameters do not need to be changed in order to compile a simple program. Please refer to the CodeVisionAVR's help section for further details on the other parameters. Click on the "Files" tab to continue.
10. The next step would be to add a source file. To add a source file click the 

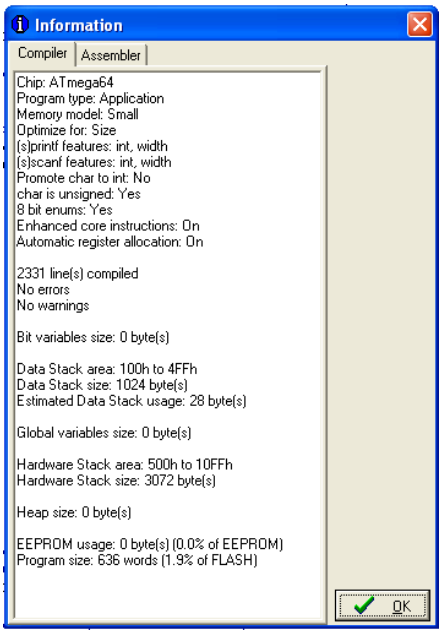
button and the following window will appear.



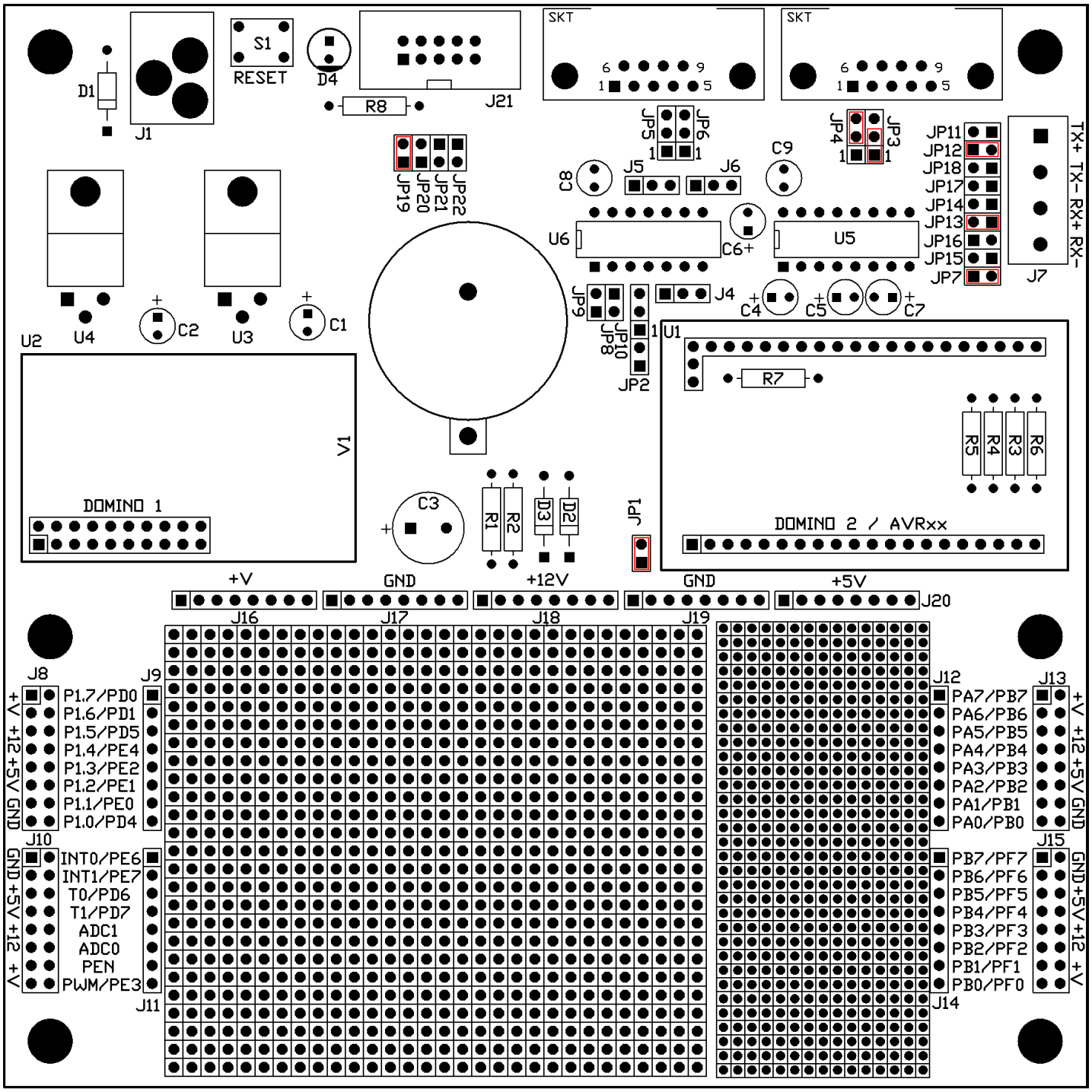
11. Find and Select "HelloWorld.c" and click "Open". HelloWorld.C can be found on the CD or on Micromint's Micro64/128 Datasheet & Application Notes webpage. http://www.micromint.com/app_notes/micro64_128.htm The following window should appear.



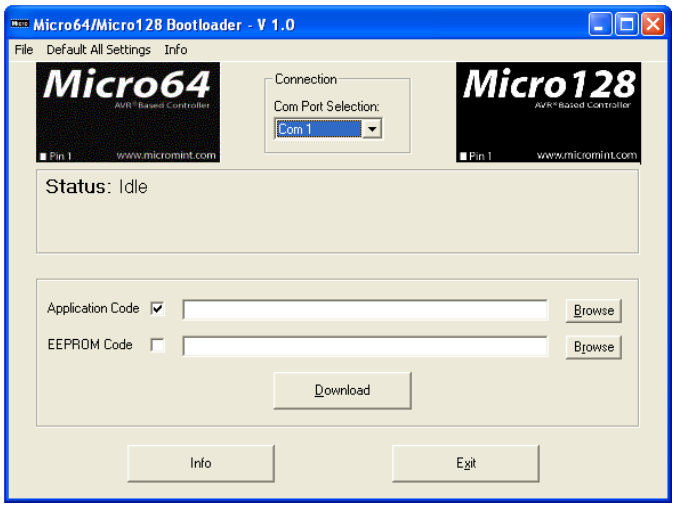
12. Click the "OK" button and then you will be ready to make the hex file.
13. To make the hex file for the program click on "Project|Make" menu option or click the  tool bar button. The following window should open.



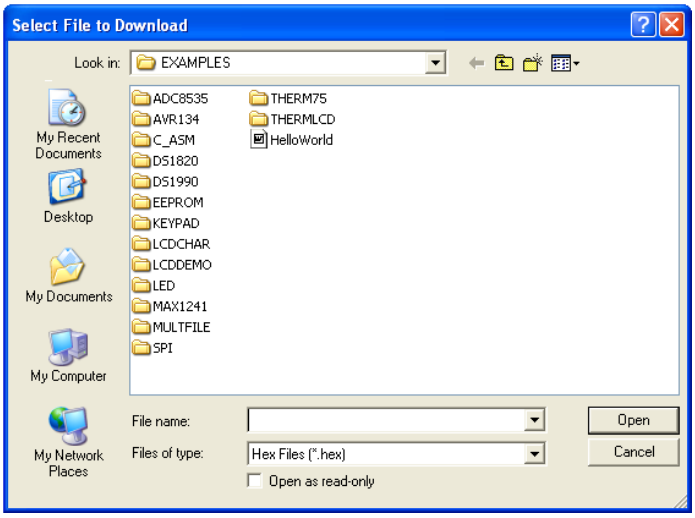
14. Click "OK". Now it is time to program the Micro64. using the Boot Loader software. Please use the following diagram to verify the minimum jumpers need for downloading programs.



15. If you already installed the Boot Loader software in section 1 then click on "START|Programs| Micromint Development Tools| Micro64_Micro128 Bootloader" and the following window will open. The boot loader uses USART1 to download programs to the Micro64/128. If you are using the development board then please refer to section 6.31 of the Micro64/128 datasheet to set-up the jumpers properly.

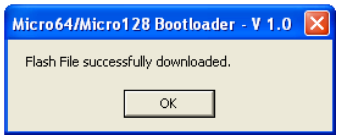


16. Click the Browse button next to "Application Code" to select the hex file that was created by the compiler. A similar window like the following window will open.

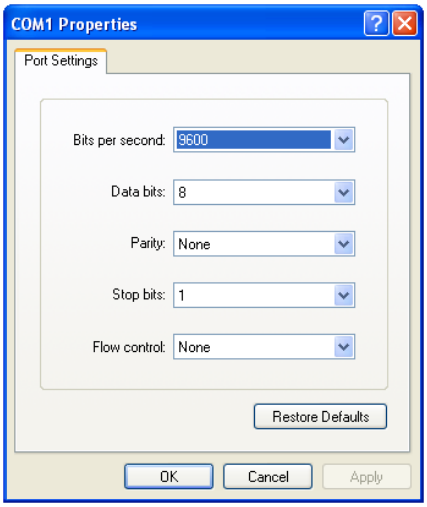


17. Find and select the file "HelloWorld.hex" and click "Open".

18. Click on the Download button to send the file through COM1 to the Micro64.
NOTE: The Boot Loader software uses the COM Port when it is sending a file to the Micro64. If the software says it is unable to open the COM port there can be a few reasons why.
    1. The COM port selected is already in use. Close the application using the COM port or select another one.
    2. The Micro64/128 is constantly sending information to the closed COM port. Hold the RESET button or disconnect the power the Micro64/128 then click on the Download button.
19. Press and release the RESET button on the Micro64 development board or cycle the power to the module. After the program is done downloading the file the following message box will open.



20. Click "OK" and open HyperTerminal with the following settings.



21. Press and release the RESET button or cycle the power. Wait for a moment and you should see "Hello World" constantly displaying on the screen. That is all there is to getting a program up and running.

3.0 HelloWorld.c Listing

```
/*****************************************
Program : HelloWorld Example for Micro64 or Micro128
Company : Micromint, Inc
*****************************************/

#include <mega64.h>                  // Comment this line out for Micro128
// For Micro128 make sure that you goto "Project|Configure", to the
// "C Compiler" tab and change the chip to ATmega128
//#include <mega128.h>               // Uncomment this line for Micro128
#include <stdio.h>                                                              // Standard I/O
library

// Declare your global variables here
#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7
#define TXC 6

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
#define TX_COMPETE (1<<TXC)
int COM;        // if COM = 0 then use USART0 if it = 1 then use USART1
//*************************************************************************************************
// The program lines between the ** lines, like the above line, is needed in order to use the
// printf command for USART1.

/* inform the compiler that an alternate version
   of the getchar function will be used for USART1 */

#define _ALTERNATE_GETCHAR_

/* now define the new getchar function */
char getchar(void)
{
/* write your code here */
char status,data;
    switch(COM)
    {
      case 0:
          while (1)
               {
               while (((status=UCSR0A) & RX_COMPLETE)==0);
               data=UDR0;
               if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
               return data;
               };
      case 1:
```

```
            while (1)
                {
                while (((status=UCSR1A) & RX_COMPLETE)==0);
                data=UDR1;
                if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
                return data;
                };
    };
}

/* inform the compiler that an alternate version
   of the putchar function will be used */
#define _ALTERNATE_PUTCHAR_

/* now define the new putchar function */
void putchar(char c)
{
/* write your code here */
    switch(COM)
    {
      case 0:
          while ((UCSR0A & DATA_REGISTER_EMPTY)==0);
          UDR0=c;
          break;
      case 1:
          while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
          UDR1=c;
          while ((UCSR1A & TX_COMPETE)==0);
    };
}
//*****************************************************************************************
void main(void)
{
// Declare your local variables here

// Set up USART1's Baud rate at 9600 bps with a 11.0592 MHz Crystal
UCSR1A=0x00;      // RX EN, TX EN
UCSR1B=0x18;     // RX EN, TX EN
UCSR1C=0x06;     // 8N1
UBRR1H=0x00;    // Baud rate high - 9600
UBRR1L=0x47;   // Baud rate low

COM = 1;                    // Use USART1
DDRD.6 = 0;                                                                    // Make
PORTD.6 an output
PORTD.6 = 1;                                                                   // Enable the
RS485 control line


while (1)
     {
      printf("Hello World\r\n");
     };
}
```