

# **eXtreme Electronics**

---

## **ProGFX – Embedded Graphic Library User Manual**

**<http://www.ProGFX.org>**

# ProgGFX - Embedded Graphic Library User Manual

---

## Table of Contents

---

<b>Chapter 1. Introduction.....</b>	<b>3</b>
1.1 Introduction .....	3
1.2 The JHD12864E Graphic LCD .....	3
1.3 ATmega32 Based Test Circuit .....	5
1.4 Test Circuit Fabrication Tips .....	6
<b>Chapter 2. Installing ProGFX .....</b>	<b>9</b>
2.1 Downloading ProGFX .....	9
2.2 Installing ProGFX .....	9
2.3 ProGFX Project Creation with AVR Studio .....	10
2.4 "Hello GraphicX!" Project .....	18
2.5 Compilation and Running .....	19
2.6 Troubleshooting .....	19
<b>Chapter 3. API Documentation .....</b>	<b>20</b>
3.1 Introduction .....	20
3.2 .....	

# ProgGFX - Embedded Graphic Library User Manual

---

## Introduction

---

### 1.1 Introduction

Those who are building microcontroller based project for little long must have got bored with the good old character LCDs. Whether you are bored or your application require to present more data to the user in a better way, you need Graphic LCD. Character LCD Interfacing is quite easy so every one uses it, but when I comes to Graphic LCD you need a well written and powerful graphic library. Its not enough that you read the LCD datasheet and connect it your MCU and start sending data. Because the LCD just appears to be block of memory whose contents are directly visible on screen. The datasheet can only guide you how to access this memory. You just can't do much by writing to the memory. The graphic library is a piece of software that has complex algorithms to render graphic primitives like line, rectangles, circles, images and more. It also helps load fonts and render text and numbers on screen. So it provide high level access to the LCD screen and applications can be written much more easily.

While I was researching for graphic library for the GLCDs, I found some but I was not fully satisfied by any of them. So I began to write a clean, powerful, portable and easy to use library that can handle sever different graphic LCDs and can be ported to many MCUs. In this article I will use a Graphic LCD which is based on common KS0108 type controller with AVR ATmega32 MCU. In future I will port the tutorial and library for PIC and ARM CPUs.

### 1.2 Introducing the JHD12864E Graphical LCD Module.

Their are several type of graphical LCD module in the market. In this tutorial I will introduce you to a common graphical lcd module that is easily available in most part of the globe.



**JHD12864E Graphical LCD**

# ProgGFX - Embedded Graphic LibraryUser Manual

---

The specification of this LCD are as follows.

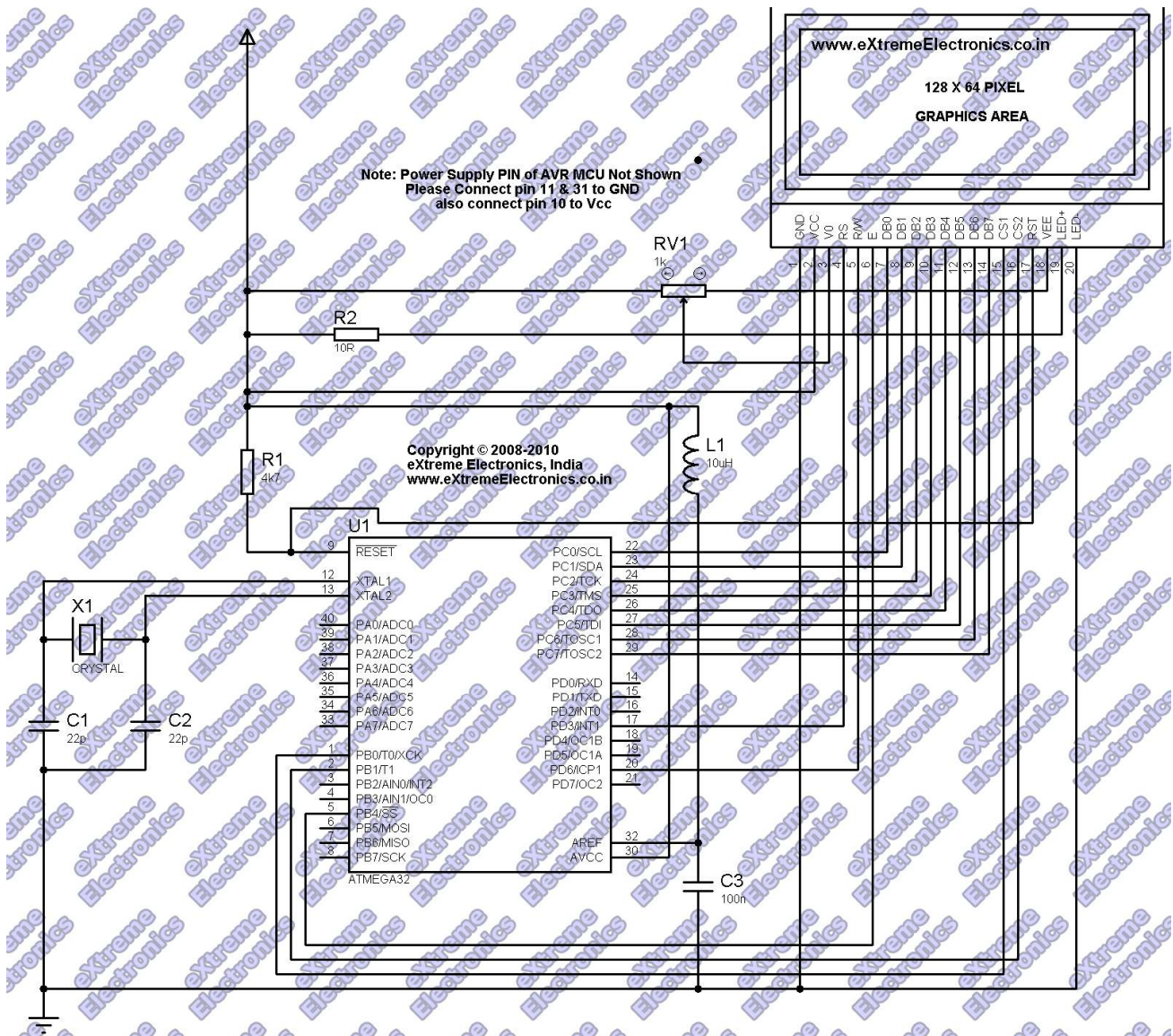
- ❖ 128 horizontal pixel and 64 vertical pixel resolution.
- ❖ Controlled based on KS0108B
- ❖ Parallel 8bit interface
- ❖ On board graphic memory.
- ❖ Available in Green backlight with dark green pixels.
- ❖ Also available in Blue backlight with light blue pixels.
- ❖ LED backlight.
- ❖ 20 PIN linear connection.

## PIN Description

PIN	Name	Function	Connection with AVR PIN
1	Vss	Ground	
2	Vcc	+5v Supply in	
3	V0	Contrast Adjust	
4	RS	Instruction/Data Register Select	PD3
5	R/W	READ/WRITE SELECTION	PD6
6	E	ENABLE SIGNAL	PB4
7	DB0	DATA IN/OUT	PC0
8	DB1	DATA IN/OUT	PC1
9	DB2	DATA IN/OUT	PC2
10	DB3	DATA IN/OUT	PC3
11	DB4	DATA IN/OUT	PC4
12	DB5	DATA IN/OUT	PC5
13	DB6	DATA IN/OUT	PC6
14	DB7	DATA IN/OUT	PC7
15	CS1	Chip Select 1	PB0
16	CS2	Chip Select 2	PB1
17	RST	RESET SIGNAL	RESET
18	VEE	NEGATIVE 10V OUT	
19	LED+	LED BACKLIGHT	
20	LED-	LED BACKLIGHT	

## 1.3 Graphic LCD Testing Circuit with AVR ATmega32

I suggest you to make a small circuit with ATmega32, its life support environment and Graphic LCD to experiment with these LCD. I have clocked the Test circuit with 16MHz crystal for maximum speed. Make the circuit as follows.



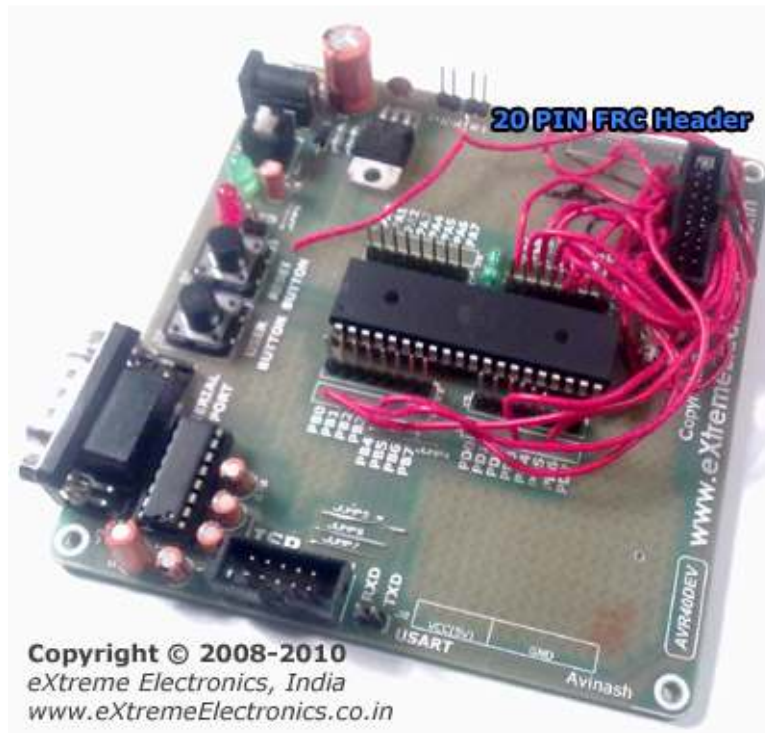
**Schematic for AVR ATmega32 and Graphic LCD Connection.**

The RV1 (which is a variable resistor of 10K) is used to adjust the contrast of the display. If no display is visible on screen use this to adjust the contrast until you see everything clearly. The AVR's Fuse bits must be set as follows to disable JTAG and enable external crystal. This step is very important and things won't work if you forget this.

- ❖ HIGH Fuse = 0xC9
- ❖ LOW Fuse = 0xFF

## 1.4 Graphic LCD Test Circuit Fabrication Tips.

I have used my Low Cost 40 PIN AVR Development Board for quick testing. I is a bare bone AVR board with AVR and its life support. The nice thing is that it has prototyping area surrounding the AVR. I used this area to mount the 20 PIN FRC Box header for connecting the GLCD.



**40 PIN AVR Devboard with Graphic LCD Connector.**

In the above image you can see the 20 PIN FRC Box header is soldered. The red wires connect the PINS of header to the MCUs I/O lines. They also supply power to the module and its led backlight.



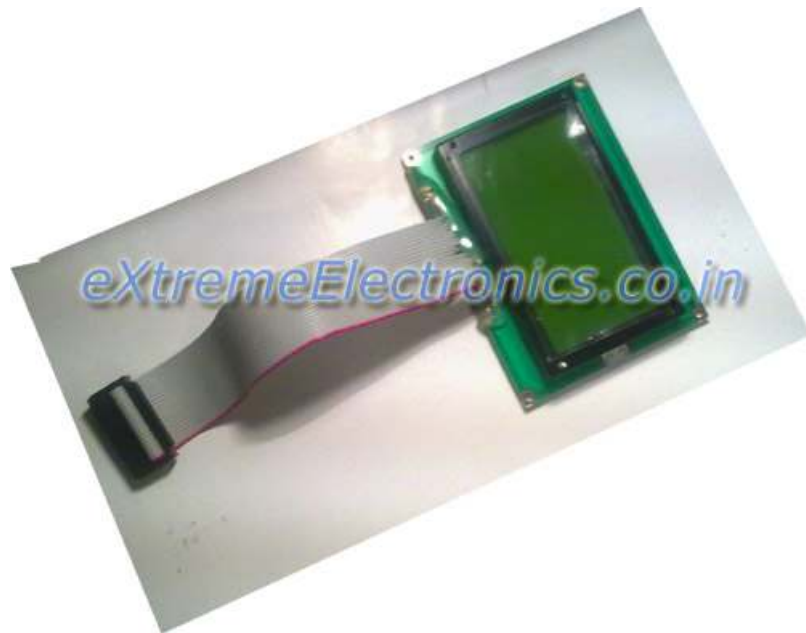
**Graphic LCD Connector**



# ProgGFX - Embedded Graphic Library User Manual

---

Take a 20 PIN FRC Cable. It has female connectors at both ends. Cut it to half and solder the wires to 20 PINs on the LCD Module. The image below illustrate the point. Now you can easily connect and disconnect the graphic lcd module from board.



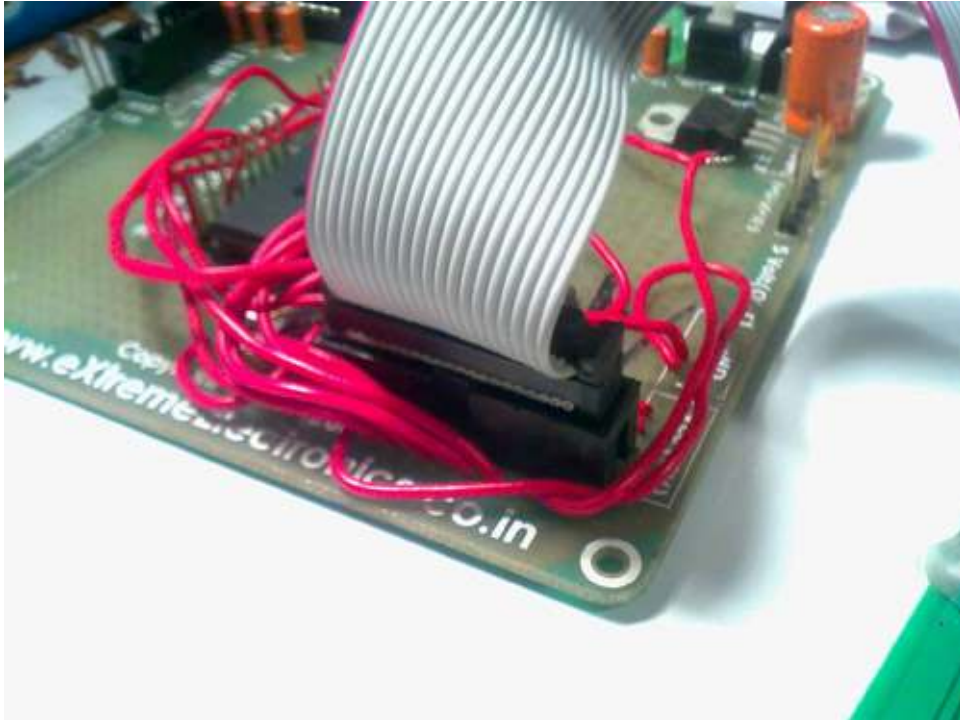
**Graphic LCD with Wire Soldered.**



**Graphic LCD with Wire Soldering.**

# ProgGFX - Embedded Graphic Library User Manual

---



*Graphic LCD Connected.*



*AVR Graphic LCD setup is ready!*



## Download & Installation

This tutorial deals with downloading and installing ProgFX graphic engine. ProgFX is a free and easy to use embedded graphic library that can control several display modules and can run on AVR, PIC and ARM MCUs. In this tutorial we will make a ProgFX graphical application with Atmel AVR ATmega32 MCU using AVR Studio and WinAVR (C Compiler). So lets begin.

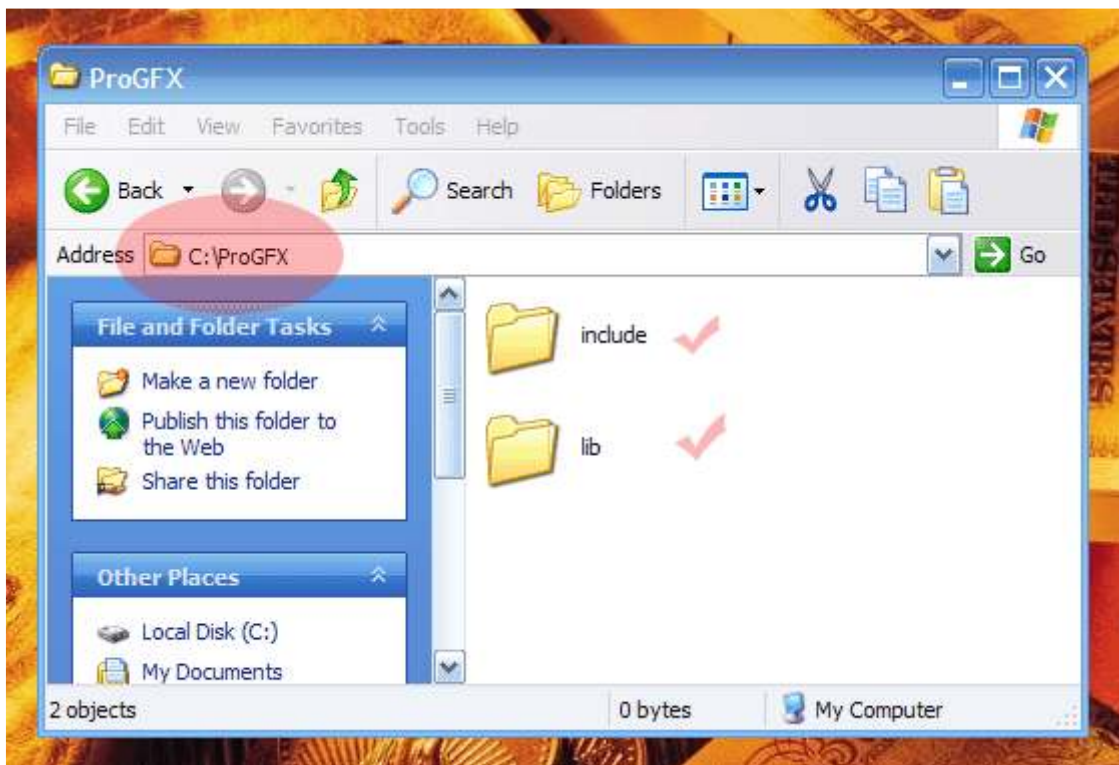
### 2.1 Downloading ProgFX

Latest version of ProgFX can be downloaded from  
<http://www.ProGFX.org>

### 2.2 Installing ProgFX

You need WinZIP to extract the files to your hard disk. Please extract it to root of any drive like "C:". Following Folders will be created.

- ❖ C:\ProGFX\include
- ❖ C:\ProGFX\lib



***Folders Created after extracting the ProgFX package***

Note that core of the engine is contained in two folders named ***“include”*** and ***“lib”*** but there may be some others folders too, like ***“Help”*** depending on your version of ProgFX engine.

# ProgGFX – Embedded Graphic Library User Manual

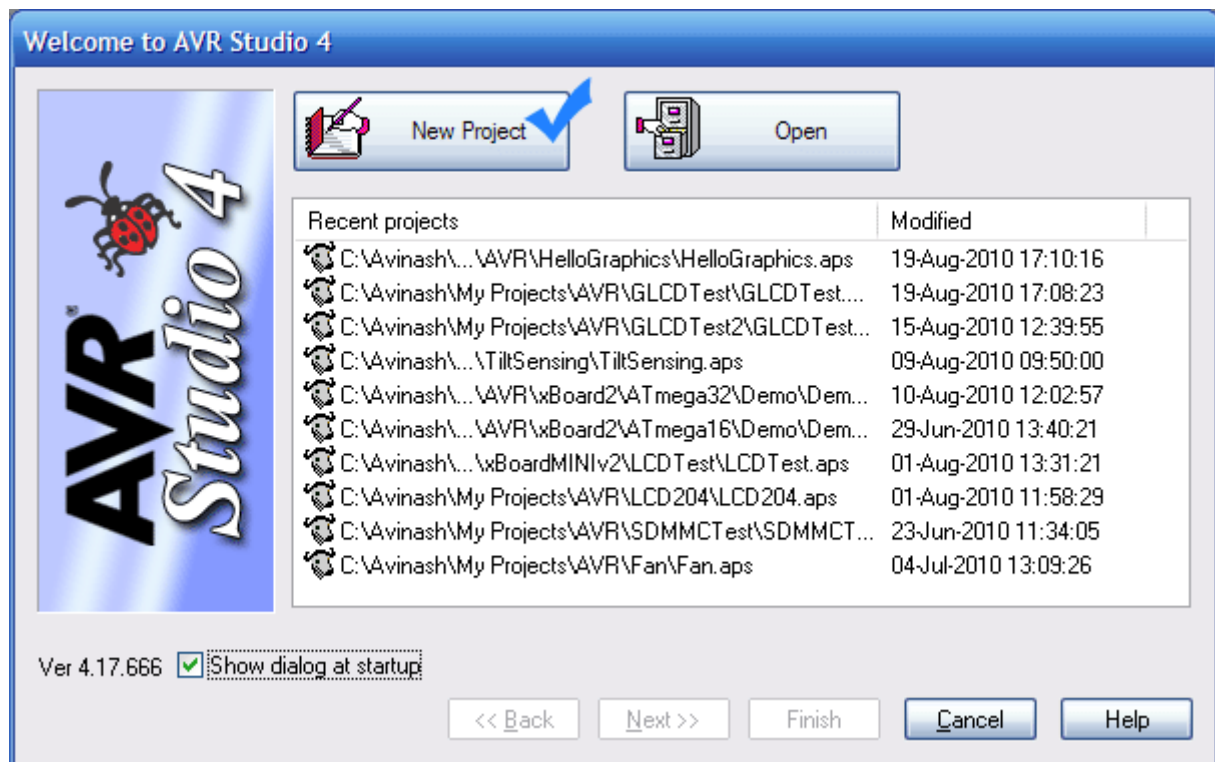
## 2.3 Creating a Graphical Project using AVR Studio.

The basic tools required for developing with AVR are

- ❖ **AVR Studio**
- ❖ **avr-gcc** (Precompiled Binary version for Windows is called WinAVR)
- ❖ **eXtreme Burner AVR**

These tools must be installed and set up correctly.

Start AVR Studio and you will be presented with the following screen.



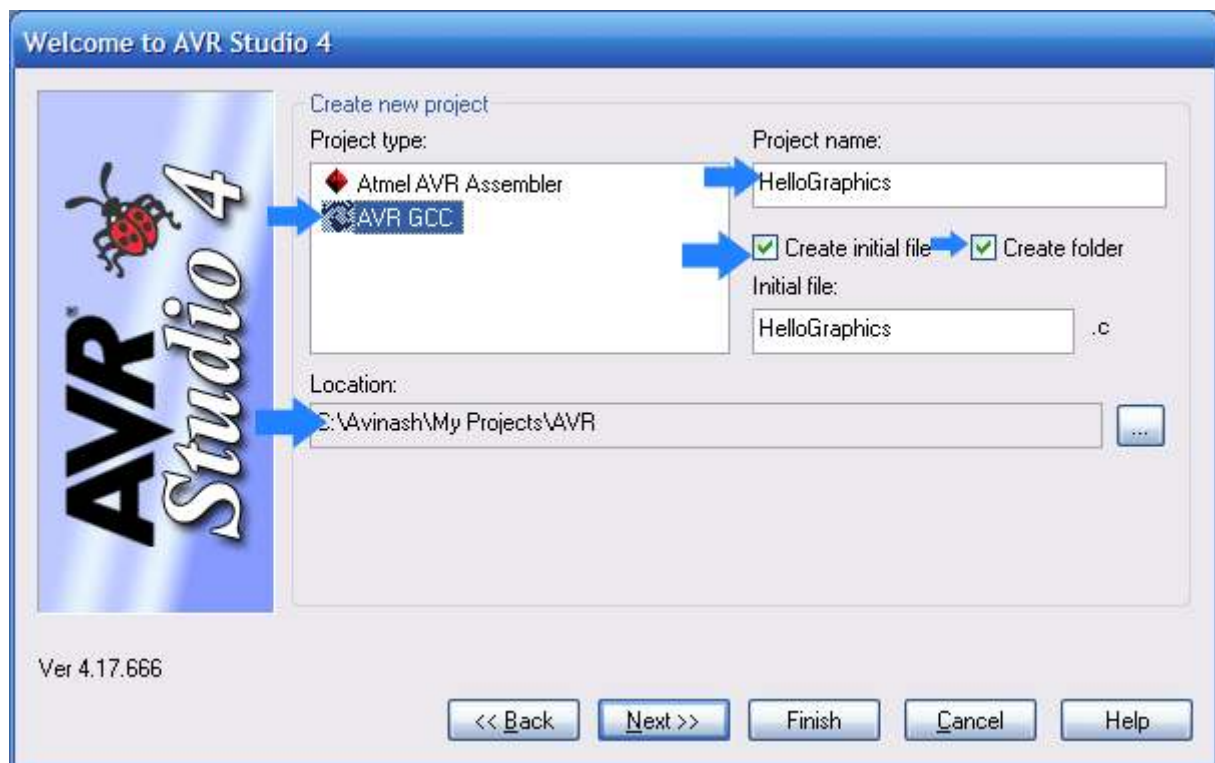
**AVR Studio Welcome Screen**

Select **"New Project"**

And on the next screen ...

# ProgGFX - Embedded Graphic Library User Manual

---



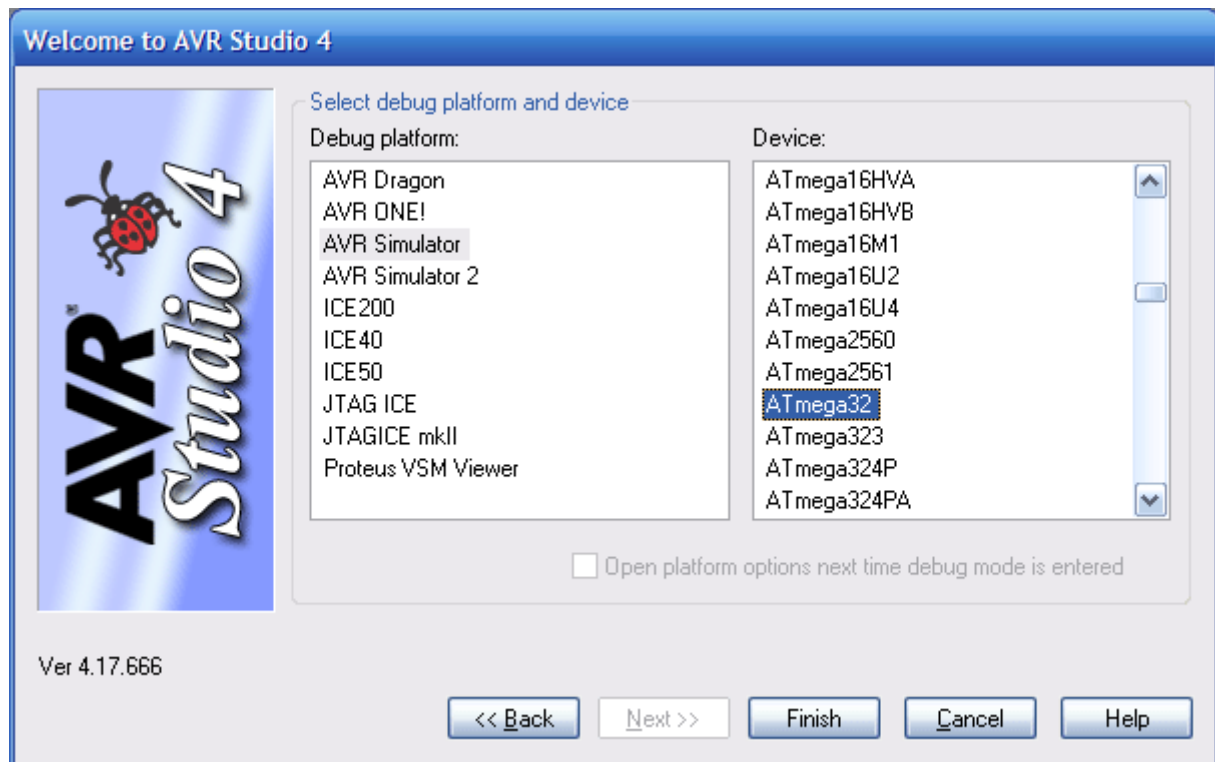
## ***Configure the Project***

Select the options as shown above.

- ❖ Project Type is AVR GCC
- ❖ Project Name is "HelloGraphics"
- ❖ Select any appropriate location for the project in your hard disk.

Now select Debug Platform as AVR Simulator and Device as ATmega32 and click Finish.

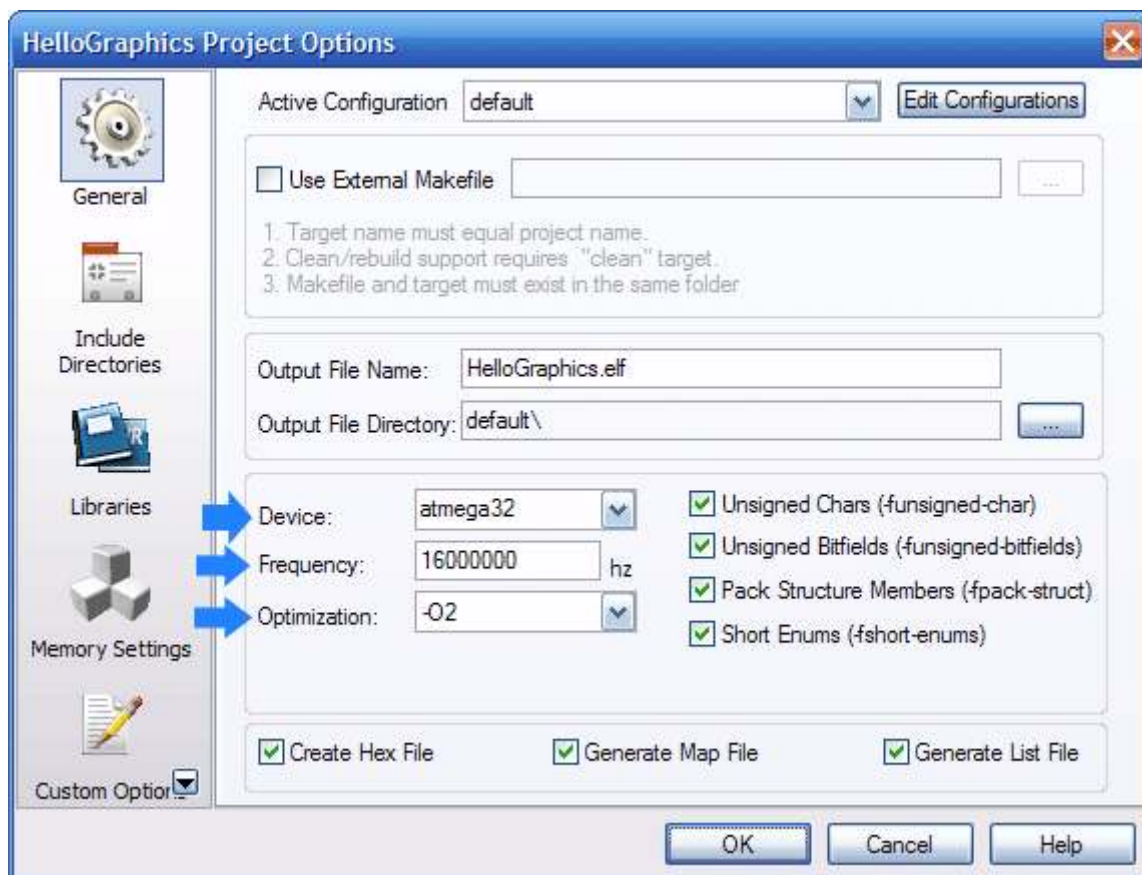
# ProgGFX - Embedded Graphic Library User Manual



## Device Selection

Now your project is ready. From the **"Project"** menu select **"Configuration Options"** you will be presented with the following screen.

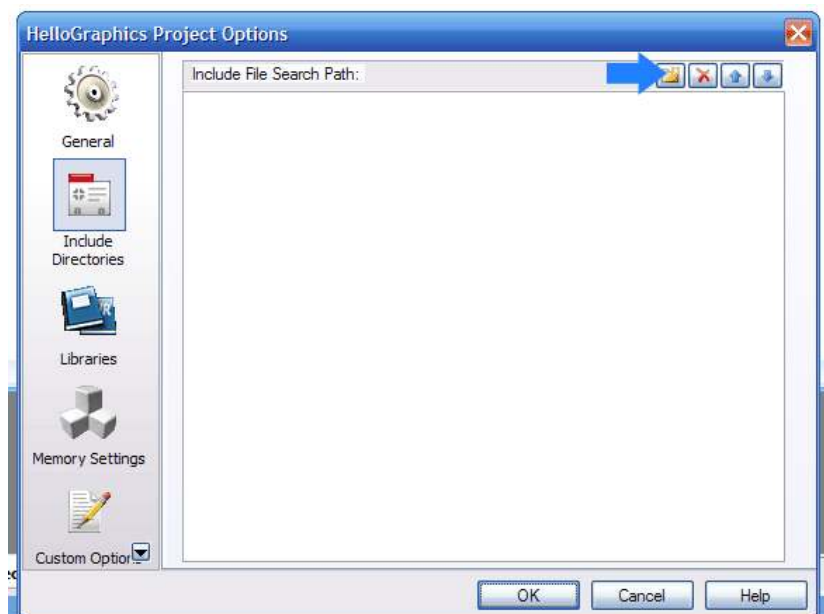
# ProgGFX - Embedded Graphic Library User Manual



## Configure the Project

Select Frequency as 16000000 (16MHz) and Optimization as "-O2". Then select **"Include Directories"** from the left hand pane. After that select **add include directory**.

The button is shown in the image below.

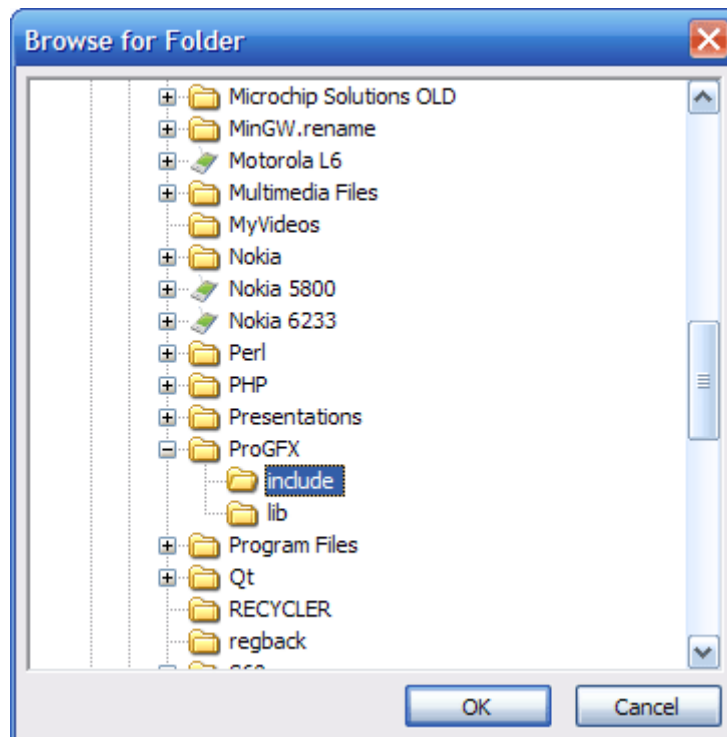


## Add new include directory



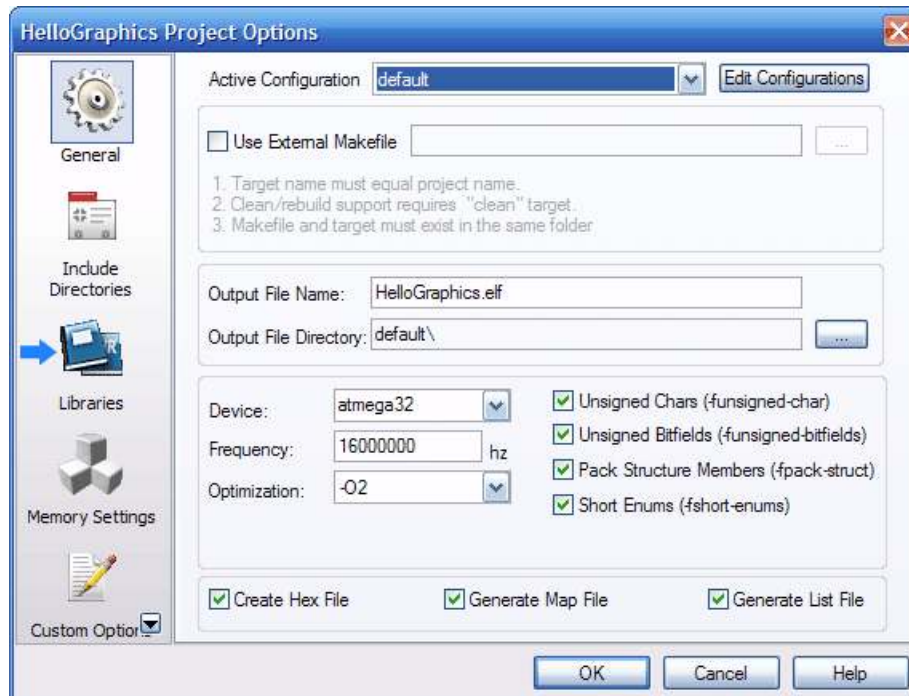
# ProgGFX - Embedded Graphic LibraryUser Manual

Now select the **"include"** folder inside the ProGFX Folder.



**Add new include directory**

Now similarly we need to add the library to our project. So go to **"Libraries"** section.

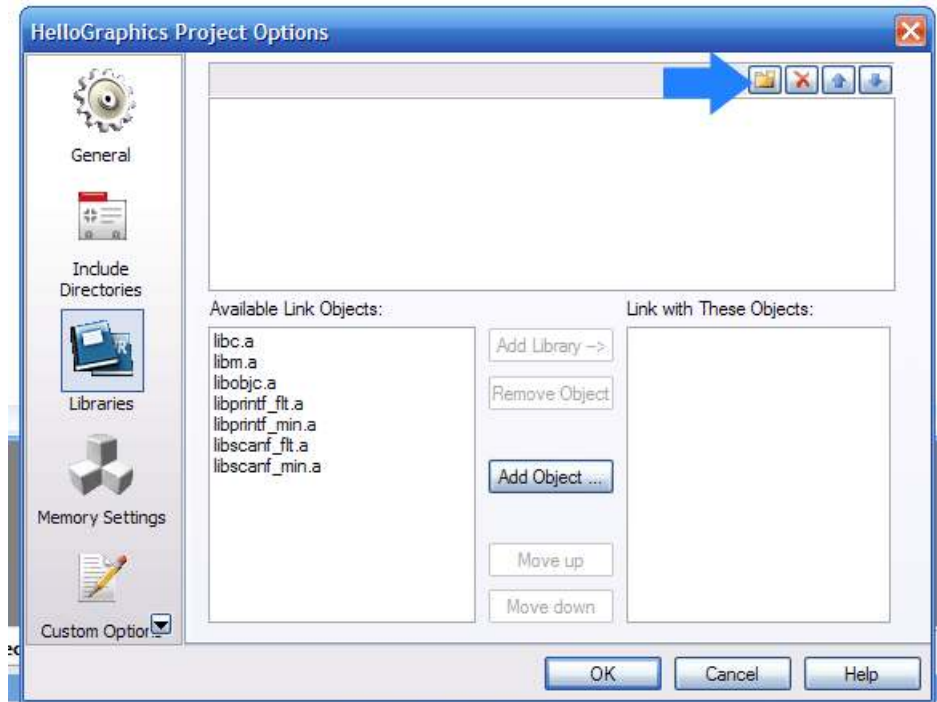


**Go To Library Section**

# ProgGFX - Embedded Graphic Library User Manual

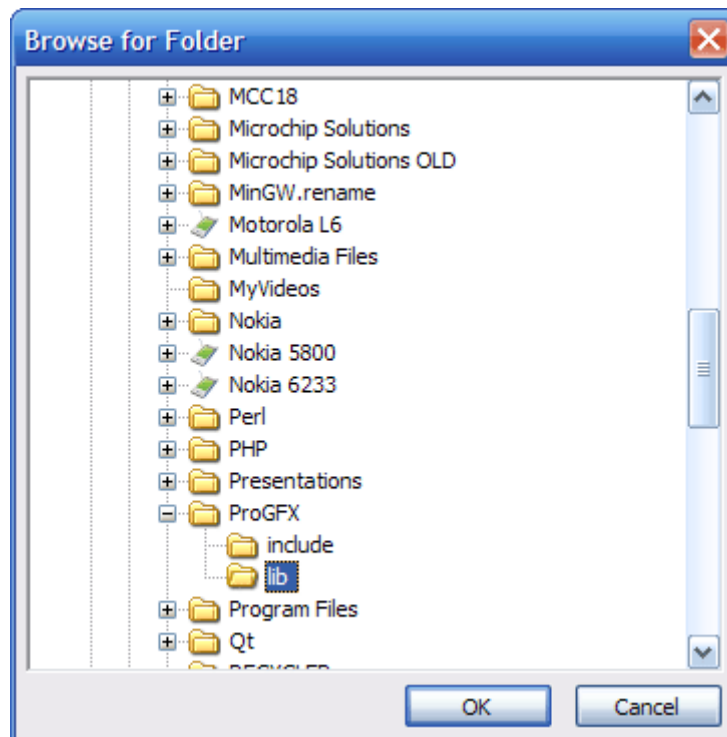
---

In library section select **"Add new library path"**. It is shown in the image below.



**Add new library**

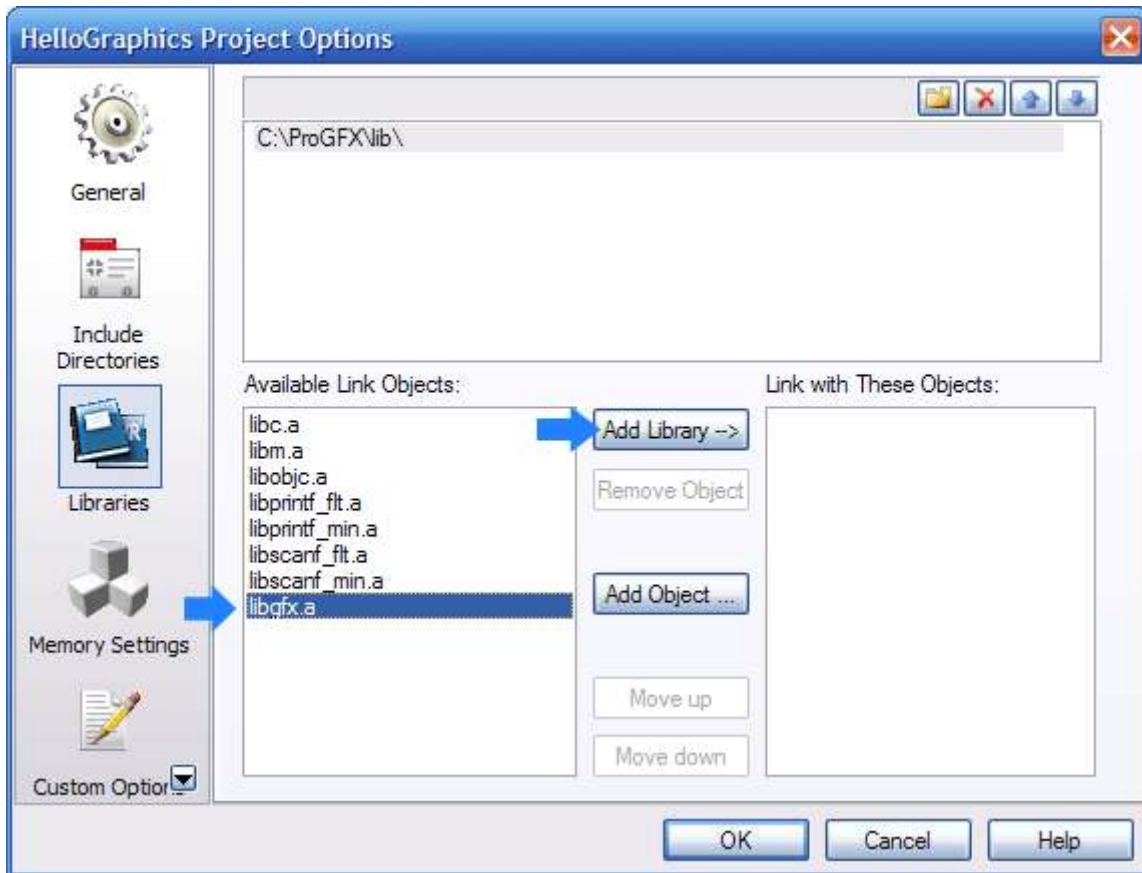
Then select the **"lib"** folder inside the ProGFX installation.



**Add new library**

# ProgGFX - Embedded Graphic Library User Manual

Now the **"Available Link Objects"** section will show "libgfx.a". Select "libgfx.a" and click "Add Library" button. Finally click OK.

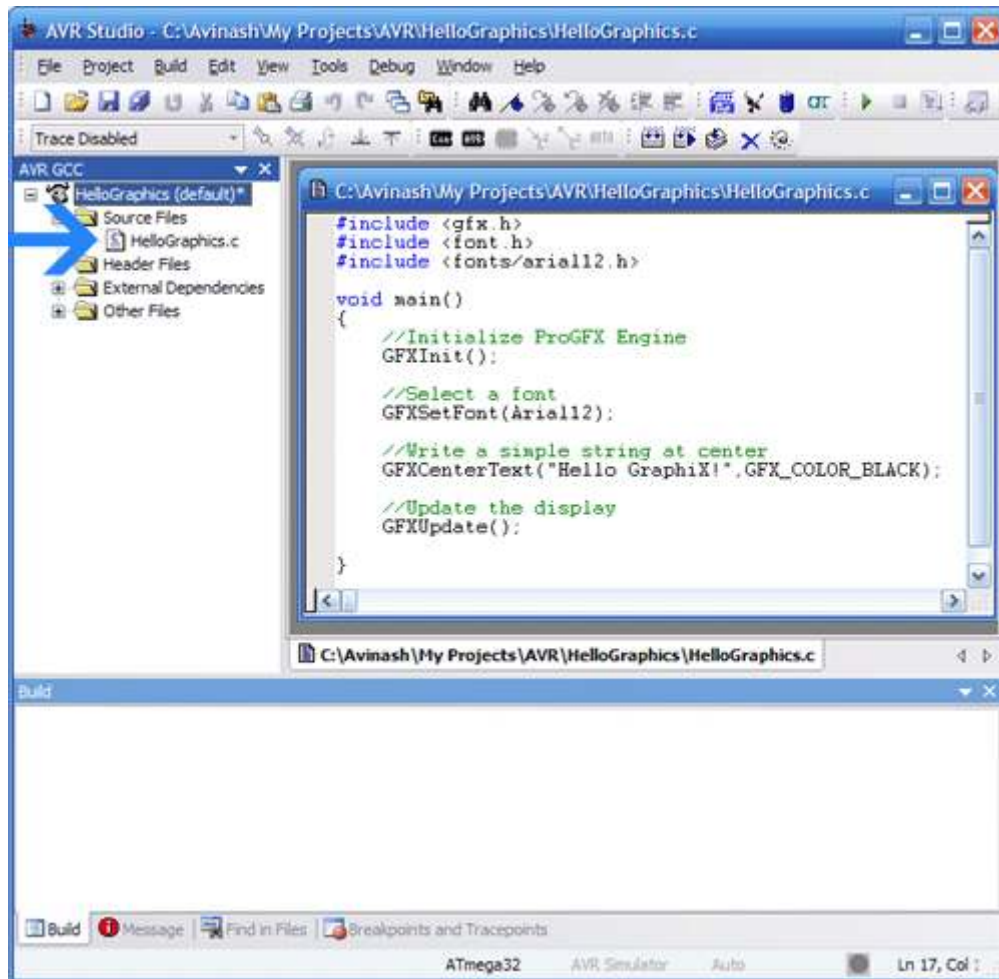


***Adding libgfx.a***

Finally Your Project is ready!

# ProgGFX - Embedded Graphic Library User Manual

---



**AVR Studio Ready For Graphical Application**

Double Click On the **"HelloGraphics.c"** from the project file area on the LEFT hand side of AVR Studio. You will be presented with a blank file. Just copy/paste the code given below.

# ProgGFX - Embedded Graphic Library User Manual

---

## 2.4 “Hello GraphiX” project

*/\*\*\*\*\**

*Hello World Project Using ProGFX Engine. The application just prints the text "Hello GraphicX!" at the center of 128x64 Graphical LCD Module.*

*For More Information See*

*<http://www.ProGFX.org>*

*ProGFX makes it easy to interface graphical LCD modules with embedded microcontrollers. Currently Atmel AVR MCUs are supported. Latter PIC and ARM MCUs will also be supported.*

*Development Environment : AVR Studio v4.17  
Compiler : WinAVR 20090313*

*Written By  
Avinash Gupta  
[me@avinashgupta.com](mailto:me@avinashgupta.com)*

*\*\*\*\*\*/*

```
#include <gfx.h>
#include <font.h>
#include <fonts/arial12.h>
```

```
void main()
{
    //Initialize ProGFX Engine
    GFXInit();

    //Select a font
    GFXSetFont(Arial12);

    //Write a simple string at center
    GFXCenterText("Hello GraphiX!",GFX_COLOR_BLACK);

    //Update the display
    GFXUpdate();
}
```



# ProgGFX - Embedded Graphic Library User Manual

---

## 2.5 Compilation and Running

After entering code save the file. Then from the **"Build"** menu select **"Rebuild All"**. If you did every steps carefully then the project will compile without any problem and you will be get a HEX file. The hex file is found inside the **"default"** folder inside your project folder. In this case the file should be **"HelloGraphics.hex"**. You can program this hex file to the MCU using any standard ISP Programmer.

After running the demo you will get an output similar similar to the images shown below.



*Atmel AVR Based Graphical LCD Demo*

## 2.6 Troubleshooting.

If you get NO output on the LCD Panel then :-

- ❖ Adjust the RV1 variable resistor until you get proper display.
- ❖ Confirm that HIGH fuse is programmed to C9 (HEX) and LOW fuse is programmed to FF
- ❖ Check all connections.
- ❖ Check that 16MHz crystal is connected to AVR MCU
- ❖ Build Circuit exactly as shown in Chapter 1

**About:** ProGFX is a free project started by eXtreme Electronics India to promote use of graphical modules in new embedded projects. The development of ProGFX project requires money and LOTS of time. So please buy Graphical LCD Modules from our online store to help us in return. Currently these modules are only supplied within India.

<http://shop.eXtremeElectronics.co.in>