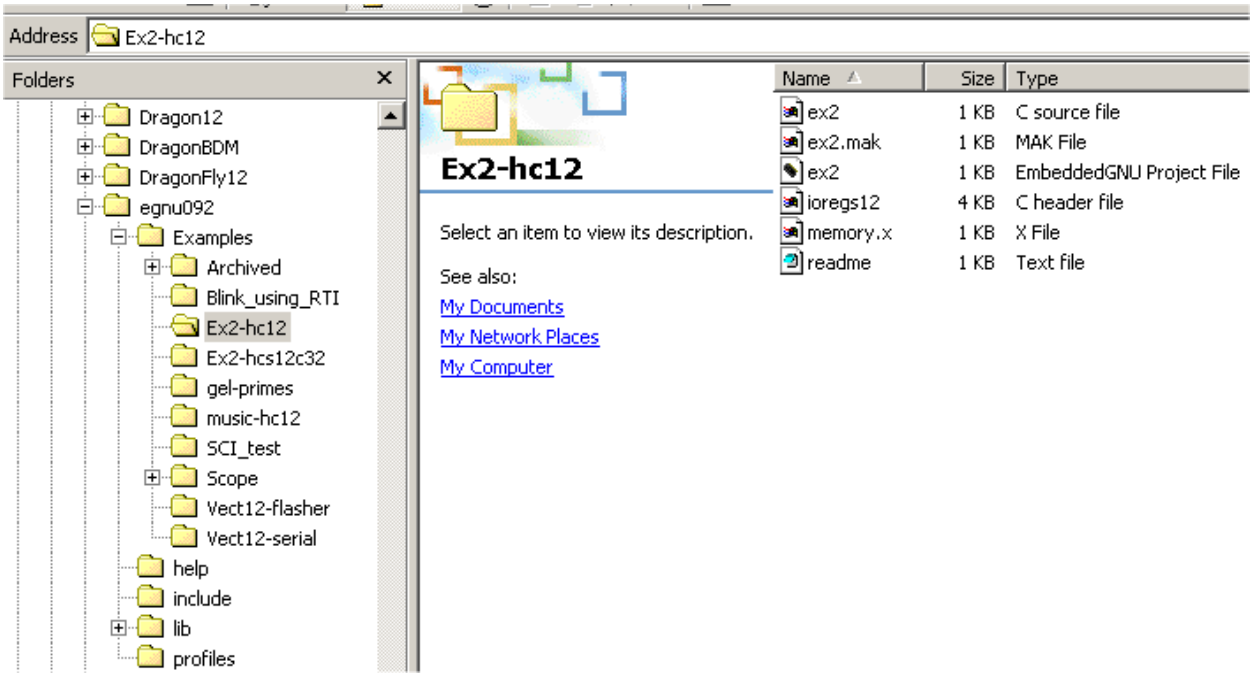
Eric Engler *EmbeddedGNU* IDE installed in F drive, GNU C-compiler toolset installed in C drive, sample C code ex2.c in F drive.



readme.txt

This is a simple program that flashes the LEDs on the **Wytec Dragon12** development board. This board uses a MC9S12DP256 (a type of 68hc12) processor.

http://www.evbplus.com/

ex2.c

```
#include  <stdio.h>
#include  <string.h>
#include  "ioregs12.h"

int  main(void);
void delay(void);

unsigned short int  speed = 0xffff;

unsigned short int        start = 0x0000;
unsigned short int        end   = 0xffff;
```

```
int main()
{
    unsigned short int n;

    COPCTL = 0x08; /* COP watchdog timer */
    COPCTL = 0x08;

    DDRB = 0xff; /* make all bits of port B outputs */

    /* DDRP = 0xff; -* make all bits of port P outputs */
    /* PTP = 0xff;  * turn-off 7 segment display */

    while (1)
    {
        for (n = start; n < end; n++)
        {
          PORTB = n;
          delay();
        }
    }
    return  0; /* not used */
}

void delay()
{
  unsigned short int i;

  for (i = 0; i < speed; i++)
  {
   /* nothing */
  }
}
```

ex2.mak

```
CC = c:/usr/bin/m6811-elf-gcc
RM = F:/egnu092/rm.exe

CFLAGS  = -m68hc12 -Os -fno-ident -fno-common -fomit-frame-pointer -
mshort
LDFLAGS = -Wl,-u,-mm68hc12elfb
OCFLAGS = -O srec

CSRCS=ex2.c

OBJS=$(CSRCS:.c=.o)

all:  ex2.elf ex2.s19

$(OBJS): %.o: %.c
```

```
	$(CC) $(CFLAGS) -c $<

ex2.elf:   $(OBJS)
	$(CC) $(CFLAGS) $(LDFLAGS) -o ex2.elf $(OBJS)

ex2.s19:   ex2.elf
	$(OC) $(OCFLAGS) ex2.elf ex2.s19

clean:
	$(RM) -f ex2.elf
	$(RM) -f ex2.s19
	$(RM) -f ex2.dmp
	$(RM) -f $(OBJS)
```

ioregs12.h

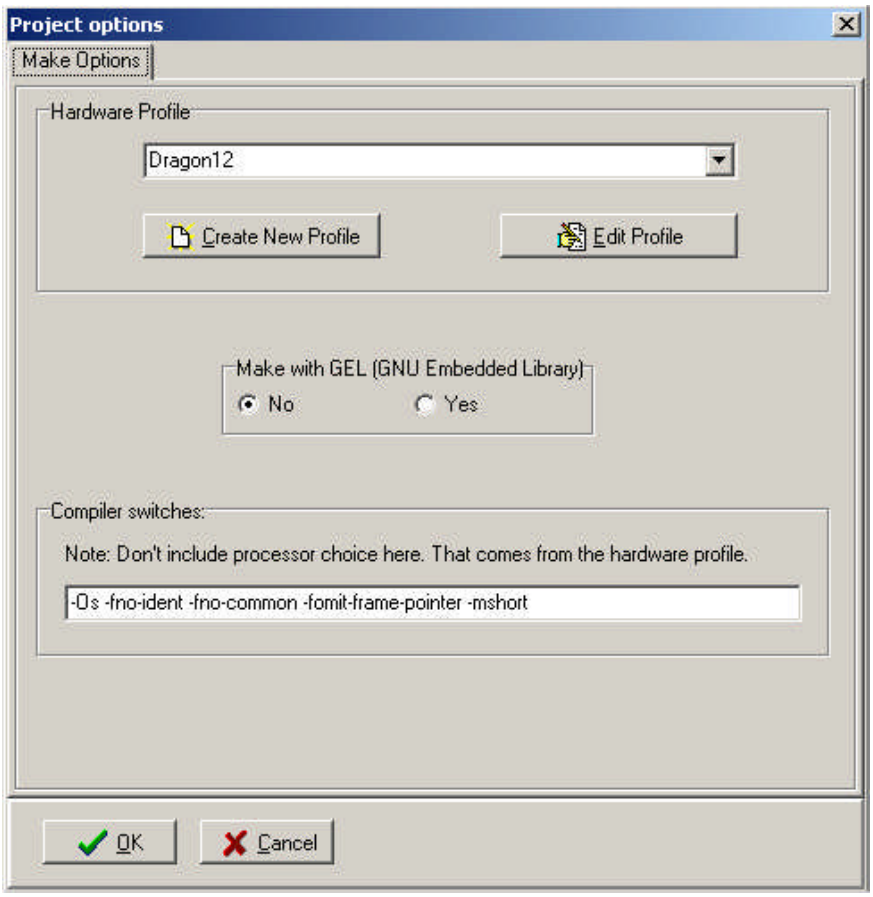
```
#ifndef  IOREGS12_H
#define  IOREGS12_H

#ifndef  IOREGS_BASE
#define  IOREGS_BASE    0x0000
#endif

#define _IO8(off)       *(unsigned char  volatile *)(IOREGS_BASE +
off)
#define _IO16(off)      *(unsigned short volatile *)(IOREGS_BASE +
off)
#define  PORTA      _IO8(0x00)      /* i/o port a */
#define  PORTB      _IO8(0x01)      /* i/o port b */
#define  DDRA       _IO8(0x02)      /* data direction reg a */
#define  DDRB       _IO8(0x03)      /* data direction reg b */
#define  PORTE      _IO8(0x08)      /* i/o port e */
#define  DDRE       _IO8(0x09)      /* data direction reg e */
#define  PEAR       _IO8(0x0a)      /* pear */
#define  MODE       _IO8(0x0b)      /* mode */
#define  PUCR       _IO8(0x0c)      /* pucr */
#define  RDRIV      _IO8(0x0d)      /* rdriv */
#define  EBICTL     _IO8(0x0e)      /* e stretch control */

#define  INITRM     _IO8(0x10)      /* initrm */
#define  INITRG     _IO8(0x11)      /* initrg */
#define  INITEE     _IO8(0x12)      /* initee */
#define  MISC       _IO8(0x13)      /* misc */
#define  MTST0      _IO8(0x14)      /* mtst0 */
#define  ITCR       _IO8(0x15)      /* itcr */
#define  ITEST      _IO8(0x16)      /* itest */
#define  MTST1      _IO8(0x17)      /* mtst1 */

#define  PARTIDH    _IO8(0x1a)      /* partidh */
#define  PARTIDL    _IO8(0x1b)      /* partidl */
#define  MEMSIZ0    _IO8(0x1c)      /* memsiz0 */
```

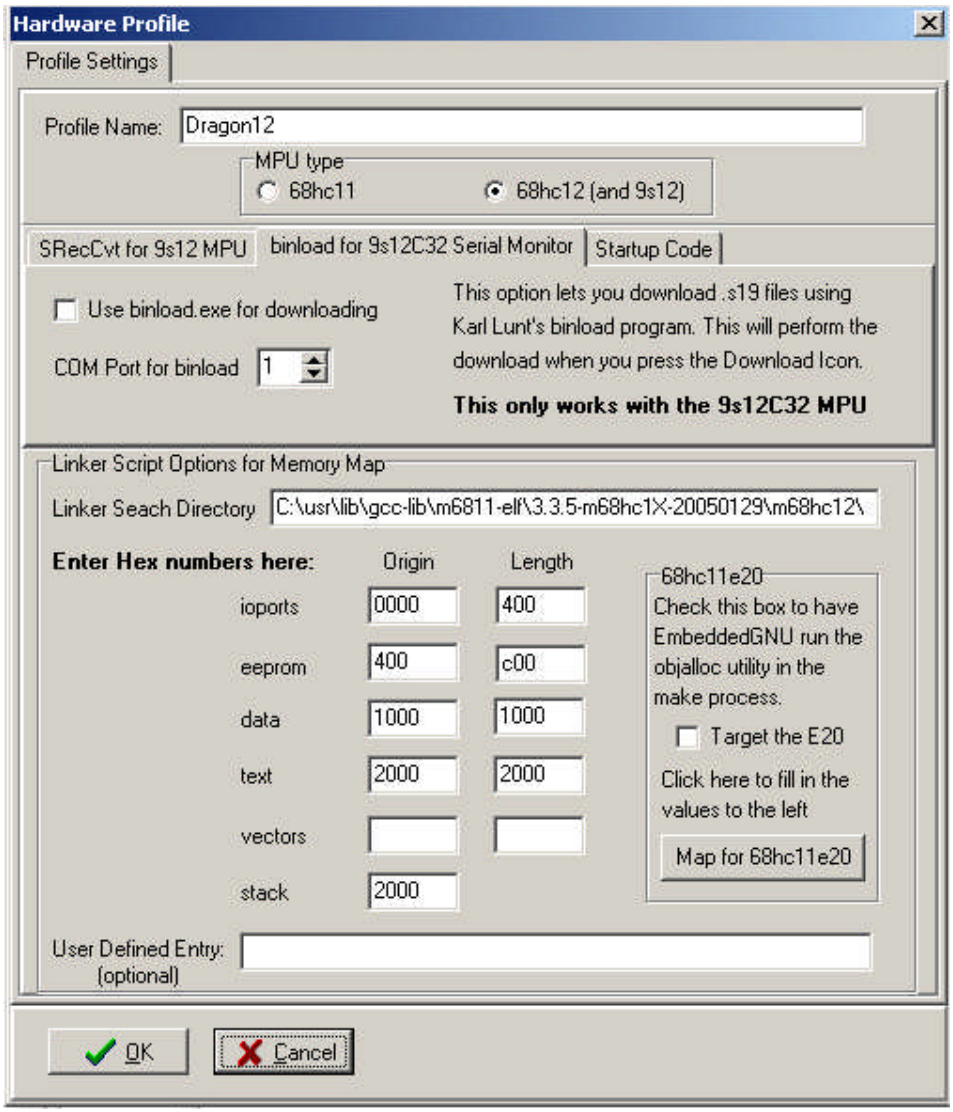
```
#define  MEMSIZ1    _IO8(0x1d)      /* memsiz1 */

#define  INTCR      _IO8(0x1e)      /* intcr */
#define  HPRIO      _IO8(0x1f)      /* hprio */

#define  BKPCT0     _IO8(0x28)
#define  BKPCT1     _IO8(0x29)
#define  BKP0X      _IO8(0x2a)
#define  BKP0H      _IO8(0x2b)
#define  BRP0l      _IO8(0x2c)
#define  BKP1X      _IO8(0x2d)
#define  BKP1H      _IO8(0x2e)
#define  BRP1L      _IO8(0x2f)

#define  PPAGE      _IO8(0x30)
#define  PORTK      _IO8(0x32)   /* Port K */
#define  DDRK       _IO8(0x33)
#define  SYNR       _IO8(0x34)
#define  REFDV      _IO8(0x35)
#define  CTFLG      _IO8(0x36)
#define  CRGFLG     _IO8(0x37)
#define  CRGINT     _IO8(0x38)
#define  CLKSEL     _IO8(0x39)
#define  PLLCTL     _IO8(0x3a)
#define  RTICTL     _IO8(0x3b)
#define  COPCTL     _IO8(0x3c)
#define  FORBYP     _IO8(0x3d)
#define  CTCTL      _IO8(0x3e)
#define  ARMCOP     _IO8(0x3f)

#define  TIOS       _IO8(0x40)
#define  CFORC      _IO8(0x41)
#define  OC7M       _IO8(0x42)
#define  OC7D       _IO8(0x43)
#define  TCNT       _IO16(0x44)  /* $44 = Hi byte, $45 = low byte */
#define  TSCR       _IO8(0x46)
#define  TTOV       _IO8(0x47)
#define  TCTL1      _IO8(0x48)
#define  TCTL2      _IO8(0x49)
#define  TCTL3      _IO8(0x4a)
#define  TCTL4      _IO8(0x4b)
#define  TMSK1      _IO8(0x4c)
#define  TMSK2      _IO8(0x4d)
#define  TFLG1      _IO8(0x4e)
#define  TFLG2      _IO8(0x4f)

#define  TC0        _IO16(0x50)
#define  TC1        _IO16(0x52)
#define  TC2        _IO16(0x54)
#define  TC3        _IO16(0x56)
#define  TC4        _IO16(0x58)
#define  TC5        _IO16(0x5a)
```

```
#define  TC6        _IO16(0x5c)
#define  TC7        _IO16(0x5e)

#define  PACTL      _IO8(0x60)
#define  PAFLG      _IO8(0x61)
#define  PACN3      _IO8(0x62)  /* Pulse Accumulator */
#define  PACN2      _IO8(0x63)
#define  PACN1      _IO8(0x64)
#define  PACN0      _IO8(0x65)
#define  MCCTL      _IO8(0x66)
#define  MCFLG      _IO8(0x67)
#define  ICPAR      _IO8(0x68)
#define  DLYCT      _IO8(0x69)
#define  ICOVW      _IO8(0x6a)
#define  ICSYS      _IO8(0x6b)
#define  TIMTST     _IO8(0x6d)

#define  PORTT      _IO8(0x240)      /* i/o port T */
#define  DDRT       _IO8(0x242)      /* data direction reg T */

#define  PORTS      _IO8(0x248)      /* i/o port S */
#define  DDRS       _IO8(0x24a)      /* data direction reg S */

#define  PORTM      _IO8(0x250)      /* i/o port M */
#define  DDRM       _IO8(0x252)      /* data direction reg M */

#define  PORTP      _IO8(0x258)      /* i/o port P */
#define  DDRP       _IO8(0x25a)      /* data direction reg P */
#endif
```

memory.x

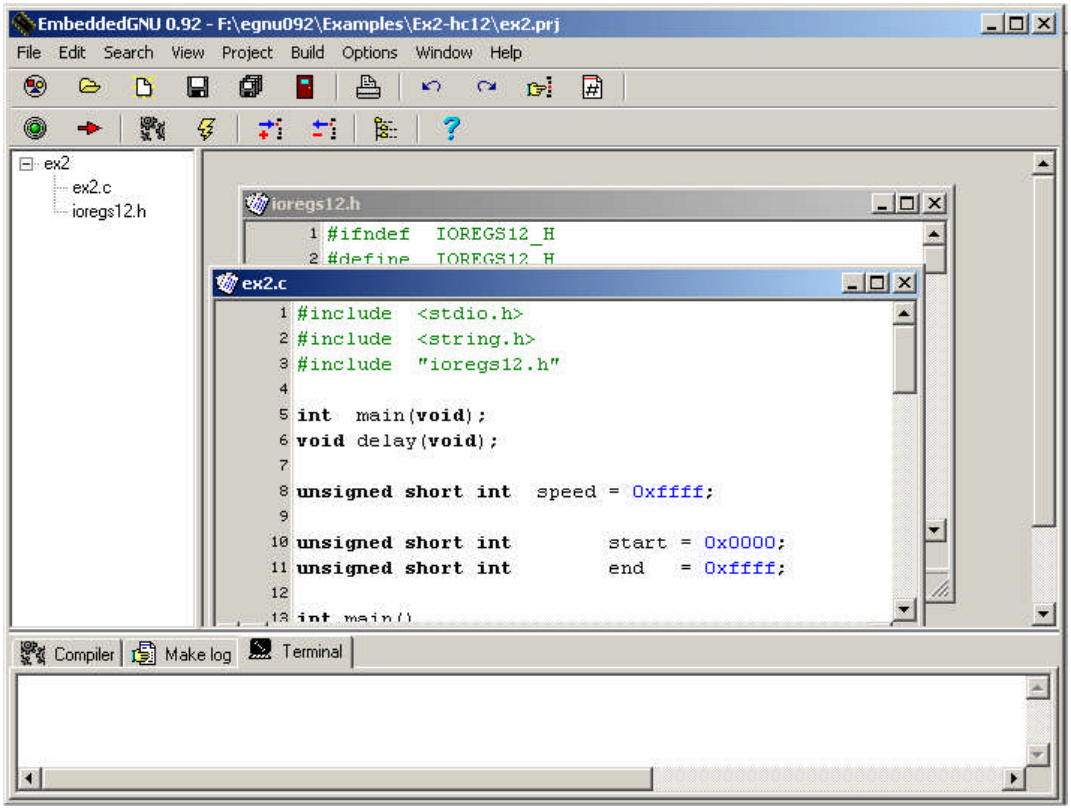
```
OUTPUT_FORMAT("elf32-m68hc12", "elf32-m68hc12",
            "elf32-m68hc12")
OUTPUT_ARCH(m68hc12)
ENTRY(_start)
SEARCH_DIR(C:\usr\lib\gcc-lib\m6811-elf\3.3.5-m68hc1X-
20050129\m68hc12\mshort)

MEMORY
{
  ioports (!x)  : org = 0x0000, l = 0x400
  eeprom  (!i)  : org = 0x400, l = 0xc00
  data    (rwx) : org = 0x1000, l = 0x1000
  text    (rx)  : org = 0x2000, l = 0x2000
}

PROVIDE (_stack = 0x2000);
```


Project options
Make Options
Hardware Profile
Dragon12
Create New Profile
Edit Profile
Make with GEL (GNU Embedded Library)
No
Yes
Compiler switches:
Note: Don't include processor choice here. That comes from the hardware profile.
-Os -fno-ident -fno-common -fomit-frame-pointer -mshort
OK
Cancel

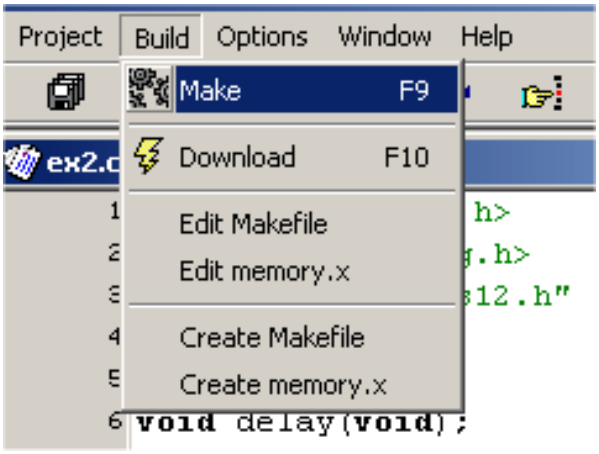

Hardware Profile
Profile Settings
Profile Name: Dragon12
MPU type
68hc11
68hc12 (and 9s12)
SRecCvt for 9s12 MPU
binload for 9s12C32 Serial Monitor
Startup Code
Use binload.exe for downloading
COM Port for binload
1
This option lets you download .s19 files using Karl Lunt's binload program. This will perform the download when you press the Download Icon.
This only works with the 9s12C32 MPU
Linker Script Options for Memory Map
Linker Seach Directory
C:\usr\lib\gcc-lib\m6811-elf\3.3.5-m68hc1X-20050129\m68hc12\
Enter Hex numbers here:
Origin
Length
ioports
0000
400
eeprom
400
c00
data
1000
1000
text
2000
2000
vectors
stack
2000
68hc11e20
Check this box to have EmbeddedGNU run the objalloc utility in the make process.
Target the E20
Click here to fill in the values to the left
Map for 68hc11e20
User Defined Entry: (optional)
OK
Cancel

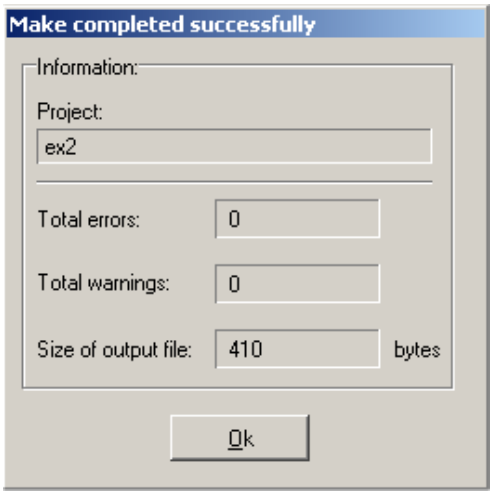
GNU C-compiler Toolset installed in C: drive

***EmbeddedGNU* IDE Linker Search Path for GNU C compiler toolset points to C drive (while sample ex2.c in F drive)**

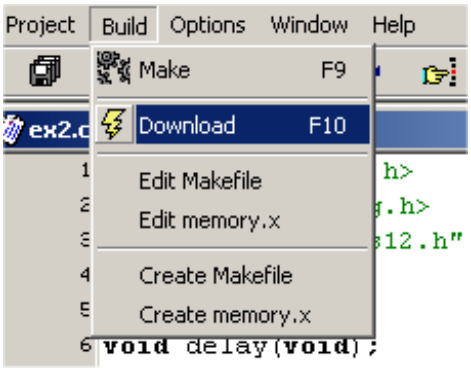


Invoke Make





Download ex2.s19 object file to DRAGON 12 & execute automatically.



On DRAGON 12 board observe the eight discrete LEDs & four 7-segment LEDs.