Bharath Muthuswamy
Email corrections, comments to: mbharat@cory.eecs.berkeley.edu

Dave & Keil Configuration for the Phytec KitCON 167

1. Introduction

The purpose of this document is to help a user **easily** configure Dave and the Keil IDE (Integrated Development Environment) for use with your microcontroller board, the Phytec KitCON 167. You use Dave to generate C functions that will be used in Keil to write C programs.

2. Why configure?

Unlike writing a program for an IBM compatible, you are coding for an embedded system[1] using a different processor, with much less memory, no hard disk etc – you get the idea! So, you have to tell your host computer (the one in which you are writing your program) what kind of target architecture to expect.

This document is simply organized into two sections. The first section shows you how to configure Dave and the next section deals with Keil. Please note: this document does not tell you how to use Dave and Keil. It just helps you to set them up.

3. Configuring Dave

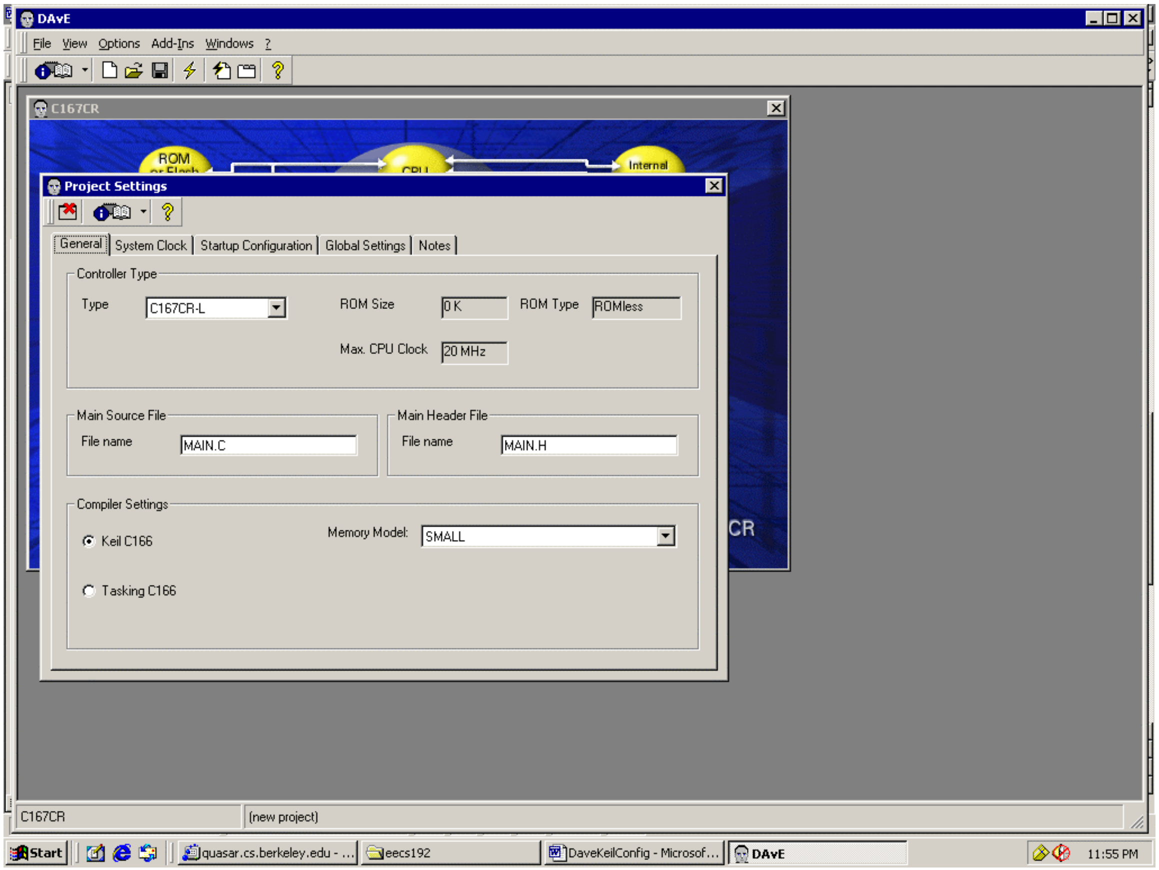
The first step is to configure Dave. Dave generates five types of files:

1. C source files: These have an extension "c". For example, timer.c and pwm.c. These files contain the C code generated by Dave.
2. C header files: These have an extension "h". For example, timer.h and pwm.h. These files contain the preprocessor directives (#define statements) and other constant declarations.
3. Documentation file: This has an extension "RTF" (it stands for Rich Text Format). This file has a description of your C functions and other stuff. Please take time to look over this file as it contains very useful information.
4. Start167.a66: This file has configuration information specific to your board. This information is set when you first create the project.
5. Dave Project File: This file has an extension "dpt". You open this file in Keil to use the code generated by Dave.

Obviously, the only file that has startup information is Start167.a66. However, it turns out that Keil needs to only the clock speed of your microcontroller. This is because the Keil IDE uses its own software to communicate with the board. However, it will be helpful to set the options (other than clock speed) correctly in Dave and understand what is going on. First, let us set the correct clock speed:

[1] This form of coding is usually called cross compiling.

1. Create a new project in Dave.
2. Choose the C167CR microcontroller.
3. You will be shown the **Project Settings** window as shown below:


DAvE
File View Options Add-Ins Windows ?
C167CR
ROM or Flash
CPU
Internal
CR
Project Settings
General | System Clock | Startup Configuration | Global Settings | Notes
C167CR
(new project)
Start
quasar.cs.berkeley.edu - ...
eecs192
DaveKeilConfig - Microsof...
DAvE
11:55 PM

Controller Type

Type: C167CR-L ROM Size: 0 K ROM Type: ROMless

Max. CPU Clock: 20 MHz

Main Source File — File name: MAIN.C

Main Header File — File name: MAIN.H

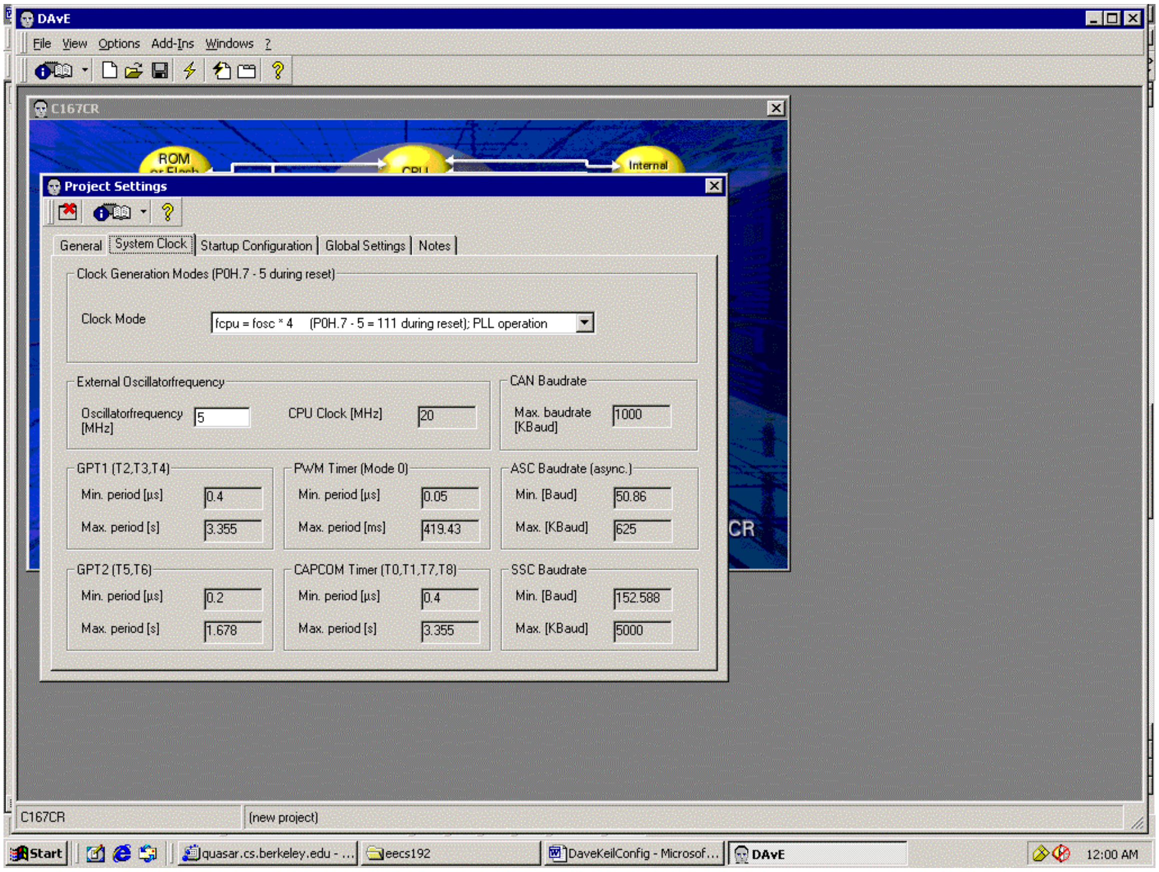
Compiler Settings

● Keil C166 Memory Model: SMALL

○ Tasking C166

Figure 1. Project Settings screen in Dave.

4. Click on the **System Clock** tab.



Clock Generation Modes (P0H.7 - 5 during reset)

Clock Mode: fcpu = fosc * 4 (P0H.7 - 5 = 111 during reset); PLL operation

External Oscillatorfrequency

Oscillatorfrequency [MHz]: 5

CPU Clock [MHz]: 20

CAN Baudrate

Max. baudrate [KBaud]: 1000

GPT1 (T2,T3,T4)

Min. period [µs]: 0.4

Max. period [s]: 3.355

PWM Timer (Mode 0)

Min. period [µs]: 0.05

Max. period [ms]: 419.43

ASC Baudrate (async.)

Min. [Baud]: 50.86

Max. [KBaud]: 625

GPT2 (T5,T6)

Min. period [µs]: 0.2

Max. period [s]: 1.678

CAPCOM Timer (T0,T1,T7,T8)

Min. period [µs]: 0.4

Max. period [s]: 3.355

SSC Baudrate

Min. [Baud]: 152.588

Max. [KBaud]: 5000

Figure 2. Change the oscillator frequency to 5 MHz.

5. Set the **Oscillator Frequency** option to 5 MHz as shown in figure 2. This helps your controller attain the maximum clock speed of 20 MHz. A little explanation is at hand. Multiplying the onboard oscillator with 1, 2, 3, 4 or 5 sets your clock mode. This allows for a variety of input clock frequencies. By default, your multiplier is set to 4. You can change this on your KitCON board using DIP switches 6, 7 or 8. Refer to figure 3 for a picture of the DIP switch. Refer to the KitCON hardware manual on setting clock modes using DIP switch S3. Please DO NOT change the DIP switch settings on the microcontroller board unless you know what you are doing!

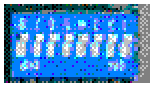
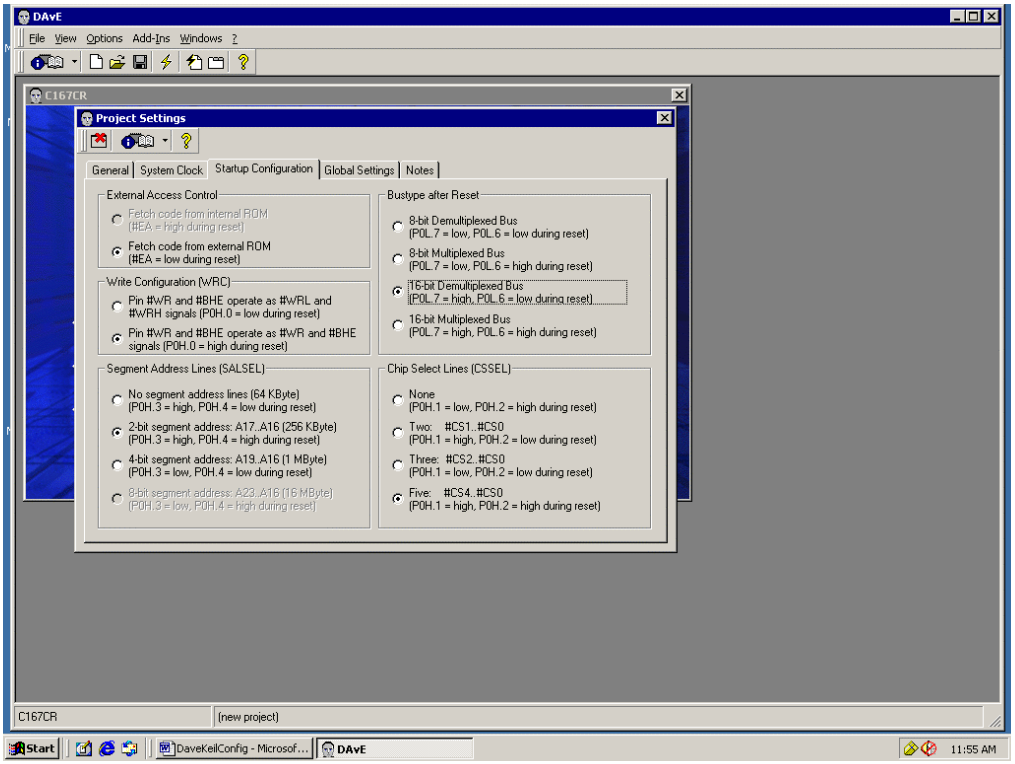


Figure 3. DIP switch S3. Switch 1 controls the boot mode (FLASH or RAM), switches 2 and 3 control the number of chip select signals, switches 4 and 5 control the number of segment address line and switches 6;7 and 8 control the clock mode. For a detailed description of the DIP switch please refer to the KitCON hardware manual.


DAvE
File View Options Add-Ins Windows ?
C167CR
Project Settings
General | System Clock | Startup Configuration | Global Settings | Notes
C167CR
(new project)
Start
DaveKeilConfig - Microsof...
DAvE
11:55 AM

External Access Control

○ Fetch code from internal ROM (#EA = high during reset)
● Fetch code from external ROM (#EA = low during reset)

Write Configuration (WRC)

○ Pin #WR and #BHE operate as #WRL and #WRH signals (P0H.0 = low during reset)
● Pin #WR and #BHE operate as #WR and #BHE signals (P0H.0 = high during reset)

Segment Address Lines (SALSEL)

○ No segment address lines (64 KByte) (P0H.3 = high, P0H.4 = low during reset)
● 2-bit segment address: A17..A16 (256 KByte) (P0H.3 = high, P0H.4 = high during reset)
○ 4-bit segment address: A19..A16 (1 MByte) (P0H.3 = low, P0H.4 = low during reset)
○ 8-bit segment address: A23..A16 (16 MByte) (P0H.3 = low, P0H.4 = high during reset)

Bustype after Reset

○ 8-bit Demultiplexed Bus (P0L.7 = low, P0L.6 = low during reset)
○ 8-bit Multiplexed Bus (P0L.7 = low, P0L.6 = high during reset)
● 16-bit Demultiplexed Bus (P0L.7 = high, P0L.6 = low during reset)
○ 16-bit Multiplexed Bus (P0L.7 = high, P0L.6 = high during reset)

Chip Select Lines (CSSEL)

○ None (P0H.1 = low, P0H.2 = high during reset)
○ Two: #CS1..#CS0 (P0H.1 = high, P0H.2 = low during reset)
○ Three: #CS2..#CS0 (P0H.1 = low, P0H.2 = low during reset)
● Five: #CS4..#CS0 (P0H.1 = high, P0H.2 = high during reset)

Figure 4. Select the correct bus type

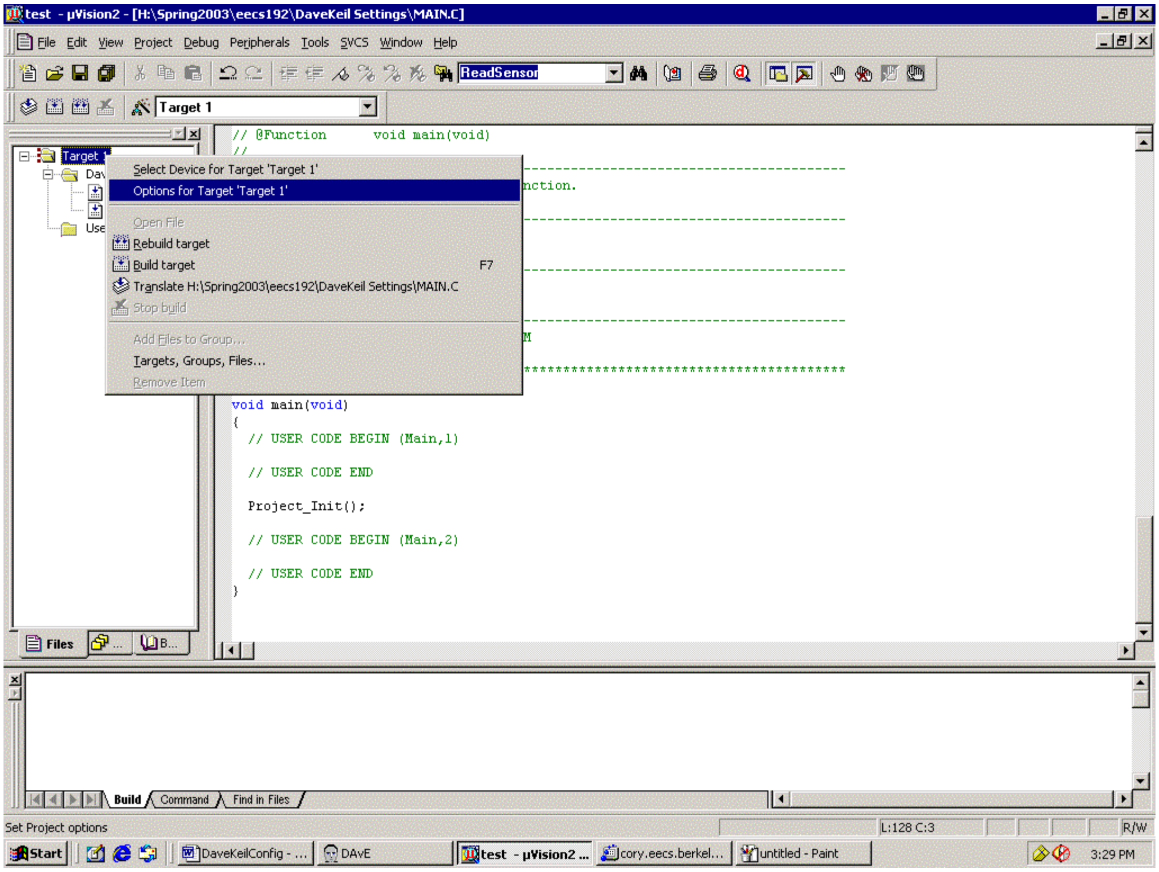
6. Click the **Startup Configuration** tab. Once again, this section is optional since Keil does not need this information. All these settings are determined by DIP S3 switches on your controller board.

<u>(a) External Access Control:</u> The C167CR does not have on-board ROM, so you have no option here.
<u>(b) Write Configuration:</u> This is hardwired to 0 by default. Refer to the KitCON hardware manual, under the section “Configuration during System Reset S3”. It will show that bit WRC of port 0 is hardwired to 0.
<u>(c) Segment Address Lines:</u> Switches 4 and 5 of DIP switch S3 control the number of segment address lines and they are set to 2 by default. Once again, refer to the KitCON hardware manual. You can find the information under the section on DIP switch S3 Segment Address Lines.
<u>(d) Bustype after Reset:</u> If you check your KitCON manual under the section “Configuration during System Reset S3”. You will notice that bits 7 and 6 of the low byte of port zero are 1 and 0 respectively. These bits control the bus type. Look up the External Bus Type section in the System Reset chapter in the C167 hardware manual. You will see that 1 0 corresponds to a 16-bit demultiplexed bus. Set this option in Dave.
<u>(d) Chip Select lines:</u> By default, DIP switch S3 sets them to five.

You are done configuring Dave. Click on ☒ to close the Project Settings screen and now you can start configuring the peripherals that you want to use. Dave will generate the C code for these peripherals that you can use in Keil, which we configure next.

4. Configuring Keil

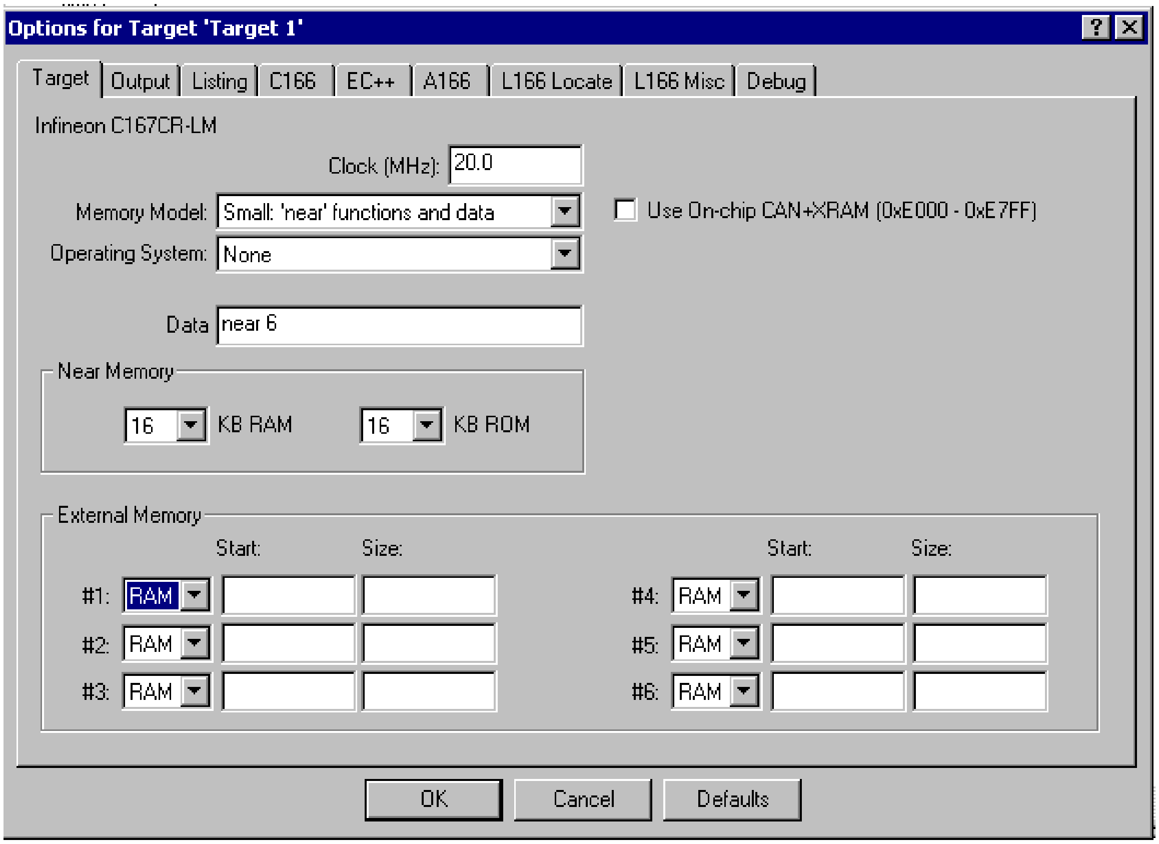
1. Once you finish configuring and setting up the peripherals in Dave, fire up Keil by double clicking on the Keil uVision 2 icon.
2. Click on **Project** and **Open Project**.
3. Go to the directory where your project is stored. Choose Files of type: Dave Project Files (*.dpt). You should see your Dave project file. Double click on it to open the project. You should see the screen shown in figure 5. Right click on Target 1 as and choose Options for Target 'Target 1' as shown figure 5.


test - µVision2 - [H:\Spring2003\eecs192\DaveKeil Settings\MAIN.C]
File Edit View Project Debug Peripherals Tools SVCS Window Help
ReadSensor
Target 1
Target 1
Dav
Use
Select Device for Target 'Target 1'
Options for Target 'Target 1'
Open File
Rebuild target
Build target F7
Translate H:\Spring2003\eecs192\DaveKeil Settings\MAIN.C
Stop build
Add Files to Group...
Targets, Groups, Files...
Remove Item
Files
B...
Build
Command
Find in Files
Set Project options
L:128 C:3
R/W
Start
DaveKeilConfig - ...
DAvE
test - µVision2 ...
cory.eecs.berkel...
untitled - Paint
3:29 PM

```
// @Function      void main(void)
//
---------------------------------------------
nction.
---------------------------------------------
---------------------------------------------
---------------------------------------------
M
*********************************************
void main(void)
{
  // USER CODE BEGIN (Main,1)

  // USER CODE END

  Project_Init();

  // USER CODE BEGIN (Main,2)

  // USER CODE END
}
```

Figure 5. Starting Keil configuration

You will see the screen in figure 6.



Infineon C167CR-LM

Clock (MHz): 20.0

Memory Model: Small: 'near' functions and data

☐ Use On-chip CAN+XRAM (0xE000 - 0xE7FF)

Operating System: None

Data near 6

Near Memory

16 KB RAM 16 KB ROM

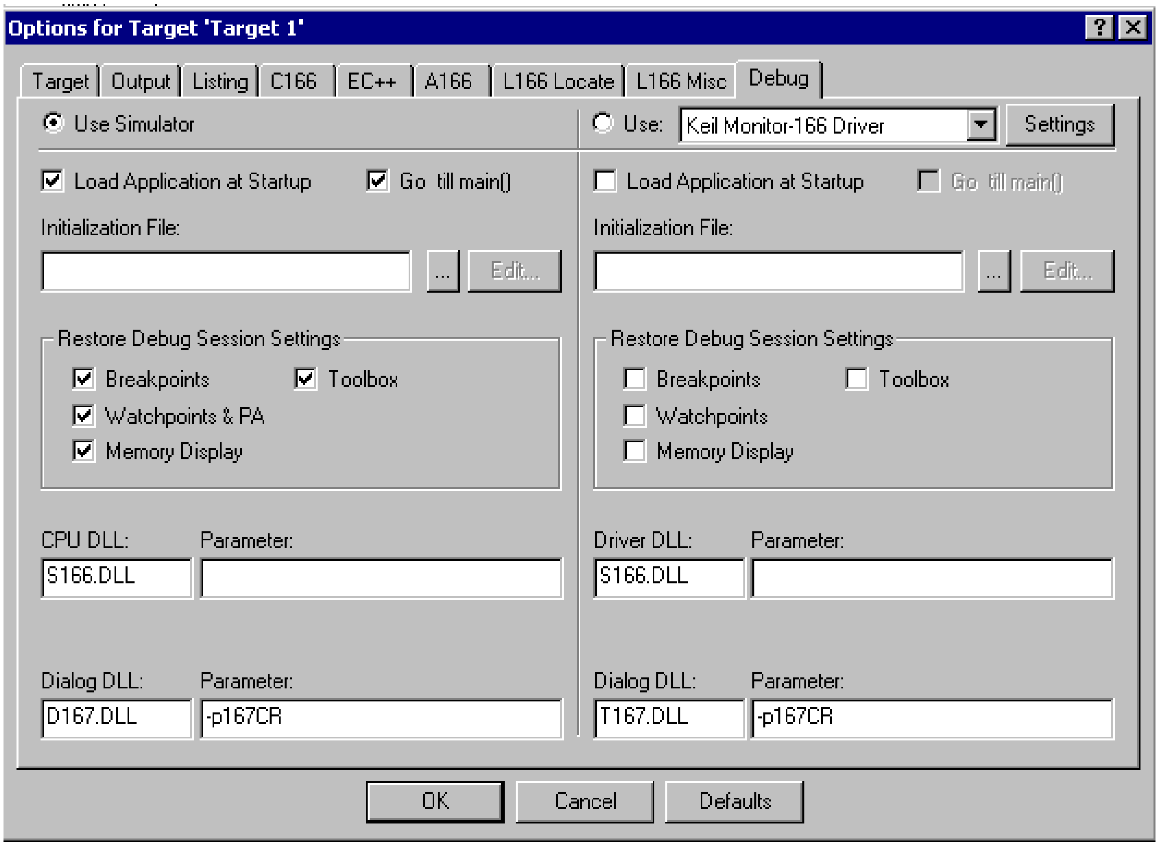
External Memory

		Start:	Size:			Start:	Size:
#1:	RAM			#4:	RAM		
#2:	RAM			#5:	RAM		
#3:	RAM			#6:	RAM		

Figure 6. Setting memory size

The first step you need to do is set up Keil to communicate with the microcontroller board:

1. Click on the **Debug** tab. You will see the screen in figure 7.



Options for Target 'Target 1'

Target | Output | Listing | C166 | EC++ | A166 | L166 Locate | L166 Misc | Debug

● Use Simulator

☑ Load Application at Startup ☑ Go till main()

Initialization File:

Restore Debug Session Settings

☑ Breakpoints ☑ Toolbox

☑ Watchpoints & PA

☑ Memory Display

CPU DLL: S166.DLL Parameter:

Dialog DLL: D167.DLL Parameter: -p167CR

○ Use: Keil Monitor-166 Driver Settings

☐ Load Application at Startup ☐ Go till main()

Initialization File:

Restore Debug Session Settings

☐ Breakpoints ☐ Toolbox

☐ Watchpoints

☐ Memory Display

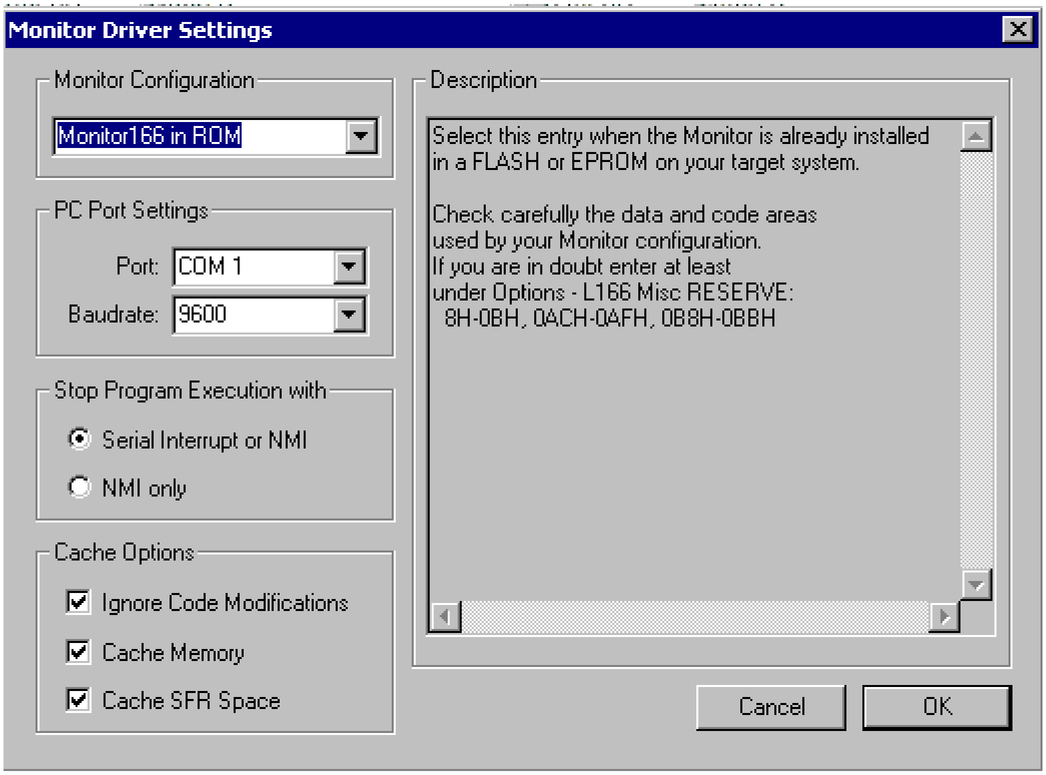
Driver DLL: S166.DLL Parameter:

Dialog DLL: T167.DLL Parameter: -p167CR

Figure 7. Choosing the monitor

You need to switch from the simulator to the monitor. The monitor is the program that helps in the communication between the microcontroller board and Keil. It is downloaded to your board RAM before your program. To choose the monitor, click on **Keil Monitor-166 Driver**. Check **Load Application at Startup**, **Go till main()** and check all the options under **Restore Debug Session Settings**.

Once you have told Keil to use the monitor, you have to select the correct monitor for your board. Click the **Settings** button shown in figure 7. You will see the Monitor Driver Settings window (see figure 8).



Monitor Driver Settings

Monitor Configuration: Monitor166 in ROM

PC Port Settings

Port: COM 1

Baudrate: 9600

Stop Program Execution with

● Serial Interrupt or NMI

○ NMI only

Cache Options

☑ Ignore Code Modifications

☑ Cache Memory

☑ Cache SFR Space

Description

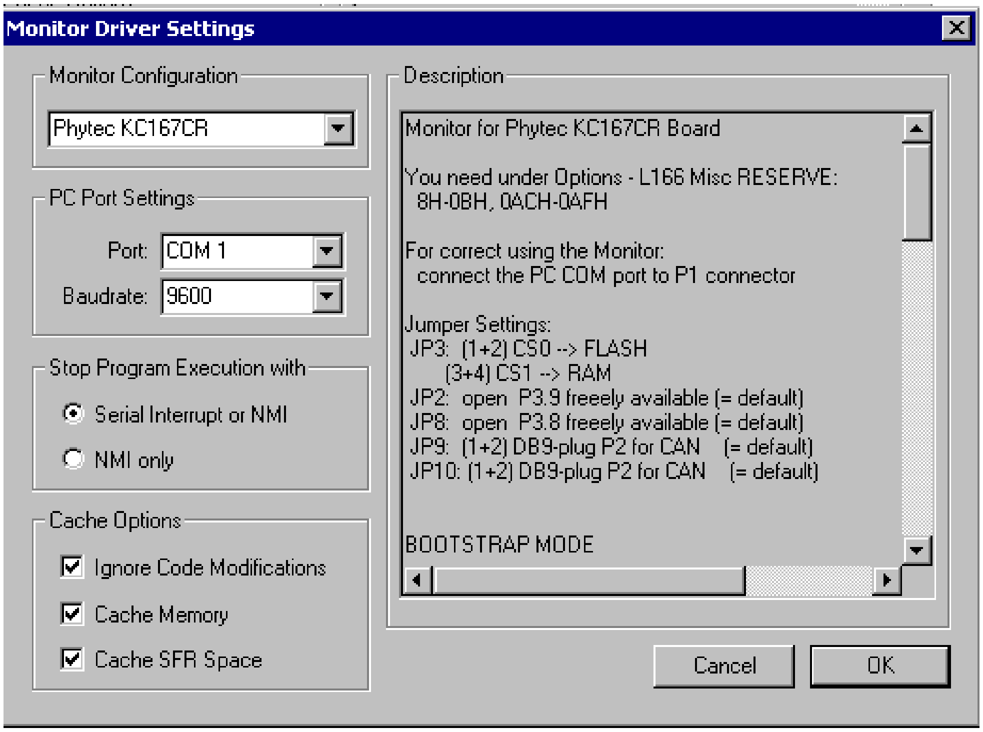
Select this entry when the Monitor is already installed
in a FLASH or EPROM on your target system.

Check carefully the data and code areas
used by your Monitor configuration.
If you are in doubt enter at least
under Options - L166 Misc RESERVE:
 8H-0BH, 0ACH-0AFH, 0B8H-0BBH

Figure 8. Incorrect monitor

In the drop box that has "Monitor166 in ROM", change it to "Phytec KC167CR". Change the **Stop Program Execution with** option from "NMI only" to "Serial Interrupt or NMI" as shown in figure 9. NMI stands for Non-Maskable Interrupt. If you choose NMI only, then you will not be able to stop the program from Keil. The only way to stop the program is to reset the board by hitting the black reset button on the microcontroller board. If you choose the "Serial Interrupt or NMI" option, you will be able to stop program execution from Keil.

In the Description box, you will see explanations for further options you should set. The main idea you must understand is the memory model. Scroll down in the Description box and notice the memory mapping under "Bootstrap mode". You have a total of 64 kB of RAM on board. This region is mapped in the Keil IDE into ROM (for code and constant data) and RAM (for volatile data like variables). You must also reserve some memory on the microcontroller board for the monitor program that you are going to download to the board.



Monitor Driver Settings

Monitor Configuration: Phytec KC167CR

PC Port Settings
Port: COM 1
Baudrate: 9600

Stop Program Execution with
● Serial Interrupt or NMI
○ NMI only

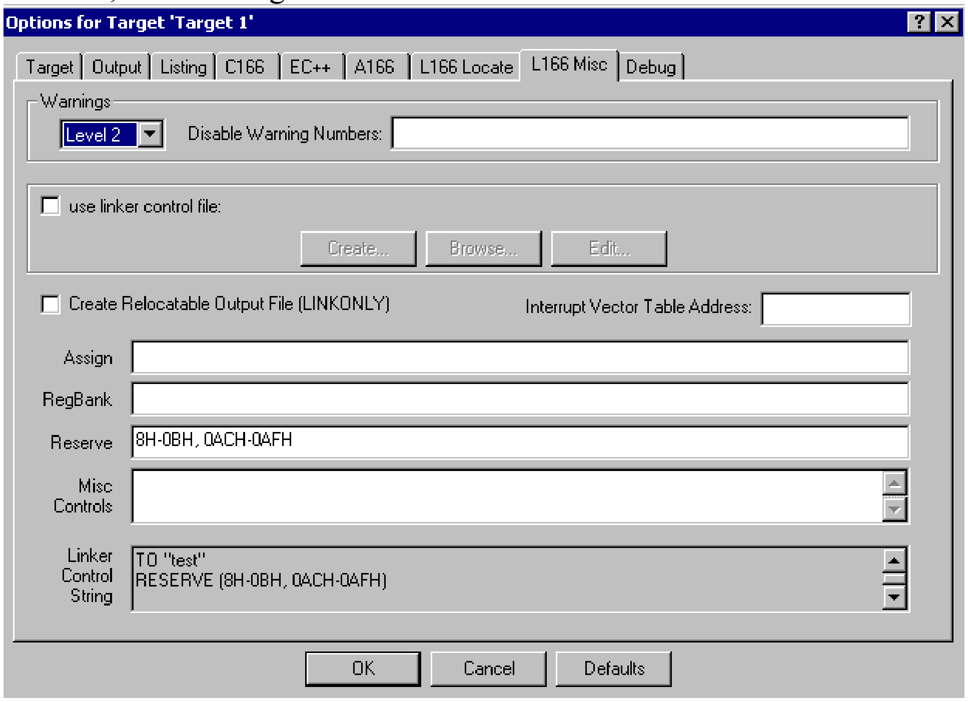
Cache Options
☑ Ignore Code Modifications
☑ Cache Memory
☑ Cache SFR Space

Description

```
Monitor for Phytec KC167CR Board

You need under Options - L166 Misc RESERVE:
  8H-0BH, 0ACH-0AFH

For correct using the Monitor:
  connect the PC COM port to P1 connector

Jumper Settings:
 JP3:  (1+2) CS0 --> FLASH
       (3+4) CS1 --> RAM
 JP2:  open  P3.9 freeely available (= default)
 JP8:  open  P3.8 freeely available (= default)
 JP9:  (1+2) DB9-plug P2 for CAN    (= default)
 JP10: (1+2) DB9-plug P2 for CAN    (= default)


BOOTSTRAP MODE
```

Figure 9. Correct monitor settings

1. First, let us reserve some memory for the monitor. Click on OK to close the Monitor Driver Settings window and click on the **L166 Misc.** tab. Type the values in the Reserve field, shown in figure 10.



Options for Target 'Target 1'

Target | Output | Listing | C166 | EC++ | A166 | L166 Locate | L166 Misc | Debug

Warnings: Level 2 Disable Warning Numbers:

□ use linker control file:

Create...
Browse...
Edit...

□ Create Relocatable Output File (LINKONLY)

Interrupt Vector Table Address:

Assign:

RegBank:

Reserve: 8H-0BH, 0ACH-0AFH

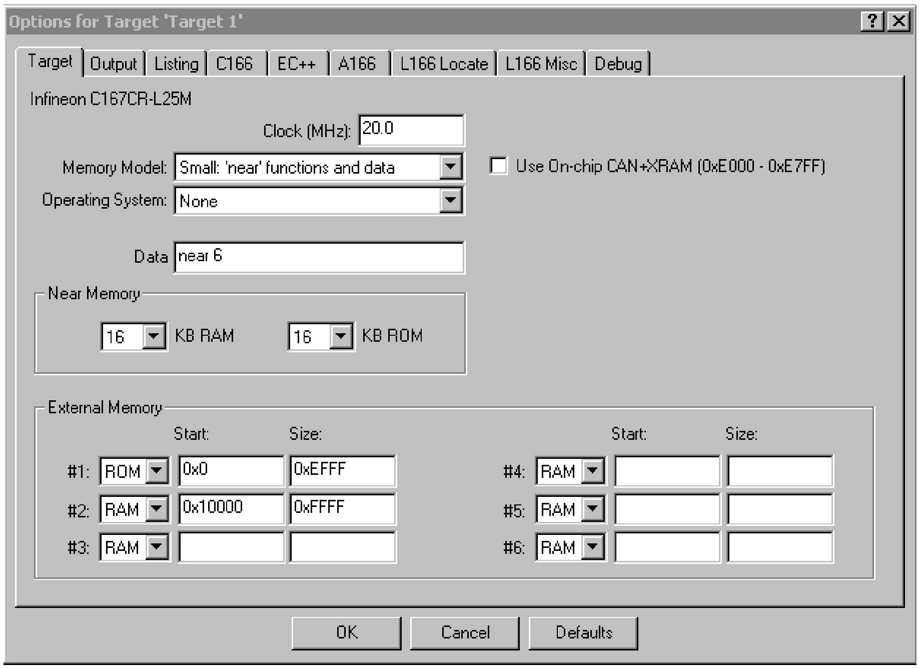
Misc Controls:

Linker Control String:

```
TO "test"
RESERVE (8H-0BH, 0ACH-0AFH)
```

Figure 10. Reserving memory for the monitor

3. Now, we just have to reserve memory for code and data. Click on the **Target** tab and change the options under Target to match figure 11.


Options for Target 'Target 1'
Target | Output | Listing | C166 | EC++ | A166 | L166 Locate | L166 Misc | Debug
OK
Cancel
Defaults

Infineon C167CR-L25M

Clock (MHz): 20.0

Memory Model: Small: 'near' functions and data

□ Use On-chip CAN+XRAM (0xE000 - 0xE7FF)

Operating System: None

Data near 6

Near Memory

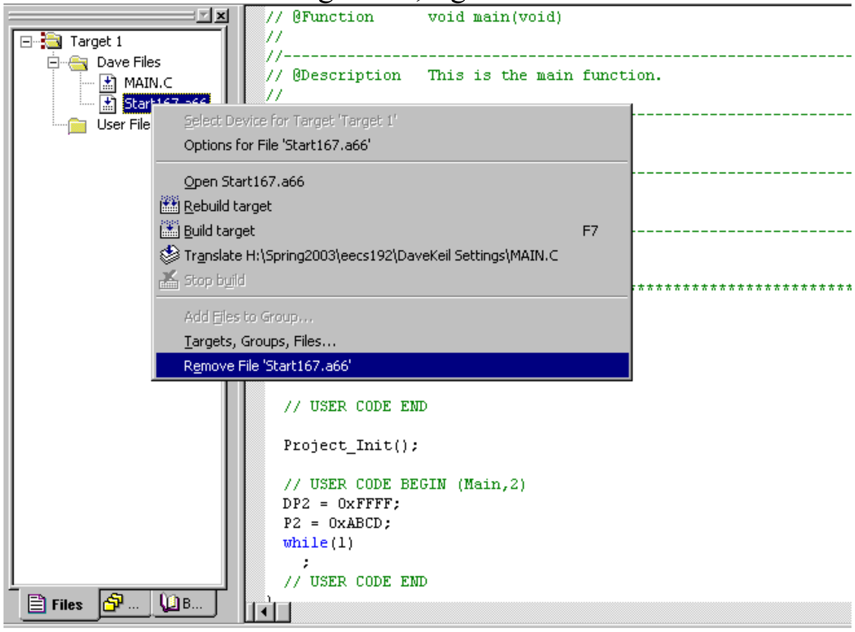
16 KB RAM 16 KB ROM

External Memory

		Start:	Size:			Start:	Size:
#1:	ROM	0x0	0xEFFF	#4:	RAM		
#2:	RAM	0x10000	0xFFFF	#5:	RAM		
#3:	RAM			#6:	RAM		

Figure 11. Reserving memory for code and data

Click on OK to accept all the changes. The final step is to remove the Start167.a66 file from your project. You don't need this file since the Keil Monitor takes care of other configuration stuff. As shown in figure 12, right click on Start167.a66 and remove it.


Target 1
Dave Files
MAIN.C
Start167.a66
User File
Select Device for Target 'Target 1'
Options for File 'Start167.a66'
Open Start167.a66
Rebuild target
Build target F7
Translate H:\Spring2003\eecs192\DaveKeil Settings\MAIN.C
Stop build
Add Files to Group...
Targets, Groups, Files...
Remove File 'Start167.a66'
Files
...
B...

```
// @Function      void main(void)
//
//----------------------------------------------------------------
// @Description   This is the main function.
//
```

```
  // USER CODE END

  Project_Init();

  // USER CODE BEGIN (Main,2)
  DP2 = 0xFFFF;
  P2 = 0xABCD;
  while(1)
    ;
  // USER CODE END
```

Figure 12. Configuration complete!