Configuring the Internal and External Oscillators

Relevant Devices

This application note applies to the following devices:

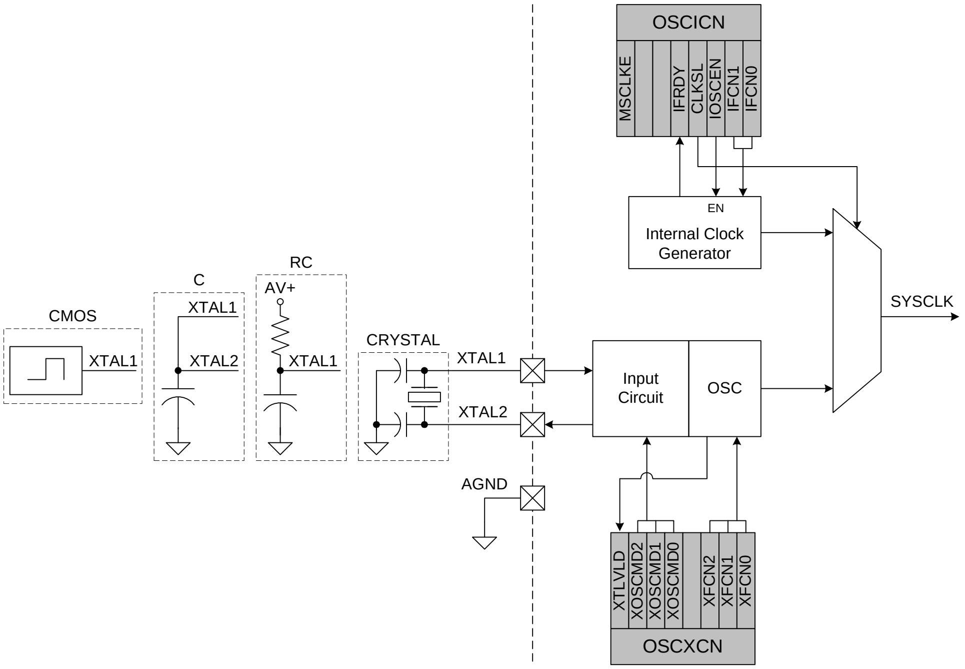
C8051F000, C8051F001, C8051F002, C8051F005, C8051F006, C8051F010, C8051F011, C8051F012, C8051F012, C8051F015, C8051F016, C8051F017, C8051F018, C8051F019, C8051F020, C8051F021, C8051F022, C8051F023, C8051F040, C8051F041, C8051F042, C8051F043, C8051F060, C8051F061, C8051F062, C8051F063, C8051F120, C8051F121, C8051F122, C8051F123, C8051F124, C8051F125, C8051F126, C8051F127, C8051F206, C8051F220, C8051F221, C8051F226, C8051F230 and C8051F236.

Introduction

The purpose of this application note is to describe how to configure and use the internal and external oscillators. Configuration descriptions, setup examples, and sample code are provided.

Key Points

- The internal oscillator in divide-by-8 mode is enabled and selected automatically when the device is reset.
- The system clock can be easily switched between the internal and external oscillators.
- It is legal to select the External Oscillator as the System Clock and disable the Internal Oscillator in the same write, if possible on the device. If running off the external oscillator, it is legal to enable the internal oscillator and select the internal oscillator as the system clock in the same write.



- It is legal to change the frequency of the internal oscillator while using the internal oscillator as the system clock.
- In all oscillator modes, /SYSCLK, a buffered version of the system clock, can be output on a port pin by enabling it in the Crossbar. On C8051F2xx devices,
  /SYSCLK may be enabled in the Port 1 MUX.
- If the Missing Clock Detector is enabled, a RESET will occur if the system clock drops below about 10 kHz.
- On some devices, the Crystal Oscillator Valid flag can be used to generate an interrupt when the crystal oscillator stabilizes, allowing the interrupt handler to switch to the external oscillator.

Clocking Options

All devices covered by this note have an internal and an external oscillator. The internal oscillator frequency is configurable from user software and has four settings that that can divide the oscillator frequency by 1, 2, 4, or 8. It addition to the four main settings, some devices have a calibrated ± 2 % internal oscillator with a fine-tunable frequency in steps of approximately 50 kHz. The external oscillator provides even more flexibility in frequency selection, power consumption, and accuracy using its 4 modes of operation.

The system clock can be freely switched between the internal and external oscillators. Furthermore, you can leave the external oscillator enabled while the internal oscillator is selected to avoid startup delays when the system clock is switched back to the external oscillator.

The operation of the internal and external oscillators is governed by SFR registers described in the ‘Oscillators’ chapter of the appropriate datasheet.

Internal Oscillator

At reset, the internal oscillator divided by 8 is selected as the system clock. The internal oscillator settings and typical frequencies are shown in Table 1 . The internal oscillator divide factor can be changed on the fly by changing the IFCN bits. The frequency change happens almost instantaneously

Table 1. Internal Oscillator Frequency Control Bits in the OSCICN Register

IFCN	Typical Frequency (16 MHz Oscillator)	Typical Frequency (24.5 MHz Calibrated Osciillator)
00b	2 MHz	3.0625 MHz
01b	4 MHz	6.125 MHz
10b	8 MHz	12.25 MHz
11b	16 MHz	24.5 MHz

The power consumption of the internal oscillator itself is independent of the selected frequency; however, the power consumption of the entire device is frequency dependent.

The accuracy of the internal oscillator is ± 20 % across process, power supply, and temperature variations on devices with an uncalibrated 16 MHz oscillator and ± 2 % on devices with a calibrated 24.5 MHz oscillator.

External Oscillator

The external oscillator is highly configurable, offering many choices to the system designer. The time base can be derived from an external CMOS-level clock source, an attached crystal or ceramic resonator, an attached RC combination, or an external capacitor.

External CMOS Clock Mode

The system clock can be supplied from an external CMOS-level clock source tied to the XTAL1 pin, such as a crystal oscillator module or the clock sig-

nal from another MCU.
Note: the XTAL1 pin is NOT 5V-tolerant.

External Crystal Mode

In general, a crystal is called for when an accurate time base is needed, for example when the absolute sampling rate of the ADC is critical or a standard UART baud rate must be generated. Note that on devices with a calibrated 24.5 MHz internal oscillator, a crystal is not needed for UART communication. Alternately, a low-frequency tuning-fork crystal, e.g. a 32.768 kHz watch crystal, can be used to operate the device in a low-power mode, then control can be switched to the high-frequency internal oscillator as required by the system.

The accuracy and stability of the crystal oscillator is governed almost exclusively by the attached crystal or ceramic resonator, provided that the loading capacitance parameters and oscillator drive level are set appropriately.

The crystal oscillator circuit is designed to be used with parallel-mode-specified crystals.

External RC

The external oscillator time base can also be derived from an external series RC combination connected as shown in Figure 1. When the capacitor voltage (Vc) is less than VDD/3, the capacitor charges through the resistor. Once the capacitor voltage reaches VDD/3, the comparator creates a path to AGND and discharges the capacitor.

This operation produces a saw-tooth type waveform at XTAL1, shown in Figure 2, whose period is dominated by the rise time of the voltage across the cap; the discharge time is less than 10 ns for a 100 pF capacitor. The output of the comparator is buffered and fed to a divide-by-two stage, the output of which becomes the system clock.

The accuracy of the time base in external RC mode is dominated by the tolerances of the R and C components.

Figure 1. RC Mode Overview

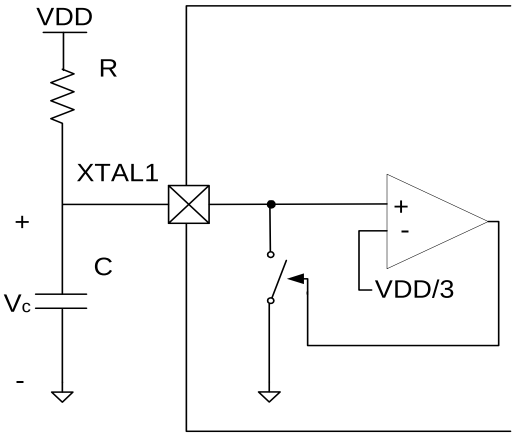
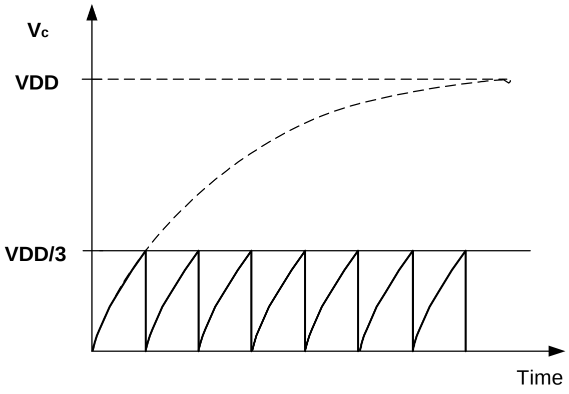


Figure 2. RC Mode Waveform Generation



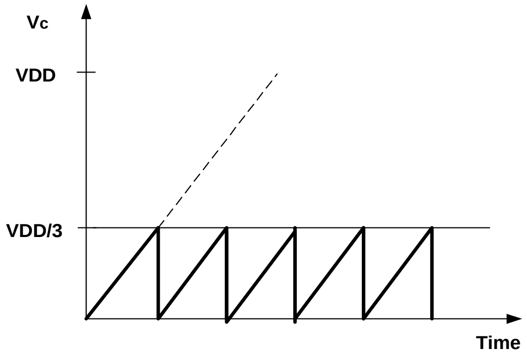
External C Mode

This mode is similar in operation to the external RC mode above, except the charging current for the capacitor is supplied by an internal programmable current source at XTAL2. This is the least accurate time base mode; however, it is the most flexible in that a single passive component can provide up to eight different operating frequencies, the highest frequency being almost a factor of 3000 greater than the lowest frequency.

The external oscillator in C mode generates a signal by constantly charging and discharging the capacitor connected to XTAL2. As Figure 3 shows, the capacitor charges linearly from a constant current source. When the voltage on the capacitor reaches VDD/3, the comparator creates a path to ground, discharging the capacitor. Once the capacitor is discharged, the comparator opens the switch and the cycle repeats. The resulting waveform is shown in Figure 4.

The accuracy of the time base in external C mode is dominated by the tolerance on the capacitor and the accuracy of the internal current source at XTAL2. The accuracy of the internal current source is on the order of ± 30 % across process, power supply, and temperature variations.

Figure 4. C Mode Waveform Generation



Internal Oscillator Configuration Example

At reset, the internal oscillator divided by 8 is selected as the system clock.

Hardware Connection: If the system design uses the internal oscillator exclusively **AND does not use the external oscillator,** the XTAL1 pin should be grounded externally as shown in Figure 5, or grounded internally by setting the XOSCMD bits (OSCXCN.6-4) to '000'. If the system calls for having the /RST line of the device held low for long periods of time, then grounding XTAL1 externally is recommended.

Figure 3. C Mode Overview

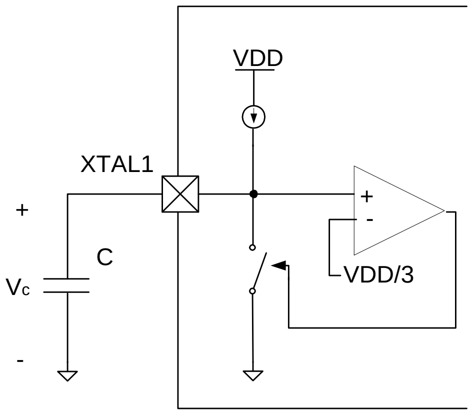
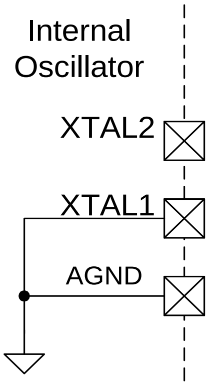


Figure 5. Optional Internal Oscillator Connection



The IFCN bits (OSCICN.1-0) program the internal oscillator frequency. Four divide settings are selectable as shown in Table 1 . If the device has a calibrated 24.5 MHz internal oscillator, the frequency may be fine-tuned by incrementing or decrementing the OSCICL register.

The startup and stabilization time for the internal oscillator is nearly instantaneous. Also, the internal oscillator's frequency can be changed arbitrarily at any time. Since the internal oscillator stabilizes at its newly programmed frequency before the next instruction is executed, IFRDY polling is not required.

External Oscillator Configuration Examples

The external oscillator supports four different configurations: CMOS clock, crystal , RC, and C modes. The external oscillator can be configured using the OSCXCN register. Once the external oscillator has been configured and has stabilized, the system clock can switch from the internal oscillator to the external oscillator using the CLKSL bit(s). Check the appropriate datasheet for more information and the location of the CLKSL bit(s).

In crystal-based designs, it can take up to several milliseconds to start the crystal oscillator, depending on the crystal frequency. Slower crystals tend to have a longer startup time than faster crystals. The XTLVLD (Crystal Oscillator Valid) flag can be used to determine when the external oscillator has stabilized.

In external RC and external C modes, the startup time of the external oscillator is instantaneous.

External CMOS Clock

The external oscillator clock can be supplied from an external CMOS-level source tied to the XTAL1 input. In this configuration, XTAL2 should be left floating, as shown in Figure 6. The SYSCLK frequency can be derived from the incoming signal as-is, or passed through a divide-by-two stage.

Note: Unlike some port I/O pins, the XTAL1 and XTAL2 pins are NOT 5-V tolerant. The voltage at these pins should be kept between AV+ and AGND.

External Crystal

The external oscillator timebase can be derived from a crystal or ceramic resonator connected across the XTAL1 and XTAL2 pins, as shown in Figure 7. The external oscillator can be configured to to use the crystal frequency as-is, or divided by two. Also, XFCN should be set based on the crystal frequency as discussed later in the text.

The XTLVLD (Crystal Oscillator Valid) flag can be used to determine when the external oscillator has

Figure 6. External CMOS Clock Connection

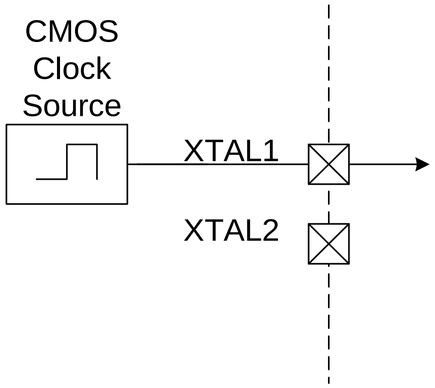
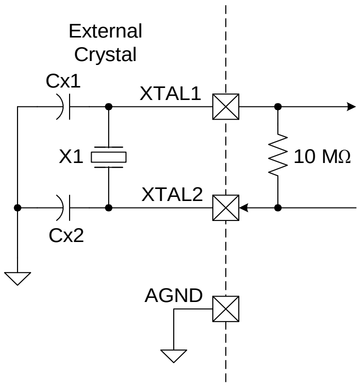


Figure 7. External



stabilized. The XTLVLD detection circuit requires a settling time to acheive proper bias. Introducing a 1 ms delay between enabling the oscillator and checking the XTLVLD flag will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before it has fully stabilized can cause unpredicable behavior.

Note: the loading capacitors (Cx1 and Cx2 in Figure 7) should be tied to the analog ground plane. **Also note that the feedback resistor for the crystal oscillator inverter is supplied on-chip, making an external resistor unnecessary.**

Crystal Startup Procedure

1. Configure the OSCXCN register to select the desired external oscillator mode.
2. Wait at least 1ms.
3. Poll for XTLVLD => ‘1’.
4. Switch the system clock to the external oscillator.

Determining XFCN

XFCN controls the drive level of the crystal oscillator driver. In essence, the drive level should be strong enough to excite the crystal to oscillation, but not so strong as to stress the crystal to cause it to degrade prematurely.

For 3 MHz crystals and above, degradation based on an XFCN setting that is too high is typically not an issue, and the maximum XFCN value can be used, though this will result in a higher operating current for the oscillator. For low-frequency tuning-fork crystals, 32.768 kHz and 100 kHz for example, overdriving the crystal does present a reliability concern. Additionally, if the drive level is too high, the tuning-fork crystals may not oscillate at all.

If the design uses a normal quartz crystal, Table 2 and Table 3 can be used to quickly determine XFCN based on the crystal frequency. .

Table 2. Crystal Mode XFCN Selection for ‘F00x, ‘F01x, ‘F02x and ‘F2xx Devices

XFCN	Crystal Frequency
000b	f < 12 kHz
001b	12 kHz < f ≤ 30 kHz
010b	30 kHz < f ≤ 95 kHz
011b	95 kHz < f ≤ 270 kHz
100b	270 kHz < f ≤ 720 kHz
101b	720 kHz < f ≤ 2.2 MHz
110b	2.2 MHz < f ≤ 6.7 MHz
111b	f > 6.7 MHz

Table 3. Crystal Mode XFCN Selection for ‘F04x, ‘F06x, and ‘F12x Devices

XFCN	Crystal Frequency
000b	f < 32 kHz
001b	32 kHz < f ≤ 84 kHz
010b	84 kHz < f ≤ 225 kHz
011b	225 kHz < f ≤ 590 kHz
100b	590 kHz < f ≤ 1.5 MHz
101b	1.5 kHz < f ≤ 4 MHz
110b	4 MHz < f ≤ 10 MHz
111b	10 MHz < f ≤ 30 MHz

External RC Network

Figure 5 shows the connection diagram for external RC mode. Note that the series RC connection is made between the analog power supply and analog ground. Equation 1 gives the external oscillator frequency in RC mode.

Equation 1. External Oscillator Frequency in RC Mode

$$\mathrm{Fosc} = \frac{1.23\times10^{3}}{R \times C}$$

Where:

Fosc = external oscillator frequency in MHz

C = capacitor value in pF

R = resistor value in kΩ

To configure the device for RC mode, the External Oscillator Mode bits in the OSCXCN register should be set to ‘10x’. This setting is valid for both RC and C modes. When using these modes, an internal divide-by-two stage is always enabled and is accounted for in Equation 1. Furthermore, the XFCN drive current selection should be set according to Table 4 .

Figure 8. External RC Connection

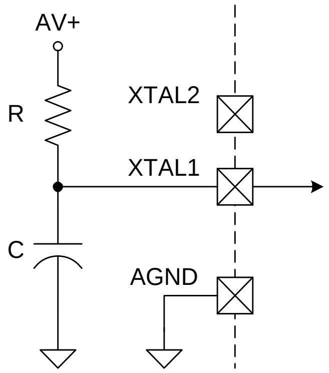


Table 4. RC Mode XFCN Selection

XFCN	Oscillation Frequency
000b	f < 25 kHz
001b	25 kHz < f ≤ 50 kHz
010b	50 kHz < f ≤ 100 kHz
011b	100 kHz < f ≤ 200 kHz
100b	200 kHz < f ≤ 400 kHz
101b	400 kHz < f ≤ 800 kHz
110b	800 kHz < f ≤ 1.6 MHz
111b	1.6 MHz < f ≤ 3.2 MHz

XFCN can always be set to a number higher than indicated in Table 3, but will result in a higher operating current for the oscillator. If XFCN is set too low, the oscillator frequency will be lower than predicted by Equation 1.

Note: the startup time for the external oscillator in RC mode is nearly instantaneous. The XTLVLD flag is undefined in this mode.

The supplied capacitor should be between 10 pF and 100 pF, keeping in mind that at lower capacitance values the parasitic capacitance will have a greater impact on the final frequency. We illustrate with a few examples:

RC Example: 100 kHZ

We start with C = 33 pF. The total capacitance including stray capacitance is 33 + 6, or 39 pF.

Assuming we want a frequency of oscillation of 100 kHz, we solve Equation 3 to find R:

$$\text{Fosc} = \frac{1.23 \times 10^3}{R \times C}$$

$$R = \frac{1.23 \times 10^3}{\text{Fosc} \times C}$$

$$R = \frac{1.23 \times 10^3}{0.1 \times 39}$$

$$R = 315\ \text{k}\Omega$$

From Table 3, we set XFCN to at least '010'.

RC Example: 3.2 MHz

It is possible to operate the external oscillator in RC mode at frequencies greater than 3.2 MHz, but this practice is not recommended. There will be a significant deviation of the oscillation frequency from the equation-predicted value as the frequency increases. Specifically, the actual frequency of oscillation will tend to be lower than the predicted value due to fixed internal delays, which become appreciable as the oscillation period decreases.

Using a 33 pF capacitor which results in a total capacitance of 39 pF, we solve Equation 1 to find R:

$$\text{Fosc} = \frac{1.23 \times 10^3}{R \times C}$$

$$R = \frac{1.23 \times 10^3}{\text{Fosc} \times C}$$

$$R = \frac{1.23 \times 10^3}{\mathbf{3.2} \times 39}$$

$$R = 9.86\ \text{k}\Omega$$

From Table 4 , we set XFCN to a value of '111'.

RC Example: 1 kHz

If A/D converter performance is not critical, the system clock frequency can be made arbitrarily slow, provided that the missing clock detector is disabled. If the missing clock detector is enabled, it will generate a system reset if the system clock falls below about 10 kHz.

In this example, we use a C value of 100 pF, resulting in a total capacitance of 106 pF. Solving Equation 3 for R:

$$\text{Fosc} = \frac{1.23 \times 10^3}{R \times C}$$

$$R = \frac{1.23 \times 10^3}{\text{Fosc} \times C}$$

$$R = \frac{1.23 \times 10^3}{\mathbf{0.001} \times \mathbf{106}}$$

$$R = 11.6\ \text{M}\Omega$$

From Table 3, XFCN can be set to '000'.

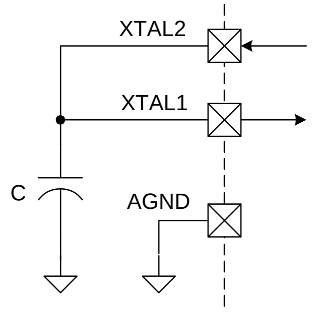
External Capacitor

Figure 6 shows the connections for external C mode. Note that the XTAL1 and XTAL2 pins are tied together, and that the supplied capacitor is tied between the XTAL pins and the analog ground plane. Equation 2 gives the external oscillator frequency in C mode.

Table 5 lists K Factor vs. XFCN setting.

External C mode is similar in operation to external RC mode, except that the charging current for the capacitor is supplied by a programmable current source at XTAL2. The supplied capacitor should be between 10 pF and 100 pF, keeping in mind that at

Figure 9. External Capacitor Connection



Equation 2. Oscillator Frequency in C Mode

$$\text{Fosc} = \frac{\text{KF}}{\text{C} \times \text{AV+}}$$

Where:

$\text{Fosc} =$ external oscillator frequency MHz

$\text{KF} =$ The K Factor, a unit-less quantity

$\text{C} =$ capacitor value in pF (includes stray capacitanc

$\text{AV+} =$ analog power supply voltage in Volts

Table 5. C Mode XFCN Selection

XFCN	KF for ‘F00x, ‘F01x, ‘F02x, and ‘F2xx Devices	KF for ‘F04x, ‘F06x, and ‘F12x Devices
000b	0.44	0.87
001b	1.4	2.6
010b	4.4	7.7
011b	13	22
100b	38	65
101b	100	180
110b	420	664
111b	1400	1590

lower capacitance values the parasitic capacitance will have a greater impact on the final frequency.

The actual frequency of oscillation is highly variable due to capacitor tolerance, parasitic capacitance, power supply voltage, and the variance of the internal current source, which is about ± 30 %.

Note: the startup time for the external oscillator in C mode is nearly instantaneous. The XTLVLD flag is undefined in this mode.

We illustrate external C mode with some examples:

C Example: 100 kHz

Assuming a 3.0V analog power supply and a 33 pF external capacitor, the total capacitance including parasitic capacitance is 39 pF. We solve Equation 2 for KF:

$$\text{KF} = \text{Fosc}\,(\text{C} \times \text{AV+})$$

$$\text{KF} = 0.1(39 \times 3.0)$$

$$\text{KF} = 11.7$$

Referencing Table 5 , the closest KF value is 13, for a device that has a 16 MHz internal oscillator. Solving Equation 2 with a KF of 13 yields an oscillation frequency of about 111 kHz. Increasing our capacitor value to 37 pF (43 pF total) and solving Equation 2 for frequency, we find that we can achieve 101 kHz. We set XFCN to '011' to select a KF value of 13.

Adjacent XFCN values result in K Factors which are about a factor of three apart. For example, increasing XFCN to '011' in the above example would increase the oscillation frequency from about 100 kHz to about 300 kHz.

Setting XFCN to its highest value '111', results in a KF of 1400. Using this setting with the above capacitance results in a frequency of oscillation of about 10 MHz as predicted by Equation 4. As in the external RC mode, the equations are accurate for frequencies that are 3 MHz and below, and become less accurate as the frequency increases.

C Example: 3 MHz

Using an external capacitor of 50 pF (56 pF total capacitance) and 3.0 V AV+, we solve Equation 4 for KF:

$$KF = Fosc\ (C \times AV+)$$

$$KF = 3\ (56 \times 3.0)$$

$$KF = 504$$

The closest KF in Table 4 for devices with a 16 MHz internal oscillator is 420. This results in a frequency of oscillation around 2.5 MHz so we set XFCN to '110'.

Software Configuration Examples

The following examples show how to configure the internal and external oscillators on each device family.

C8051F00x and C8051F01x Example

```
//-----------------------------------------------------------------------------
// F0xx_Osc_Init.c
//-----------------------------------------------------------------------------
// Copyright 2003 Cygnal Integrated Products, Inc.
//
// AUTH: FB
// DATE: 27 DEC 02
//
// This program shows an example of configuring the internal
// and external oscillators.
//
// Target: C8051F00x and C8051F01x
// Tool chain: KEIL C51 6.03 / KEIL EVAL C51
//

//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------

#include <c8051f000.h>                 // SFR declarations

//-----------------------------------------------------------------------------
// 16-bit SFR Definitions for ‘F00x
//-----------------------------------------------------------------------------

sfr16 DP       = 0x82;                 // data pointer
sfr16 TMR3RL   = 0x92;                 // Timer3 reload value
sfr16 TMR3     = 0x94;                 // Timer3 counter
sfr16 ADC0     = 0xbe;                 // ADC0 data
sfr16 ADC0GT   = 0xc4;                 // ADC0 greater than window
sfr16 ADC0LT   = 0xc6;                 // ADC0 less than window
sfr16 RCAP2    = 0xca;                 // Timer2 capture/reload
sfr16 T2       = 0xcc;                 // Timer2
sfr16 DAC0     = 0xd2;                 // DAC0 data
sfr16 DAC1     = 0xd5;                 // DAC1 data

//-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------

void SYSCLK_IntOsc_Init (void);
void SYSCLK_Crystal_Init (void);
void SYSCLK_C_RC_Init (void);
void SYSCLK_CMOS_Init (void);

//-----------------------------------------------------------------------------
// MAIN Routine
//-----------------------------------------------------------------------------

void main (void) {
   WDTCN = 0xde;                       // disable watchdog timer
   WDTCN = 0xad;

   // select one of the following functions to initialze the system clock
   SYSCLK_IntOsc_Init ();
// SYSCLK_Crystal_Init ();
// SYSCLK_C_RC_Init ();
// SYSCLK_CMOS_Init ();

   while (1);
}

//------------------------------------------------------------------------------
// Initialization Subroutines
//------------------------------------------------------------------------------
//------------------------------------------------------------------------------
// SYSCLK_IntOsc_Init
//------------------------------------------------------------------------------
//
// This routine initializes the system clock to use the internal oscillator
// at its maximum frequency.
//
void SYSCLK_IntOsc_Init (void)
{
   OSCICN = 0x87;                      // Set internal oscillator to
                                       // maximum frequency and enable missing
                                       // clock detector
}

//------------------------------------------------------------------------------
// SYSCLK_Crystal_Init
//------------------------------------------------------------------------------
//
// This routine initializes the system clock to use a 22.1184MHz crystal
// as its clock source. Assumes a 22.1184 MHz crystal and associated loading
// capacitors are connected at XTAL1 and XTAL2.
//
void SYSCLK_Crystal_Init (void)
{
   int i;                              // delay counter

   OSCXCN = 0x67;                      // start external oscillator with
                                       // 22.1184MHz crystal

   for (i=0; i < 256; i++) ;           // XTLVLD blanking interval (>1ms)

   while (!(OSCXCN & 0x80)) ;          // Wait for crystal osc. to settle

   OSCICN = 0x88;                      // select external oscillator as SYSCLK
                                       // source and enable missing clock
                                       // detector
}

//------------------------------------------------------------------------------
// SYSCLK_C_RC_Init
//------------------------------------------------------------------------------
//
// This routine initializes the system clock to use an external RC network
// or a single capacitor as its clock source. Assumes an RC network is
// connected to XTAL1 or a single capacitor is connected to XTAL1 and
// XTAL2.
//
void SYSCLK_C_RC_Init (void)
{
   OSCXCN = 0x47;                      // start external oscillator in
                                       // C/RC mode with XFCN = 7

   OSCICN = 0x88;                      // select external oscillator as SYSCLK
                                       // source and enable missing clock
                                       // detector
}

//-----------------------------------------------------------------------------
// SYSCLK_CMOS_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to the external oscillator in
// CMOS clock mode. Assumes a CMOS clock generator is connected to XTAL1.
//
void SYSCLK_CMOS_Init (void)
{
   OSCXCN = 0x20;                      // start external oscillator in
                                       // CMOS clock mode.

   OSCICN = 0x88;                      // select external oscillator as SYSCLK
                                       // source and enable missing clock
                                       // detector
}
```

C8051F02x Example

```
//-----------------------------------------------------------------------------
// F02x_Osc_Init.c
//-----------------------------------------------------------------------------
// Copyright 2003 Cygnal Integrated Products, Inc.
//
// AUTH: FB
// DATE: 27 DEC 02
//
// This program shows an example of configuring the internal
// and external oscillators.
//
// Target: C8051F02x
// Tool chain: KEIL C51 6.03 / KEIL EVAL C51
//

//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------

#include <c8051f020.h>                 // SFR declarations

//-----------------------------------------------------------------------------
// 16-bit SFR Definitions for ‘F02x
//-----------------------------------------------------------------------------

sfr16 DP       = 0x82;                 // data pointer
sfr16 TMR3RL   = 0x92;                 // Timer3 reload value
sfr16 TMR3     = 0x94;                 // Timer3 counter
sfr16 ADC0     = 0xbe;                 // ADC0 data
sfr16 ADC0GT   = 0xc4;                 // ADC0 greater than window
sfr16 ADC0LT   = 0xc6;                 // ADC0 less than window
sfr16 RCAP2    = 0xca;                 // Timer2 capture/reload
sfr16 T2       = 0xcc;                 // Timer2
sfr16 RCAP4    = 0xe4;                 // Timer4 capture/reload
sfr16 T4       = 0xf4;                 // Timer4
sfr16 DAC0     = 0xd2;                 // DAC0 data
sfr16 DAC1     = 0xd5;                 // DAC1 data

//-----------------------------------------------------------------------------
// Global CONSTANTS
//-----------------------------------------------------------------------------

sbit LED = P1^6;                       // LED=’1’ means ON
sbit SW1 = P3^7;                       // SW1=’0’ means switch pressed

//-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------

void SYSCLK_IntOsc_Init (void);
void SYSCLK_Crystal_Init (void);
void SYSCLK_C_RC_Init (void);
void SYSCLK_CMOS_Init (void);

//-----------------------------------------------------------------------------
// MAIN Routine
//-----------------------------------------------------------------------------
```

```
void main (void) {

   WDTCN = 0xde;                        // disable watchdog timer
   WDTCN = 0xad;

   // select one of the following functions to initialze the system clock
   SYSCLK_IntOsc_Init ();
// SYSCLK_Crystal_Init ();
// SYSCLK_C_RC_Init ();
// SYSCLK_CMOS_Init ();

   while (1);
}

//-----------------------------------------------------------------------------
// Initialization Subroutines
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
// SYSCLK_IntOsc_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use the internal oscillator
// at its maximum frequency.
//
void SYSCLK_IntOsc_Init (void)
{
   OSCICN = 0x87;                       // Set internal oscillator to
                                        // maximum frequency and enable missing
                                        // clock detector
}

//-----------------------------------------------------------------------------
// SYSCLK_Crystal_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use a 22.1184MHz crystal
// as its clock source. Assumes a 22.1184 MHz crystal and associated loading
// capacitors are connected at XTAL1 and XTAL2.
//
void SYSCLK_Crystal_Init (void)
{
   int i;                               // delay counter

   OSCXCN = 0x67;                       // start external oscillator with
                                        // 22.1184MHz crystal

   for (i=0; i < 256; i++) ;            // XTLVLD blanking interval (>1ms)

   while (!(OSCXCN & 0x80)) ;           // Wait for crystal osc. to settle

   OSCICN = 0x88;                       // select external oscillator as SYSCLK
                                        // source and enable missing clock
                                        // detector
}

//-----------------------------------------------------------------------------
// SYSCLK_C_RC_Init
//-----------------------------------------------------------------------------
```

```
//
// This routine initializes the system clock to use an external RC network
// or a single capacitor as its clock source. Assumes an RC network is
// connected to XTAL1 or a single capacitor is connected to XTAL1 and
// XTAL2.
//
void SYSCLK_C_RC_Init (void)
{
   OSCXCN = 0x47;                       // start external oscillator in
                                        // C/RC mode with XFCN = 7

   OSCICN = 0x88;                       // select external oscillator as SYSCLK
                                        // source and enable missing clock
                                        // detector
}

//-----------------------------------------------------------------------------
// SYSCLK_CMOS_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to the external oscillator in
// CMOS clock mode. Assumes a CMOS clock generator is connected to XTAL1.
//
void SYSCLK_CMOS_Init (void)
{
   OSCXCN = 0x20;                       // start external oscillator in
                                        // CMOS clock mode.

   OSCICN = 0x88;                       // select external oscillator as SYSCLK
                                        // source and enable missing clock
                                        // detector
}
```

C8051F04x Example

```
//-----------------------------------------------------------------------------
// F04x_Osc_Init.c
//-----------------------------------------------------------------------------
// Copyright 2003 Cygnal Integrated Products, Inc.
//
// AUTH: FB
// DATE: 27 DEC 02
//
// This program shows an example of configuring the internal
// and external oscillators.
//
// Target: C8051F04x
// Tool chain: KEIL C51 6.03 / KEIL EVAL C51
//

//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------

#include <c8051f040.h>                  // SFR declarations

//-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------

void SYSCLK_IntOsc_Init (void);
void SYSCLK_Crystal_Init (void);
void SYSCLK_C_RC_Init (void);
void SYSCLK_CMOS_Init (void);

//-----------------------------------------------------------------------------
// MAIN Routine
//-----------------------------------------------------------------------------

void main (void) {

   WDTCN = 0xde;                        // disable watchdog timer
   WDTCN = 0xad;

   // select one of the following functions to initialze the system clock
   SYSCLK_IntOsc_Init ();
// SYSCLK_Crystal_Init ();
// SYSCLK_C_RC_Init ();
// SYSCLK_CMOS_Init ();

   while (1);
}

//-----------------------------------------------------------------------------
// Initialization Subroutines
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
// SYSCLK_IntOsc_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use the internal oscillator
// at its maximum frequency.
```

```
//
void SYSCLK_IntOsc_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCICN = 0x83;                      // Set internal oscillator to
                                       // maximum frequency

   CLKSEL = 0x00;                      // Select internal oscillator as
                                       // SYSCLK source

   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_Crystal_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use an 22.1184MHz crystal
// as its clock source. Assumes a 22.1184 MHz crystal and associated loading
// capacitors are connected at XTAL1 and XTAL2.
//
void SYSCLK_Crystal_Init (void)
{
   int i;                              // delay counter
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCXCN = 0x67;                      // start external oscillator with
                                       // 22.1184MHz crystal (XFCN = 7)

   for (i=0; i < 256; i++) ;           // XTLVLD blanking interval (>1ms)

   while (!(OSCXCN & 0x80)) ;          // Wait for crystal osc. to settle


   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                      // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                      // select external oscillator as SYSCLK
                                       // source

   OSCICN = 0x00;                      // disable internal oscillator


   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_C_RC_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use an external RC network
// or a single capacitor as its clock source. Assumes an RC network is
// connected to XTAL1 or a single capacitor is connected to XTAL1 and
// XTAL2.
//
```

```
void SYSCLK_C_RC_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCXCN = 0x47;                      // start external oscillator in
                                       // C/RC mode with XFCN = 7
   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                      // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                      // select external oscillator as SYSCLK
                                       // source

   OSCICN = 0x00;                      // disable internal oscillator

   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_CMOS_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to the external oscillator in
// CMOS clock mode. Assumes a CMOS clock generator is connected to XTAL1.
//
void SYSCLK_CMOS_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCXCN = 0x20;                      // start external oscillator in
                                       // CMOS clock mode.

   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                      // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                      // select external oscillator as SYSCLK
                                       // source

   OSCICN = 0x00;                      // disable internal oscillator

   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}
```

C8051F06x Example

```
//-----------------------------------------------------------------------------
// F06x_Osc_Init.c
//-----------------------------------------------------------------------------
// Copyright 2003 Cygnal Integrated Products, Inc.
//
// AUTH: FB
// DATE: 27 DEC 02
//
// This program shows an example of configuring the internal
// and external oscillators.
//
// Target: C8051F06x
// Tool chain: KEIL C51 6.03 / KEIL EVAL C51
//

//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------

#include <c8051f060.h>                  // SFR declarations

//-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------

void SYSCLK_IntOsc_Init (void);
void SYSCLK_Crystal_Init (void);
void SYSCLK_C_RC_Init (void);
void SYSCLK_CMOS_Init (void);

//-----------------------------------------------------------------------------
// MAIN Routine
//-----------------------------------------------------------------------------

void main (void) {

   WDTCN = 0xde;                        // disable watchdog timer
   WDTCN = 0xad;

   // select one of the following functions to initialze the system clock
   SYSCLK_IntOsc_Init ();
// SYSCLK_Crystal_Init ();
// SYSCLK_C_RC_Init ();
// SYSCLK_CMOS_Init ();

   while (1);
}

//-----------------------------------------------------------------------------
// Initialization Subroutines
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
// SYSCLK_IntOsc_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use the internal oscillator
// at its maximum frequency.
```

```
//
void SYSCLK_IntOsc_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;         // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;               // Set SFR Page

   OSCICN = 0x83;                       // Set internal oscillator to
                                        // maximum frequency

   CLKSEL = 0x00;                       // Select internal oscillator as
                                        // SYSCLK source

   SFRPAGE = SFRPAGE_SAVE;              // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_Crystal_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use an 22.1184MHz crystal
// as its clock source. Assumes a 22.1184 MHz crystal and associated loading
// capacitors are connected at XTAL1 and XTAL2.
//
void SYSCLK_Crystal_Init (void)
{
   int i;                               // delay counter
   char SFRPAGE_SAVE = SFRPAGE;         // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;               // Set SFR Page

   OSCXCN = 0x67;                       // start external oscillator with
                                        // 22.1184MHz crystal (XFCN = 7)

   for (i=0; i < 256; i++) ;            // XTLVLD blanking interval (>1ms)

   while (!(OSCXCN & 0x80)) ;           // Wait for crystal osc. to settle


   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                       // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                       // select external oscillator as SYSCLK
                                        // source

   OSCICN = 0x00;                       // disable internal oscillator


   SFRPAGE = SFRPAGE_SAVE;              // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_C_RC_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use an external RC network
// or a single capacitor as its clock source. Assumes an RC network is
// connected to XTAL1 or a single capacitor is connected to XTAL1 and
// XTAL2.
//
```

```
void SYSCLK_C_RC_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCXCN = 0x47;                      // start external oscillator in
                                       // C/RC mode with XFCN = 7
   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                      // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                      // select external oscillator as SYSCLK
                                       // source

   OSCICN = 0x00;                      // disable internal oscillator

   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_CMOS_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to the external oscillator in
// CMOS clock mode. Assumes a CMOS clock generator is connected to XTAL1.
//
void SYSCLK_CMOS_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCXCN = 0x20;                      // start external oscillator in
                                       // CMOS clock mode.

   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                      // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                      // select external oscillator as SYSCLK
                                       // source

   OSCICN = 0x00;                      // disable internal oscillator

   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}
```

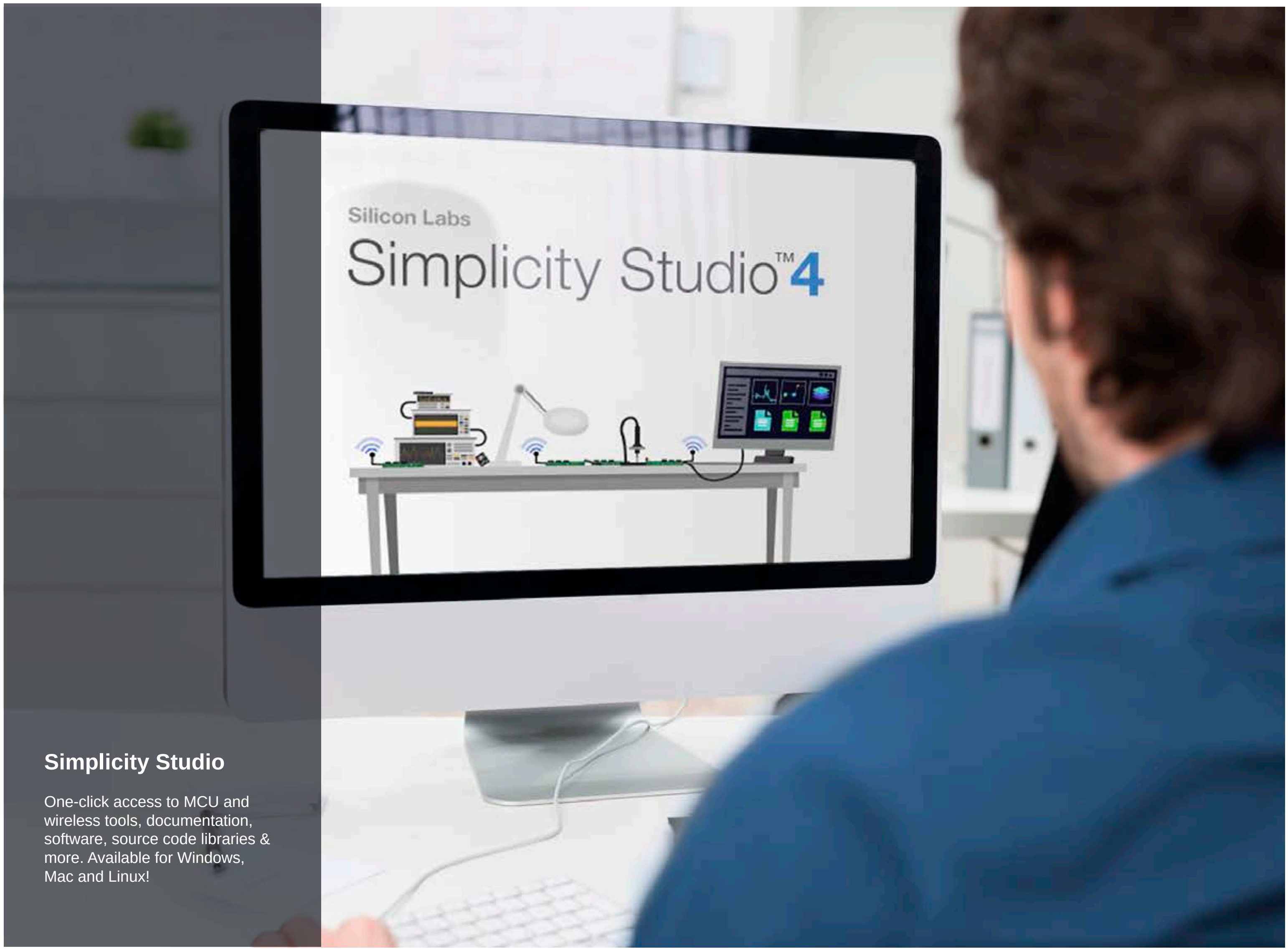
C8051F12x Example

```
//-----------------------------------------------------------------------------
// F12x_Osc_Init.c
//-----------------------------------------------------------------------------
// Copyright 2003 Cygnal Integrated Products, Inc.
//
// AUTH: FB
// DATE: 27 DEC 02
//
// This program shows an example of configuring the internal
// and external oscillators.
//
// Target: C8051F12x
// Tool chain: KEIL C51 6.03 / KEIL EVAL C51
//

//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------

#include <c8051f120.h>                 // SFR declarations

//-----------------------------------------------------------------------------
// 16-bit SFR Definitions for ‘F12x
//-----------------------------------------------------------------------------

sfr16 DP       = 0x82;                 // data pointer
sfr16 ADC0     = 0xbe;                 // ADC0 data
sfr16 ADC0GT   = 0xc4;                 // ADC0 greater than window
sfr16 ADC0LT   = 0xc6;                 // ADC0 less than window
sfr16 RCAP2    = 0xca;                 // Timer2 capture/reload
sfr16 RCAP3    = 0xca;                 // Timer3 capture/reload
sfr16 RCAP4    = 0xca;                 // Timer4 capture/reload
sfr16 TMR2     = 0xcc;                 // Timer2
sfr16 TMR3     = 0xcc;                 // Timer3
sfr16 TMR4     = 0xcc;                 // Timer4
sfr16 DAC0     = 0xd2;                 // DAC0 data
sfr16 DAC1     = 0xd2;                 // DAC1 data
sfr16 PCA0CP5  = 0xe1;                 // PCA0 Module 5 capture
sfr16 PCA0CP2  = 0xe9;                 // PCA0 Module 2 capture
sfr16 PCA0CP3  = 0xeb;                 // PCA0 Module 3 capture
sfr16 PCA0CP4  = 0xed;                 // PCA0 Module 4 capture
sfr16 PCA0     = 0xf9;                 // PCA0 counter
sfr16 PCA0CP0  = 0xfb;                 // PCA0 Module 0 capture
sfr16 PCA0CP1  = 0xfd;                 // PCA0 Module 1 capture

//-----------------------------------------------------------------------------
// Global CONSTANTS
//-----------------------------------------------------------------------------

sbit LED = P1^6;                       // LED=’1’ means ON
sbit SW1 = P3^7;                       // SW1=’0’ means switch pressed

//-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------

void SYSCLK_IntOsc_Init (void);
void SYSCLK_Crystal_Init (void);
void SYSCLK_C_RC_Init (void);
void SYSCLK_CMOS_Init (void);

//-----------------------------------------------------------------------------
// MAIN Routine
//-----------------------------------------------------------------------------

void main (void) {

   WDTCN = 0xde;                       // disable watchdog timer
   WDTCN = 0xad;

   // select one of the following functions to initialze the system clock
   SYSCLK_IntOsc_Init ();
// SYSCLK_Crystal_Init ();
// SYSCLK_C_RC_Init ();
// SYSCLK_CMOS_Init ();

   while (1);
}

//-----------------------------------------------------------------------------
// Initialization Subroutines
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
// SYSCLK_IntOsc_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use the internal oscillator
// at its maximum frequency.
//
void SYSCLK_IntOsc_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCICN = 0x83;                      // Set internal oscillator to
                                       // maximum frequency

   CLKSEL = 0x00;                      // Select internal oscillator as
                                       // SYSCLK source

   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_Crystal_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use an 22.1184 MHz crystal
// as its clock source. Assumes a 22.1184 MHz crystal and associated loading
// capacitors are connected at XTAL1 and XTAL2.
//
void SYSCLK_Crystal_Init (void)
{
   int i;                              // delay counter
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page
   OSCXCN = 0x67;                      // start external oscillator with
                                       // 22.1184MHz crystal (XFCN = 7)

   for (i=0; i < 256; i++) ;           // XTLVLD blanking interval (>1ms)

   while (!(OSCXCN & 0x80)) ;          // Wait for crystal osc. to settle


   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                      // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                      // select external oscillator as SYSCLK
                                       // source

   OSCICN = 0x00;                      // disable internal oscillator


   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_C_RC_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use an external RC network
// or a single capacitor as its clock source. Assumes an RC network is
// connected to XTAL1 or a single capacitor is connected to XTAL1 and
// XTAL2.
//
void SYSCLK_C_RC_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCXCN = 0x47;                      // start external oscillator in
                                       // C/RC mode with XFCN = 7
   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                      // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                      // select external oscillator as SYSCLK
                                       // source

   OSCICN = 0x00;                      // disable internal oscillator

   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}

//-----------------------------------------------------------------------------
// SYSCLK_CMOS_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to the external oscillator in
// CMOS clock mode. Assumes a CMOS clock generator is connected to XTAL1.
//
void SYSCLK_CMOS_Init (void)
{
   char SFRPAGE_SAVE = SFRPAGE;        // Save Current SFR page
   SFRPAGE = CONFIG_PAGE;              // Set SFR Page

   OSCXCN = 0x20;                      // start external oscillator in
                                       // CMOS clock mode.

   SFRPAGE = LEGACY_PAGE;
   RSTSRC = 0x04;                      // enable missing clock detector

   SFRPAGE = CONFIG_PAGE;
   CLKSEL = 0x01;                      // select external oscillator as SYSCLK
                                       // source

   OSCICN = 0x00;                      // disable internal oscillator

   SFRPAGE = SFRPAGE_SAVE;             // Restore SFR page
}
```

C8051F2xx Example

```
//-----------------------------------------------------------------------------
// F2xx_Osc_Init.c
//-----------------------------------------------------------------------------
// Copyright 2003 Cygnal Integrated Products, Inc.
//
// AUTH: FB
// DATE: 27 DEC 02
//
// This program shows an example of configuring the internal
// and external oscillators.
//
// Target: C8051F2xx
// Tool chain: KEIL C51 6.03 / KEIL EVAL C51
//

//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------

#include <c8051f200.h>                 // SFR declarations

//-----------------------------------------------------------------------------
// 16-bit SFR Definitions for ‘F2xx
//-----------------------------------------------------------------------------

sfr16 DP       = 0x82;                 // data pointer
sfr16 ADC0     = 0xbe;                 // ADC0 data
sfr16 ADC0GT   = 0xc4;                 // ADC0 greater than window
sfr16 ADC0LT   = 0xc6;                 // ADC0 less than window
sfr16 RCAP2    = 0xca;                 // Timer2 capture/reload
sfr16 T2       = 0xcc;                 // Timer2

//-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------

void SYSCLK_IntOsc_Init (void);
void SYSCLK_Crystal_Init (void);
void SYSCLK_C_RC_Init (void);
void SYSCLK_CMOS_Init (void);

//-----------------------------------------------------------------------------
// MAIN Routine
//-----------------------------------------------------------------------------

void main (void) {

   WDTCN = 0xde;                       // disable watchdog timer
   WDTCN = 0xad;

   // select one of the following functions to initialze the system clock
   SYSCLK_IntOsc_Init ();
// SYSCLK_Crystal_Init ();
// SYSCLK_C_RC_Init ();
// SYSCLK_CMOS_Init ();

   while (1);
```

```
}

//-----------------------------------------------------------------------------
// Initialization Subroutines
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
// SYSCLK_IntOsc_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use the internal oscillator
// at its maximum frequency.
//
void SYSCLK_IntOsc_Init (void)
{
   OSCICN = 0x87;                      // Set internal oscillator to
                                       // maximum frequency and enable missing
                                       // clock detector
}

//-----------------------------------------------------------------------------
// SYSCLK_Crystal_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use an 22.1184 MHz crystal
// as its clock source. Assumes a 22.1184 MHz crystal and associated loading
// capacitors are connected at XTAL1 and XTAL2.
//
void SYSCLK_Crystal_Init (void)
{
   int i;                              // delay counter

   OSCXCN = 0x67;                      // start external oscillator with
                                       // 22.1184MHz crystal

   for (i=0; i < 256; i++) ;           // XTLVLD blanking interval (>1ms)

   while (!(OSCXCN & 0x80)) ;          // Wait for crystal osc. to settle

   OSCICN = 0x88;                      // select external oscillator as SYSCLK
                                       // source and enable missing clock
                                       // detector
}

//-----------------------------------------------------------------------------
// SYSCLK_C_RC_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to use an external RC network
// or a single capacitor as its clock source. Assumes an RC network is
// connected to XTAL1 or a single capacitor is connected to XTAL1 and
// XTAL2.
//
void SYSCLK_C_RC_Init (void)
{
   OSCXCN = 0x47;                      // start external oscillator in
                                       // C/RC mode with XFCN = 7

   OSCICN = 0x88;                      // select external oscillator as SYSCLK
                                       // source and enable missing clock
                                       // detector
}

//-----------------------------------------------------------------------------
// SYSCLK_CMOS_Init
//-----------------------------------------------------------------------------
//
// This routine initializes the system clock to the external oscillator in
// CMOS clock mode. Assumes a CMOS clock generator is connected to XTAL1.
//
void SYSCLK_CMOS_Init (void)
{
   OSCXCN = 0x20;                      // start external oscillator in
                                       // CMOS clock mode.

   OSCICN = 0x88;                      // select external oscillator as SYSCLK
                                       // source and enable missing clock
                                       // detector
}
```



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer
Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information
Silicon Laboratories Inc.® , Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com