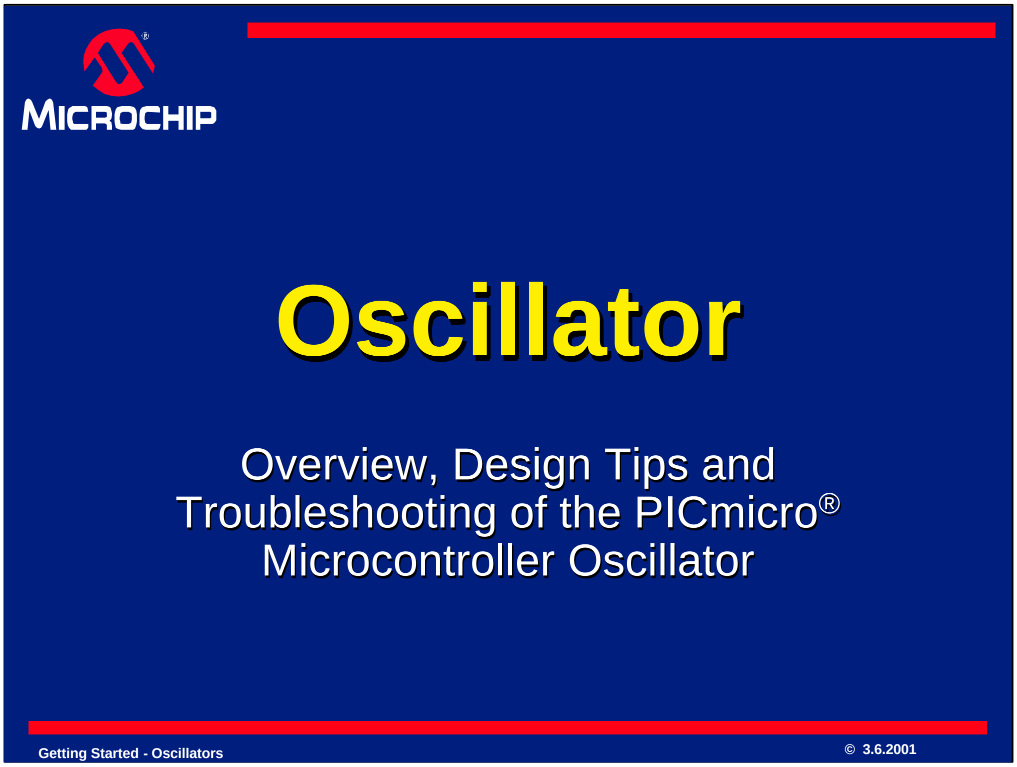


Oscillator

Overview, Design Tips and Troubleshooting of the PICmicro® Microcontroller Oscillator

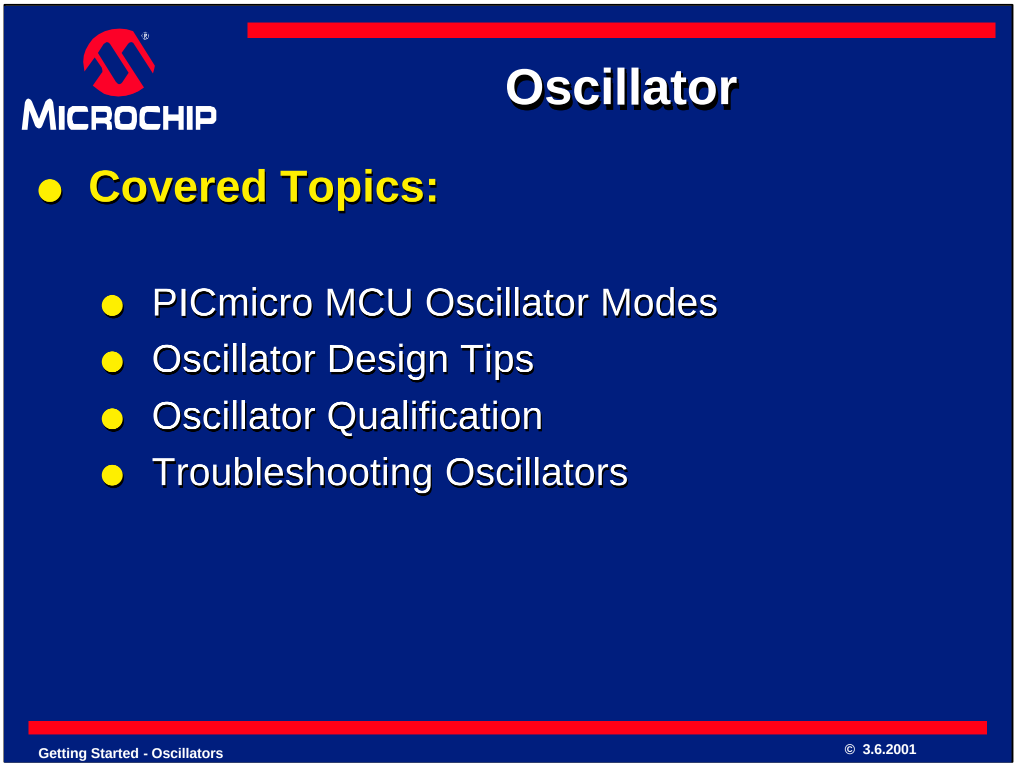
Getting Started - Oscillators

© 3.6.2001

In this presentation, we will explore the different oscillators and modes available on PICmicro® microcontrollers (MCUs).

While oscillator design and usage is simple in concept, the selection of a wrong oscillator for an application or oscillators that are not qualified properly can lead to difficulties.

This presentation explores these issues and other so you will have a greater understanding of oscillator design in general.



Oscillator

- Covered Topics:
  - PICmicro MCU Oscillator Modes
  - Oscillator Design Tips
  - Oscillator Qualification
  - Troubleshooting Oscillators

Getting Started - Oscillators

© 3.6.2001

Topics that will be covered include:

- PICmicro MCU Oscillator Modes -

Microchip Technology’s PIC® microcontroller is capable of supporting many different oscillator types. The types supported depend on the device, and some are more common than others. We will look at all modes that exist at the time this presentation was created (December 2000)

- Oscillator Design Tips -

Next we will explore each of the different oscillators and how best to use them. Tips will be broken down to suit each type of oscillator. The pros and cons of each will also be described.

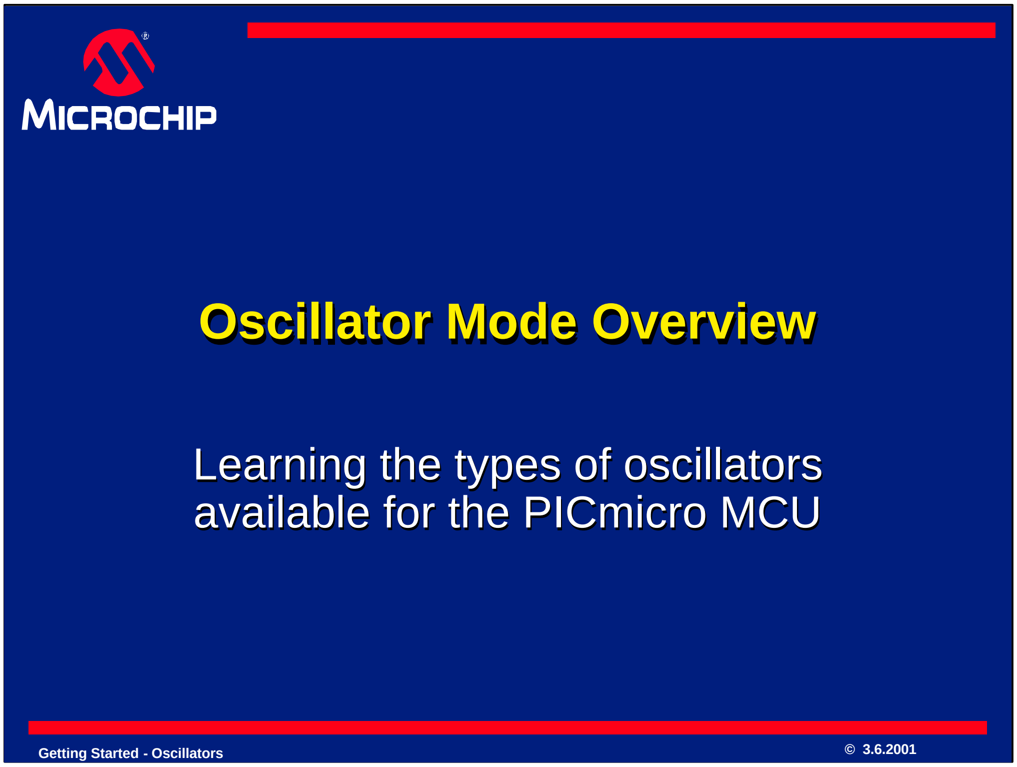
- Oscillator Qualification -

Then we will consider the conditions that are difficult on oscillators and how best to deal with them. It happens that creating these conditions in the lab is an excellent way to help qualify the oscillator component of your design.

- Troubleshooting Oscillators -

Finally, we will explore some of the more common mistakes in oscillator design and testing. At the end of this section you will find a listing for more resources on oscillators.

And now...



MICROCHIP

Oscillator Mode Overview

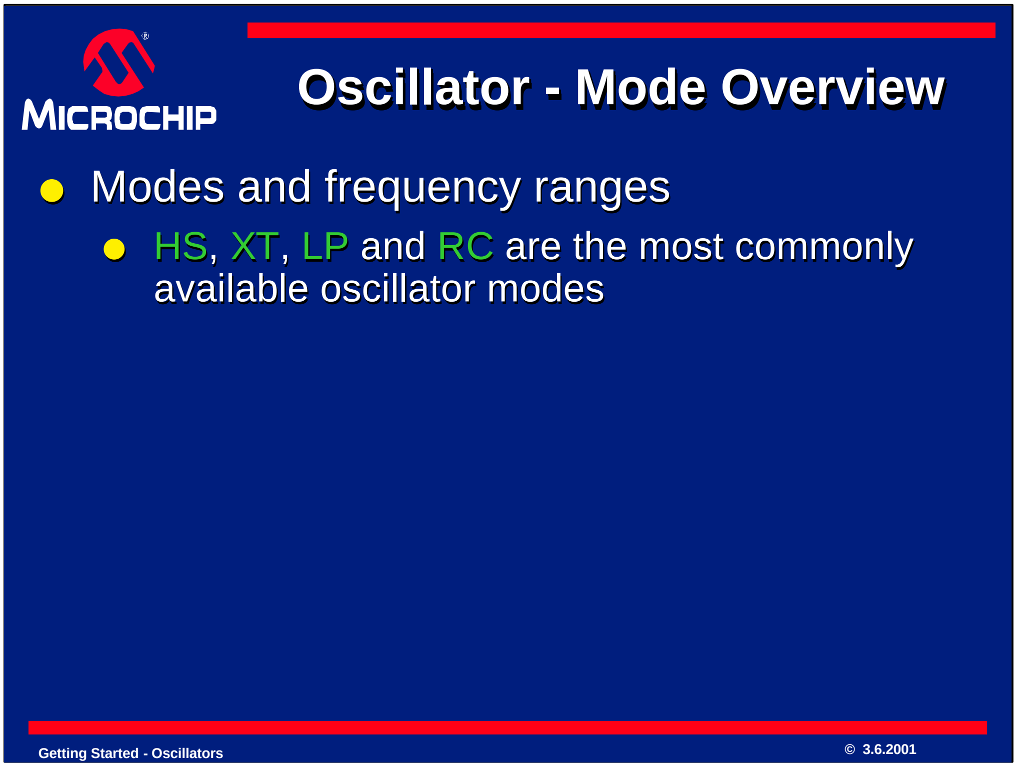
Learning the types of oscillators
available for the PICmicro MCU

Getting Started - Oscillators

© 3.6.2001

... an overview of PICmicro Oscillator Modes.

First, we will look at the most commonly available oscillator modes.



Oscillator - Mode Overview

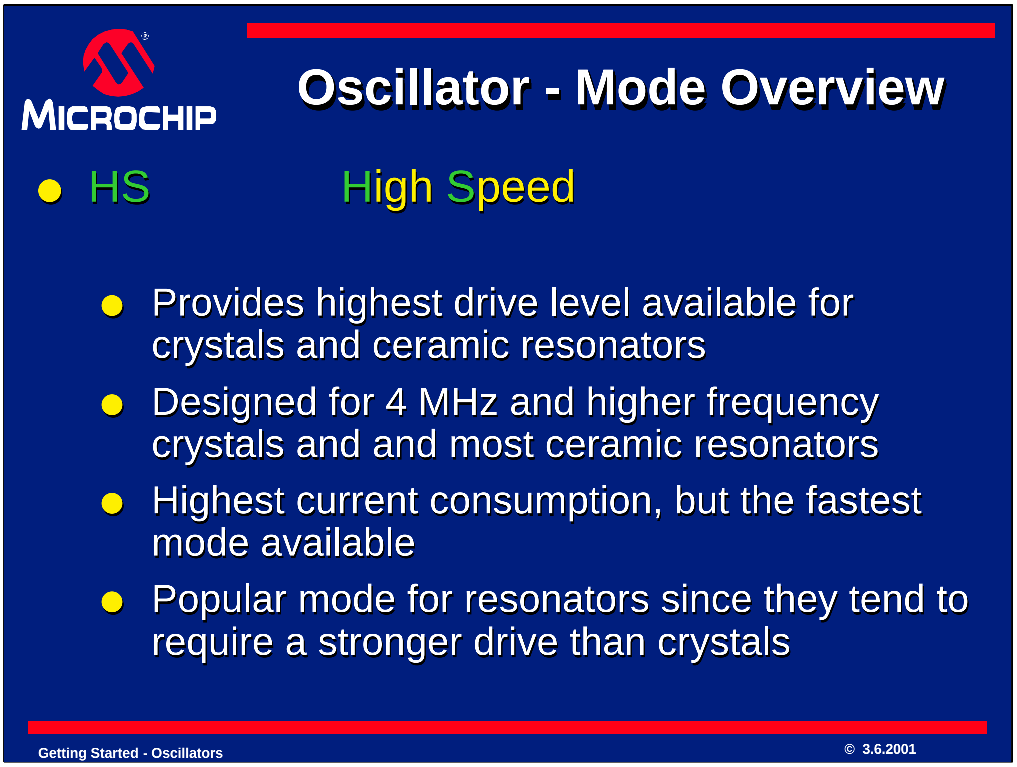
- Modes and frequency ranges
  - HS, XT, LP and RC are the most commonly available oscillator modes

Getting Started - Oscillators

© 3.6.2001

There are 4 common oscillator modes that are available on most PICmicro devices. HS, XT, LP and RC.

These modes support crystals, canned oscillator modules, some resonators or the use of an external resistor and capacitor as a clock source. When using a crystal or resonator, other components such as capacitors may be needed. We will look at this later.



Oscillator - Mode Overview

- HS High Speed
  - Provides highest drive level available for crystals and ceramic resonators
  - Designed for 4 MHz and higher frequency crystals and and most ceramic resonators
  - Highest current consumption, but the fastest mode available
  - Popular mode for resonators since they tend to require a stronger drive than crystals

Getting Started - Oscillators © 3.6.2001

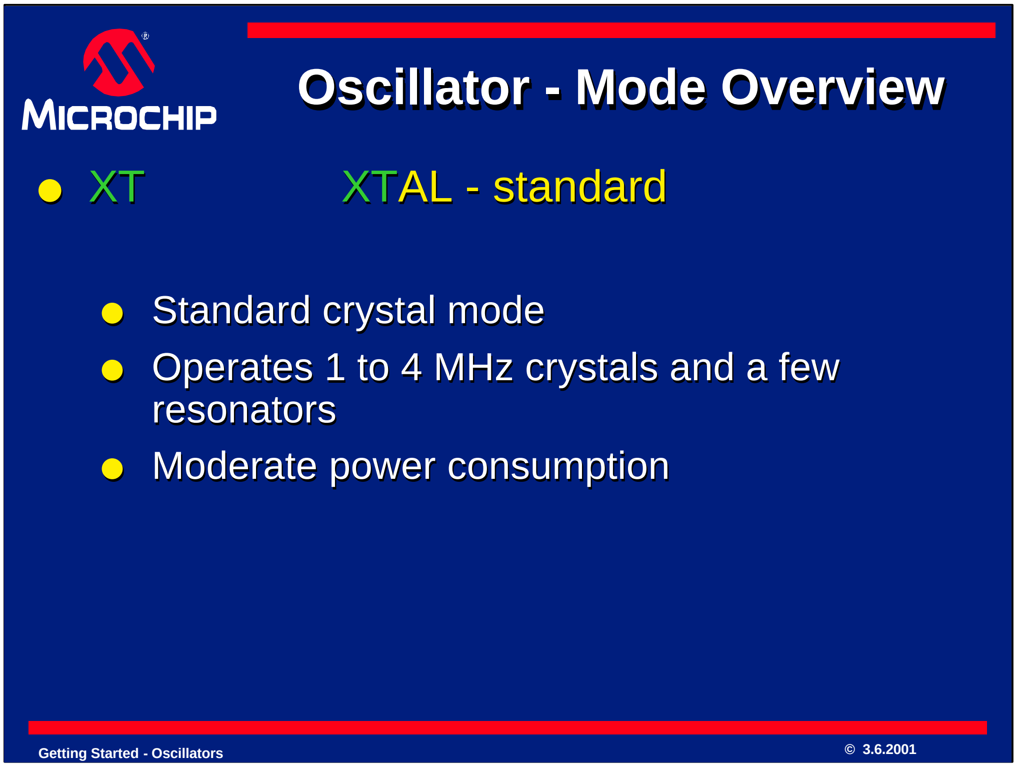
The HS mode stands for “High Speed” mode. It is designed to be used with crystals and resonators with a frequency of 3 to 4 MHz or more. However, this is only a guideline. The important thing to remember about HS mode is that it provides the highest drive level available.

Crystals and Resonators must be DRIVEN by a signal to work. The gain on this signal controls whether an oscillation will occur and how strong it will be. Care must be taken not to underdrive or overdrive the crystal or resonator.

If there is too much drive, crystals suffer long-term damage, often failing in a matter of weeks. With too little drive, the product may not start up correctly or may fail under certain conditions. If these are problems you have encountered, be sure to see the following sections and try modifying your drive level.

As a general rule, most resonators require more drive than crystals, so you will find that HS mode is commonly used for resonators.

Two additional tips: Be sure to qualify your design to insure it is stable and does not damage components. And remember, that as the frequency increases, the drive level required tends to increase. This is why HS mode is designed for use with resonators and crystals of 3 to 4MHz and faster.



Oscillator - Mode Overview

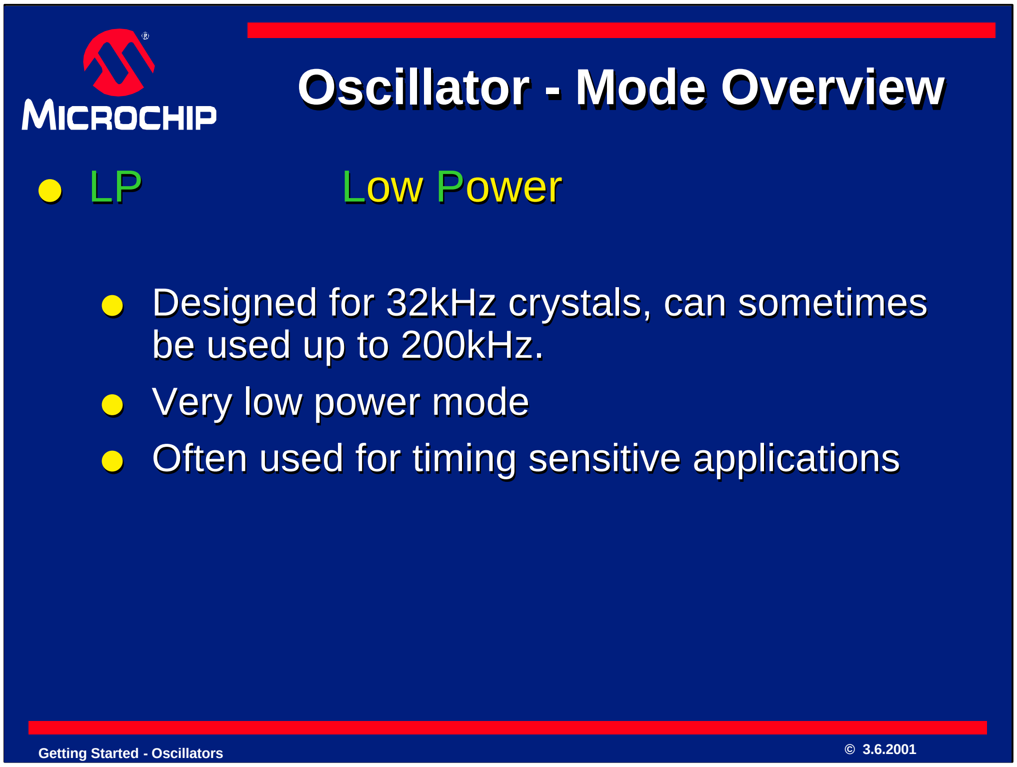
- XT XTAL - standard
  - Standard crystal mode
  - Operates 1 to 4 MHz crystals and a few resonators
  - Moderate power consumption

Getting Started - Oscillators © 3.6.2001

The XT mode stands for “Crystal” mode and will produce a medium drive level.

It is designed to be used with crystals and resonators of 1 to about 4 MHz. Again, if having problems starting up in this mode, you may want to switch to HS mode and requalify your design with the new mode.

XT mode has moderate power consumption since its drive level is lower than that of HS mode, and because a lower clock speed is produced. Remember, as a rule: the faster the clock used, the more current the application will require.



Oscillator - Mode Overview

- LP Low Power
  - Designed for 32kHz crystals, can sometimes be used up to 200kHz.
  - Very low power mode
  - Often used for timing sensitive applications

Getting Started - Oscillators

© 3.6.2001

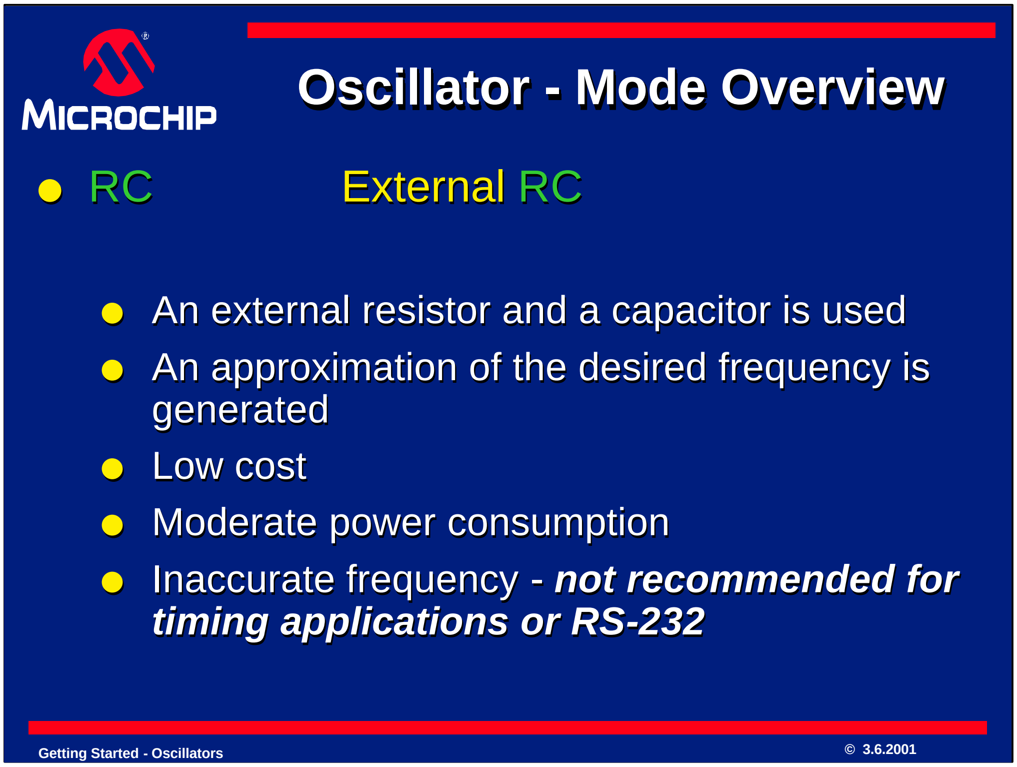
The LP mode stands for "Low Power” mode.

This mode is useful for circuits that require the lowest power possible. LP mode is engineered for 32.768kHz crystal operation, and it can function at any frequency below 200kHz. LP mode is most commonly used for 32.768kHz operation.

LP mode will produce the slowest clock rate, and as a result, the lowest power consumption of all the modes.

LP mode is ideal for timing sensitive applications since these same crystals are used as a time base in wrist-watches.

Note, however, that to produce accurate timing, the system should be as stable as possible. This refers to stability over temperature, voltage and other factors.



MICROCHIP

Oscillator - Mode Overview

- RC External RC
  - An external resistor and a capacitor is used
  - An approximation of the desired frequency is generated
  - Low cost
  - Moderate power consumption
  - Inaccurate frequency - ***not recommended for timing applications or RS-232***

Getting Started - Oscillators © 3.6.2001

The RC mode stands for “External RC” mode.

It is important to note that this is an _EXTERNAL_ RC mode, as some PICmicro devices have an _INTERNAL_ RC mode, which will be discussed later.

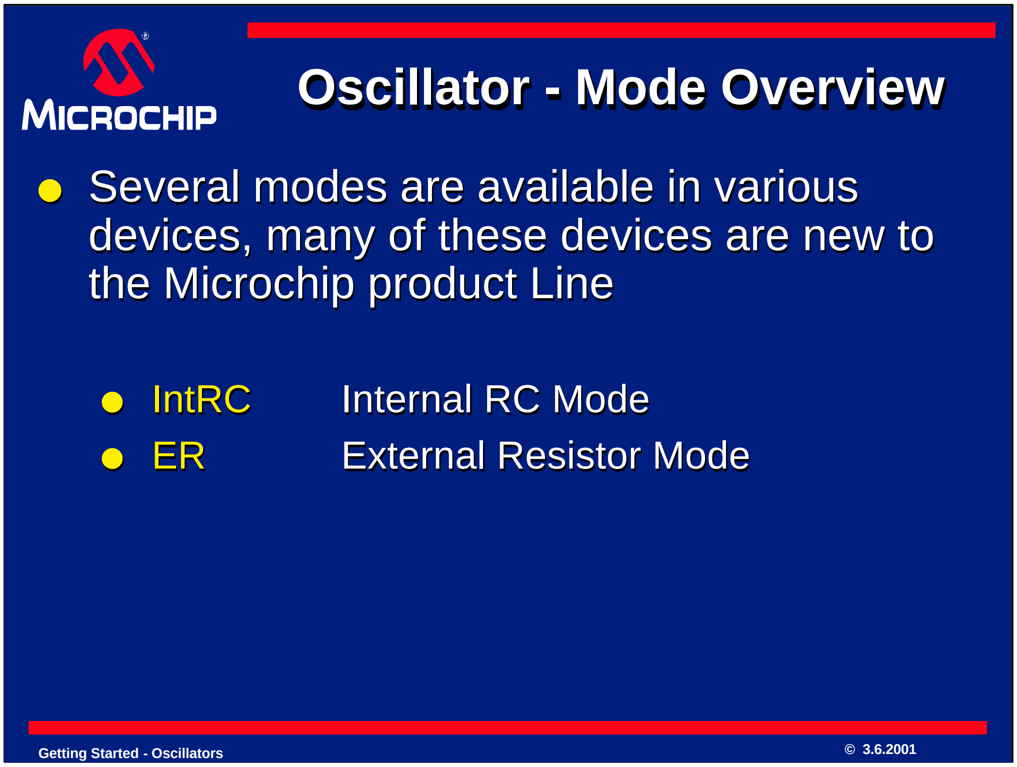
RC mode uses a resistor-capacitor network connected to the OSC1 pin. When the device is configured for external RC mode, these components are automatically driven to produce a frequency which will run the PICmicro MCU.

RC mode is designed for very low-cost applications. The power consumed will vary due to the wide range of frequencies that can be created using this mode. For the lowest available power, consider the LP mode discussed previously.

It is extremely important to note that RC mode will produce an inaccurate clock source. In some applications this won’t matter, but applications that are timing sensitive should not be used with this mode. For this reason, RC mode is not recommended for timing sensitive applications or for RS-232 communication.

Timing will vary due to process variance in the device, capacitor, resistor used and the printed circuit board. Humidity, proximity to other objects and temperature all affect the frequency. However, RC mode is one of the lowest cost modes available and far more rugged than crystals or resonators to impacts.

Also note that RC mode will often generate a clock of 4 times slower frequency on OSC2. This is known as a "divide by 4 clock". See the supporting device datasheet for details.



MICROCHIP

Oscillator - Mode Overview

- Several modes are available in various devices, many of these devices are new to the Microchip product Line

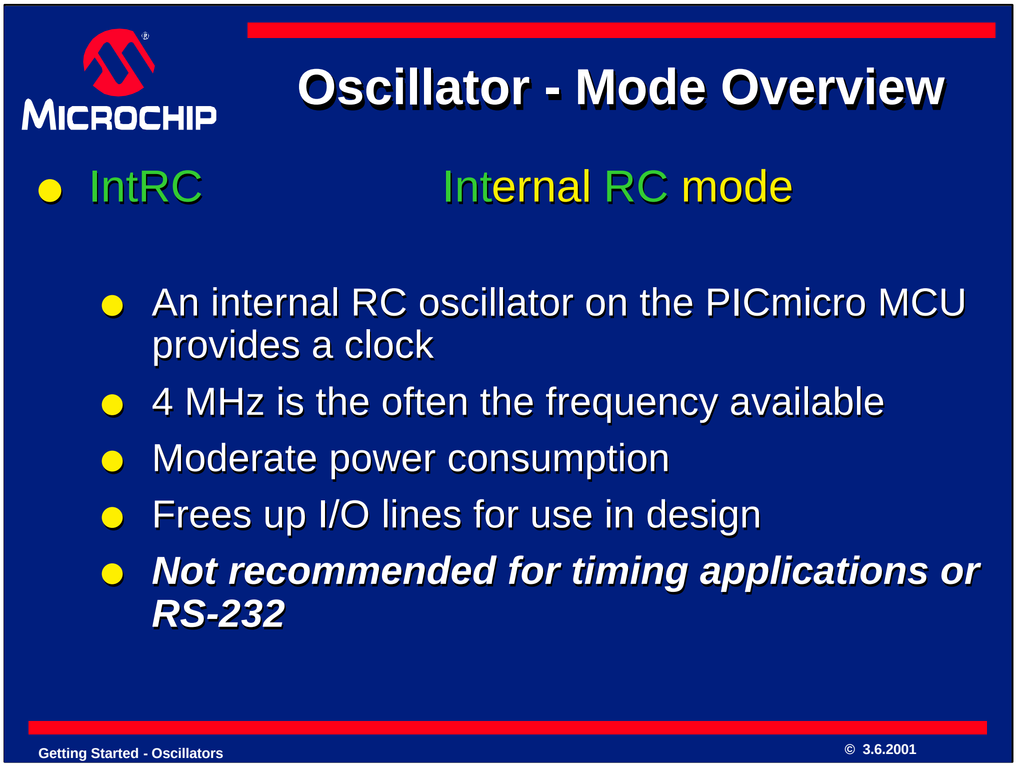
● IntRC	Internal RC Mode
● ER	External Resistor Mode

Getting Started - Oscillators

© 3.6.2001

That completes the discussion on the 4 primary oscillator modes available, HS, XT, LP and RC.

As new PICmicro devices have been introduced, new oscillator modes have also emerged. The new modes are Internal RC and External Resistor, and will be discussed next.



Oscillator - Mode Overview

- IntRC Internal RC mode
  - An internal RC oscillator on the PICmicro MCU provides a clock
  - 4 MHz is the often the frequency available
  - Moderate power consumption
  - Frees up I/O lines for use in design
  - ***Not recommended for timing applications or RS-232***

Getting Started - Oscillators

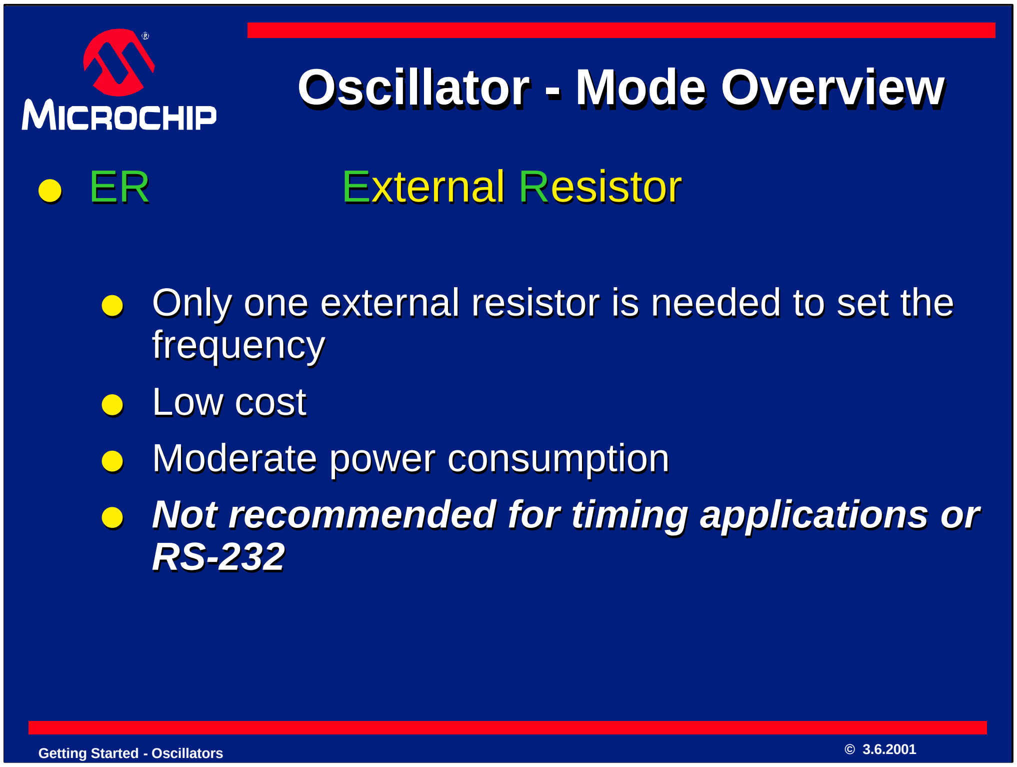
© 3.6.2001

The IntRC mode stands for “Internal RC” mode and functions much like the standard External RC mode.

Unlike External RC mode, in IntRC mode the resistor and capacitor are already provided. Microcontrollers with this feature have an on-chip RC oscillator. Current designs run at approximately 4 MHz.

IntRC mode is the least expensive oscillator available since no external components are needed. It is also useful, because devices in this mode can often use the OSC1 and OSC2 lines for general purpose I/O. This feature makes the PIC12CXXX and other 8-pin PICmicro devices very popular.

Often PICmicro devices with IntRC mode contain an OSCCAL register. This register is loaded by user software to calibrate the IntRC oscillator as accurately as possible. As with External RC mode, use IntRC mode for timing sensitive applications. RS-232 is not recommended due to the inherent inaccuracies of an RC oscillator. Details on this can be found in the supporting device datasheet.



MICROCHIP

Oscillator - Mode Overview

- ER External Resistor
  - Only one external resistor is needed to set the frequency
  - Low cost
  - Moderate power consumption
  - ***Not recommended for timing applications or RS-232***

Getting Started - Oscillators

© 3.6.2001

The ER mode stands for “External Resistor” mode.

ER mode is very similar to External RC mode, but no capacitor is needed. One resistor controls the frequency produced.

Similar to RC mode, ER mode has moderate power consumption, and is low cost. However, due to the nature of RC oscillators, it is not recommended for timing sensitive applications or for RS-232 use.

For information on the frequency resulting from various resistor values, refer to the device datasheet.

MICROCHIP

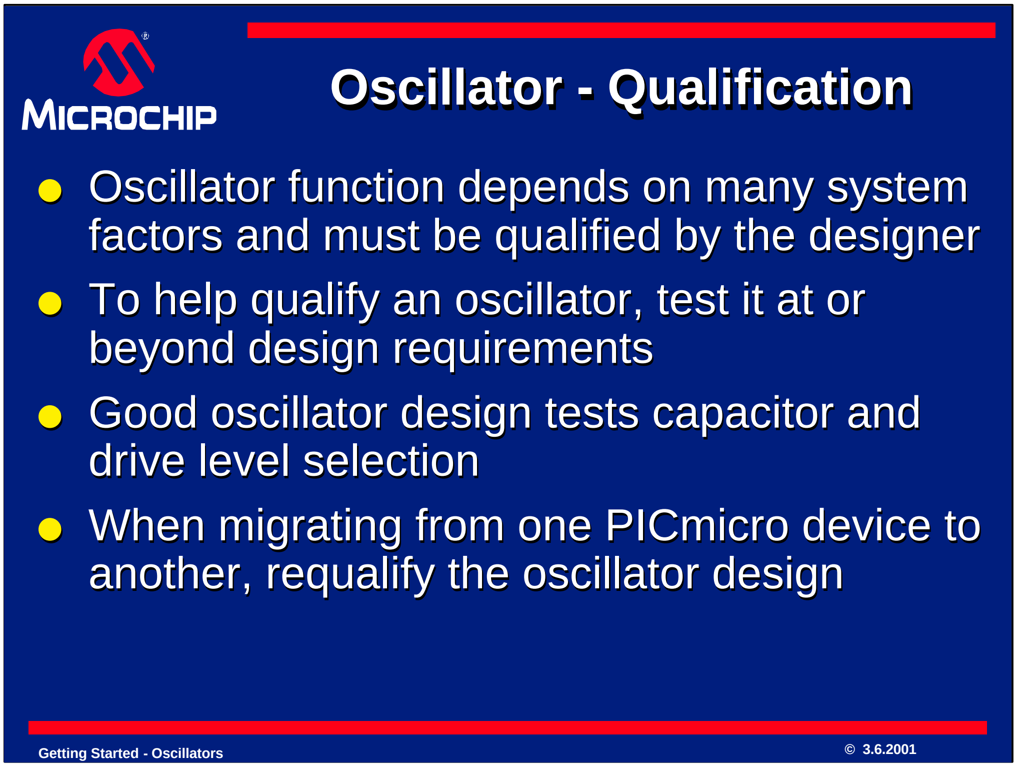
Oscillator Design and Qualification

Tips on designing and qualifying oscillator circuits for the PICmicro® microcontroller

Getting Started - Oscillators

© 3.6.2001

Now we are ready to explore the details of using crystals, resonators and other devices as oscillators. Each item is explored in detail. We also will look at the important topic of Qualifying Oscillators.



Oscillator - Qualification

- Oscillator function depends on many system factors and must be qualified by the designer
- To help qualify an oscillator, test it at or beyond design requirements
- Good oscillator design tests capacitor and drive level selection
- When migrating from one PICmicro device to another, requalify the oscillator design

Getting Started - Oscillators

© 3.6.2001

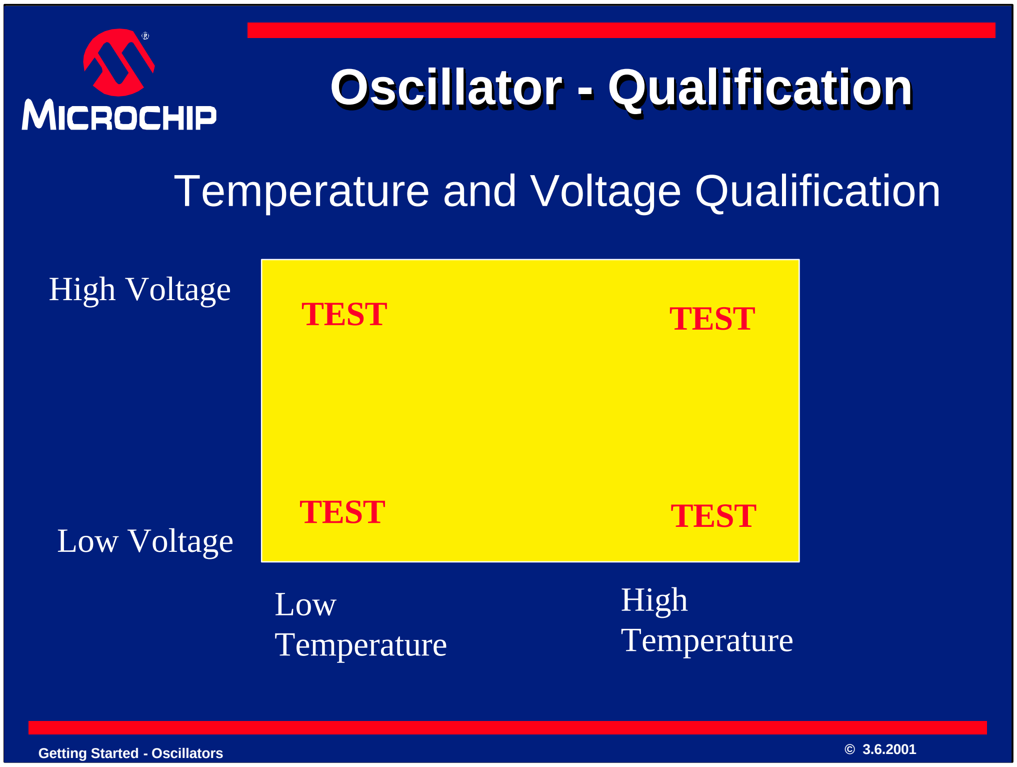
When an oscillator is designed, it should be qualified. The purpose of qualifying and requalifying an oscillator is to ensure it will work under all the conditions where it is expected and specified to perform. Oscillator designs should be tested just like your software and other hardware.

When qualifying an oscillator, it is a necessary to test it at least to the limits of its intended purpose. However, it is much better to test it beyond its intended use, to help ensure that it will function correctly in your design.

If using a crystal or resonator, make sure that your capacitor and drive level selection are correct. You should observe OSC2 with an oscilloscope. You should see a sine wave with an amplitude as large as possible without clipping.

Remember, since the probe has some capacitance, measuring the signal will affect it. You should use the highest power probe you can, such as a x100 or FET amplified probe, for testing oscillator circuits. x10 probes can be used, but they will load down the circuit considerably and the results may be inaccurate.

If you migrate your design from one PICmicro device to another, it is recommended that you requalify your oscillator design. Common examples of migration are a: PIC16C74 to a PIC16F877, or a PIC12C508 to a PIC12C508A, or a PIC16C54 to a PIC16C54C.



Oscillator - Qualification

Temperature and Voltage Qualification

High Voltage

TEST TEST

TEST TEST

Low Voltage

Low Temperature

High Temperature

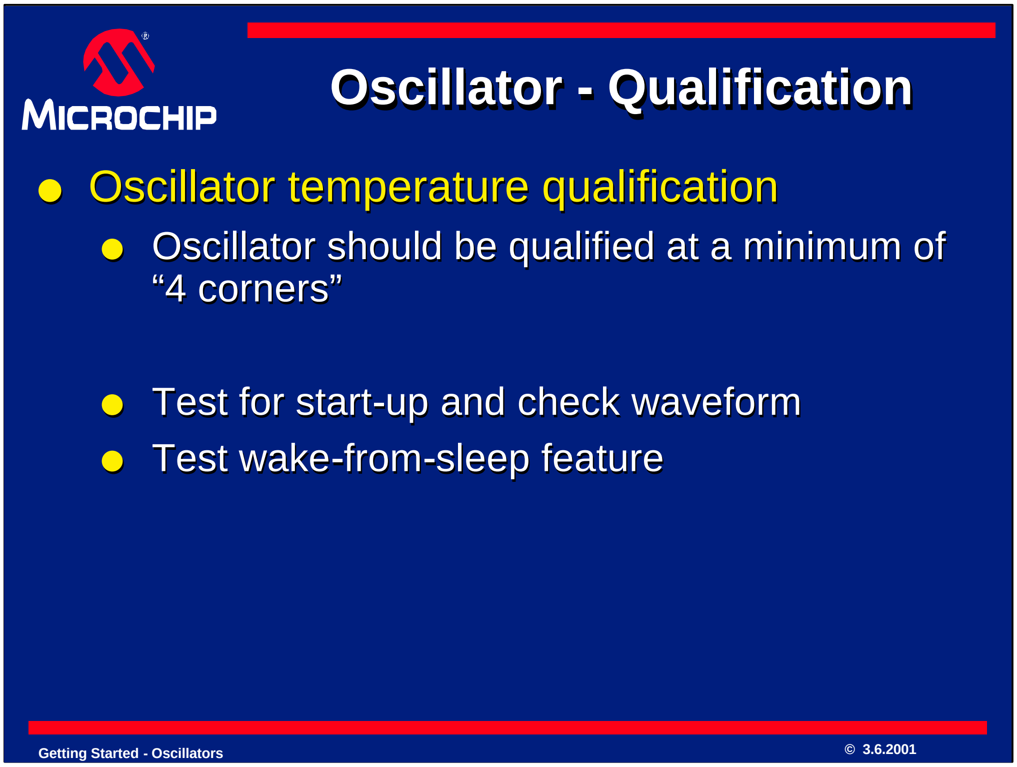
Getting Started - Oscillators

© 3.6.2001

This diagram represents the range of voltages and temperatures you can expect to operate your circuit in. The actual values will depend on what the PICmicro MCU is specified for, what the circuit is designed for, and what you are willing to test.

However, the general concept is this: The most important places to test the oscillator design is at the “corners.” A corner is a maximum and/or minimum of temperature and voltage.

The four sections labeled “TEST” in the diagram above represent the ideal places to test, and are referred to as the “4 corners.” It is certainly reasonable to test more, and recommended.



Oscillator - Qualification

- Oscillator temperature qualification
  - Oscillator should be qualified at a minimum of “4 corners”
  - Test for start-up and check waveform
  - Test wake-from-sleep feature

Getting Started - Oscillators

© 3.6.2001

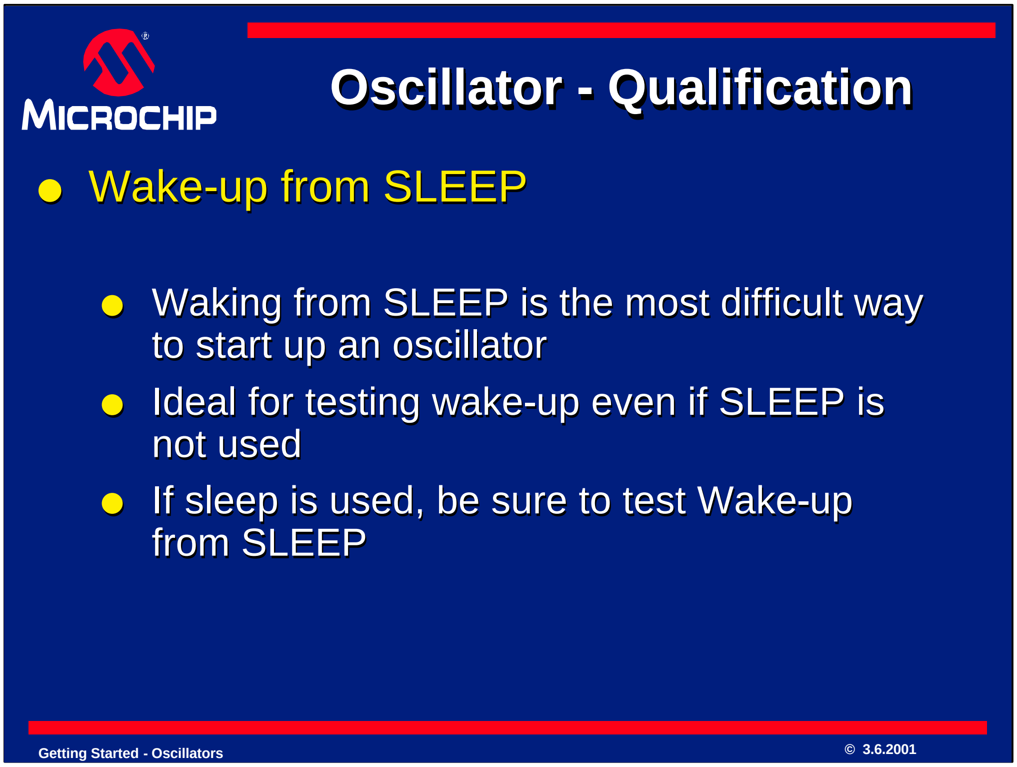
As mentioned in the previous diagram, a “corner” represents a minimum and/or maximum of voltage and temperature.

First test high voltage with low temperature, and low voltage with high temperature. This part of the four-corner test tends to check the highest and lowest drive levels the circuit will encounter. Next, test high voltage with high temperature, and low voltage with low temperature.

The signal on OSC2 should be a sinusoidal wave of maximum achievable amplitude without clipping. The peaks should approach the ground and VDD potentials of your application. The device datasheet for the PICmicro MCU you are testing will specify what a valid high and low is. Your oscillator should meet at least these specifications.

Interfere with the oscillator as little as possible by isolating the test from the circuit as much as possible. A x100 or FET amplified probe is a good start, and other methods can be devised.

Testing for "wake from SLEEP" is also recommended, which we will explore next.



Oscillator - Qualification

- Wake-up from SLEEP
  - Waking from SLEEP is the most difficult way to start up an oscillator
  - Ideal for testing wake-up even if SLEEP is not used
  - If sleep is used, be sure to test Wake-up from SLEEP

Getting Started - Oscillators

© 3.6.2001

Waking from SLEEP is the most difficult task for an oscillator to perform. Oscillators work by amplifying noise for starting up, and then resonating at the desired frequency.

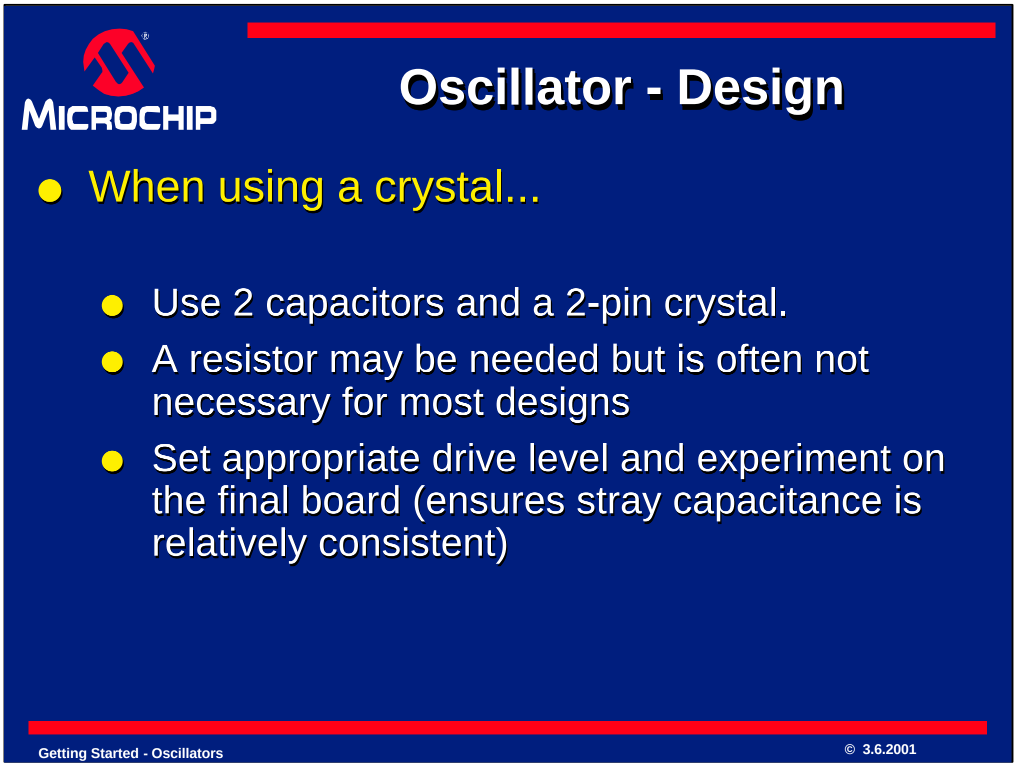
The oscillator circuit needs *some* noise in order to start up properly. A power-on spike (generated when your circuit is first switched on) is ideal for this, which explains why it is easier for an oscillator to start up when first powered up.

When the microcontroller is put to SLEEP, there is very little noise to allow the oscillator to start up. However, properly designed and qualified circuits should not have a problem starting up.

High temperature and low voltage also make waking from SLEEP difficult.

Even if you are not using SLEEP in your application, it makes a good “worst case” test. SLEEP, particularly at high temperature and low voltage, should also be tested using the corner method described earlier.

These tips will give you insight into the type of testing that can be done to aid qualification even if you determine that it is not necessary to test this much or this often.



Oscillator - Design

- When using a crystal...
  - Use 2 capacitors and a 2-pin crystal.
  - A resistor may be needed but is often not necessary for most designs
  - Set appropriate drive level and experiment on the final board (ensures stray capacitance is relatively consistent)

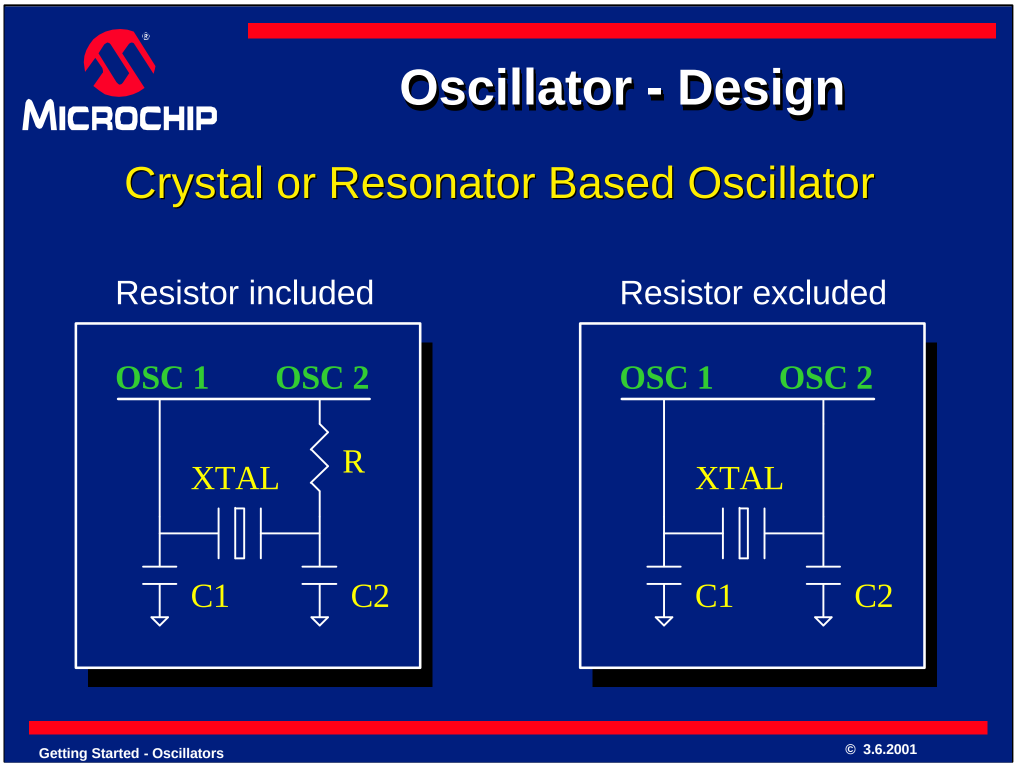
Getting Started - Oscillators

© 3.6.2001

When using a crystal, you will need at least a 2-pin crystal and 2 small capacitors. Although a resistor is not normally needed, it is still a good idea to leave the pads, and if it is not needed, a 0 or 1 ohm resistor can be put in its place. The resistor is needed if you see that clipping of your signal and adjustment of the capacitors was not enough.

It is a good idea to do all testing and qualification on the final board, since the board will have stray capacitance and other issues which may affect the oscillator. By doing the testing on the final board design, it helps to ensure that the behavior will be similar on all manufactured boards.

Remember, you want to see a sinusoidal wave on OSC2 that is not clipped and of as maximum an amplitude as possible. If you observe a clipped signal you may need to add the resistor mentioned above to reduce gain. Again, remember that measuring the OSC2 pin will load it down a small amount.



MICROCHIP

Oscillator - Design

Crystal or Resonator Based Oscillator

Resistor included

Resistor excluded

Getting Started - Oscillators

© 3.6.2001

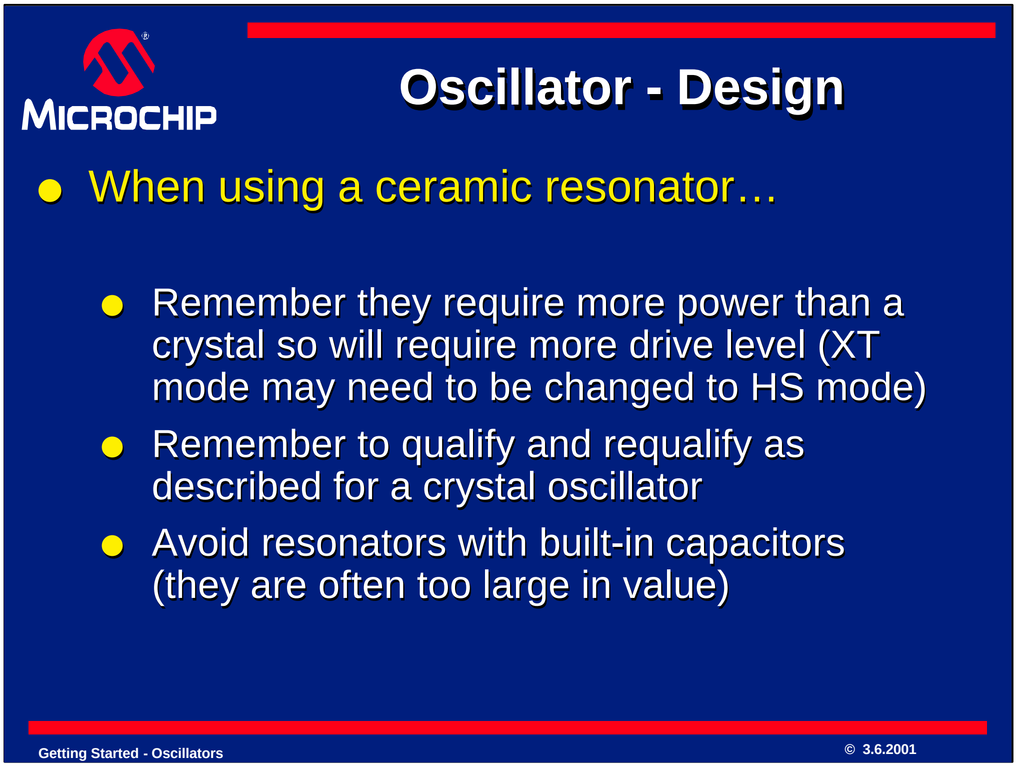
These two diagrams show how a crystal or resonator is connected to a PICmicro device. Only the OSC1 and OSC2 pins of the microcontroller are shown here.

Here we see the connections to the OSC1 and OSC2 pins of the PICmicro microcontroller.

The capacitor C1 is connected between the OSC1 pin and ground. Capacitor C2 is connected between the bottom of resistor “R” and ground. If no resistor is used, it is connected directly between OSC2 and ground instead. The crystal, named X-T-A-L in this diagram, is connected to OSC1 and the bottom of the resistor. If there is no resistor present, then it connects directly between OSC1 and OSC2.

To measure the oscillator, it should be tested with an oscilloscope connected to OSC2. Remember that the oscilloscope probe should be a x100 probe or FET amplified.

C2 controls the overall gain of the oscillator, while C1 mainly controls the phase shift. Often these capacitors are of equal value, but some adjustment of these may be needed in your design.



Oscillator - Design

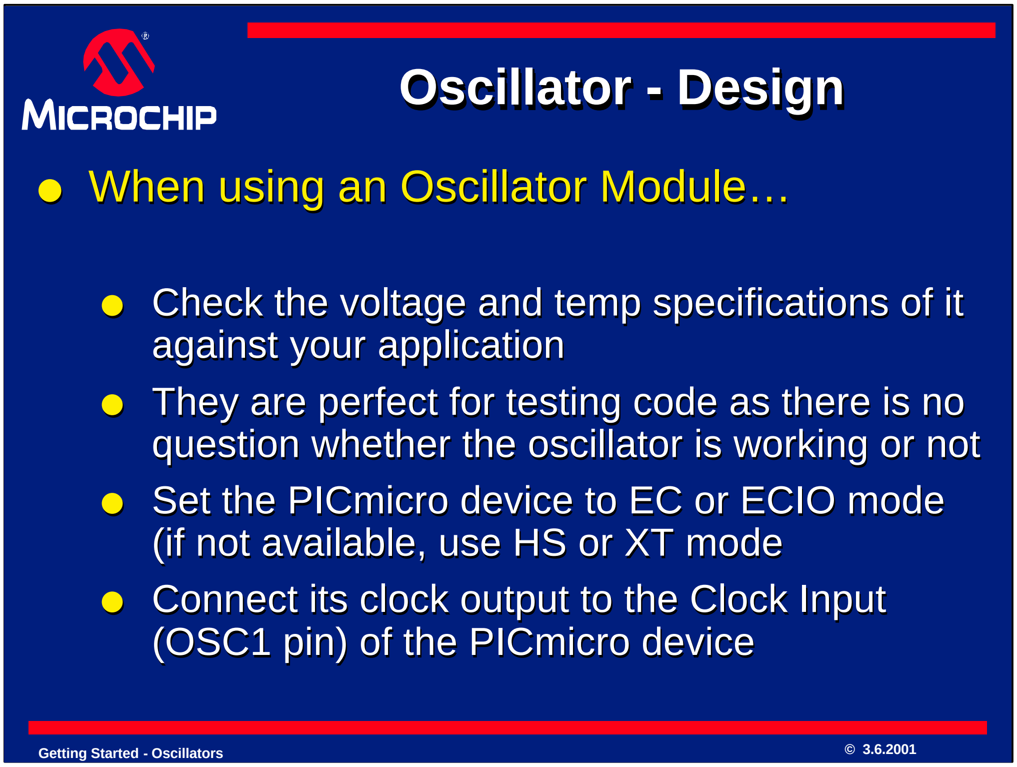
- When using a ceramic resonator…
  - Remember they require more power than a crystal so will require more drive level (XT mode may need to be changed to HS mode)
  - Remember to qualify and requalify as described for a crystal oscillator
  - Avoid resonators with built-in capacitors (they are often too large in value)

Getting Started - Oscillators

© 3.6.2001

When using a ceramic resonator, the same rules for crystals apply. Ceramic resonators are often used as they are lower in cost and less prone to vibration damage. However, as mentioned earlier, they tend to require more drive level. Resonators with a frequency greater than 3 to 4 MHz can be difficult to drive, due to limitations in the available drive level.

As a general rule, avoid resonators with built-in capacitors. Using built-in capacitors will prevent you from being able to make changes easily during the qualification process. Plus, many resonators use capacitors that are too large for the drive level supplied by the microcontroller.



Oscillator - Design

- When using an Oscillator Module…
  - Check the voltage and temp specifications of it against your application
  - They are perfect for testing code as there is no question whether the oscillator is working or not
  - Set the PICmicro device to EC or ECIO mode (if not available, use HS or XT mode
  - Connect its clock output to the Clock Input (OSC1 pin) of the PICmicro device

Getting Started - Oscillators

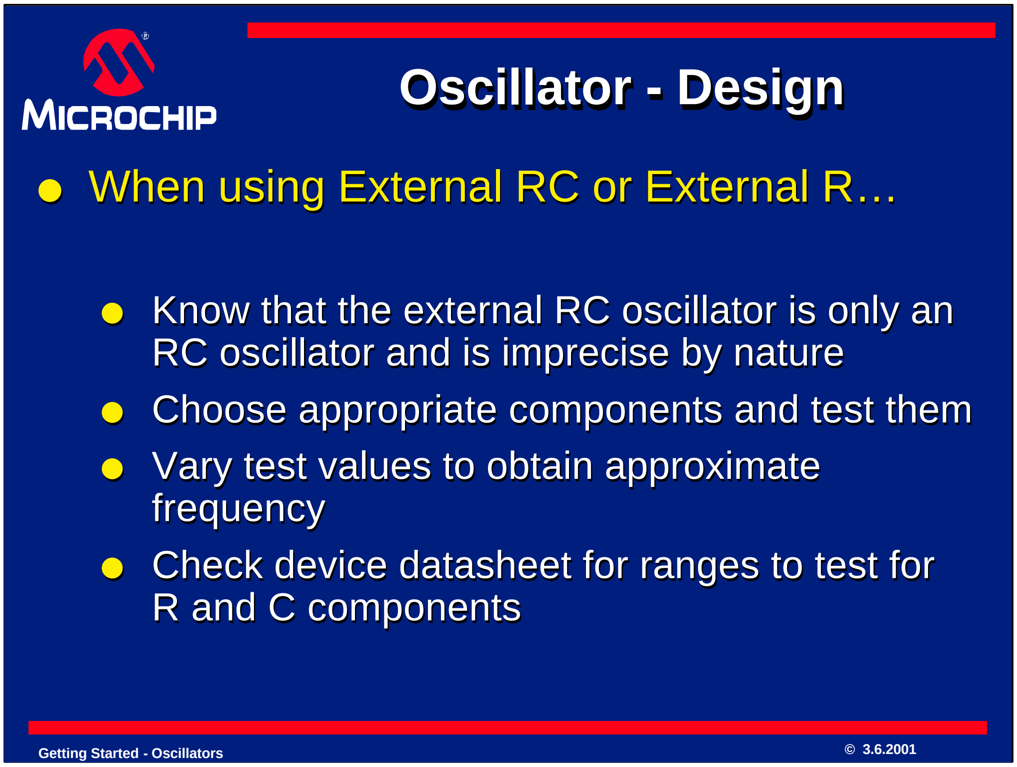
© 3.6.2001

External oscillator modules are one of the easiest items to use. Place the PICmicro device into HS or XT as described earlier and feed the signal into the OSC1 pin. OSC2 can be left floating. See the device datasheet for details.

As with any component in your design, it is important to ensure that the oscillator module is designed to function within the desired range. There are a large number of modules available, some are designed for 5V, others are designed for 3.3V operation. The temperature range can also vary widely. Ensure the oscillator module is suitable to your design.

Oscillator modules are ideal for testing code, and for products that you do not wish to require a lot of oscillator qualification effort. Oscillator module drawbacks are their higher current consumption, size, and cost. Also, unlike a crystal-based design, an oscillator module can drive more than just one microcontroller. Frequently it can drive several PICmicro devices as well as other components of your circuit. They are ideal for designs that require an oscillator clock in several places.

If you have a design that is not working, try swapping in an oscillator module. You may find that the oscillator was the cause of your problem. And, if nothing else, it will tell you that the oscillator is fine, and the problem really does lie within the code or other parts of your hardware.



Oscillator - Design

- When using External RC or External R…
  - Know that the external RC oscillator is only an RC oscillator and is imprecise by nature
  - Choose appropriate components and test them
  - Vary test values to obtain approximate frequency
  - Check device datasheet for ranges to test for R and C components

Getting Started - Oscillators

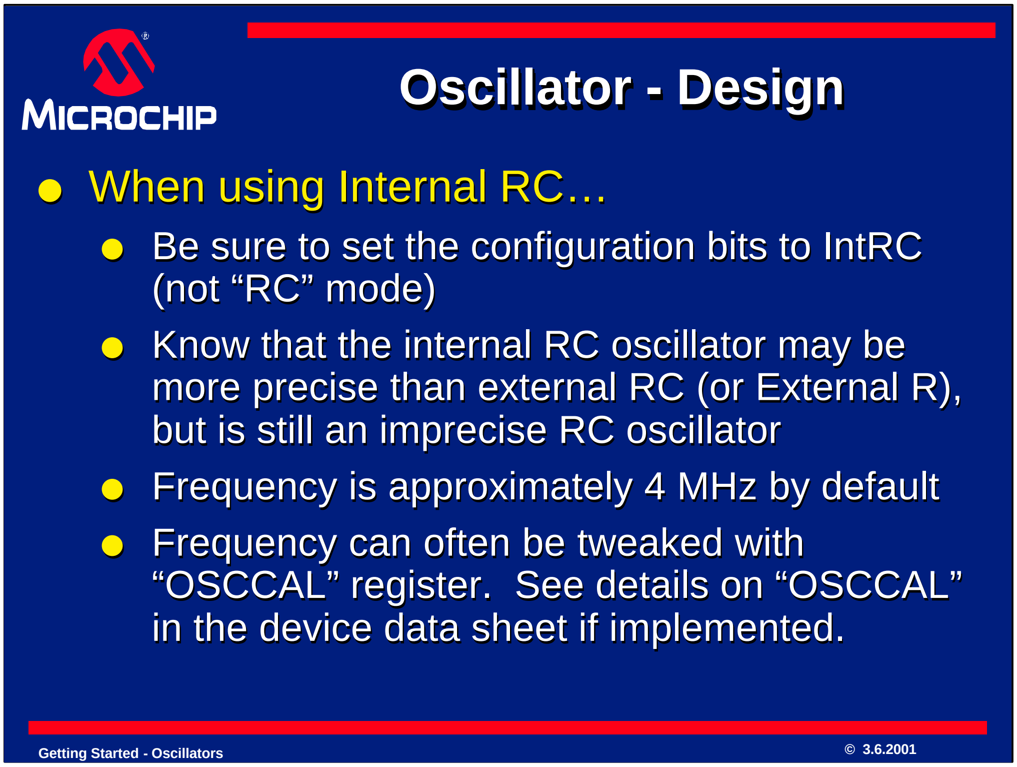
© 3.6.2001

The External RC or External R mode is simple to use.

As with any RC oscillator, it is not a precise time base and should not be used for RS-232 or other timing applications. However, it is an inexpensive way of getting almost any clock frequency up to 4MHz maximum.

Due to drift in temperature, process variation, and other factors, specifications of what frequency is produced by what combination of R and C are often not available. This mode requires some trial and error to find the values for the desired frequency. However, the device datasheet will provide you with a range of values that are valid for use with the external RC or external R mode. Since there is little issue with loading and drive level, the frequency can be easily tested with an oscilloscope or frequency counter.

To see the clock rate when using External RC mode, test the OSC2 pin. It will output one-quarter the clock rate the R-C oscillator is producing.



Oscillator - Design

- When using Internal RC…
  - Be sure to set the configuration bits to IntRC (not “RC” mode)
  - Know that the internal RC oscillator may be more precise than external RC (or External R), but is still an imprecise RC oscillator
  - Frequency is approximately 4 MHz by default
  - Frequency can often be tweaked with “OSCCAL” register. See details on “OSCCAL” in the device data sheet if implemented.

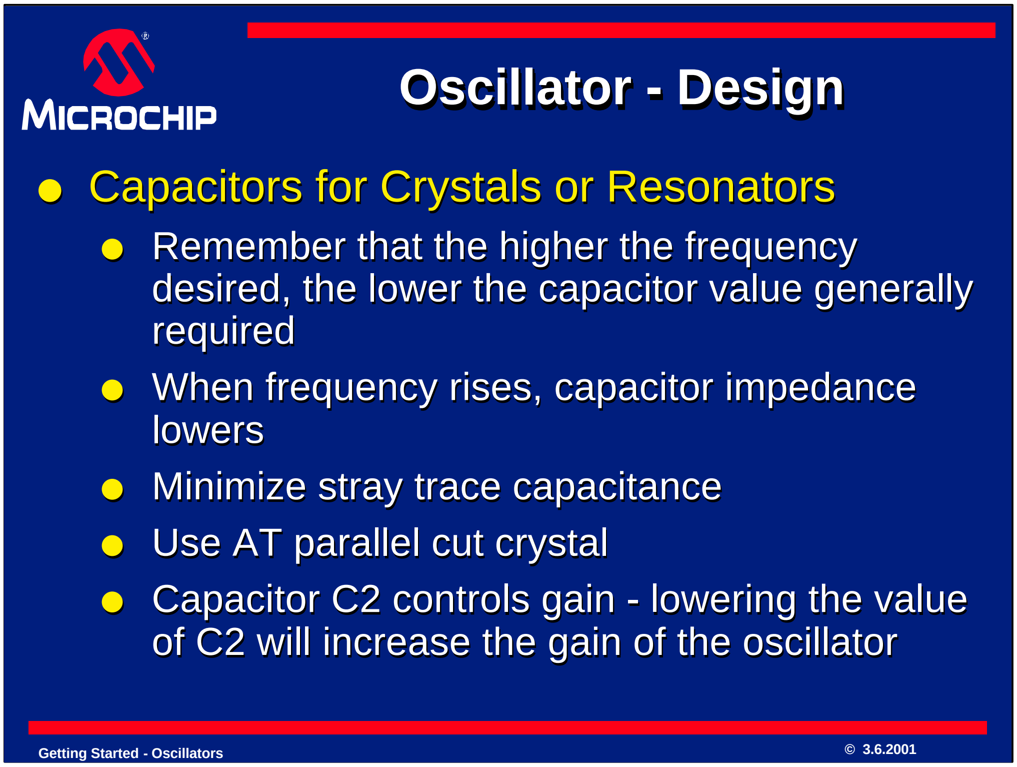
Getting Started - Oscillators

© 3.6.2001

Internal RC is another very simple mode to use. First be sure you set the mode to be Internal and not External RC. Since they are both called RC modes of one type or another, this can be easy to do.

Again, due to the nature of R-C oscillators, this mode is not recommended for timing sensitive applications or RS-232 use. The frequency from most internal RC modes is approximately 4 MHz. Other options may be available, consult the device data sheet for details.

Often, the oscillator can be calibrated. While this will not prevent frequency drift, a calibrated oscillator will be closer to the desired 4 MHz or other frequency than one that is not calibrated. See index entry, “OSCCAL” in your device data sheet for more information on calibration if available on your device.



Oscillator - Design

- Capacitors for Crystals or Resonators
  - Remember that the higher the frequency desired, the lower the capacitor value generally required
  - When frequency rises, capacitor impedance lowers
  - Minimize stray trace capacitance
  - Use AT parallel cut crystal
  - Capacitor C2 controls gain - lowering the value of C2 will increase the gain of the oscillator

Getting Started - Oscillators

© 3.6.2001

Some additional tips to remember when choosing capacitors when you are using a crystal or resonator:

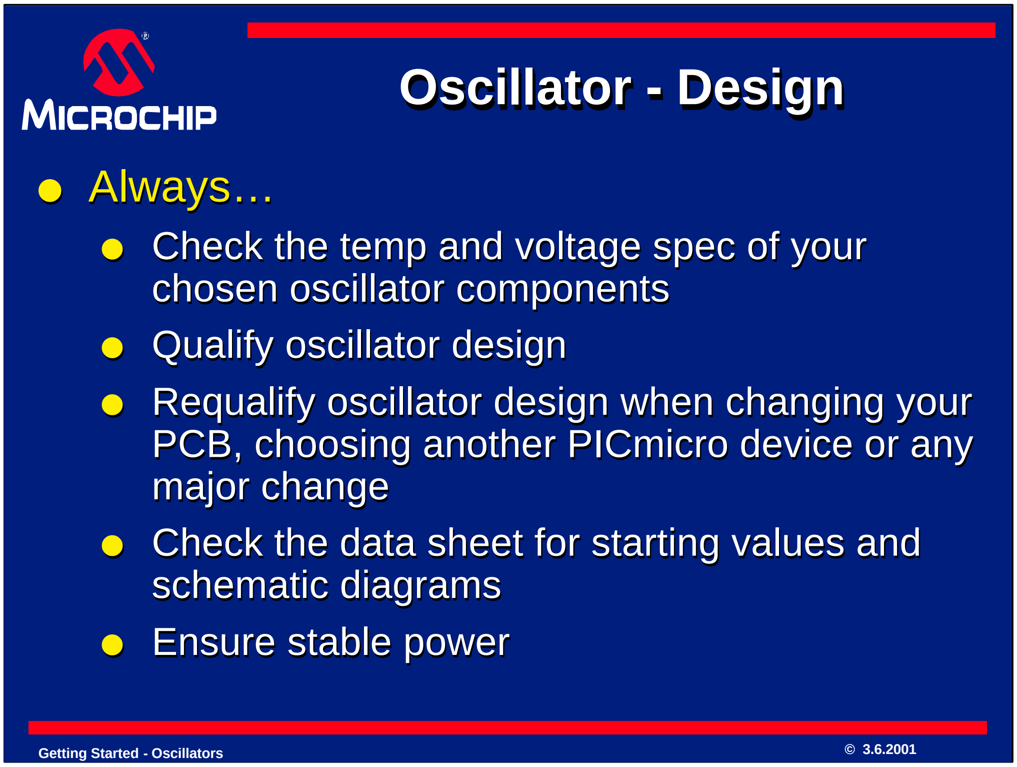
Consult the device datasheet of both the PICmicro MCU and the manufacturer of the crystal or resonator. They will both give reasonable ranges and suggested values. As a rule, keep within these ranges during your tests. If using a higher frequency, lean more toward lower capacitance values.

Smaller caps are more appropriate for higher frequencies due to the law of capacitive reactance. This law states that for any given capacitor, the higher the frequency, the more easily it will pass though that capacitor. Oscillators require some of the current to pass though, but not too much. As a general rule, the current must remain fairly constant, so the capacitor changes to match the change in frequency.

It is a good idea in general to minimize stray capacitance on your board, particularly in the oscillator area. This helps prevent unwanted capacitance from affecting your oscillator.

Use an AT cut crystal. The gain profile of the PICmicro oscillator is designed to use AT cut crystals.

When having drive level problems, try adjusting the capacitors. C2 controls the gain of the oscillator. The lower the value of C2, the higher the gain that can be expected. Note that C2 is the capacitor which connects to the OSC2 pin of the PICmicro microcontroller.



Oscillator - Design

- Always…
  - Check the temp and voltage spec of your chosen oscillator components
  - Qualify oscillator design
  - Requalify oscillator design when changing your PCB, choosing another PICmicro device or any major change
  - Check the data sheet for starting values and schematic diagrams
  - Ensure stable power

Getting Started - Oscillators © 3.6.2001

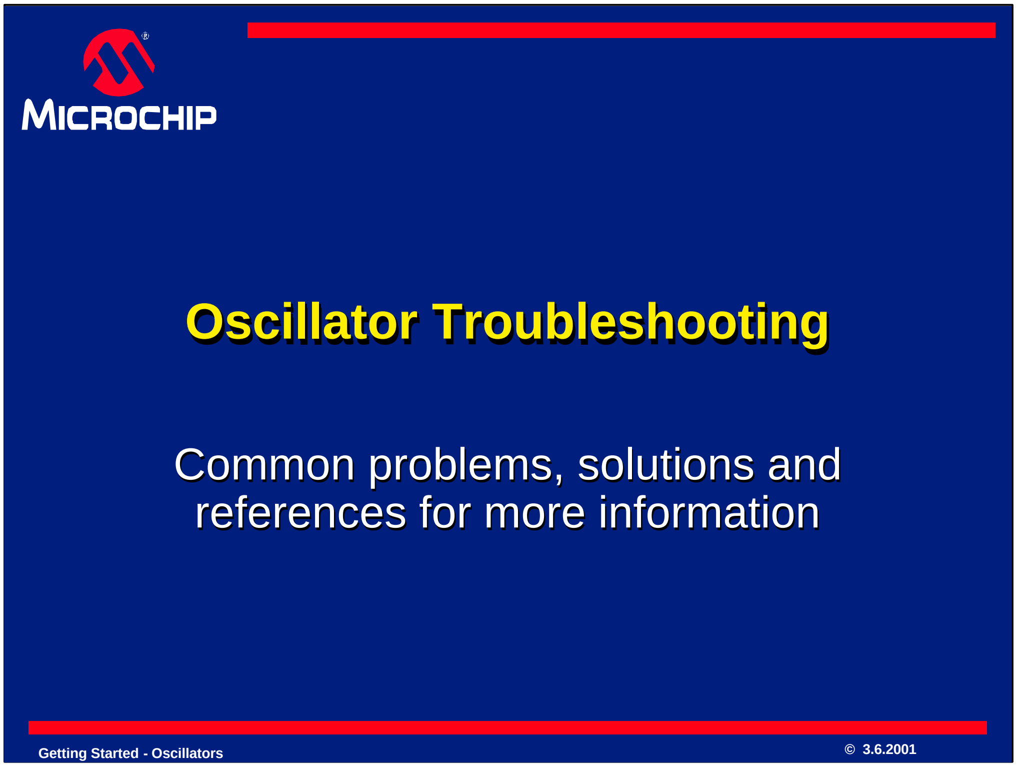
When designing oscillators of any kind, always remember:

* Check the temperature and voltage specifications of all components, including the oscillator components.

* Always qualify your initial design, and if you change the oscillator circuit be sure you requalify your design. This is true even if you change just the microcontroller. Changing the PICmicro device includes migrations, such as migrating from a PIC16F84 to a PIC16F84A.

* Check the data sheets for the devices involved to get an idea of what are good starting values. Use your knowledge of oscillator design to experiment and modify as appropriate.

* Oscillators will not function correctly if the power supply is not stable. Ensure the stability of your power supply and use bypass capacitors as appropriate to your design.



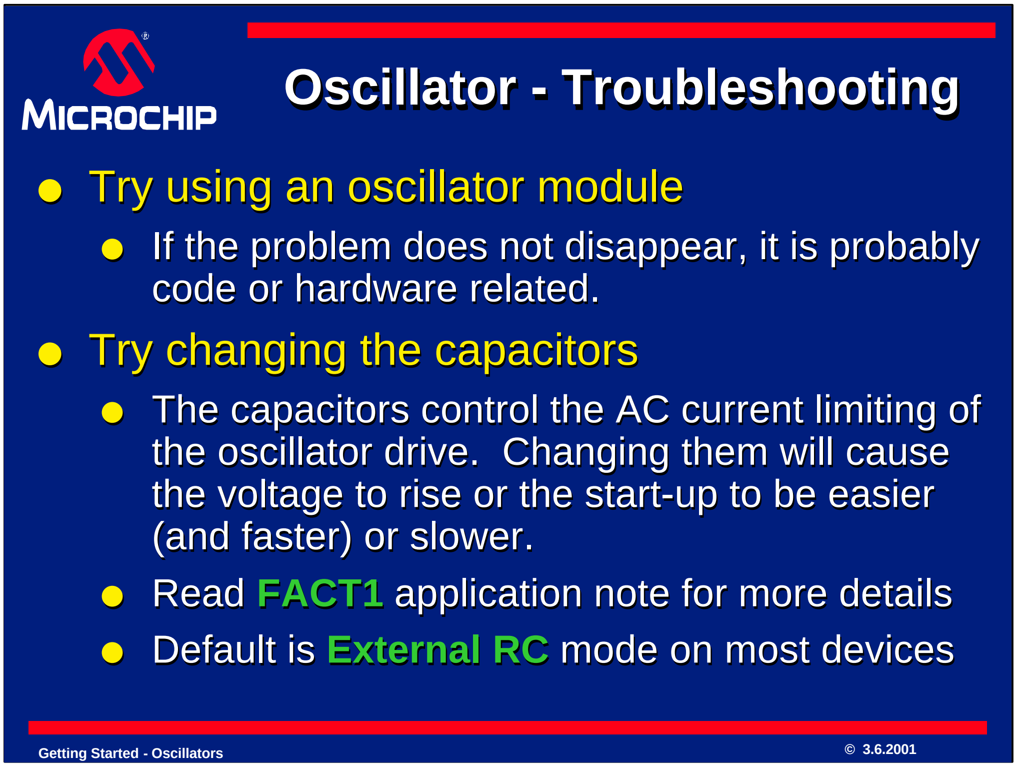
Oscillator Troubleshooting

Common problems, solutions and references for more information

Getting Started - Oscillators

© 3.6.2001

This final section will discuss a few troubleshooting tips and leave you with some resources to look at for more information.



Oscillator - Troubleshooting

- Try using an oscillator module
  - If the problem does not disappear, it is probably code or hardware related.
- Try changing the capacitors
  - The capacitors control the AC current limiting of the oscillator drive. Changing them will cause the voltage to rise or the start-up to be easier (and faster) or slower.
  - Read **FACT1** application note for more details
  - Default is **External RC** mode on most devices

Getting Started - Oscillators

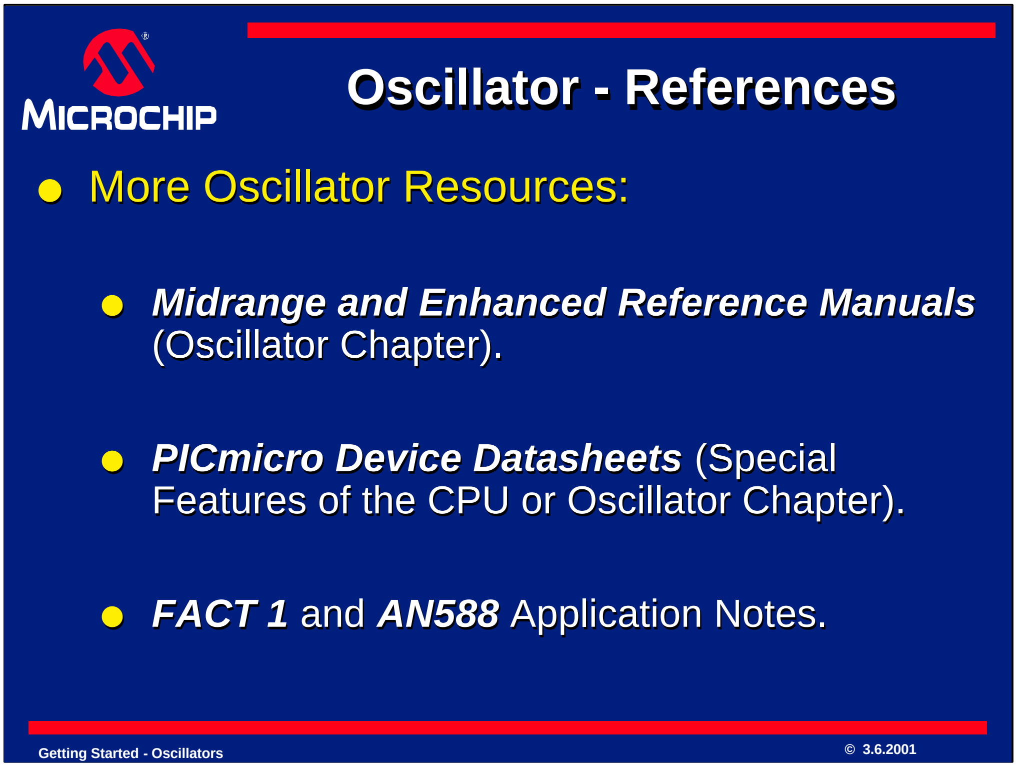
© 3.6.2001

When troubleshooting oscillator designs, always remember to make sure it really is the oscillator that is at fault. This is tested by swapping in an external oscillator module or an appropriately configured function generator.

If your design doesn't start up properly, fails after some time, or acts erratically and you are using a crystal or resonator, you may have a drive level problem. Experiment with the capacitor values and drive level as appropriate. Chances are start-up and erratic operation is an under-driven oscillator and the capacitors need to be decreased. Failure after some time may suggest that the crystal is being overdriven. Overdriving a crystal will permanently damage it and the damage can take weeks or months to become apparent.

The ratio of the two capacitors can also be changed. This is discussed in detail in an application document titled FACT 1, available from the Microchip website. FACT 1 and other application notes contain more tips that are not covered here. This guide was created to get you started with your oscillator design and familiarize you with the options available and key points.

Note that the default mode of most PICmicro devices is External R-C mode. If you do not program the PICmicro MCU, it will likely be in External RC mode.



Oscillator - References

- More Oscillator Resources:
  - ***Midrange and Enhanced Reference Manuals*** (Oscillator Chapter).
  - ***PICmicro Device Datasheets*** (Special Features of the CPU or Oscillator Chapter).
  - ***FACT 1*** and ***AN588*** Application Notes.

Getting Started - Oscillators

© 3.6.2001

This concludes the brief examination of PICmicro MCU Oscillator considerations.

Additional information on this topic can be found in the following literature available in pdf format on the Website:

Midrange and Enhanced Reference Manuals (Oscillator Chapter).

PICmicro Device Datasheets (Special Features of the CPU or Oscillator Chapter).

FACT 1, “Basic PIC16/17 Oscillator Design” and Application Note *AN588, “PICmicro Microcontroller Oscillator Design Guide.”*

Visit: “http://www.microchip.com” to locate these documents, or contact your local Microchip Sales Office.

“Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Inc. with respect to the accuracy of such information, or infringement of patents arising from any such use of otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.”