

LECTURE-4

What is an Oscillator?

Load the code onto the chip and the ATmega16 runs that code, but how is it running? And how fast? Any micro needs a clock source. Think of it like a type of 'musical beat' that the micro uses to execute its code in a set manner. Without a clock, the micro doesn't know how to run the code, and with a sloppy clock (one that varies a lot) the code runs at an undetermined rate.

There are many ways to generate a 'beat' for the microcontroller. The ATmega16 gives many options. Here is a quick breakdown of the different types of beat generators:

- **External RC** - This is generally used for *very low cost* applications. Using a resistor and a capacitor, the charge/discharge rate can be used as a clock input.
- **Internal RC** - This is the super cool oscillator. Found on newer micros, you can just ask the ATmega16 to generate its own clock! The silicon has a built in oscillator. Unfortunately it's not very accurate.
- **External Oscillator** - This is the defacto standard. Attach a quartz crystal ('crystal' for short) to the two osc pins and the code executes at this given frequency.
- **External Resonator** - A resonator is a bit cheaper than an oscillator but has worse tolerances.
- **External Clock** - Instead of an oscillator, one can use a powered clock driver. Handy if have multiple devices that need to run on the same frequency.

Internal RC –

ATmega16 is shipped preconfigured to use the 1MHz internal oscillator. The ATmega16 (and AVRs in general) can operate at 1 instruction per clock cycle. This means that every time the oscillator goes through one cycle, one instruction is completed (this is roughly true - there are some instructions that take more than one clock cycle). Because we are using stock ATmega16s, we are running the blink code at 1MHz or 1MIPS (million instructions per second). That's pretty impressive. What's the problem with the internal osc? It has a tolerance of +/-5% and a max speed of 8MHz. +/-5% tolerance means ATmega16 might run at $1,000,000 * 1.05 = 1,050,000$ IPS while other ATmega16 runs at $1,000,000 * 0.95 = 950,000$ IPS. This may not sound like much difference but in the digital world, this is huge! Also, the ATmega16 has a max speed of 20MHz (the internal osc runs max up to 8MHz) so if one really wanna push this IC to the max, need to use an external oscillator.

External Oscillator –

This one is most common type of clock source.



Quartz crystal oscillators come in all different flavors and frequencies. Some of the more common freq are 20MHz, 16MHz, 10MHz, 4MHz. There are also some frequencies like 14.7456MHz, 9.216MHz, and 32.768kHz that are available because these frequencies are multiples of speeds needed for serial communications and for timing. For example in need of really accurate 9600bps serial communications, $9.216\text{MHz} \div 960 = 9600$. There is no integer that divides nicely into 16MHz

to get 9600. So serial communications at 9.216MHz will be very accurate while serial at 16MHz will always have some small amount of error.

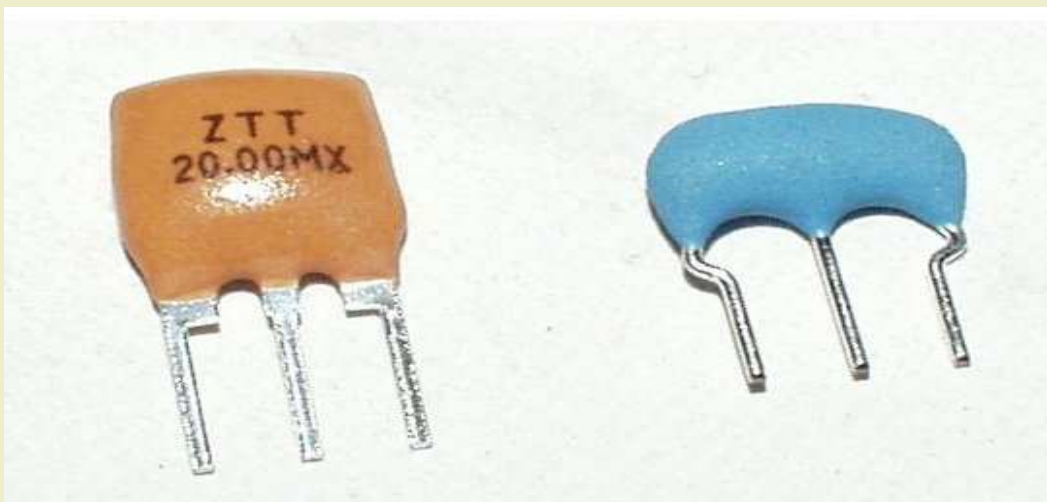
Inside the metal housing is a small piece of quartz crystal that has been precisely cut in size so that the piece of crystal vibrates at a specific frequency. The ATmega's internal osc was +/-5% tolerant. On the other hand, a crystal is normally '+/-20ppm'. This means the frequency is accurate within +/-20 parts per million! This is equivalent to +/-0.00000125%. The crystal is 4 million times more accurate than the internal oscillator!

Some negatives of crystals:-

- Crystals are a bit expensive compared to a resonator.
- They cannot be made as small as a resonator (crystals take up more PCB area).
- Crystals require 'load capacitors'. Load capacitors start the crystal oscillating. Without the load caps, crystal may function today, but someday it may not. Load caps are cheap, but they take more room on the PCB.

External Resonator –

Resonators fall in between the internal RC and a crystal.



A resonator is a piece of ceramic that is manufactured in such a way to oscillate at a given frequency. Unfortunately this process is difficult to do well. Resonators have a standard tolerance of $\pm 0.5\%$. So resonators are 10 times more accurate than the internal oscillator but they are still a bit 'loose' compared to crystals.

Resonators tend to be cheaper than crystals. Resonators tend to be lower frequency than crystals. Resonators are cool because they have the 'load caps' built into the 3-pin device! Resonators can be made *very small* and can minimize your PCB area.

NOTE - As mentioned earlier, the ATmega16 ships configured with an internal 1MHz osc. But we can push it faster - and better! Let's setup the ATmega16 to run at 16MHz with an external crystal.

What are some of the side-effects of running the ATmega168 at this higher frequency? You will not be able to run the IC at this higher freq at lower voltages (such as 3.3V or 2.8V). Since we are dealing with a 5V supply, this is not a deal breaker. At 20MHz the micro will consume more power than at 1MHz. These are all things to consider when developing your own system.