Microprocessor – Microcontroller

W3.3 – Basic hardware foundations

Nguyen Tran Huu Nguyen

D: Computer Engineering

E: nthnguyen@hcmut.edu.vn

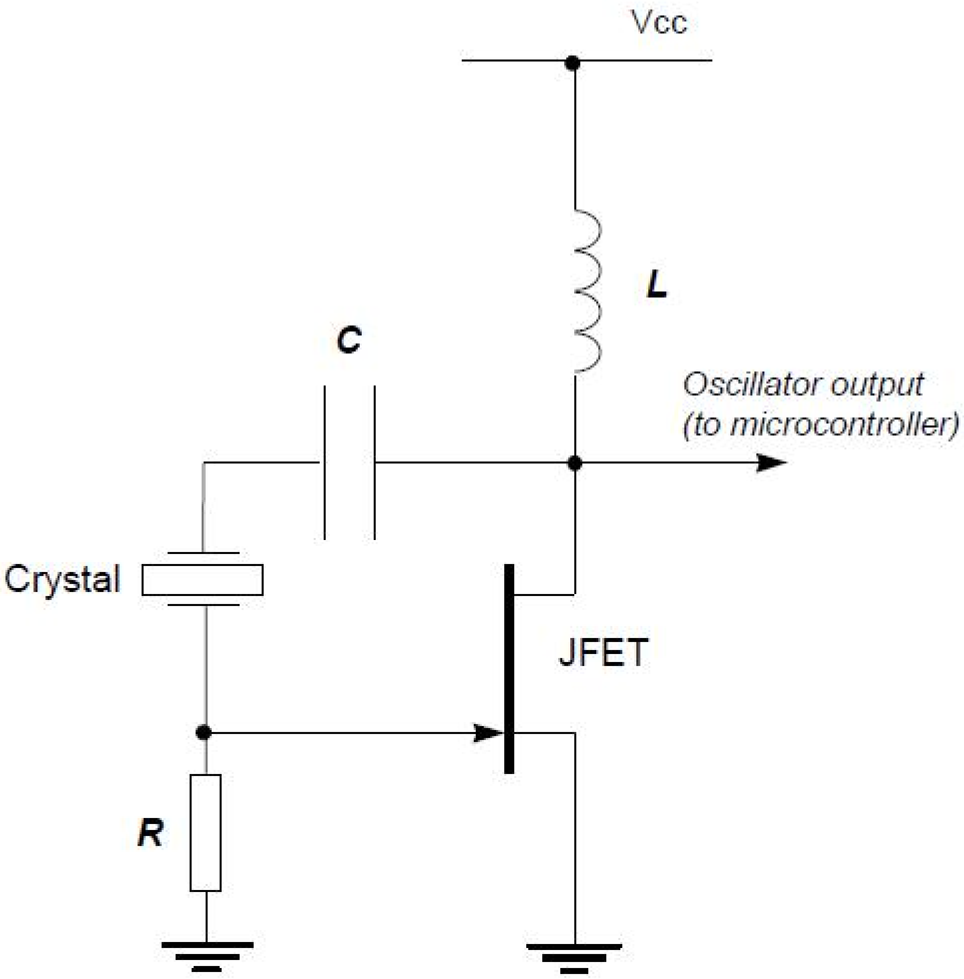
Contents

- Oscillator circuits
- Reset circuits

Oscillator Circuits

- All digital computer systems are driven by some form of oscillator circuits
- This circuits is the heartbeat of the system and is crucial to correct operations
- For example:
  - If the oscillator fails, the system will not function at all
  - If the oscillator runs irregularly, any timing calculation performed by the system will be inaccurate.

Crystal oscillator

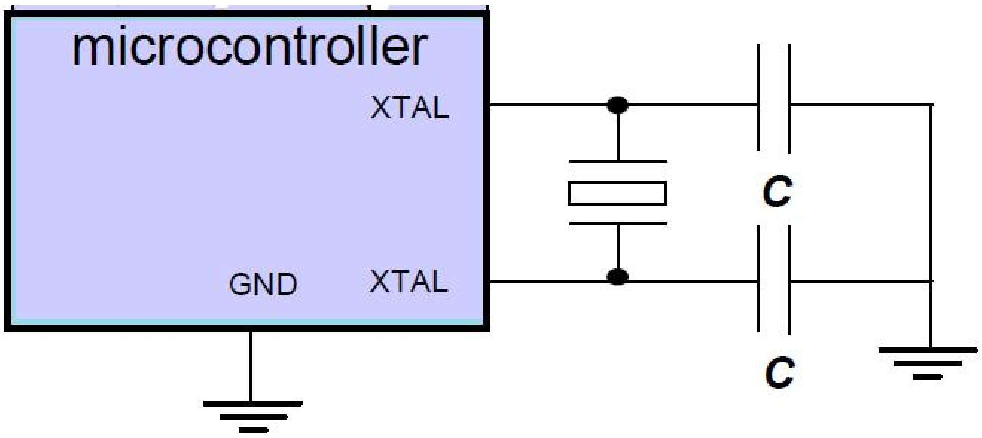


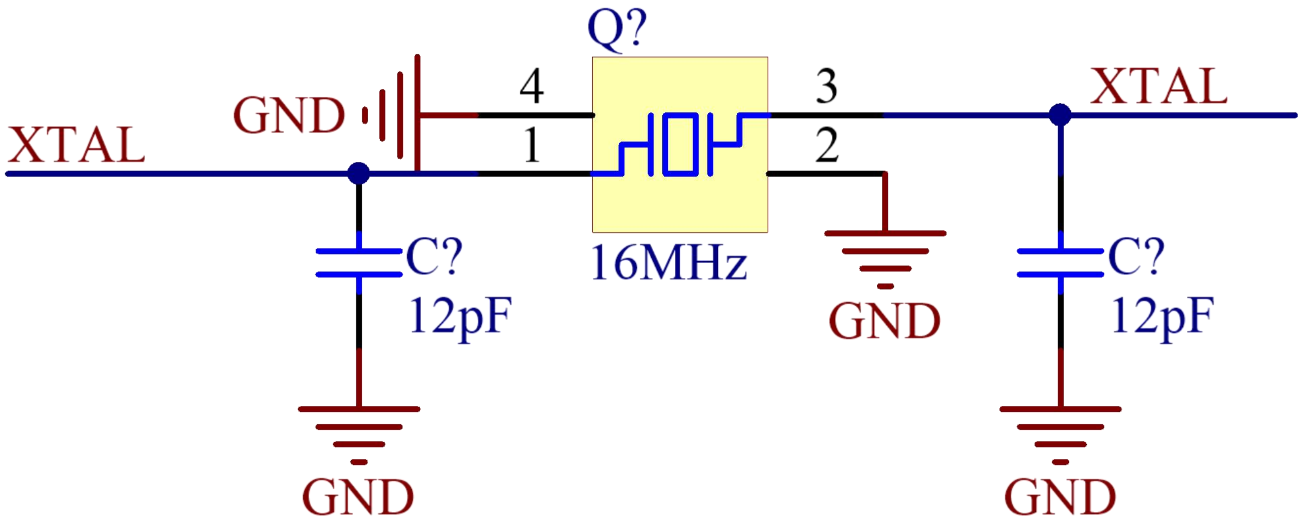
Crystals may be used to generate a popular form of oscillator circuit knows as a **Pierce oscillator**.

Crystal oscillator

- A variant of the Pierce oscillator is common in the MCU.
- To create such an oscillator, most of the components are included on the MCU itself.
- The user of the MCU must generally only supply the crystal and two small capacitors to complete the oscillator implementation.

How to connect a crystal to an MCU





Oscillator frequency and machine cycle period

- In the original members of the 8051 family, the machine cycle takes 12 oscillator periods
- Infineon C515C: 6 periods
- Dallas 89C420: 1 period
- P18F8722: 4 periods

Keep the clock frequency as low as possible

- Many developers select an oscillator/resonator frequency that is at or near the maximum value supported by a particular device
- This can be a mistake:
  - Many application do not require the levels of performance that an MCU can provide.
  - The electromagnetic interference (EMI) generated by a circuit increases with clock frequency
  - In most modern MCUs, there is almost linear relationship between the oscillator frequency and the power-supply current. As a result, by using the lowest frequency necessary it is possible to reduce the power requirement: this can be useful in many applications.
  - When accessing low-speed peripherals (such as slow memory or LCD displays), programming and hardware design can be greatly simplified, and the cost of peripherals components can be reduced

Keep the clock frequency as low as possible

In general, you should operate at the lowest possible oscillator frequency compatible with the performance needs of your application.

Stability issues

- A key factor in selecting an oscillator for your system is the issue of oscillator stability.
- In most cases, oscillator stability expressed in figures such as ‘+/- 20 ppm’: 20 parts per million
- To see what this mean in practice, consider that there are approximately 32 million seconds in a year. In every million seconds, your crystal may gain (or lose) 20 seconds. Over a year, a clock based on a 20-ppm crystal may gain (or lose) about 32x20 second, or around 10 minutes

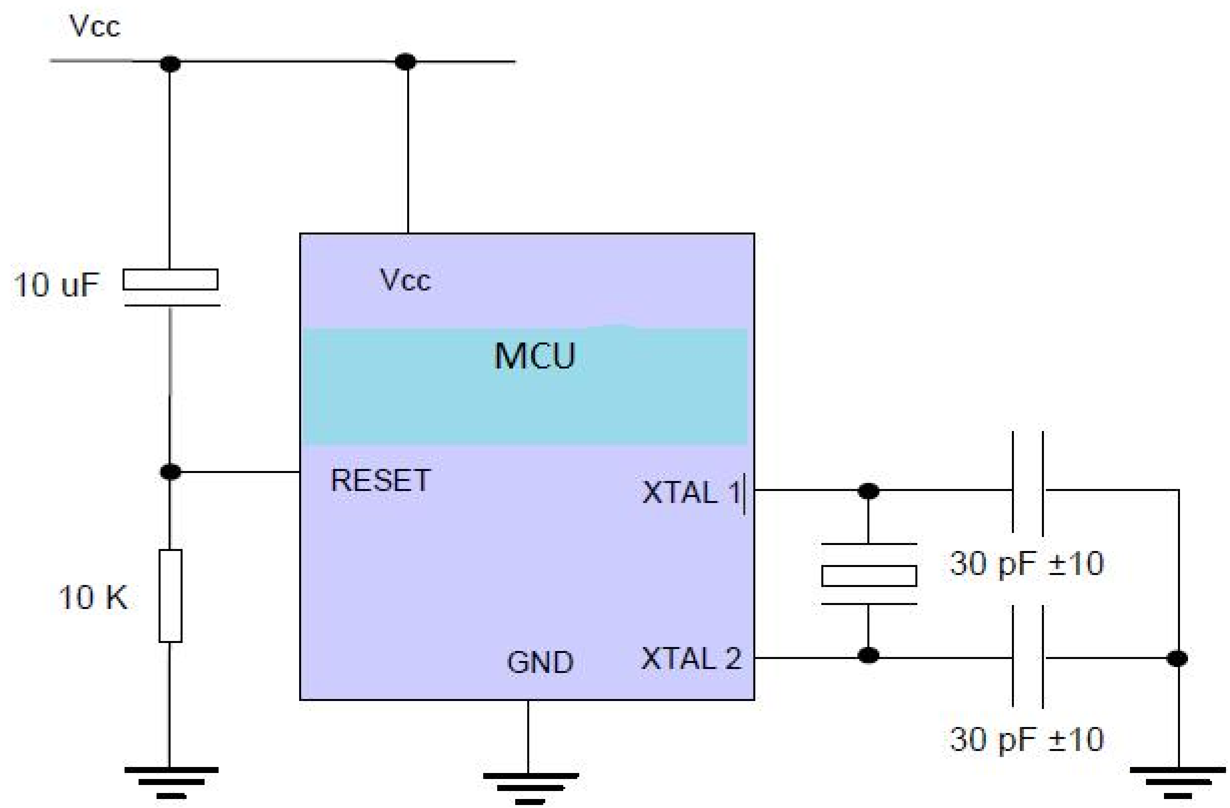
Improving the stability of a crystal oscillator

- If you want a general crystal-controlled embedded system to keep accurate time, you can choose to keep the device in an oven (or fridge) at a fixed temperature, and fine-tune the software to keep accurate time → this is impractical
- Temperature compensated crystal Oscillator (TCXO) are available that provide - in an easy to use package - a crystal oscillator and circuitry that compensates for changes in temperature
  - +/- 0.1 ppm – no more than 1 minute every 20 years
  - TCXOs can cost in excess of $100 per unit.
- One practical alternative is to determine the temperature-frequency characteristics for your chosen crystal, and include this information in your application

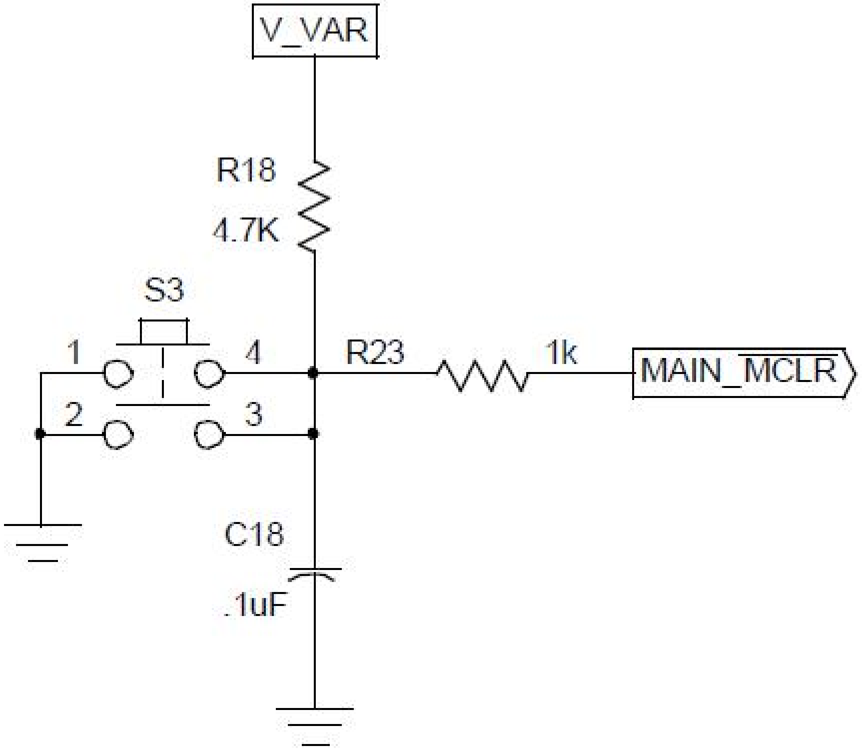
Reset circuits

- The process of starting any MCU is a non-trivial one.
- The underlying hardware is complex and a small, manufactured-defined ‘reset routine’ must be run to place this hardware into an appropriate state before it can begin executing the user program. Running this reset routine takes time and requires that the MCU’s oscillator is operating.
- An RC reset circuit is usually the simplest way of controlling the reset behavior.

Reset circuits



Reset circuits



More robust reset circuit