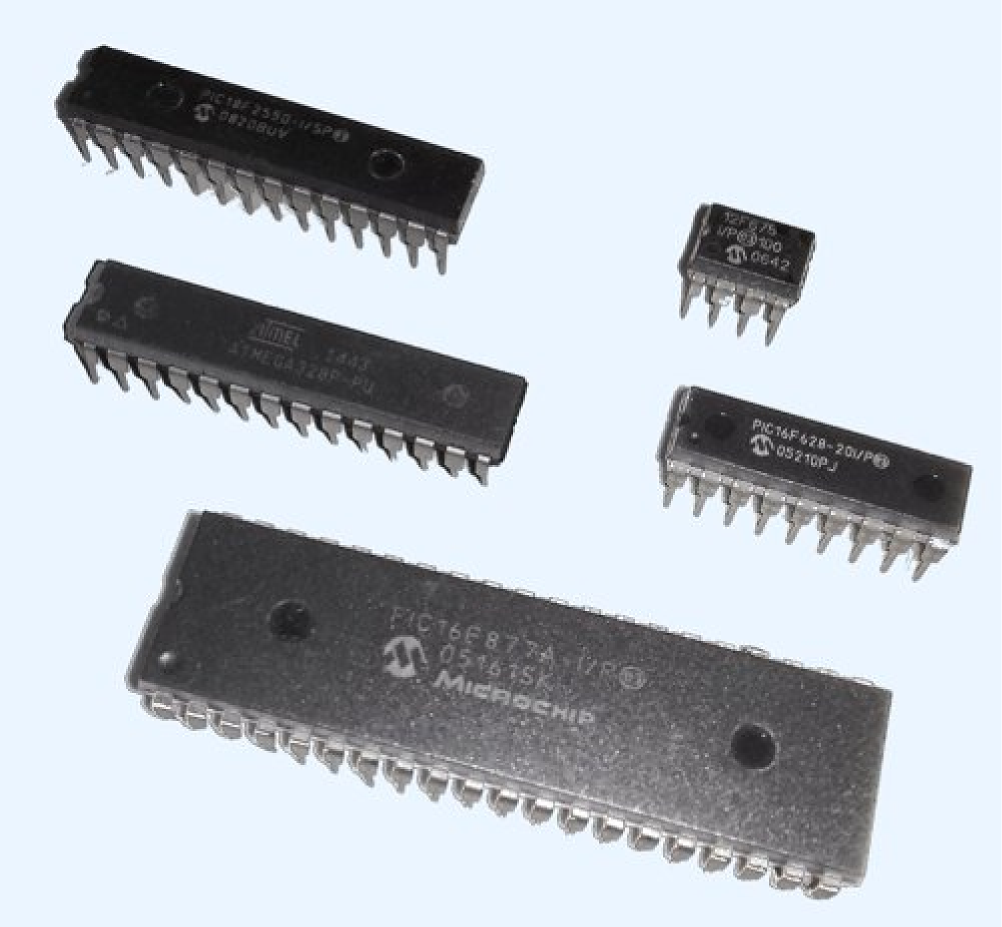
Best Microcontroller Projects"

Here you can find microcontroller Projects and Tutorials designed by me...a Professional Electronics Design engineer. Find out :

- What they are,
- What's inside them,
- Why you need them,
- How to use them in your projects.

Here are 5 different microcontrollers:



Clockwise from top left is 18F2550, 12F675, 16F628, 18F877A, AtMega328P.

These chips are packaged in standard PDIP (Plastic dual In Line) formats but you can also get them in surface mount formats. PDIP is easier for breadboarding though.

The AtMega328P is used in the Arduino Uno R3; This one is comparable to the PIC 18F2550 (except the 18F2550 has built in USB). As you can see they come in many different shapes, sizes and abilities (there are even PIC pin head versions in SOT packages!)

Once you add one of these re-programmable elements to your project you can increase the functionality of your projects a hundred fold. They are packed with lots of different hardware modules that make creating projects far easier.

The smallest one here (12F675) has 2 Timers (one 16 bit and one 8 bit), 4 Analogue inputs (10 bit), an internal oscillator and an analogue comparator along with the standard memory and RAM. All the rest have much more (even so you can do a lot with the 12F675).

For instance:

- You can add an LCD display screen (simple text display output)..

- You can time events easily from the very long: hours and minutes, to the very short: 100's of microseconds.
- You can measure analogue inputs for whatever sensor you need to check, humidity, temperature, pressure.
- You can connect a GPS unit to detect your location.
- You can connect a radio module for short range comms e.g. Bluetooth or a dedicated 433MHz comms system or even connect to a Ziggbee network.
- You can also control servo motors or a stepper motors to make robots and interactive systems.
- Much more!...

You can use microcontrollers for many, many different applications and all you need to do is learn how to program them using the information on this site.

A microcontroller is a circuit element that is completely self-contained, with many hardware blocks such as counters, comparators, even ADCs built in, so there is no need to wire up too much external hardware to make useful projects.

These days you don't even have to use an external crystal for the clock as you can turn on the internal oscillator!

Microcontroller Projects

There are many **FREE** projects on this site (with source code in C) that you can use to learn about microcontrollers including an ultrasonic distance meter, an led matrix driver, and more.

Each Project includes source code, description and schematics which you can use as a basis for starting your own projects or just use them stand-alone - to obtain the source code - just sign up for the free newsletter in the forms on any page..

Just Starting out ?

Check out the tutorial section where you'll find tutorials on building the circuits, device programming, prototyping and using high level languages.

Need some extras ?

The tips and techniques section gives information on designing specific circuits that either are difficult or that save you time and effort.

Did you know ?...

...98% of the processors used today are *microcontollers* - that means for every 2 desktop computers you can see in an office there are 98 others that you don't see!. They are in everything toasters, microwave ovens, security alarms, mobile phones, petrol pumps, jogging machines, key fobs... the list is endless.

The reason is simple - **reprogrammability** - they are effectively reconfigurable hardware blocks. This is important as you don't have to re-wire the circuit when you want a different operation from an existing circuit. You can change their operation very quickly - even in circuit!

All of them have at least one built in timer, often several, as well as other goodies such as comparators, PWM modules and ADCs and more. These are all internal hardware units making it simple to create almost any project you can think of. All you need to do is decide what to use, and then activate the appropriate modules.

On this site you can learn how to use these compact devices to create any type of project you can dream up, and you don't have to work in low level assembler - all the projects use high level C Programming which makes it easier to update and understand what is going on.

The navigation bar shows you some of the projects, all of which include **free C code and description**.

Microchip and Atmel (Arduino) are very popular

There are many different types of microcontroller from many different manufacturers and two popular types are from Microchip^{TM} and Atmel^{TM}: In fact Microchip has bough Atmel (2016) to broaden their product range - but you won't see any difference yet they still look like separate companies.

Microchip^{TM} (Industry popular PIC range)

Microchip^{TM} specialises in supplying almost bespoke solutions (microcontrollers) for a problem and creates hundreds of different variants. The idea is that you create a solution to a problem and you want to cost reduce it so what you really want to do is chop out all the hardware that you do not need.

There are many variants with added built-in modules such as a GLCD lcd driver, a motor controller, a capacitive touch sensor etc. So you choose a device based on what hardware you want to control. You can even get 6 pin SMD (SOT-23) microcontrollers!

On this site only a few devices are used 12F675, 16F88, 16F877A, 18F2550 because these devices offer a broad range of internal peripherals found in almost all in the devices in the same family (12F,16F 18F). You can learn here about these devices and be able to move easily to another, since the internal hardware is controlled in the same way.

There are also examples of using Arduino for building projects wit the main device in these being the AtMega328p which is broadly similar to the 18F2550.

Atmel^{TM} : Made extremely popular by the Open source Arduino^{TM} group.

You have probably heard of the Arduino^{TM} which is not a microcontroller but rather an open source movement that uses Atmel (and other) microcontrollers to offer a simplified way of connecting hardware (shields) and control devices (software). Because the hardware and software is open source many people have contributed free software so it makes it easy to quickly setup and create something.

Be warned though that just because you can use other software and connect it quickly may not mean that you understand it - and that means that any bugs in the software are up to you to find and correct.