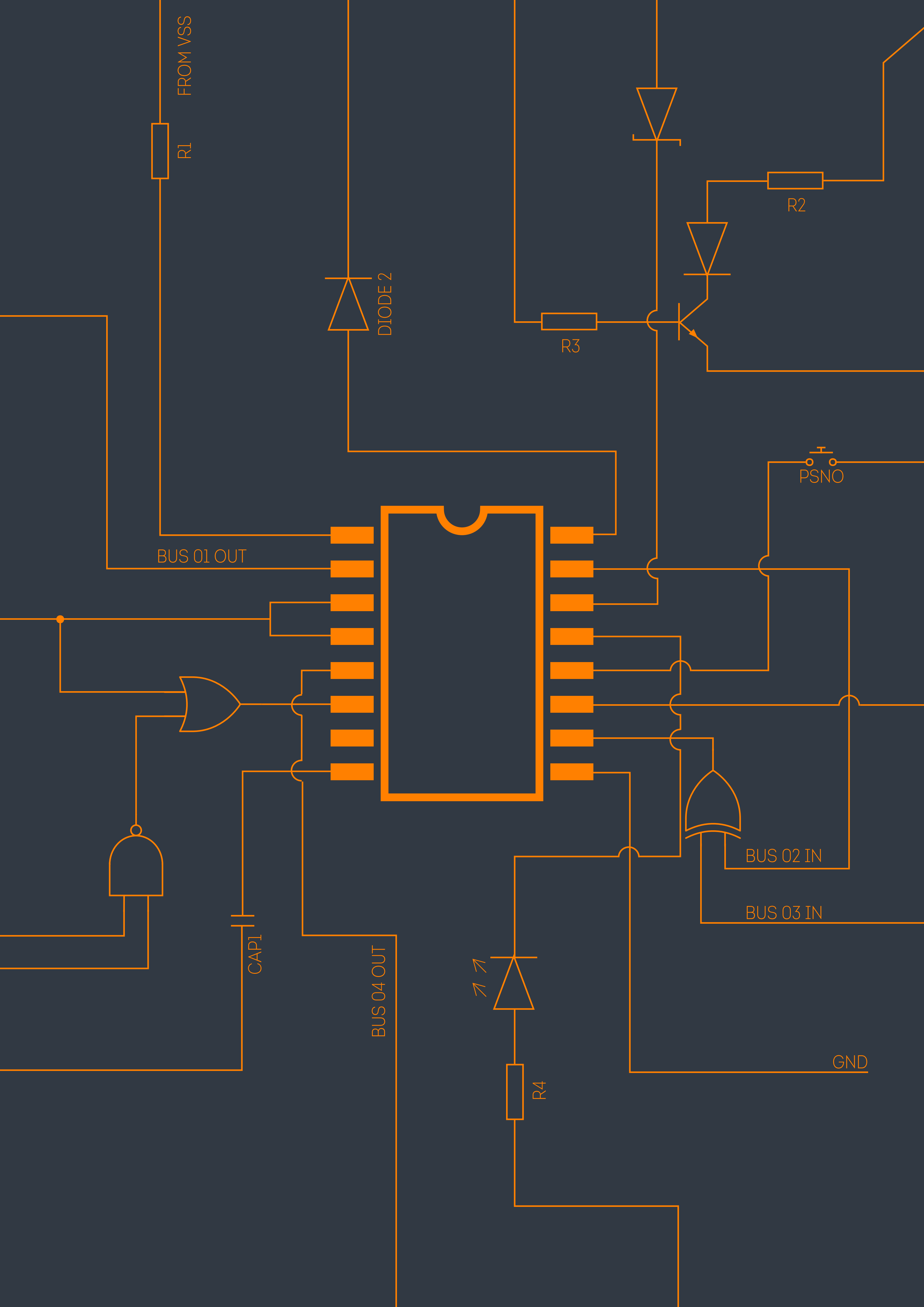

FROM VSS
R1
DIODE 2
R2
R3
PSNO
BUS 01 OUT
BUS 02 IN
BUS 03 IN
CAP1
BUS 04 OUT
R4
GND

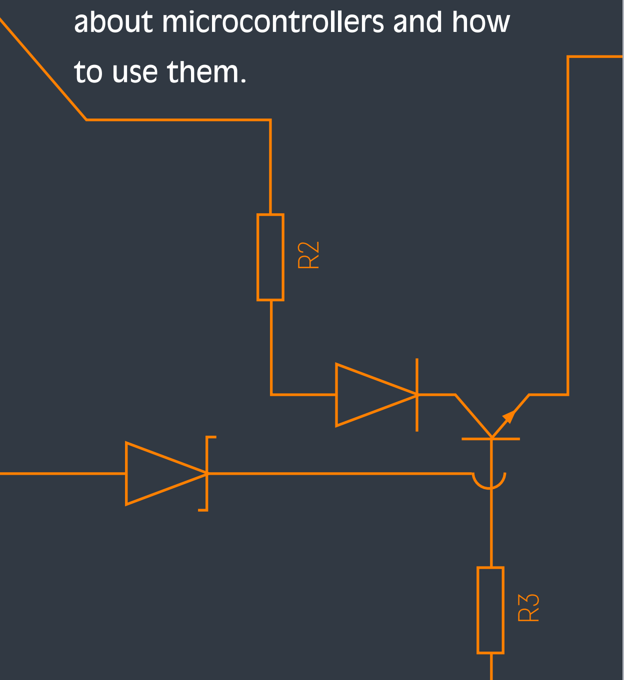
MICROCONTROLLERS

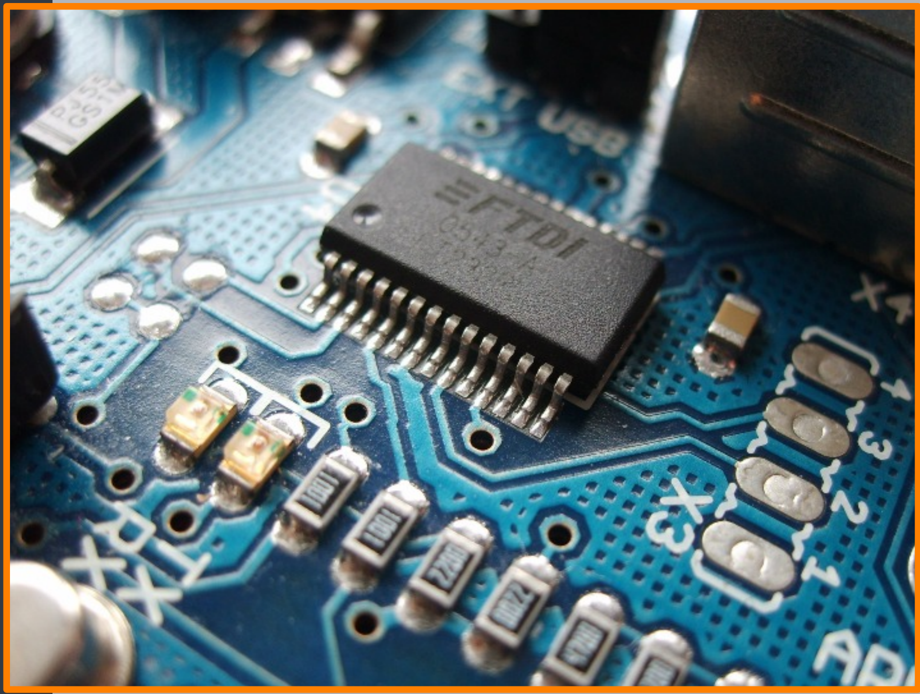
Microcontrollers are taking over everything, you just haven't noticed. You may have heard of the Internet-of-Things, or seen TV commercials advertising how you can control various devices with your smart phone. All these devices involve microcontrollers.

Microcontrollers - sometimes called PIC (Peripheral Interface Controllers) or PLC (Programmable Logic Controllers), are all small micro-processors capable of following a human-made programme. They can handle signals from various inputs and activate a number of output devices as defined by the programme.

If the programme doesn't respond the way you want, or you want to change how the device works, you can simply amend your programme.

In this unit, you will learn more about microcontrollers and how to use them.





- I can describe the advantages to using a microcontroller.
- I can identify products that may use a microcontroller.
- I can describe the Universal System in context of a microcontroller.
- I can identify different symbols used in a flowchart.
- I can sketch a flowchart using correct symbols.
- I can manually code a programme for a microcontroller.
- I can explain the term: SUBROUTINE
- I can explain the term: ALU
- I can explain the term: BUS
- I can explain the term: RAM
- I can explain the term: ROM
- I can explain the term: EEPROM
- I can explain the term: FOR...NEXT LOOPS
- I can explain the term: CLOCK
- I can explain the term: INFINITE LOOPS

WHERE, WHEN, WHY, HOW

It was during 1970 and 1971 when Intel was working on inventing the world's first microprocessor, that Gary Boone of 'Texas Instruments' was working on quite a similar concept and invented the microcontroller. Boone designed a single integrated circuit chip that could hold nearly all the essential circuits to form a calculator - only the display and the keypad needed added.

Surprisingly, this exceptional breakthrough in the field of electronics was given a rather mundane name of TMS1802NC; however, the device wasn't ordinary. It had five thousand transistors so it was possible to program it to perform a range of functions. The microcontoller was born...

ADVANTAGES

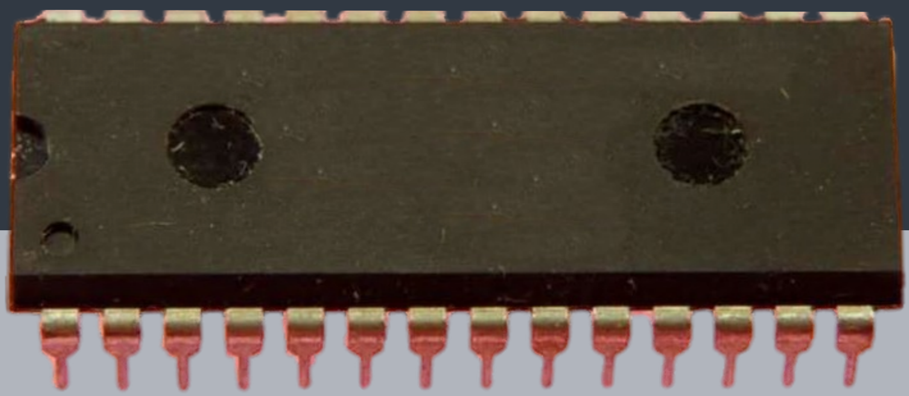
1. Microcontrollers are relatively inexpensive to buy - certainly cheaper than buying lots of discrete digital logic components and wiring them up.
2. Microcontrollers are tiny and can fit into almost any product.
3. Microcontrollers can be connected to the internet - meaning you can interface with them from anywhere in the world.
4. Don't like how the product is working? Simple. Change the code...

DISADVANTAGES

1. Internet connected microcontrollers can be vulnerable to people wanting to hack the code...
2. You do need to learn how to code - if you are an electrical or mechanical engineer, this is another skill you need to become familiar with and some code can get very complex.
3. Microcontrollers are vulnerable to static shocks (called a 'frazzle') or from voltage spikes. This can make the system go haywire!
4. Choice. There are so many different microcontrollers with different features. You need to know your stuff to choose the right controller...

Microcontrollers are fantastically small and have lots of different components built in. Each of these components has a particular function required to make the processor work.

You need to understand each of these component areas of the microcontroller so that you can make effective programmes. A list and description of the components is shown below.



ALU

Arithmetic Logic Unit is the part of the microcontroller that will process the instructions of the code. The ALU coordinates between all the other parts of the controller, such as reading/writing to RAM and sending instructions to the output controller

RAM

Random Access Memory is a temporary storage area. This can be used to name IO ports, remember numbers or data-log information coming in. All data is lost when the power goes off...

CLOCK

The clock sends a 'pulse' to the ALU and this keeps all the components 'in-time', processing one instruction at a time. The clock in modern PC is often above 3Ghz. A microcontroller doesn't require such processing speed and is often between 3Mhz and 30Mhz.

ROM

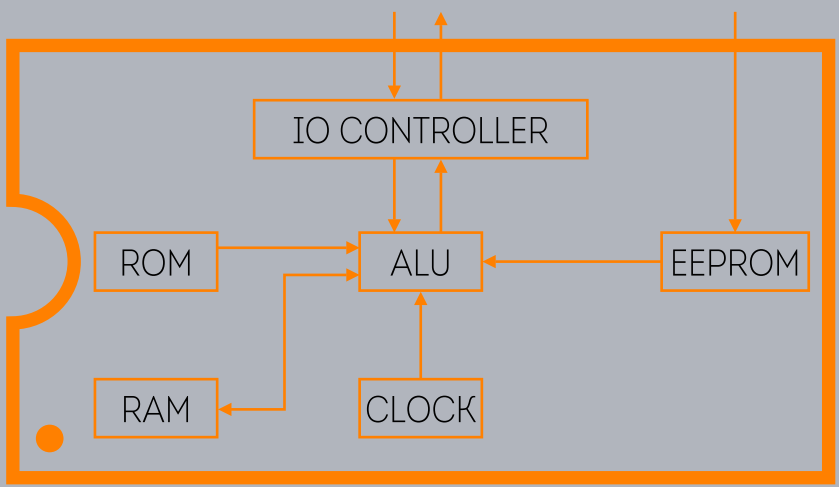
Read Only Memory contains a base set of instructions that allow the microcontoller to function. The microcontroller will read this memory when it first switches on and will then know how to read the user-made programme.

I/O CONTROLLER

The IO controller is responsible for all the pins of the chip that connect to the ALU. It will ensure input signal are sent to the ALU correctly and that any signal sent from the ALU as an output has enough voltage to be received by another component.

EEPROM

Electronically Erasable Programmable Read Only Memory is where the user programme is stored. This cannot be written over by the ALU, but by connecting the microcontroller to another computer, the user can delete or amend the programme.



The best way of thinking about any engineering problem is by using the 'Universal Systems' approach. This means thinking about all the possible input signals, think about the desired output is and what conditions are required before a particular output happens.

A system is a collection of parts that performs a function. A systems diagram is a representation of how a system will work. These are often used during the design process to generate ideas for electronic systems.

A simple systems diagram contains three boxes:

Input: the input starts the system, such as a switch or sensor.

Process: the mind of the system, which considers the inputs and decides what to do.

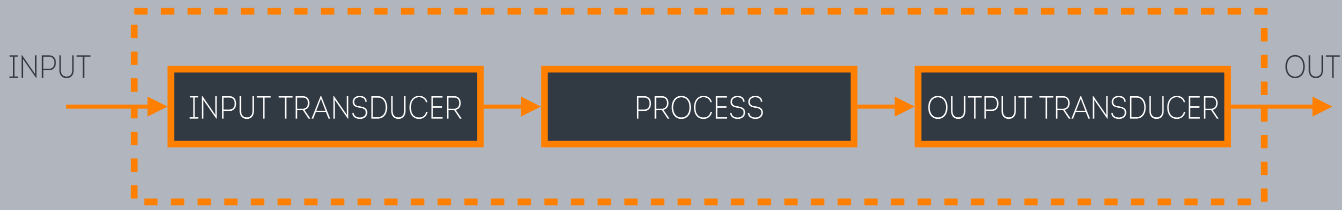
Output: the system's response to being activated, for example a flashing light or a buzzer.

Universal systems diagrams are not only used by designers and engineers, but by scientists, bankers and business leaders.

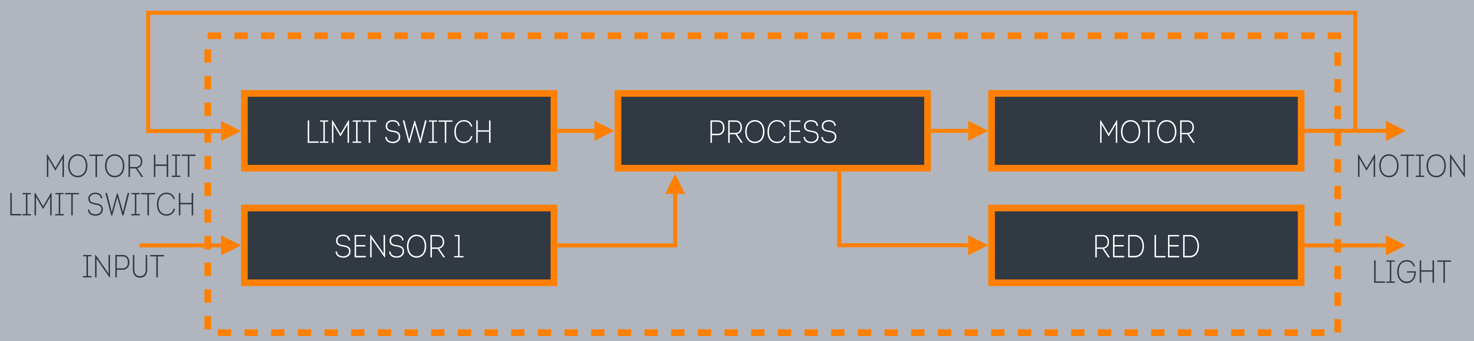
SOMETHING IN, SOMETHING OUT.

THE UNIVERSAL SYSTEM

BASIC UNIVERSAL SYSTEMS DIAGRAM



MORE DETAILED UNIVERSAL SYSTEMS DIAGRAM



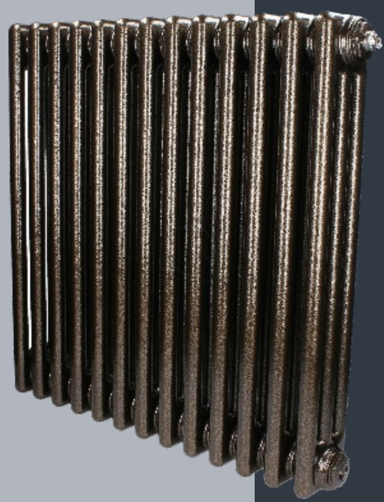
DESIGN CHALLENGE: VENDING MACHINE

Vending machines are common items in shops, schools and cinemas. They have a number of inputs, but only one major output - your chocolate, crisps or juice. (Don't worry about any returned money). Create a universal systems diagram for a vending machine, following the specification below.

DESIGN SPECIFICATION

- A number pad for making a selection
- 6 motors, one for each product
- A sensor to detect money inserted
- A sensor to detect when a product has fallen into the collection tray

DESIGN CHALLENGE: CENTRAL HEATING



It is hard to imagine not having central heating in your home - but that was the case not so long ago. Central heating must keep a room at a particular temperature but also can be turned up or down depending on a user. Create a universal systems diagram for central heating, following the specification below.

DESIGN SPECIFICATION

- A sensor to detect if the room is too hot or too cold
- A number pad for increasing the temperature
- A heating element to heat the water

DESIGN CHALLENGE: TIMER OUTLET

Going on holiday? Criminals can often burgle houses when they think the owners away. People use a range of technologies to protect their homes, including using automatic timers for lights in the house. Create a universal systems diagram for an automatic timer, following the specification below.

DESIGN SPECIFICATION

- A sensor to detect if it is dark
- A timer
- An override switch
- A relay to turn on a high powered circuit

DIGITAL OR ANALOGUE

Signals can be either Digital (sometimes called ‘Binary’), or a Analogue. A microcontroller needs to be able to receive signals from both digital and analogue input transducers and also run digital or analogue output transducers.

You need to know about the differences with these signal types and be able to programme a microcontroller to process them.

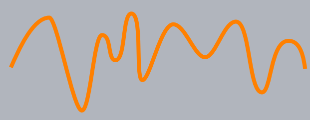
Here, we look at the differences.

Advantages:

- Simple to transmit through the air (radio), light (fibre-optic) or electrically.
- Easy for microcontrollers to process.
- Direct Current (DC) is digital
- Many inputs are simple ‘on’ or ‘off’

Disadvantages:

- Special coding required to mimic an analogue signal.



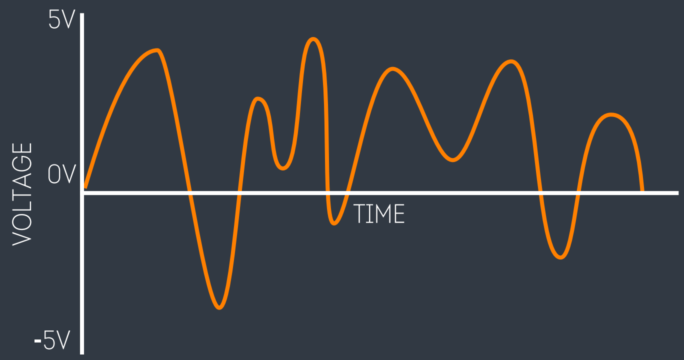
Advantages:

- Many sources of data are naturally analogue - light levels, sound and temperature can be mapped to a graph.
- Many outputs are analogue in nature - light level, sound or temperature.
- Alternating Current (AC) is analogue.

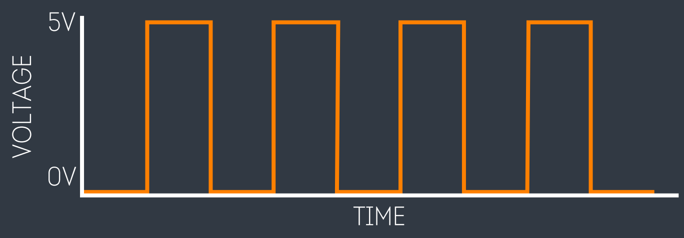
Disadvantages

- Analogue signals must be converted into digital for a microcontroller to process.

Analogue signals are in a wave format, meaning they can be a range of levels, usually both positive (+) and negative (-) and this can change quickly over time.



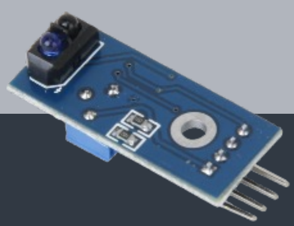
Digital signals are either on or off - really simple. This can change over time too. Microcontrollers like digital signals - they are simple to process and understand.



SENDING THE RIGHT SIGNAL

All input sensors are described as ‘Input Transducers’. These devices will react to specific environmental conditions or data coming from a different source. Input transducers can be neatly broken into two camps: Digital transducers and analogue transducers.

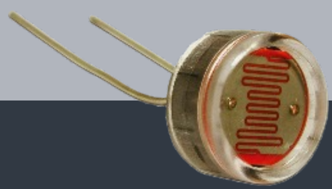
You should be able to select the correct type of input transducer for a specific system. The eight sensors shown below are some of the most common, but there are many more types available to designers and engineers.



IR SENSOR

IR - or InfraRed - sensors are analogue. IR sensors are used in remote controls for TVs and other devices. IR can also be used to measure distance, by emitting a beam and reading how much is bounced back.

A beam of infrared light is emitted at a particular frequency that is then read by the receiver unit.



LDR

Light Dependant Resistors are analogue devices. The resistance level of an LDR will change depending on the level of light falling upon it.

These are typically used in voltage-divider circuits, with a potentiometer used to improve accuracy.

THERMISTOR

Similar to an LDR, the resistance of a Thermistor will change in proportion to its’ temperature.

These are also used in voltage-divider circuits, with a potentiometer used to improve accuracy.

SOUND SENSOR

Sounds sensors are not the same as microphones - they are often not as sensitive or as accurate for reproducing sound.

They are used however for detecting specific frequen-cies of sound and can be particularly useful in securi-ty systems, for instance lis-tening for the sound of a drill trying to break into a safe...

MICROSWITCH

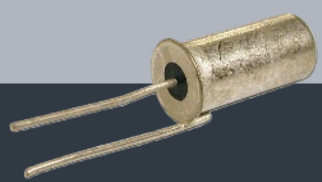
A microswitch is a simple Single Pole, Single Throw (SPST) switch device.

When something presses on the long metal arm a small button is pushed down and closes the circuit. Microswitches are used everywhere, from refrigerator doors to robot arms.

REED SWITCH

A reed switch if used to de-tect if a magnet is present. When a magnet is close to a reed switch, the thin metal plates make contact and close the circuit.

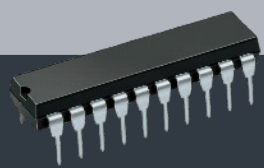
Reed switches are popular in security systems as they can be very sensitive.



TILT SWITCH

A tilt switch is a digital device that can detect if the component is at an angle.

A tilt switch is very simple, relying on a small metal ball rolling in a chamber and making a connection between two contacts.



ANOTHER CONTROLLER

Microcontrollers are digital devices and it is possible to a have a series of controllers working together to process various things.

This technique is used in very complex systems, or where part of a system could get damaged and can be swapped out with a replacement

GETTING THE RIGHT RESULT

We build machine to do stuff for us. We've created microcontrollers to make decisions based on a programme for us. What we need are devices that can put decisions into actions. Output Transducers are devices that physically actually do something.

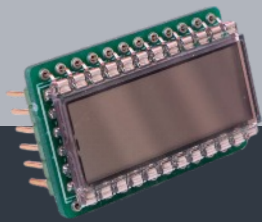
These output transducers can also be either digital or analogue. Microcontrollers have some special techniques for dealing with some analogue outputs, such a DC motors and light bulbs. Below are eight useful outputs, but there are many more.

DC MOTOR

A DC motor is an analogue device - varying the voltage will change the RPM (revolutions per minute). DC motors come in all sizes, speeds and powers - very few DC motor can be powered by the output voltage of a microcontroller. It is common to use a NPN transistor or relay to supply the required voltage.

BUZZER

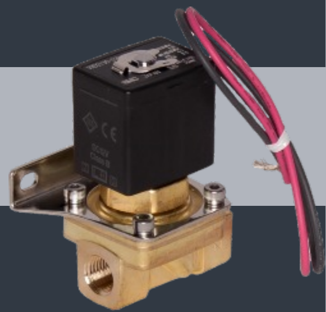
A buzzer is very annoying! Their screeching sound can anger the most patient person. Buzzers are analogue devices often used in small alarms. Changing the supply voltage will change the tone and volume of a buzzer. Microcontrollers can sometimes supply enough power to activate a buzzer. Otherwise and NPN transistor amplifier can be used.



SCREEN

A variety of screens are available, from small, simple LCD (liquid crystal display) to full colour, bright, sharp LED. (More money you've got, the better screen you can buy)

A screen is a digital device and a number of output pins from a microcontroller. Screens will have their own controllers that your microcontroller can 'talk' to.

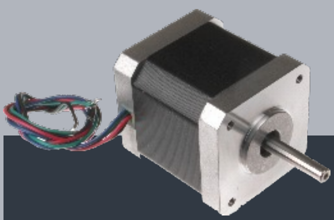


CONTROL VALVE

A control valve is used to control fluids of gases from flowing. Some control valve a digital - fully open or fully closed - and some are analogue, allowing a microcontroller to partially open a valve to set a specific flow rate. Control valves need a high-powered signal, meaning the microcontroller is usually attached to a NPN transistor amplifier or relay circuit.

SERVO

Servo-motors are a digital motor type that are very fast and have a very high torque (pulling power). It is possible to control the position a servo will turn to, but this is quite complex to build. Servos are often used in expensive remote-control models because of their pulling power. Servos are also good in products that vibrate a lot. They are however, very expensive.



STEPPER MOTOR

A stepper motor is a very useful type of digital motor. An electronic 'pulse' will make the stepper motor move a set angle. By sending a stream of these pulses, you can control how far the motor will rotate. You can also 'hold' the position of the motor. Stepper motors don't quite have the torque of a servo, but are far cheaper to buy.

LEDS

Light Emitting Diodes are fantastic. They come in a range of colours, can be very bright, use a small voltage and can last far longer that traditional lamps.

LEDs are analogue; varying the voltage can change the brightness. Be careful though, too high a voltage will blow-up the LED...

RELAY

A relay is a component that uses a small magnetic switch to turn on a high powered circuit from a low powered circuit.

This is excellent for microcontrollers, which can only output a couple of volts. The microcontroller can be connected to an NPN transistor which in turn can activate the relay.

DESIGN CHALLENGE: HOVER BOARD

Hover boards don't actually hover, but they do allow a user to travel quickly on two wheels. By leaning forward or back a user can control their speed and by leaning side to side, steer in the direction they want. Create a universal systems diagram for the hover board and list the input and output transducers, following the design specification below.

DESIGN SPECIFICATION

- Four sensors. Forward, backward, left and right
- Two motors, one for the left wheel and one for the right
- A 'kill' switch. If a user is not standing on the device (falls off), the board stops

DESIGN CHALLENGE: TOY ARCADE

Toy 'grabber' arcade machines can be found in any place you find children. These machines are attractive to children and they will nag at their parents to play… Create a universal systems diagram for the toy arcade and list the input and output transducers, following the design specification below.

DESIGN SPECIFICATION

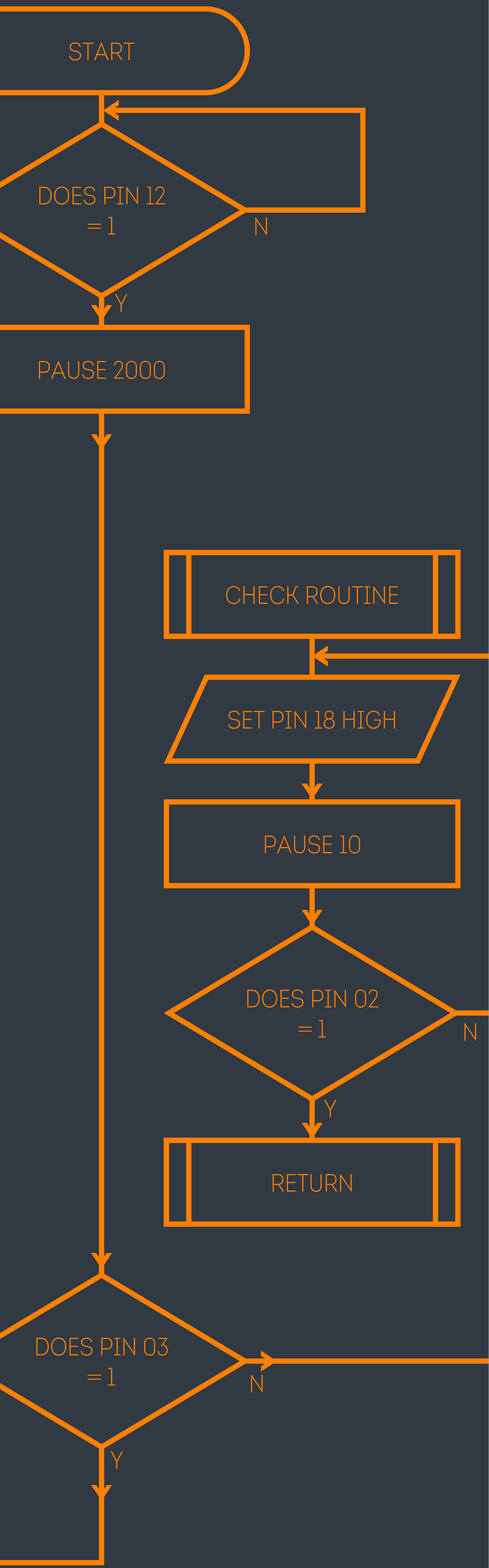
- Five controls (Move left, right, forward, back and 'grab')
- Four motors. (Left/Right) (Forward/Back) (Up/Down) (Claw open/close)
- A coin sensor, to start the game

DESIGN CHALLENGE: AUTO NIGHTLIGHT

Scared of the dark? An automatic night light can help. Many night lights will glow brighter if they detect movement and then gradually fade to a faint light. The nightlight will deactivate in the morning as the sun rises. Create a universal systems diagram for the auto nightlight and list the input and output transducers, following the design specification below.

DESIGN SPECIFICATION

- Six bright, low power light sources
- A way of detecting movement
- A way of detecting how bright or dark the room is.



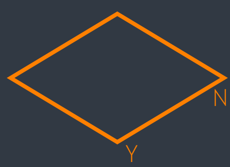
GO WITH THE FLOW | FLOWCHARTS

Flow charts are brilliant and can be used almost anywhere you want people (or machines) to follow a strict sequence of operations. They allow you to set output actions, ask simple questions (most questions can be broken down into simple yes or no answers...)

Designers and engineers use flow charts to help **plan** how a computer programme should run. It doesn't matter what type of microcontroller you are using or what programming language you are typing, the different symbols used with flowcharts can be attributed to specific lines of code.

Before writing a programme, creating a flowchart will help. You need to be able to create flowcharts and tie these to lines of code. The language you learn will depend upon your teacher (or your own interest, if you want to Google a specific controller (we're fans of Arduino and PICAXE chips)

'Terminator', used only at the start and end of a flowchart. Nothing else.



'Decision', used to check an input or test something in the memory of the microcontroller. The question has to be one where there is a yes or no answer.

'Internal Process', used to process a statement such as a time delay, or commit something to the memory of the microcontroller.

'Subroutine', used to call up another flowchart or return to the main flowchart.

'Set Output', used to set an output pin either high or low (on or off).

SPEAK IN TONGUES | CODING

Coding is the challenging part of using microcontrollers. Different controllers will use different languages.
Your teacher will tell you which language you will learn in class. Below is two examples, PBASIC (picBasic) and Arduino.

Code can be complex, and the longer the code, the more complex it will appear. Designers and Engineers use many ways to simplify coding. A very basic technique comes from Graphic Communication - using 'white space' between lines or groups of code to make it easier to follow and read.

PBASIC CODE

```
main:
                ' set name of the routine
high B.13
                ' switch pin 7 on
pause 1000
                ' wait for 2 seconds
low B.13
                ' switch pin 6 on
pause 1000
                ' wait for 1 second
goto main
                ' jump back to start
```

ARDUINO CODE

```
void setup() {
// the setup function runs once when you press reset
or power the board

  pinMode(13, OUTPUT);
 // initialise digital pin 13 as an output.

}
 // closes the setup function

void loop() {
// the loop function runs over and over again forever

  digitalWrite(13, HIGH);
// turn the LED on (HIGH is the voltage level)

  delay(1000);
// wait for a second

  digitalWrite(13, LOW);
// turn the LED off by making the voltage LOW

  delay(1000);
// wait for a second

}
 // closes the loop function
```

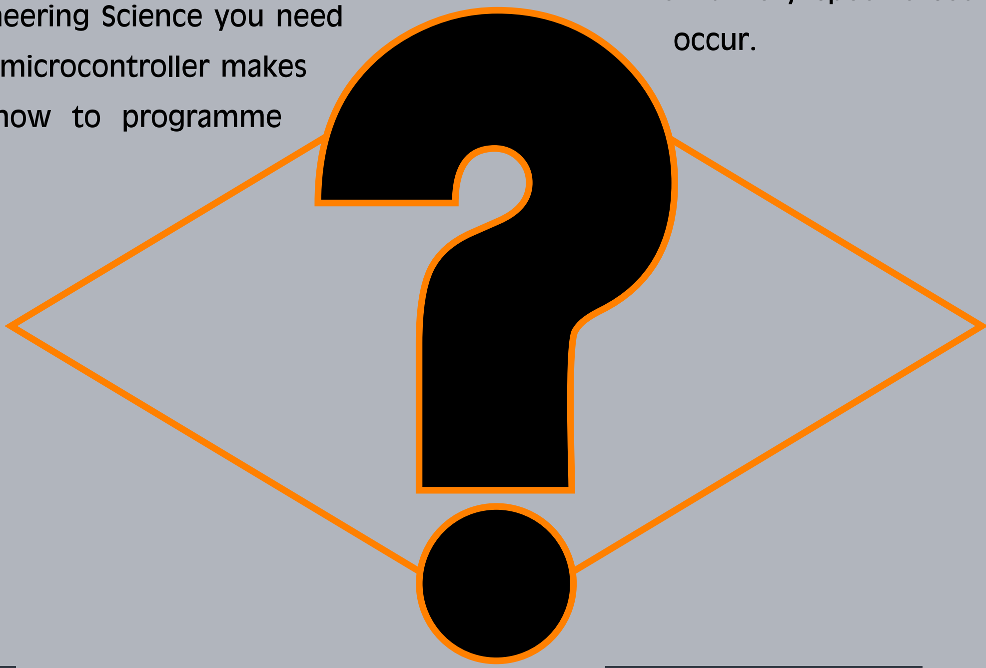
```
CalcI_Drive:
 IF (Ki <> 0) THEN
  SignBit =Err.BIT15
  TempDrive = ABS(Err)/20
  IF SignBit = 1 THEN TempDrive = TempD
  I_Edt = I_Edt + (TempDrive * DriveTime)
  SignBit = I_Edt.BIT15
  I_Edt = ABS(I_Edt) MAX 31000
  IF SignBit = 1 THEN I_Edt = I_Edt * -1
  TempDrive = ABS(I_Edt)/10
'   %Integral Drive based on total integrat
  TempDrive = TempDrive * Ki / (Ki_Scalar
  IF SignBit = 1 THEN TempDrive = TempD
 ELSE
'   Ki = 0 then reset total integrated error
  I_Edt = 0
  TempDrive = 0
 ENDIF
'   Update StampPlot and add to total driv
  DEBUG SDEC I_Edt,","             ' Send Su
  DEBUG SDEC TempDrive,","            ' Senc
  DriveTotal = DriveTotal + TempDrive
RETURN
CalcD_Drive:
 TempDrive = Err - LastErr
 signBit = TempDrive.BIT15
 TempDrive = ABS(TempDrive)/ DriveTime
 IF SignBit = 1 THEN TempDrive=TempDriv
 IF Kd = 0 THEN TempDrive = 0
 DEBUG SDEC TempDrive,","             ' Senc
 TempDrive = ABS(TempDrive) * Kd / Kd_S
 IF signBit = 1 THEN TempDrive=TempDriv
 DEBUG SDEC TempDrive,","             ' Senc
```

AH, DECISIONS, DECISIONS...

Microcontrollers are especially useful as they can be used to make decisions without a human. Microcontrollers can be relied upon to make decisions based upon their programme at anytime - 24 hours-per-day.

As part of Engineering Science you need to know how a microcontroller makes decisions and how to programme them to do so.

Microcontrollers can only make binary, 'Yes or no' decisions, so any questions you want to make need to be created using this way of answering. This can make for a long programme if you need something to happen when a very specific set of conditions occur.



PBASIC CODE

```
IF pin 1 = HIGH THEN
        {
                HIGH b.6
                PAUSE 1000
                LOW b.76
                PAUSE 1000
        }
        ELSE
        {
                HIGH b.7
                PAUSE 1000
                LOW b.7
                PAUSE 1000
        }
        END IF
```

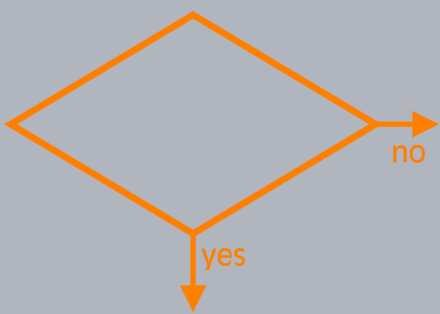
TYPES OF DECISION

A decision can be used to test an input for a signal. For instance; "is pin1 high?". This is a way of checking they physical environment.

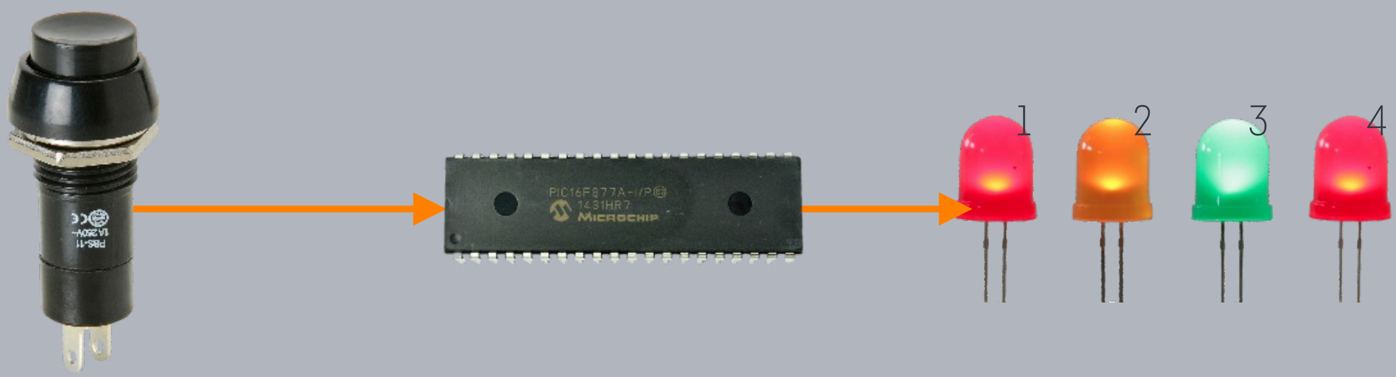
A decision can also be used to check a value in the memory of the microcontroller. For instance; "Does X = 15?". This is used to check memory in the microcontroller.

When using a flowchart, the diamond symbol is used. Two legs come from the diamond, one for 'yes', the other for 'no', A yes or no question is entered in the centre of the diamond.

You can choose which arm is marked yes and which is marked no.



OKAY, TRY THIS...



INPUT PROCESS OUTPUT

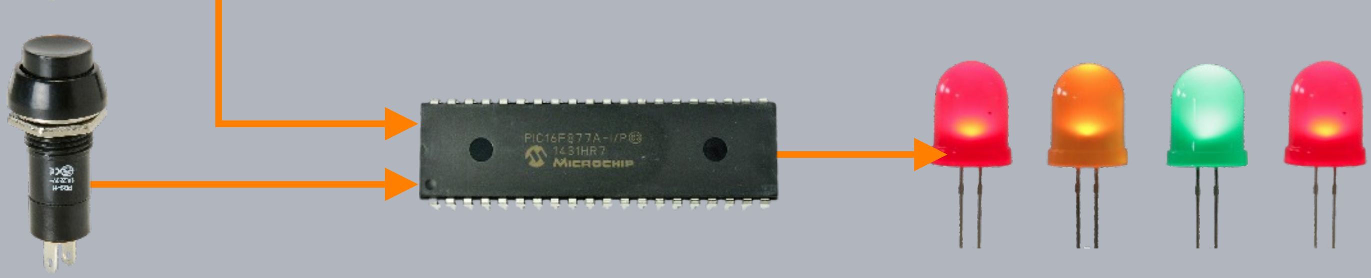
Design and make an 'LED chaser circuit'.

When a push-to-make button is pressed the LEDs should appear to flicker in one direction. When the button is pressed again the LEDs should appear to reverse direction.

DESIGN SPECIFICATION

- One push-to-make switch to the microcontroller.
- When the push-to-make switch is pressed, the LEDs flash in a different direction.
- The circuit loops forever.

Design a programme when a user can use one button to set how many times an LED will flash and a second button that will start the actual sequence.



INPUT PROCESS OUTPUT

DESIGN SPECIFICATION

- One push-to-make switch connected to the microcontroller for 'adding 1' to the memory every time it is pressed.
- Another push-to-make switch connected to the microcontrolle starts the LEDs flashing. They flash as many times as the value in the memory.
- When the LEDs have finished flashing, the memory should reset to zero.
- The circuit loops forever.

INFINITE LOOPS - INFINITE LOOPS - INFINITE LOOPS - INFINITE LOOPS - INFINITE LOOPS - INFINITE LOOPS - INFINITE LOOPS -

Something are better if they don't end, aren't they. Very few programmes have a complete end where nothing happens after it has completed its' function.

Most programmes will loop around and start again from the beginning. This is important as it allows machines to complete boring, repetitive tasks and free humans from doing the job.

Infinite loops work by not having a STOP terminator. Instead, the programme flows back to the very start. There are a number of good examples that use infinite loops, such as traffic-lights, cash machines and elevators.

You need to be able to be able to create flow charts and write programmes that use infinite loops.

DESIGN CHALLENGE

BRIEF

Road traffic is difficult to control. Whilst you have certainly seen traffic lights and cycle lanes before, you may not have considered the considerable design and engineering involve in keeping people safe.

In this scenario a simple T-junction is shown. The road has cycle lanes and all road vehicles should be given a stop (red) signal to allow cyclists a safe space to move.

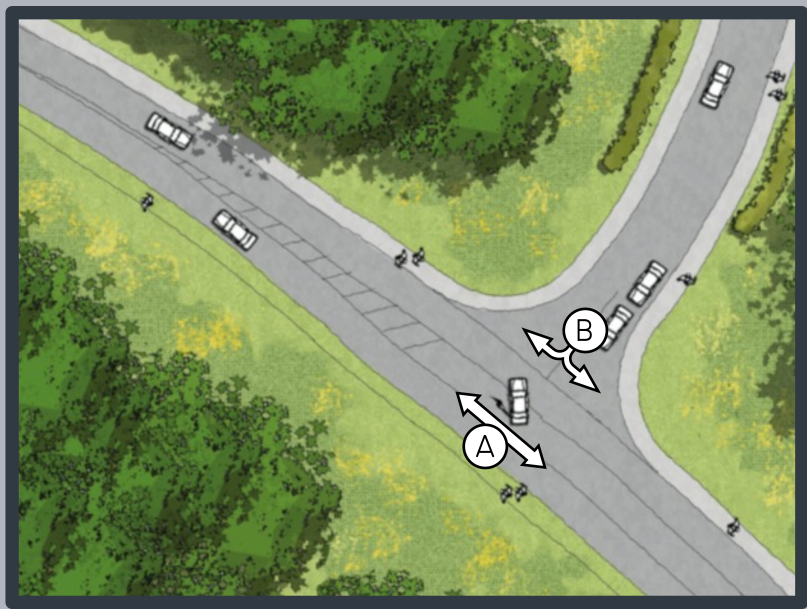
Design and simulate a traffic calming microcontroller control for the road shown below. You should create a systems diagram and write a flowchart and programme. A situation description and a detailed design specification are written below.

SITUATION

A technical sketch of the road junction is shown (left).

The main road flow is shown by arrow A.

Secondary road flow with arrow B.



Safety is critical. Pedestrians, cyclists and motorists need to be given clear instructions and plenty of time to react.

Traffic lights have a specific sequence. If you do not know this sequence, you will need to do some research.

You will also need to research timings used for pedestrians to cross roads.

DESIGN SPECIFICATION

- Main traffic is on road A. This should be kept moving as much as possible. When A is moving, lights for road B MUST be at RED.
- When B is GREEN, A MUST be at RED.
- Less often, both A and B must show RED and allow only the cycle light to show GREEN.
- If a pedestrian presses a button, A, B and Cycle lights must show RED.
- You may choose the timings. You may wish to conduct further research in this area.

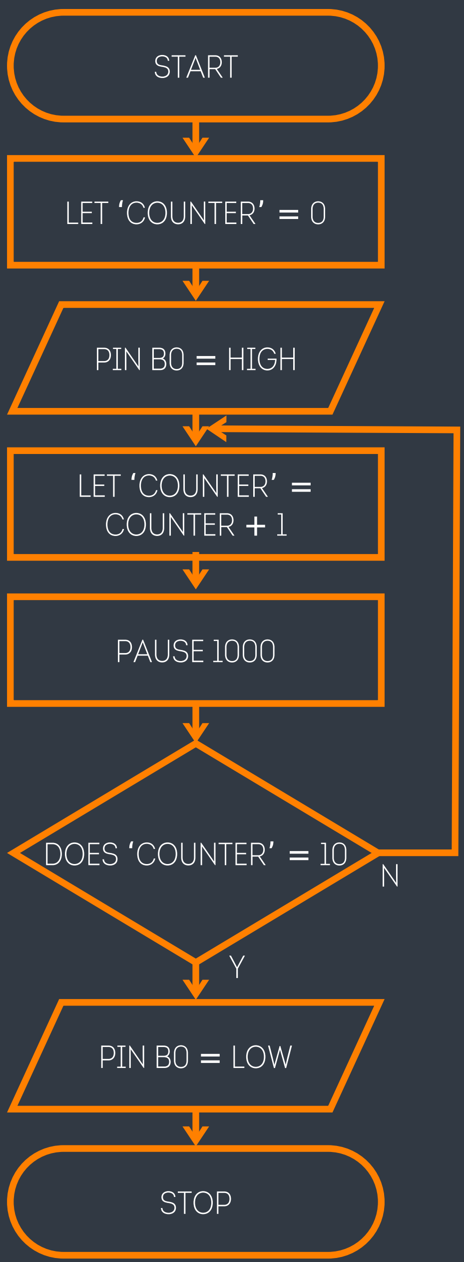
COUNTING UP - COUNTING DOWN

Getting microcontrollers to count stuff is really easy - and an important function to learn about. 'Finite' loops make a microcontroller measure whether a particular value has been achieved by counting how many loops the programme has made or how many times an input has been pressed. This can be useful when a particular task needs to be repeated a certain number of times.

To use a counter function, part of the microcontroller memory needs to be given to the process of remembering the 'count'. This memory can be named to make it easier for the programmer to remember.

On the example (left), the memory has been called 'COUNTER' and the value of '0' has been entered.

The programme then sets an output pin on and enters the counting part of the programme. The memory allocation is then told to recall the value in memory and add one to it. This new value is then stored in the memory. After a short pause (1 second), the microcontroller is asked whether the memory allocation has reached the number 10. If not, the programme loops round and another '1', is added to the memory. This continues until the value of 10 is achieved. The pin is then set low and the programme ends.



DESIGN CHALLENGE: AUTOMATIC DOOR

Automatic doors do make us lazy. However, you are tasked with designing one. The door should open if a sensor detects a person on either side of the door. Once a person is detected the door should open and stay open for a set time. The door should check no one is there before trying to close. Write a systems diagram, detail the input and output actuators, create a flow chart and write a programme to the specification below.

DESIGN SPECIFICATION

- Sensors to detect a person.
- One motor, connected to pins 5 and 6 of a push-pull driver.
- Counter to keep the door open for eight seconds.

DESIGN CHALLENGE: WINDOW WIPERS

Many modern cars have automatic window wipers that will activate as soon as rain is detected on the windscreen. These window wipers can also be switched on manually by the driver. Write a systems diagram, detail the input and output actuators, create a flow chart and write a programme to the specification below.

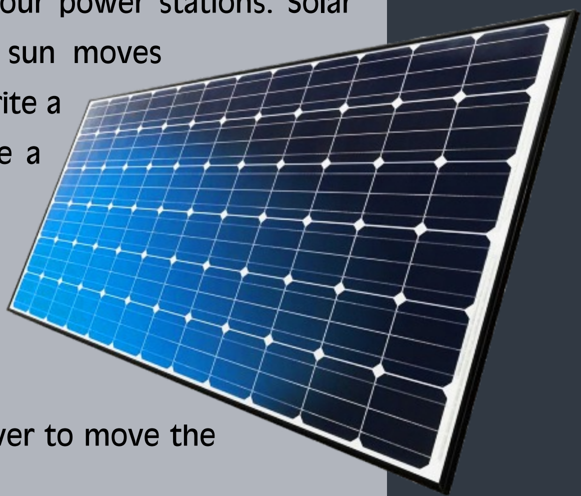
DESIGN SPECIFICATION

- Sensors to detect rain
- Switch to manually turn on
- One motor, connected to pins 5 and 6 of a push-pull driver to move wipers

---

DESIGN CHALLENGE: SOLAR PANELS

Solar panels are used to convert sun light into electricity. This can help us battle climate change by reducing our dependency on fossil fuels to power our power stations. Solar panels work best when pointed directly at the sun. As the sun moves throughout the day, the angle of the panels need to change. Write a systems diagram, detail the input and output actuators, create a flow chart and write a programme to the specification below.



DESIGN SPECIFICATION

- Sensors to detect the sun light
- A programme must determine where the sun is brightest
- One motor, connected to pins 5 and 6 of a push-pull driver to move the solar panels left or right

SUBROUTINES

Sometimes, smaller is better.

A subroutine is a small programme tasked with completing one small, specific job.

A subroutine can be 'called' by the main programme or another subroutine.

You need to learn how to plan, organise and programme subroutines.

A subroutine can be called as many times as is required.

Subroutines make it easier for designers and engineers to understand and modify part of a programme.

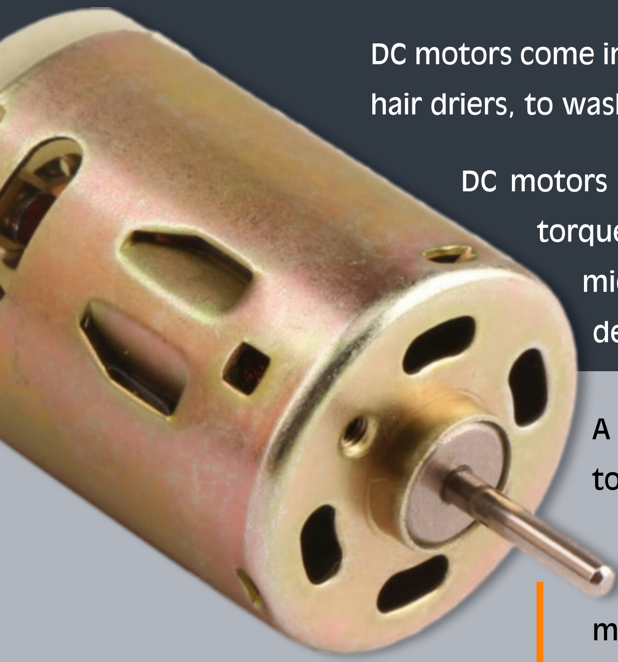
It is very common for a main programme just to be calling a series of smaller subroutines to do specific tasks.

Companies may have different design teams working on different subroutines only.

The symbol for a subroutine in a flow chart is:

Different programming languages call subroutines differently. Your teacher will show you how to do it in the language you are using.

CONTROLLING AN ELECTRIC MOTOR



DC motors come in many sizes and powers and are used in everything from toys and hair driers, to washing machines, drones and DVD players.

DC motors are analogue devices - they will move at different speeds (and torques) depending on the voltage they are supplied with. As microcontrollers are digital devices, a clever technique has been designed to allow a digital device to control the speed of a motor.

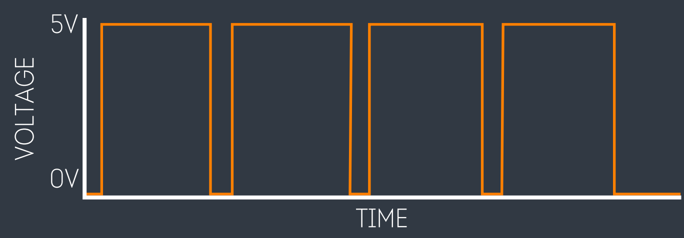
Microcontrollers use a technique called 'Pulse Width Modulation' (PWM) to control the speed of a motor. Essentially, turning it on and off very quickly!

A microcontroller works on a surprisingly small voltage - about 2.3v to 6v. This means that a microcontroller does not have the voltage to power most DC motors. A DC motor can also cause 'back electro-motive force' (Back EMF), that can damage the microcontroller.

Microcontrollers are usually attached to a hi-gain NPN transistor or a relay. This is then used to supply a higher voltage to the motor. You will learn more about powering high-powered devices in analogue and digital electronics section of this course.

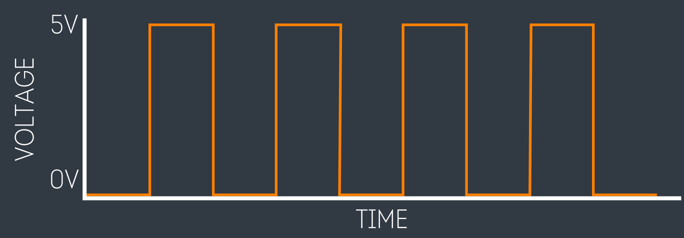
FAST

By turning the power to the motor on and off quickly, the motor receives an 'average voltage'. To make the motor spin just below full speed, the microcontroller will turn the motor on for longer than it turns off.



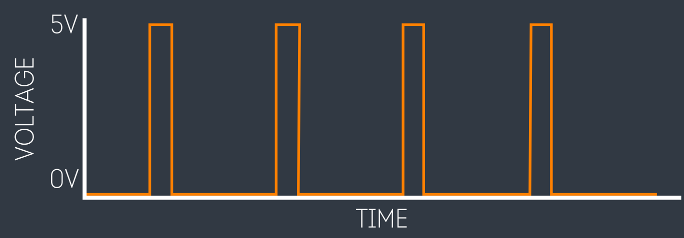
MEDIUM

If the power is on and off for an equal amount of time, the average voltage will be half the supply. This will move the motor at half-speed.



SLOW

By increasing the time when the voltage is off, the motor will move slower. You need to be careful - if the time the voltage is off is too long, the motor will stop...



DESIGN CHALLENGE

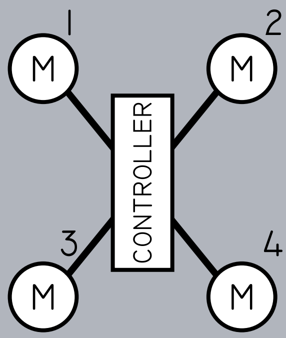
BACKGROUND

Drones have become a new tool for lots of industries and businesses. TV and film producers use them to record scenes for different shows. Emergency services use drones to help look for trapped people or different dangers. Architects and builders use them to plan new buildings.

FUNCTION

Drones have at least four rotary blades that are used to provide lift. These blades are attached to their own DC motor, capable of incredible speeds (RPM). The speed of these motors is controlled by a microcontroller, using PWM to modify speed. The microcontroller receives instructions from a radio controller.

Drones range in price. The more expensive drones will be bigger, can carry heavier loads, have better batteries and may be easier to control. However all drones work on the same basic principles...



BRIEF

Plan and build the electronics of a basic drone.

Write a systems diagram, detail the input and output actuators, create a flow chart and write a programme to the specification below.

DESIGN SPECIFICATION

Four rotary blades connected to their own motor.

Controls to make the done go up and down.

Controls to make the drone go forward and back.

Do not worry about turning round or going left or right.