Module 4

Components of a robot

After you have completed this module, you should be able to:

- differentiate between the terms *microcontroller* and *microprocessor* (single-board computer) as well as in terms of the purpose and use of each;
- explain the concept *microcontroller development board*;
- differentiate between a *microcontroller* and *microcomputer* (single-board computer);
- list examples of common microcontrollers and microcomputers (microprocessor/ single-board computer);
- list different examples of microcontroller developer and single-board computer boards;
- explain the similarities and differences between some mainstream microcontroller and single-board microprocessor computer boards;
- present a typical breakdown of the components of a microcontroller board and a microprocessor board with relation to the given list components;
- explain the purpose and use of GPIOs;
- demonstrate knowledge of various GPIO pins as part of the prescribed microprocessor (single-board computer);
- explain the purpose of the various components of a microcontroller board, e.g. Arduino;
- explain the purpose of the various components of a microprocessor board, e.g. Raspberry Pi;
- define the term *HATs* (hardware on top);
- differentiate between different examples of HATs;
- explain different ways in which power could be supplied to the development board;
- define the terms *breadboard* and *breakout*;
- differentiate between the typical use of a breadboard and breakouts;
- explain how a breadboard is configured (rows, columns, pins);
- list the basic components used as part of a breadboard prototyping project;
- explain how a breadboard could be used to present a typical circuit;
- identify typical circuit drawings and basic electrical component symbols within the given range;
- set up a breadboard connected to a GPIO board as power source with applicable components to illustrate the difference between a *series* and a *parallel circuit*;
- explain the analogies that:
  - components in parallel run side by side; and
  - components in series come one after the other;

>>

>>

- differentiate between *passive* and *active electrical components*;
- differentiate between different analogue and digital components typically used for input and output;
- define the term *sensor*;
- explain the use of sensors as part of electrical circuits and projects;
- explain the purpose of an analogue-to-digital converter; and
- incorporate various components listed under sections 4.2.4, 4.2.12 and 4.2.13 as part of the design and creation of an artefact or prototype solution for a particular scenario or problem.

Please check section refs where mentioned in text due to renumbering.

Introduction

Working with electronics can be a fascinating hobby or career. With the availability of low-cost microcontrollers, microcomputers, sensors and other components, it has never been easier to experiment with electronics and develop your own devices. Putting an electronics project together can sometimes seem daunting, but it does not have to be. If you understand the basic concepts of microcontrollers and microprocessors as well as how electrical components are used in circuits, you can design and build just about anything.

4.1 Basic concepts of microcontrollers and single-board computers

A microcontroller is a special type of computer, as you will learn in this module. Microcontrollers and microprocessors offer different ways of organising and optimising a computer system that is based on a CPU. It is important that you understand the difference between these concepts, because, depending on your project, you will need to use one or the other.

Bear in mind the concept *central processing unit* (CPU), because it is the heart of the entire computer system. Various secondary (called *peripheral*) equipment, including input/output devices and auxiliary (supplementary) storage units are linked to it. The function of the CPU is to process and execute instructions that are received as input from the user or programs. The CPU runs all the operating systems and applications.

In addition, you need remember what an *integrated circuit* (IC) is. Also called a *microelectronic circuit*, *microchip* or *chip*, an IC is an electronic circuit formed on a small piece of semiconducting material. It performs the same function as a larger circuit made up of individual (called *discreet*) electronic components.

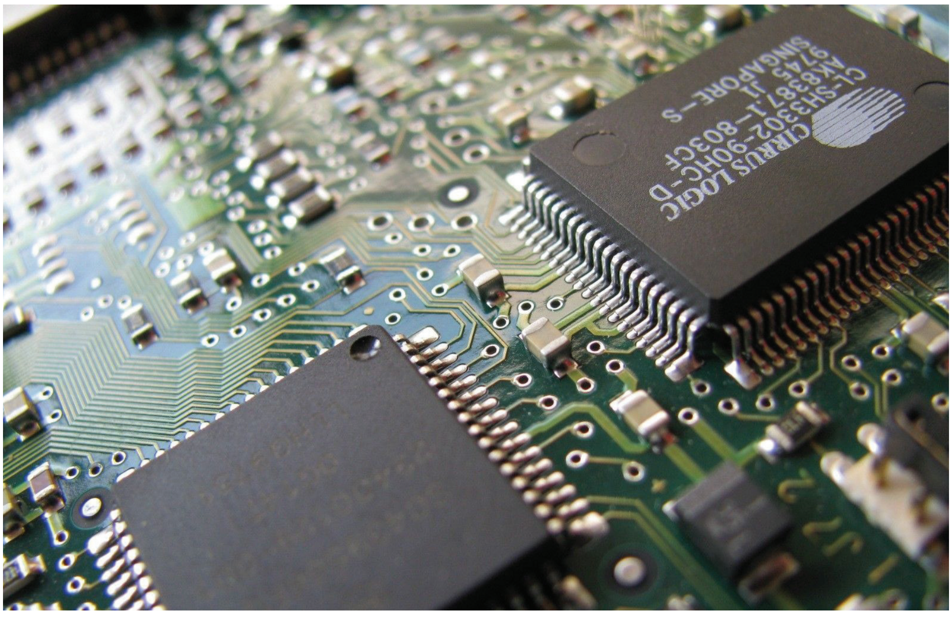


Figure 4.1: Example of ICs or main-board chips

4.1.1 Differentiating between microcontrollers and microprocessors

A **microcontroller**, also called an *embedded controller* or *microcontroller unit* (MCU), is an **embedded** system that integrates an entire computer into a single chip. Think of a microcontroller as a tiny computer. The main components are the processor (CPU), random-access memory or RAM as well as input and output (abbreviated *I/O*) pins. The microcontroller is embedded inside of a system to control a single function of the device. It takes inputs from the device it is controlling and keeps this control by sending signals to different parts of the device.

Microcontrollers usually operate at lower speeds and need to be designed to consume less power than microprocessors because they are embedded inside other devices that can have greater power consumptions in other areas.

A **microprocessor** or *single-board computer* (SBC) is a small computing device that has all the elements of a complete computer within one circuit board. It can easily be connected to other hardware if needed for a project. It works like a microcontroller, but it can do more. It can process signals, communicate wirelessly and be used in a number of applications and artificial intelligence (AI) projects while transmitting data.

NOTE

The main difference between a microcontroller and a microprocessor is that microcontrollers generally run a single program*, i.e. they perform only* one task*. Microprocessors, on the other hand, act as the heart of a computer system carrying out all the instructions of a computer program.*

VOCABULARY

Microcontroller – *compact integrated circuit (IC) designed to control one specific operation in an embedded system*

Embedded – *designed and built as an integral part of a system or device*

Microprocessor – *computer processor where the data processing logic and control are included on a single integrated circuit*

4.1.2 Purpose and use of microcontrollers as opposed to microprocessors

VOCABULARY

Bits – *binary digits, the most basic unit of information in computing*

The purpose of microcontrollers is to execute software. They gather input, process the information and output a certain action based on the information gathered. Any electrical home appliance that stores, measures, displays information or calculates has a microcontroller chip inside it. They are also used in vehicles, robots, office machines, medical devices, mobile radio transceivers and vending machines, among other devices. Microcontrollers are divided into categories according to their memory, structure, **bits** and instruction sets.

VOCABULARY

Binary system – *method of mathematical expression which uses only two symbols: 0 and 1; computers use the binary numbering system to interpret data*

I can't find the words binary system in the ms to ref t

Microprocessors combine the functions of a central processing unit on a single IC or, at most, a few ICs. They are multipurpose devices that take binary data as input, process it as per instructions stored in its memory and then produces results as output. Microprocessors are mass storage devices, and they are also an advanced form of computer, which is why they are also called *microcomputers*.

Single-board computers are found almost everywhere one interacts with a screen, e.g. computers and smartphones. Examples of uses of microprocessors include speed control in cars, traffic-light control, communication equipment, television, satellite communications, home appliances such as microwave ovens, washing machines, gaming controllers, slot machines, etc.

They are also found in more professional applications such as the following:

- ATM machines
- Industrial computers
- Medical equipment
- Automation equipment
- Cash registers.

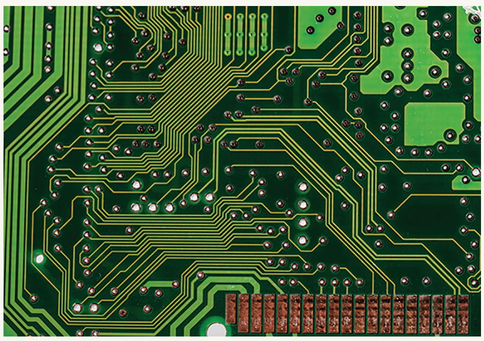
Because they can run a screen, microprocessors are much easier to use than microcontrollers. They are also less expensive and much smaller than personal computers, while they offer many more functions than the microcontrollers.

4.1.3 The microcontroller development board

As mentioned above, microcontrollers are part of most electronic products these days. A *microcontroller development board*, also called *development kit*, is basically a printed circuit board (PCB) with circuitry and hardware designed to allow you to experiment with a certain microcontroller.

NOTE

A PCB *is a board of thin insulating material that has thin strips and pads of a conducting material (usually of copper) fixed onto it. The pads support the terminals of components and the thin strips of conductive material (called* traces*) electrically connect the components in the circuit. Integrated circuits and other components are attached to the PCB.*



If you want to work on a project, you can design and build the circuit on a breadboard (refer to section 4.2). However, it is time-consuming, and you can never be completely sure about the connections. When something goes wrong, you may not be sure whether it is because of the setup or the design itself. A development board offers ready interface and can be used in various projects.

A development board has the following components:

- Power circuit – typically runs from a 9-V/12-V power supply
- Programming **interface** – to program the microcontroller through the computer
- Basic input circuitry, e.g. buttons
- Basic output circuitry, e.g. LEDs
- Input/Output (I/O) pins access that can be used for everything else, e.g. sensors, motors, etc.

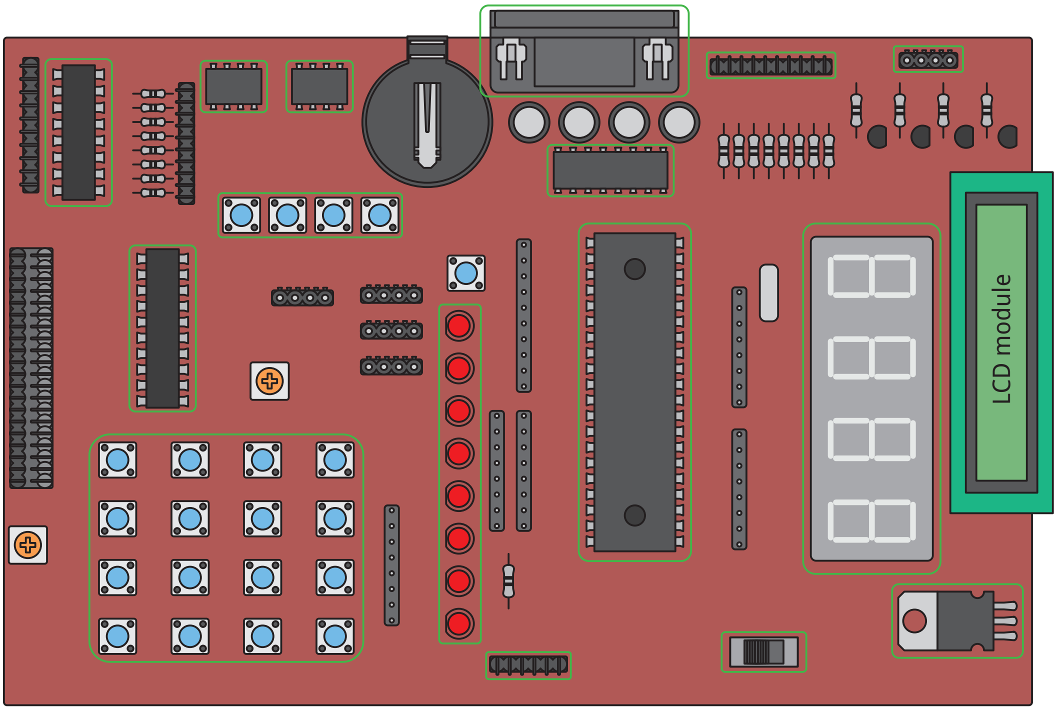


Figure 4.2: Example of a development board

VOCABULARY

Interface – *device/program enabling a user to communicate with a computer*

NOTE

Compare the development board to the breadboard and the circuits built on it in section 4.2.

4.1.4 Differentiating between microcontrollers and microcomputers

As you learned in section 4.1.1, a microcontroller is a processor on a single chip. A microcomputer is an electronic device with a microprocessor as its CPU. Personal computers were once commonly referred to as microcomputers, more specifically small digital computers whose CPU is contained on a single, integrated semiconductor chip.

4.1.5 Examples of common microcontrollers and microcomputers

The uses of microcontrollers are mentioned in section 4.1.2 above. Examples of common microcontrollers can be divided into three categories:

- 8-bit microcontroller:
  - Used for projects that:
    - do not demand very high processing power;
    - are dedicated to a single task;
    - require limited user interface; and
    - with little data processing
  - Various sizes – 16-pin devices to chips with 64 pins
  - Examples include Arduinos (refer to section 4.1.11)
- 16-bit microcontroller:
  - Faster, support more peripherals and have more memory
  - Most have input/output pins
  - Examples include the TI MSP430 series, very low-power microcontrollers
- 32-bit microcontroller:
  - Powerful devices with microprocess-like features and some advanced features, e.g. fetching subsequent instructions ahead of time
  - Examples include STM32 and ESP32

VOCABULARY

Peripheral – *an auxiliary/support device used to put information into and get information from a computer; also called an* input-output device

A *single-board computer (SBC)* is a complete computer built on a single circuit board (refer to section 4.1.1). The production of SBCs was made possible by the increasing density of integrated circuits. This computer is commonly used to demonstrate or develop systems as well as in education – to teach children (and adults) about computer hardware and software, how computers work and to create fun electronics projects.

Unlike desktop personal computers (PCs) or portable computers, single-board computers do not have expansion slots that allow for **peripheral** functions or expansion. The are built using a wide range of microprocessors.

The Raspberry PI is an excellent example of an SBC. Other popular examples include: NVIDIA, RockPi, BeagleBoard, C.H.I.P. and Cubit.

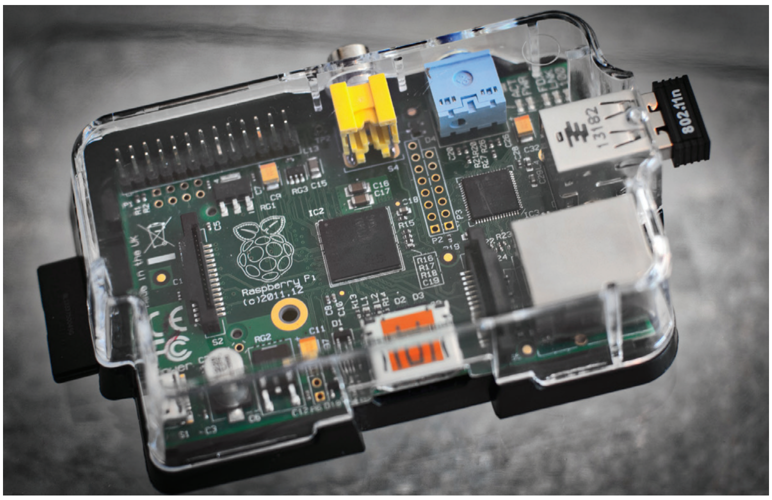


Figure 4.3: A Raspberry Pi board

Other applications of SBCs include the following:
- Programming
- Embedded applications
- Backplane – electrical connector that joins several electrical circuits
- Robotics
- Home automation.

4.1.6 Examples of microcontroller developer and single-board computer boards

We have already looked at the development board for microcontrollers (refer to section 4.1.3).

[I cannot find information on SBC boards. Please supply.]

4.1.7 Similarities and differences between mainstream microcontrollers and single-board microprocessor computer boards

In section 4.1.1, we differentiated between microcontrollers and microprocessors or single-board computers. As stated there, the main difference between a microcontroller and a microprocessor is that microcontrollers generally run a single program, i.e. they perform only one task. Microprocessors, on the other hand, act as the heart of a computer system carrying out all the instructions of a computer program.

[I can't find info on single-board microprocessor computer boards ? Please supply.]

Syllabus states "In relation to: buttons
GPIO – general purpose input-output header
GPIO pins – digital I/O pins; ground pins; analogue pins
USB ports and connections
LED indicators
Voltage regulator
Power jack"

Amanda, I sorted this table in alphabetical order. Hope it's ok.

4.1.8 Components of a microcontroller board and a microprocessor board

Terminology

Before we move on to discuss the various components of MCs and SBCs, there are certain terms and/or abbreviations that you must know.

TERM	MEANING
AC	Alternating current – electrical current that reverses or alternates its direction
ADC	Analogue-to-digital converter – convert analogue signals to digital (refer to section 4.2.10 and 4.2.13)
Analogue pin	Used for input of analogue signals
DAC	Digital-to-analogue converter – converts digital signals to analogue (refer to section 4.2.10 and 4.2.13)
DC	Direct current – electrical current that flows in one direction only
Ethernet	A system for connecting a number of computer systems to form a local area network
GPIO	General-purpose input/output – an uncommitted digital signal pin on an IC or SBC; can be used as an input or output, or both; controllable by software
Ground (GND)	Point for all signals/common path in an electrical circuit from where all the voltages can be measured
I/O ports	Input and output ports
IC	Integrated circuit – an electronic circuit formed on a small piece of semiconducting material; also called a *microelectronic circuit*, *microchip* or *chip*
Input	Data that is entered into/received by microcontroller/SBC, e.g. pressing the key on a keyboard or clicking a mouse as well as data that has to be processed
Interface	A shared boundary across which two or more separate components of a computer system exchange information; the exchange can be between software, hardware, peripheral devices, humans or combinations of these
IrDA	Infrared data association – maintains a standard for infrared data transmission in many electronic devices, e.g. laptops and remote controls
LAN	Local area network – a group of computers and associated devices that share a common communications line or wireless link
LED indicator	Light-emitting diode – prompts use to the status of a hardware device; used in electrical circuits (refer to section 4.2.4)
Logic	Sequence of operations performed by hardware or software
MMC card	Multimedia card – tiny flash-memory card designed to make storage portable among various devices, e.g. cell phones, smartphones, digital cameras, etc.
Oscillator	Electronic device that works on the principles of oscillation: a periodic fluctuation between two things based on change in energy
Output	Any data/information processed by and sent out from a microcontroller/SBC, e.g. sound from the SBC's speaker
PCB	Printed circuit board; board of thin insulating material that has thin strips and pads of a conducting material (usually of copper) fixed onto it

TERM	MEANING
Peripheral device	Any hardware component that is connected to a computer and controlled by the computer system but does not contribute to its functioning
POE	Power over ethernet – a technology that delivers power and data over a single ethernet cable to power devices
Port	Serves as an interface between the computer and peripheral devices
POS accessories	Point-of-sale accessories which include the CPU
Power jack	Connector that gives device access to a stable electric current and allows you to charge the battery
RAM	Random-access memory – short-term memory (explained later)
ROM	Read-only memory – receives data and permanently writes it on the chip
SBC	Single-board computer
SD card	Secure-digital card – tiny flash-memory card designed for high-capacity memory; used in cell phones, smartphones, digital cameras, etc.
USB	Universal serial bus; standard cable connection interface for PCs and other electronic devices
Voltage regulator	Electrical/electronic device that maintains the voltage of a power source within acceptable limits
Wired interface	Deals with the method of transferring video/audio data among electronic devices using cables
Wireless interface	A network interface controller which connects to a wireless network such as Wi-Fi or Bluetooth rather than a wired network

Components of a microcontroller

As you know by now, a microcontroller is a single IC that is typically used for a specific application and designed to implement certain tasks. Microcontrollers contain elements that interface with each other for the proper functioning of a device.

[Listed in syllabus but not shown/covered here?]

The essential components of a microcontroller are the following:

- USB ports
- LED indicators
- Voltage regulator
- Power jack.

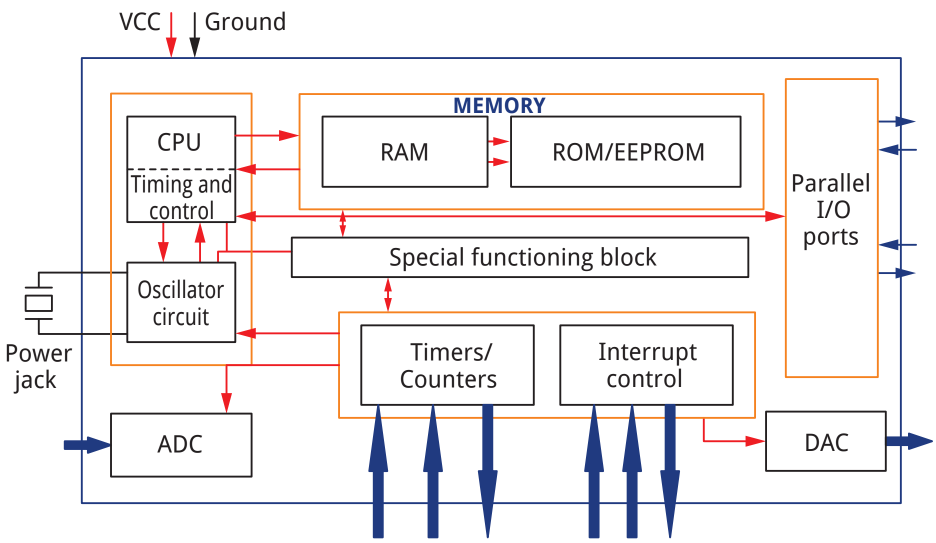


Figure 4.4: Components of a microcontroller

AQ queries on labels:
VCC = Voltage regulator???
~~Crystal~~ Power jack ?;

- **CPU (microprocessor):**
  Where the computing takes place
  - Controls all the instructions/data
  - Responsible for fetching the instruction, decoding/processing all the data and then executing the instructions
  - Two main components
    - > arithmetic logic unit (ALU) – performs arithmetic (maths) and logical operations
    - > control unit (CU) – handles all the processor’s instruction executions
- **Memory:**
  Where data or files are stored
  - *Random-access memory (RAM)* which temporarily stores the files you are working on/any variables when the CPU is operational
  - *Read-only memory (ROM)* which permanently stores instructions for your computer (the program is stored as instruction sets)
  - *Flash memory* which retains its data for an extended period
  - *EEPROM* (acronym meaning *electrically erasable programmable read-only memory*) which is like flash memory but retains data even after shutdown
- **Oscillator circuit:**
  XXX

- any special info needed?

- **Special functioning block:**
  XXX

{NOT indicated in Fig. 2.4?}

- **Serial bus interface:**
  Provides various serial interfaces between a microcontroller and other peripherals like parallel ports
  - Responsible for serial communication in the microcontroller, sending data one bit at a time
  - With microcontroller boards, it connects ICs with signal traces on the printed circuit board (PCB)
- **I/O ports:**
  Connect microcontroller to real-world applications
  - *Input ports* – receive changes in real-world (e.g. temperature or motion sensing, push buttons, etc.)
    - > Input goes to CPU where it is processed
    - > Signal is sent to output port
  - *Output ports* – execute operations, e.g. turning LED on and off, running a motor, etc.

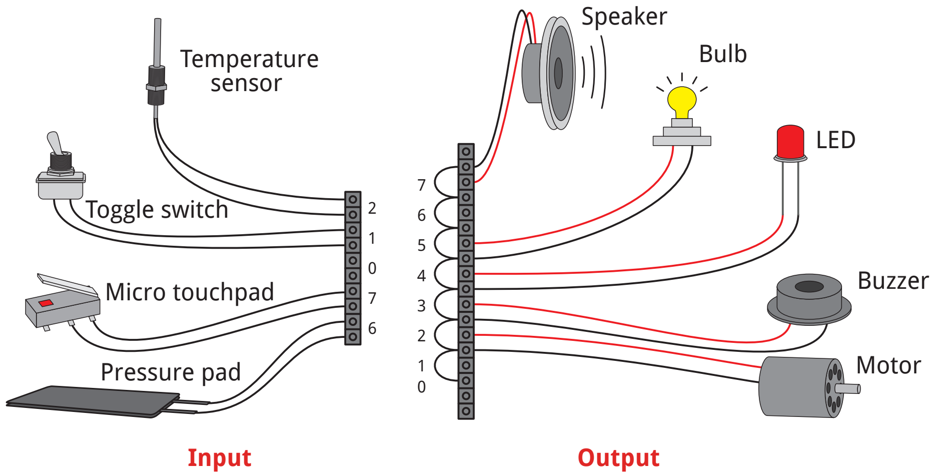


Figure 4.5: Examples of inputs and outputs

- **Converters:**
  Convert the input and output signals in the microcontroller, respectively
  - Analogue-to-digital converters (ADC) for input
  - Digital-to-analogue converters (DAC) for output
- **Timers and counters:**
  Used for all the clocking operations in the microcontroller, including functions such as pulse-width modulation, clock control and frequency measurements
- **Interrupt control:**
  Provides interruptions/delays for a working program; interruptions can occur internally or externally

Components of a single-board computer (SBC)

As explained in section 4.1.1–4.1.3, an SBC is a complete computer built on a single circuit board. It is a combination of microcontrollers and PCs. SBCs, like microcontrollers, have programmable I/O ports. The biggest difference between an SBC and a personal computer is that SBCs usually do not have expansion slots and they have a programmable I/O. Unlike microcontrollers, SBCs usually have expandable storage to store the operating system needed to run the system. It is possible to install operating systems that run the bare minimum, but SBCs will never be as fast as microcontrollers for special purpose projects and that is why they are rarely used for this purpose.

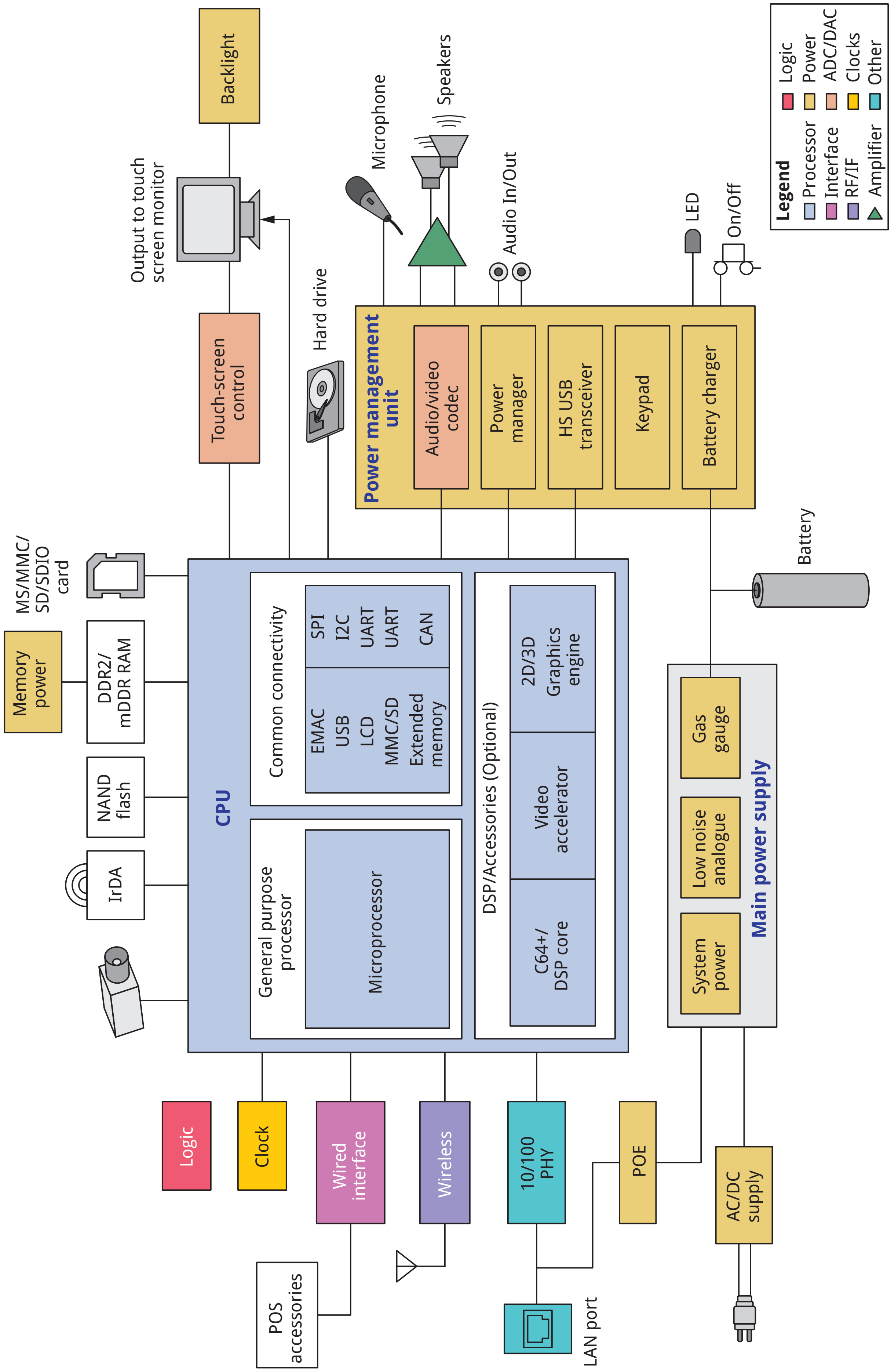


Figure 4.6: Block diagram indicating the components of an SBC

As you can tell from the block diagram above, an SBC looks like a combination of a microprocessor unit (MPU) and multi-controller unit (MCU). The main components are the following:

- **CPU:**
  The brain of the SBC, responsible for fetching the instruction, decoding it and then executing it
- **Memory:**
  Like in the microcontroller discussed above, the memory is used to store data and programs
  - There are RAM and ROM or flash memories for storing data and/or program source codes
- **Clock/Timer:**
  Provides all timing and counting functions inside the microcontroller; used to perform clock functions, modulations, etc.
- **I/O ports:**
  Connect microcontroller to real-world applications
  - Input ports (on the left of the block diagram above) – receive changes in real-world (e.g. temperature or motion sensing, push buttons, etc.)
    > Input goes to CPU where it is processed
    > Signal is sent to output port
  - Output ports (on the right of the block diagram above) – execute operations, e.g. turning LED on and off, use of microphone, speakers, etc.

[not indicated in block diagram?]

- **Serial ports:**
  Provide various serial interfaces between a SBC and other peripherals like parallel ports
- **AC/DC:**
  Alternating or direct current which is the main power supply
- **Wireless connectivity:**
  Consists of Bluetooth and Wi-Fi

4.1.9 Purpose and use of GPIOs

VOCABULARY

GPIO pin – *a generic pin whose value consists of one of two voltage settings – high or low*

GPIO is the abbreviation used for *general-purpose input-output*. It is basically a set of pins on an integrated circuit, electronic circuit board or add-on card. The pins can be used as input or output or both and are controlled by software. **GPIO pins** have no predefined purpose. They basically serve to send and receive electrical signals based on the programmed software. You are in charge of what each pin does, as explained in the following section.

One of the powerful features of the Raspberry Pi is the row of GPIO pins along the top edge of the board. These pins are a physical interface between the Raspberry Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output). There are 40 pins on the Raspberry Pi (26 pins on earlier models), and they provide various functions. The pins are numbered, as shown in the figure below. The inner row consists of uneven-numbered pins from pin 1 to pin 39. The outer row consists of even-number pins, i.e. pin 2 to pin 40.

GPIOs are used in a variety of applications. They are most commonly used to operate custom electronics, e.g. when you are building your own robot arm or prototype circuits. The GPIO interface allows you to customise signals so that they operate your equipment the way you want it to run. They are generally used with a breadboard (refer to section 4.2.7). You can add, remove or move electronic components by connecting or disconnecting specific pins. The pins can be connected to react to switches and turn LEDs on or off, etc.

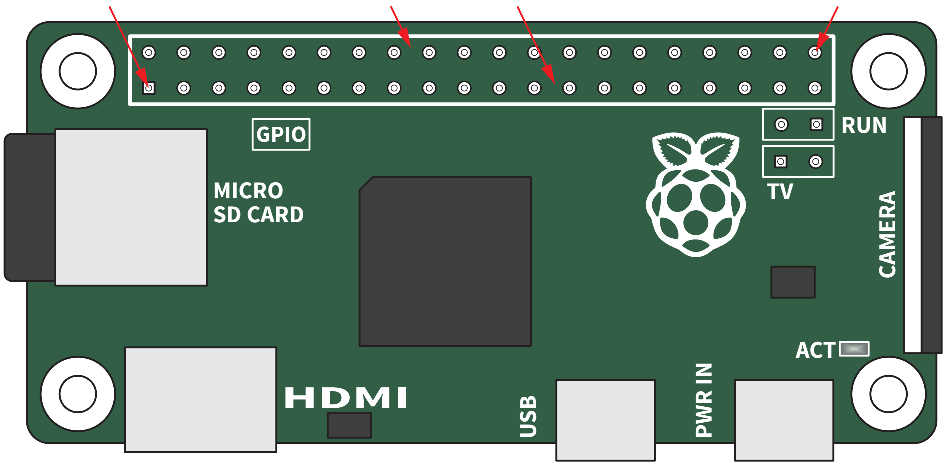


Figure 4.7: GPIO of the Raspberry Pi

NOTE

If you follow the instructions, experimenting with the GPIO pins is safe and can be fun. However, randomly plugging wires and power sources into your Raspberry Pi may destroy it, especially when you are using the 5-V pins.

4.1.10 Various GPIO pins as part of microprocessor

Syllabus: Name pins; describe function of each; Use pins as connectors; Use indicators and ports

Name of pins and their functions

The following GPIO pins are found when you use the Raspberry Pi:

- Pins that provide power (3,3 V or 5 V) – power connected devices such as an LED that does not have its own power source
- Ground pins – necessary to complete certain circuits but do not have output power
- Pins that can be configured to send/receive electrical signals
- Special-purpose pins.

A, ek't die ook in alfabetiese order gesort. OK?

POWER PINS AND RESET	
NAME OF PIN	**FUNCTION**
EN	3,3-V regulator enable pin; tie to ground to disable
GND	Common ground for power and logic
RST	Reset; tie to GND to reset (or press *Reset* button)
VBAT	Positive voltage from backup battery, if connected; 3,7 V
VCC	3,3-V output from the onboard regulator; can supply up to 500 mA
VUSB	Positive voltage from USB port if connected; 5 V

Using pins as connectors

The diagram below shows the different types of GPIO pins of a Raspberry Pi. Pins serve different purposes.

- Pins labelled GND, 3,3-V or 5-V supply power

NOTE

Pins labelled 3,3 V have output levels of between 0 and 3,3 V. The inputs should not be higher than 3,3 V. They are not safe to use at 5 V. The pins labelled GND have 0 V. These pins cannot be configured.

- The other pins can be used as either digital inputs or outputs. Output pins can be set to 3,3 V (high) or 0 V (low); input pins can be read as 3V3 (high) or 0 V (low).
- Various input pins include the following:
  - SDA and SCL: these are two inter-integrated circuit connections
  - PWM: Pulse-width modulation – used with components such as motors and LEDs by sending short pulses to control how much power devices receive
  - GPCLK0: General-purpose clock pin – can be sit to output a fixed frequency
  - MO SI: Master output slave input – transmits data from a master to a slave
  - MI SO: Master input slave output – transmits data from a slave to a master
  - SCLK: Serves as a clock for SPI communication
  - ID_SD and ID_SC: Dedicated pins for setting up HATs (refer to section 4.1.13)
  - PCM_FS and PCM_CLK: Used to provide a clock signal to an external audio device
  - SPI: Serial peripheral interface – method of communicating with devices
  - TX and RX: These are UART pins; UART stands for *universal asynchronous receiver/transmitter*, a serial communication protocol, meaning that it takes byes of data and transmits the individual bits in sequence
  - CE1 and CE0: Can be used as interrupt lines, which allow a peripheral board connected via multiple pins to signal to the primary embedded board that it requires attention
  - PCM_DIN: Provide data input signal
  - PCM_DOUT: Provide data output signal

NOTE

I2C *is the abbreviation for* inter-integrated circuit *– it is like an SPI, but generally easier to set up and use. An I2C can sustain as many different devices as needed, provided they each have unique address places on the I2C bus.*

[pic source: www. reaspberrypi.org]

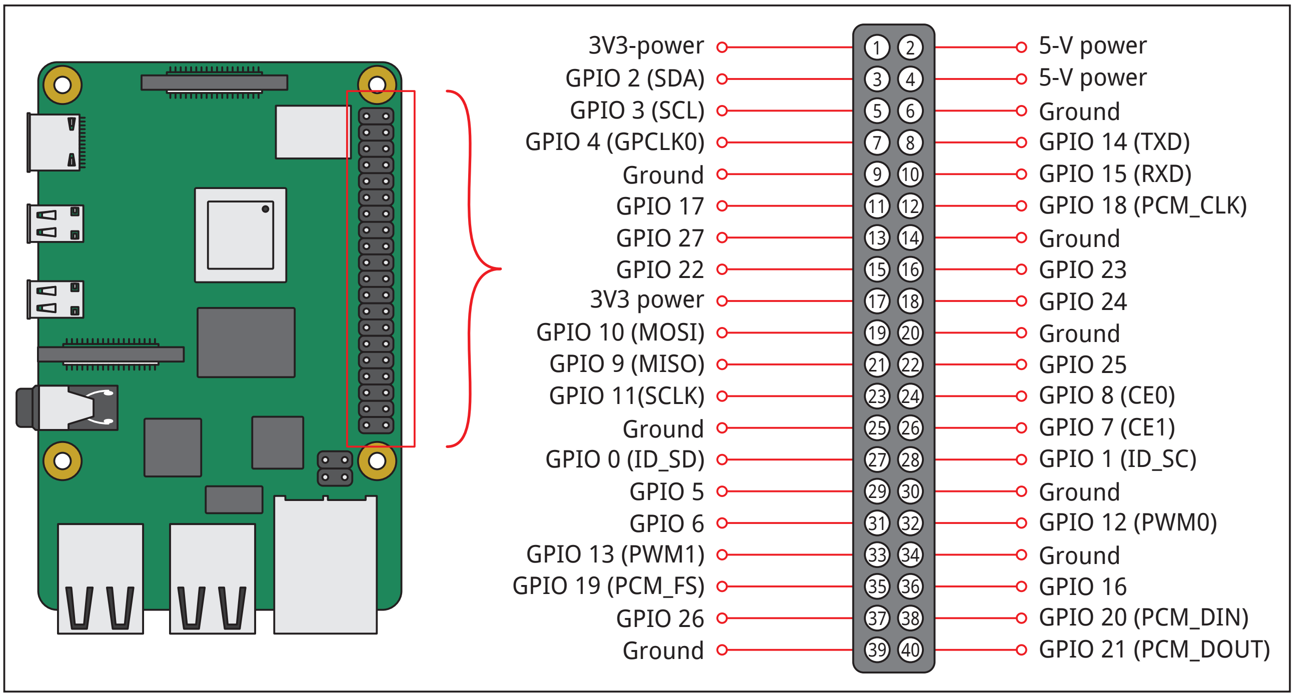


Figure 4.8: GPIO pins

GPIO indicators and ports

[Please provide info on indicators]

A *GPIO port* is a platform-defined grouping of GPIO pins – often four or more pins. These pins cannot be controlled individually as GPIO pins. To use a specific port or pin, an application should first be opened and a *GPIOin* or *GPIOPort* obtained. This is done using the numerical ID, name, type (interface) or properties of the port or pin.

4.1.11 Purpose of components of a microcontroller board

Refer to section 4.1.8 where the components and the purpose of each are discussed in detail.

4.1.12 Purpose of components of microprocessor board

Refer to section 4.1.8 where the components and the purpose of each are discussed in detail.

4.1.13 Hardware attached on top (HATs)

VOCABULARY

HAT – *an extension board that connects to the GPIO pins of a Raspberry Pi*

HAT is the abbreviation used for *hardware attached on top*. It is an expansion or add-on board that has a specific set of rules to make it easier for users to connect the set of GPIO pins, thereby extending the abilities of the Raspberry Pi, e.g. adding LEDs, motors, sensors, etc. HATs include a system that allows the Raspberry Pi to identify a connected HAT and automatically configure the GPIOs and drivers for the board.

A HAT is basically a 65 × 65 mm, rectangular board with four mounting holes in the corners. The holes align with the mounting holes on the Raspberry for a strong connection. It has a 40 W GPIO header, and the add-on board tells the Raspberry Pi which I/O pins it is using, how they need to be configured and what drivers must be loaded. In other words, it supports special automatic configuration by using two dedicated GPIO pins – ID_SD and ID_SC on the GPIO header.

The advantages of HATs include:
- Can just be plugged into the Raspberry Pi, no soldering required
- Four mounting holes ensure a strong connection
- Auto-configuration ability.

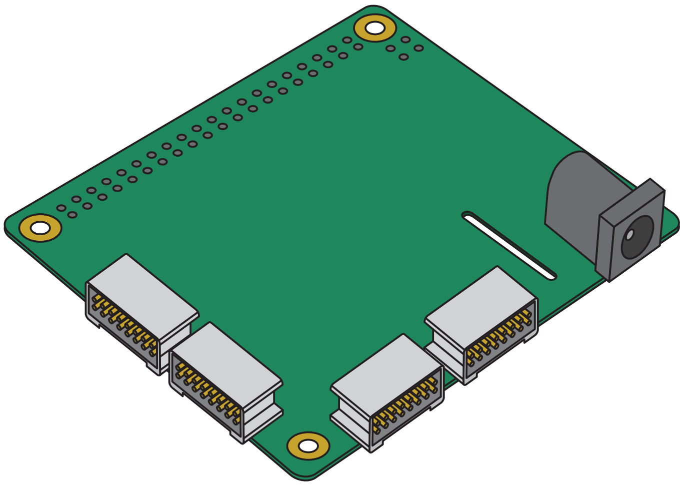


Figure 4.9: Example of a HAT

4.1.14 Examples of HATs

Examples of HATs include the following:
- Voice HATs, e.g. ReSpeaker 2-Mics Pi HAT and ReSpeaker 4-Mic Array for Raspberry Pi
- Grove Base HAT for Raspberry Pi – provides a range of digital, analogue, I2C, PWM and UART ports to meet expansion needs
- Raspberry Pi breakout board v1.0 – prototype board that can combine the Raspberry Pi with other components/mlodules
- Display HATS, e.g. Triple-Color E-Ink Display for Raspberry Pi
- ADC HATs, e.g. 4-Channel 16-Bit ADC for Raspberry Pi
- Raspberry Pi Motor Board v1.0 – can control inductive loads with currents up to 5,0 A and lets you drive two DC motors with certain Raspberry Pi models

4.1.15 Methods of supplying power to the development board

NOTE

It is recommended that you revise Module 3, specifically the concepts voltage*,* current*,* resistance *and* Ohm's law*.*

In this section, you will learn more about the various ways you can power electronic projects.

[+More info needed]

Some of the most common methods used for powering projects include the following:

- USB power – supplied by a USB cable
- AC to DC wall adapters (like the SBC) – can be used after a circuit has been tested; various wall adapters are available with set voltage and current output, so make sure the adapter matches the specifications of the project you will be powering
- Batteries – good choice for mobile projects or if you are in a remote location that is off the main power grid
  - Huge variety to choose from, including alkaline, rechargeable, lithium, etc.
  - Depending on the board and components you have used, an output of 9 V or 10 V may be needed.

Activity 4.1 **Work on your own**

1. Differentiate between a *microcontroller* and a *microprocessor*.

2. What is a *microcontroller development board*?

3. Match the explanation/definition given in COLUMN B with the term given in COLUMN A.

COLUMN A	COLUMN B
3.1 USB	**A** short-term memory that temporarily stores files
3.2 RAM	**B** component that permanently writes data on the chip
3.3 Input	**C** alternating or direct current
3.4 ADC	**D** standard cable connection interface for electronic devices
3.5 Integrated circuit	**E** another term for *microprocessor*
3.6 PCB	**F** data received/entered by pressing keyboard key
3.7 LED	**G** board of insulated material with copper traces and pads fixed onto it
3.8 SBC	**H** example of an output device
	I analogue-to-digital converter
	J electronic circuit on a tiny piece of semiconducting material called *chip*

Activity 4.1 (continued)

4. Explain why microcontrollers have input and output ports.
5. 5.1 What does the abbreviation *CPU* mean?
   5.2 Discuss the importance of the CPU in microcontrollers and microprocessors.
6. What is the function of GPIOs?
7. 7.1 Define the term *HAT*.
   7.2 Give a short description of a HAT.
   7.3 List the advantages of using HATs.

4.2 Introduction to breadboards, breakouts and solderless prototyping

4.2.1 Definition of a breadboard

VOCABULARY

Breadboard – *thin, plastic board with tiny holes for building an experimental model of an electric circuit*

A **breadboard** is a rectangular, plastic board with small holes to which electronic components and connecting wires can be attached. Underneath the plastic top, there are horizontal rows of metal strips. The tops of the metal rows have little clips which hold the wire or the leg of a component that you insert into the exposed holes in place.

The connections you make on a breadboard are not permanent, so it is easy to remove a component if you make a mistake or to start a new project. This makes breadboards great for beginners, but they can also house very complex circuits. You can use breadboards to build circuits and test all sorts of fun electronic projects, from different types of robots to an electronic drum set.

DID YOU KNOW

The term **breadboard** *comes from the early days of electronics, when people would literally drive nails or screws into actual breadboards to connect their circuits.*

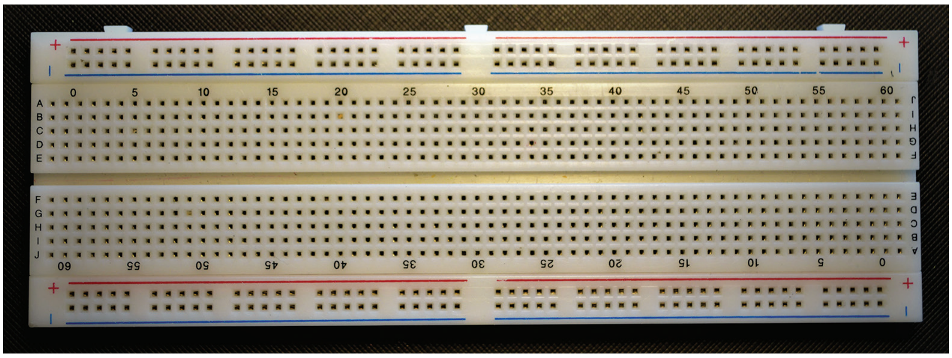


Figure 4.10: Breadboard

As you can see in the figure above, the rows are numbered, and letters are used to label the columns. This helps you identify specific holes on the breadboard when you want to insert components or wire a circuit. The two top and two bottom rows are typically marked by red and blue (or red and black) lines, with plus (+) and minus (–) signs, respectively. This is where the electrical power is connected.

Refer to section 4.2.4 for details regarding the configuration of the rows, columns and pins and the internal connections of a breadboard.

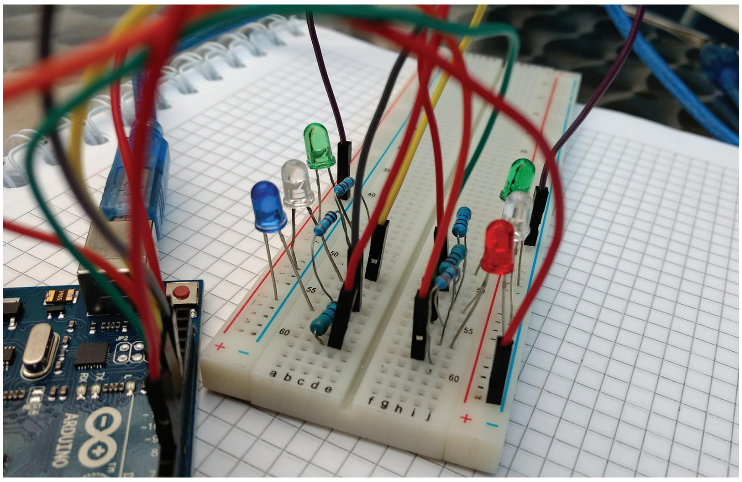


Figure 4.11: Breadboard with electronic components connected

Modern breadboards are made of plastic and are available in various shapes and sizes, and even in different colours. Most breadboards also have tabs and notches on the sides to allow you to snap multiple boards together. However, a single, half-sized breadboard is sufficient for many beginner-level projects.

Technically, the breadboards illustrated in the figures above are called *solderless* breadboards because they do not require soldering to make connections. (*Soldering* can be defined as a low-temperature joining method in which the solder, the joining material, has a much lower melting point than the surfaces to be joined.) Electronic components can be soldered together directly, but more commonly they are soldered onto PCBs. However, engineers may use solderless breadboards to prototype and test a circuit before building the final, permanent design on a PCB.

4.2.2 Definition of a breakout board

[Please check. Author copied entire text from website.]

A *breakout board* is a small board that has its own perfectly spaced pins for solderless prototyping. It makes it easier to connect a single electrical component or module to the rest of the project circuit. ICs have pins that can perform specific tasks, e.g. supplying power, providing a ground, receiving input or sending output. We can say that a breakout board 'breaks out' these pins onto an IC that has through-hole sized pins for a solderless connection.

As you can see in the figure below, breakout boards resemble integrated circuits (revise section 4.1). A breakout board uses a tiny version of an integrated circuit called a *surface mounted device* (SMD). The board is placed between the microprocessor/computer and the motor drivers.

It serves two purposes, namely:

- circuit protection; and
- signal distribution.

There are many types of breakout boards on the market, but the majority are made for various kinds of sensors, like temperature sensors, PIR motion sensors, etc. (Sensors are discussed in section 4.2.12 and 4.2.13.) Like a breadboard, the rows and columns are numbered and labelled, respectively. This makes hooking up the breakout board to an Arduino very easy.

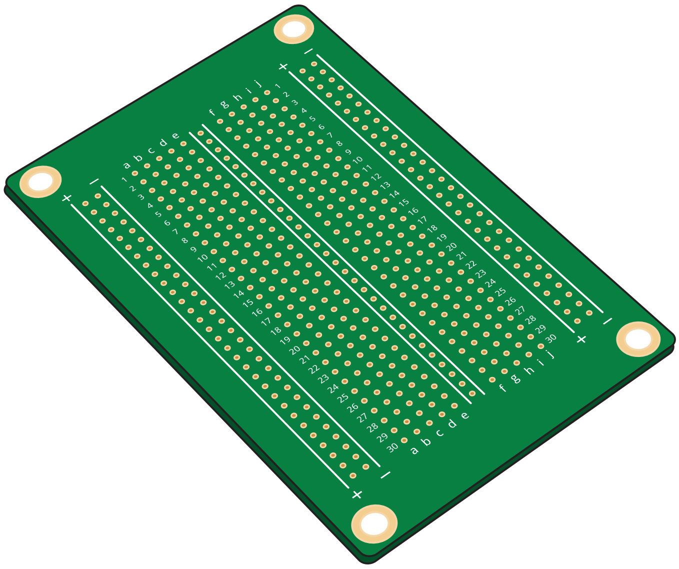


Figure 4.12: Example of a breakout board

Many breakout boards are sold as kits. Usually, the only components you must solder are the header pins that allow the breakout board PCB to plug into a breadboard. This is easy to do.

Finally, make sure you know what voltage-supply pin it needs to be hooked up to. The Arduino has two voltage-out pins, one of 3,3 V and one of 5 V. Many breakout boards use a supply voltage around 3,3 V. The supply voltage is sometimes printed on the PCB next to the associated pin. When the voltage is not indicated and the pins are simply labelled “VCC”, check the specifications of the breakout board before you connect the power supply.

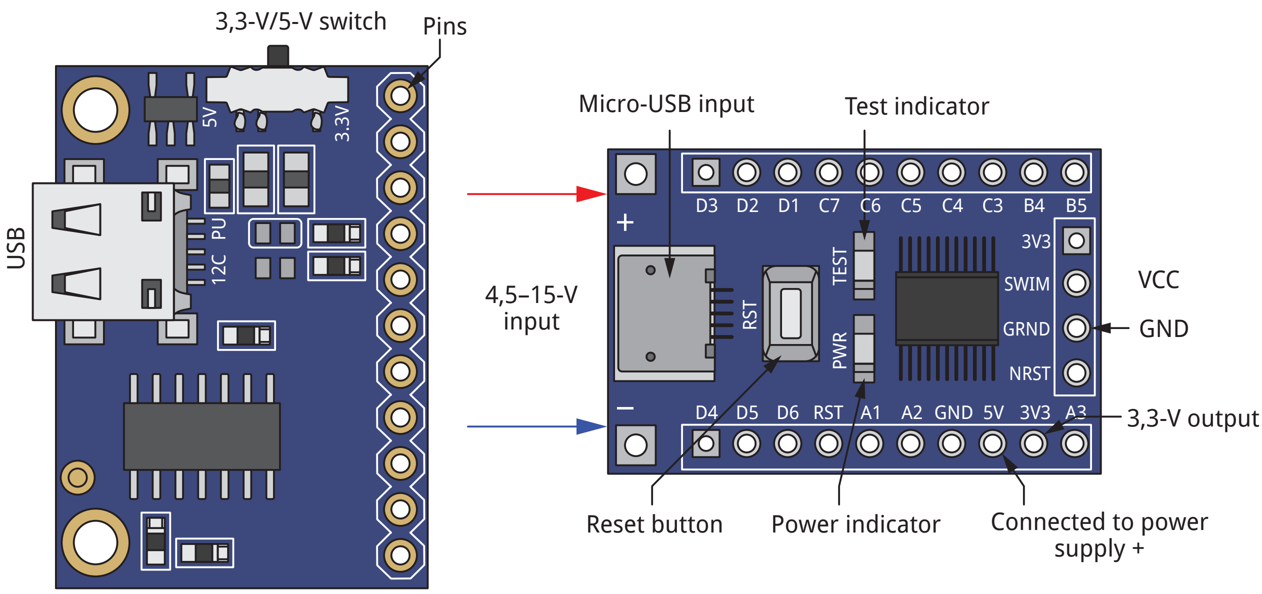


Figure 4.13: Other examples of breakout boards

4.2.3 Differentiate between typical uses of breadboards and breakouts

A breadboard is usually used as the first step in building an electrical circuit. You can change and move components that would otherwise be permanently soldered onto a PCB.

VOCABULARY

Prototyping – *the process of testing out an idea by creating a preliminary model from which other forms are develop or copied*

Breadboards are typically used:
- by beginners because they can build solderless circuits that make it easy to remove or insert components or change connections
- to test new parts, such as ICs
- for **prototyping** – to see how a circuit will react under given parameters
- to duplicate faulty circuits in order to figure out the problem.

A breakout board breaks out each conductor of a bundled cable to a terminal that can easily accept a hook-up wire for distribution to another device. They, too, are used in electronic projects and allow easy, neat installation of electronic devices. The illustration below shows how the breakout board is used to connect three motors to a microprocessor.

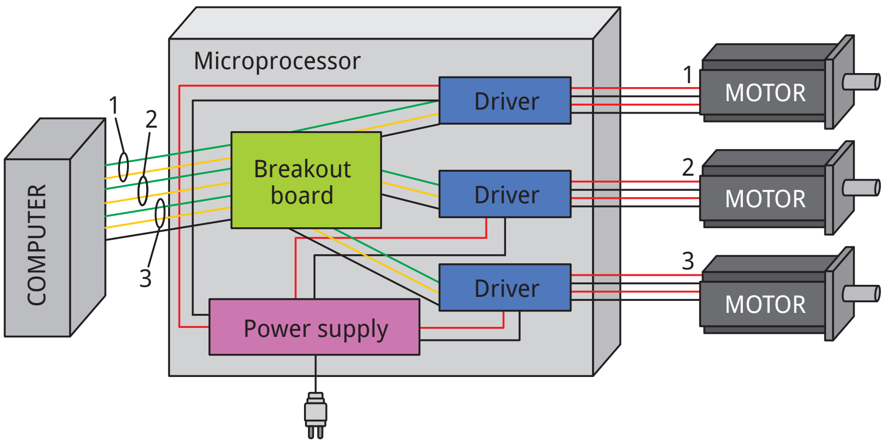


Figure 4.14: Block diagram to show the use of a breakout board

4.2.4 Breadboard configurations

As you learned in section 4.2.1, a breadboard is a plastic board with small holes arranged in a series of labelled rows and columns. The easiest way to explain how a breadboard works is to take it apart and look at the inside. Remember that the underside of the breadboard is made up of horizontal rows of metal strips. The tops of these strips have little clips that hide under the plastic holes. The strips are spaced apart a standard distance of 2,54 mm.

Each set of five holes forms a half-row (columns A–E or columns F–J) that is electrically connected. This means that, e.g. hole A1 is electrically connected to holes B1, C1, D1, and E1. It is *not* connected to hole A2, because that hole is in a different row, with a separate set of metal clips. It is also *not* connected to holes F1, G1, H1, I1, or J1, because they are on the other 'half' of the breadboard—the clips are not connected across the gap in the middle. Once inserted, the component will be electrically connected to anything else placed in that row. This is because the metal rows are conductive and allow current to flow from any point in that strip.

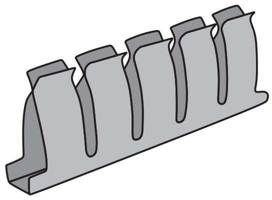


Figure 4.15:
Metal strip with clips

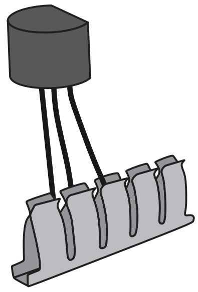


Figure 4.16:
Component connected to clips

Breadboards also have vertical strips called *power rails*. They are metal strips that are identical to the ones that run horizontally. However, unlike all the main breadboard rows which are connected in sets of five, the power rails typically run the entire length of the breadboard (note that there are exceptions). The power rails give you easy access to power wherever you need it in your circuit. They are labelled '+' and '–' and have a blue/black and a red line to indicate the positive and negative side. As the name suggests, power rails supply electrical power to the circuit when you connect them to a battery or other external power supply.

NOTE

There is no physical difference between the positive and negative rails and using them for positive or negative connections respectively, is not a requirement. The labels and colours just make it easier to organise your circuit, similar to colour-coding the wires used in the circuit.

POSITIVE	NEGATIVE
Power	Ground
Positive symbol (+)	Negative symbol (–)
Red	Blue or black

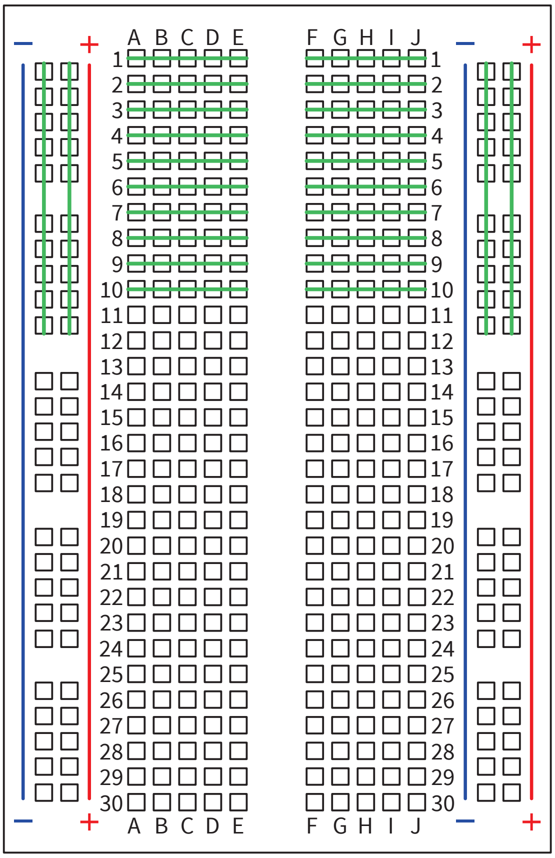


Figure 4.17: Internal connections of a breadboard

Power rails on opposite sides of the breadboard are *not* connected. To connect power and ground to both sides of the breadboard, you would connect the rails with jumper wires, as illustrated in the figure below. Make sure to connect positive to positive and negative to negative.

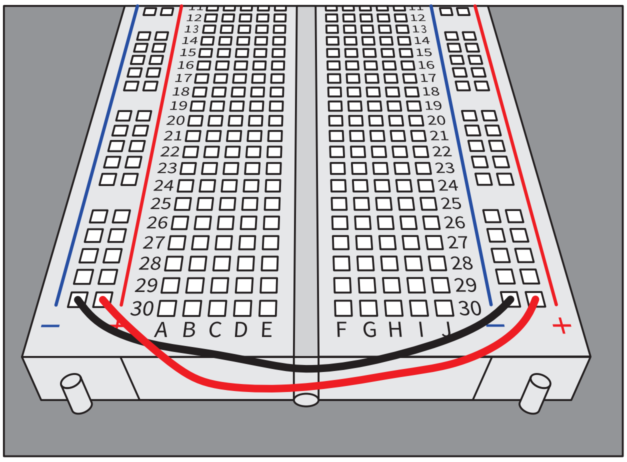


Figure 4.18: Connecting power rails on opposite sides of the breadboard

NOTE

The exact configurations might vary from breadboard to breadboard. For example, some breadboards have the labels printed in landscape orientation instead of portrait. In some the power rails are divided in half along the length of the breadboard (useful when you need to supply a circuit with two different voltage levels). Most mini breadboards do not have power rails or labels printed on them at all.

4.2.5 Components as part of breadboard prototyping

{Can't find much info – is this a battery holder? }

Breadboards can be used to build circuits that comprise various components. In this section you will learn more about the components that are generally used as part of breadboard prototyping projects.

Battery snap/Battery box

Battery terminals are the electrical contacts used to connect a load or charger to the battery. Each battery has a positive (+) and a negative (–) terminal. A *battery snap* is a connector – it is connected to the terminals of typically 9-V batteries and has wires that are connected to the circuit.

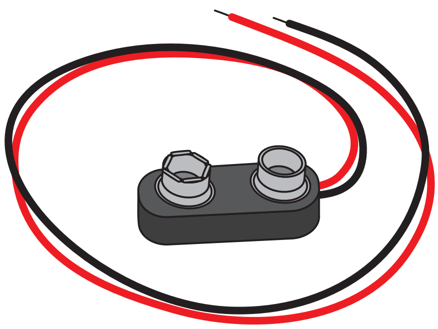


Figure 4.19: Battery snap

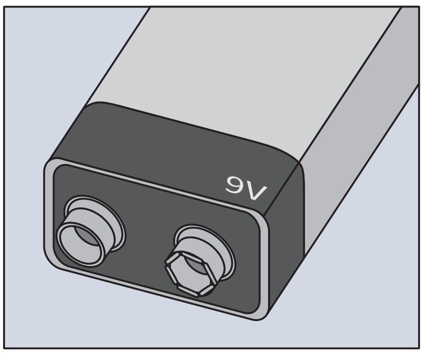


Figure 4.20: 9-V battery

A *battery holder* is used to connect batteries of lower voltages in series when a 9-V battery is needed. There are many different types of batteries that can be connected to a breadboard to power circuits. The positive terminal is almost always marked with a + symbol. Typically, battery holders will have + and – symbols printed inside them. Make sure the symbols on your batteries line up with the symbols in the battery holder.

Jumper wires

Jumper wires are used on a breadboard to make connections that create electronic circuits. They are flexible and have connectors at each end. This makes it easy to attach the wires to a breadboard, component or equipment without the need for soldering.

Several different types available, and you can also buy packs containing varying colours. This makes it easy for beginners to colour code their circuits. However, when it comes to complicated circuits, the long wires can become a tangled nest that is hard to trace (sometimes called a 'rat's nest' or 'spaghetti'). Some kits contain pre-cut lengths of wires with ends bent at 90 degrees for easy attachment to breadboards.

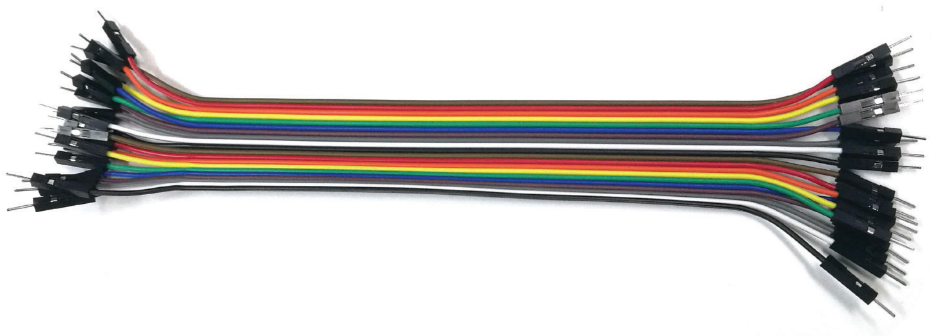


Figure 4.21: Jumper wires

You can also buy spools of solid-core, hook-up wire and a pair of wire strippers. The wire can be cut to the length needed. This is the best option if you plan on doing lots of electronics projects. You can pick the colours you want. It is also much more cost-effective than buying jumper sets.

NOTE

It is important to buy **solid-core** *wire (which is made from a single, solid strand of metal) and not* **stranded** *wire (which is made from multiple, smaller strands of wire, like a rope).*

Stranded wire is much more flexible, and the individual strands make it difficult to push the wire into connectors/breadboard holes. You also need to wire of the right **gauge***, (±0,6 mm in diameter).*

VOCABULARY

Gauge – *measurement of wire diameter; each gauge is represented by a number – the smaller the number the thicker the wire and the higher, the thinner the wires*

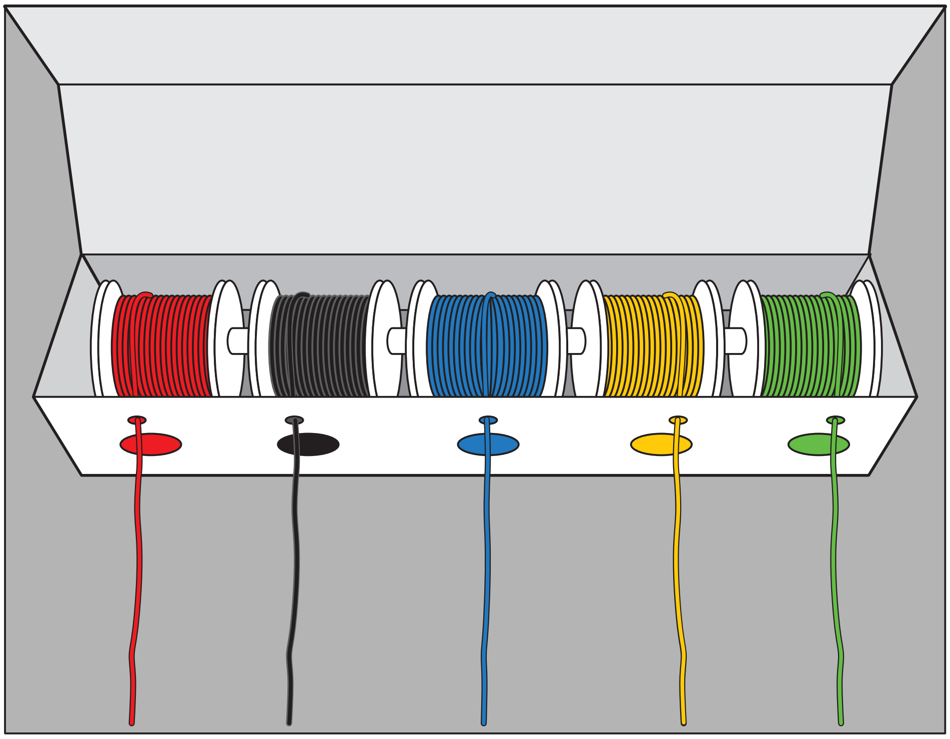


Figure 4.22: Solid-core wire

Alligator clips

These clips are used to connect two wires or to connect one wire to the positive or negative terminal of a component of device. They are a cheap, easy way to assemble or change prototypes or test circuits.

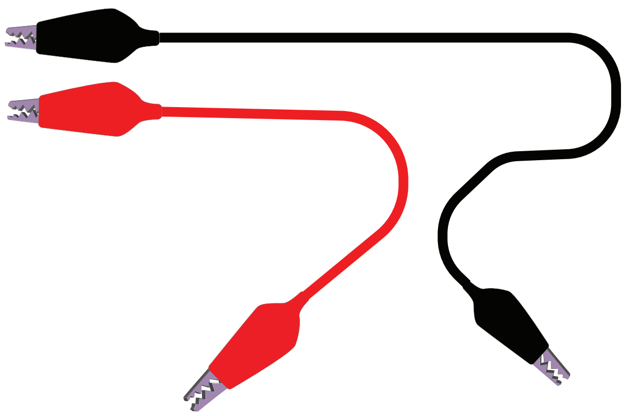


Figure 4.23: Alligator clips

DID YOU KNOW

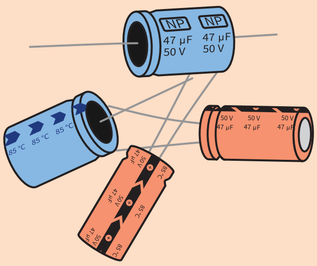
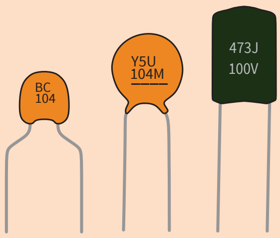
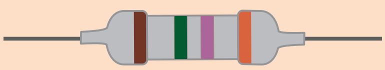
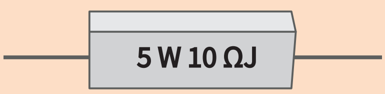
Breadboards do not work with components that have short, flat pins on their sides (called surface-mounted components). These components are designed to be soldered to the surface of a PCB instead of fixed onto a breadboard.

Electronic components

Many electronic components are used a part of breadboard projects. In this section, we will look at only a few that are most used.

Breadboards are designed to work with *through-hole* electronic components. These are electronic components with long, metal 'legs' called *leads*. The leads can easily be pushed through the holes in the breadboard. They are held firmly in place by the clips so that they do not fall out even if you turn the breadboard upside-down, but you can still easily pull on them to remove them.

The table below lists the electronic components and their uses.

COMPONENT	FUNCTION
CAPACITORS	
Have two terminals/pair of conductors separated by an insulator	
Capacitor (polarised)  **Capacitor (non-polarised)**  	Store an electrical charge temporarily so that it can be used again later
RESISTORS	
Resistor • Wire terminal at both ends • Made of copper wire covered in an insulating material • Current flows through it in any direction 0,25 W  0,5 W  1 W  2 W  5 W  	• Controls the flow of current and voltage in a circuit • Converts electrical energy into heat/light

VOCABULARY

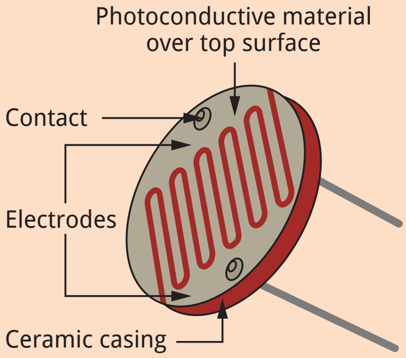
Polarised – *has a positive and a negative terminal and can only be connected in a circuit in one direction*

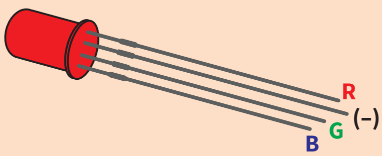
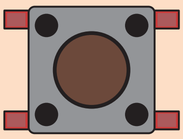
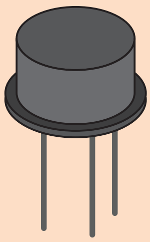
REMEMBER

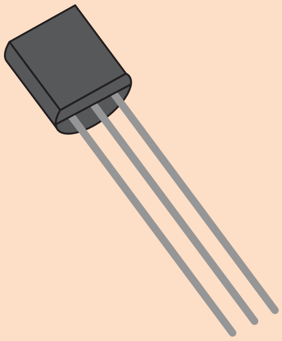
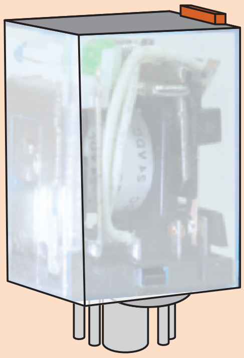
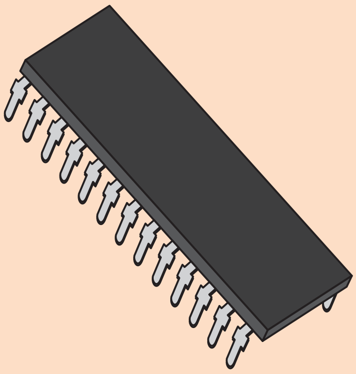
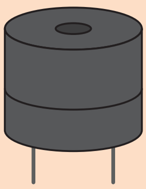
The positive terminal of a component is called the anode *and the negative terminal, the* cathode.

IMPORTANT

Various resistors may have the same value, but their physical sizes could differ. The difference lies in the amount of current each one can handle. We express these differences in watts, for example $\frac{1}{4}$ watt, $\frac{1}{2}$ watt or 1 watt, etc. The higher the current, the higher the watt rating and the bigger the component needs to be.

<table>
<tr><th>COMPONENT</th><th>FUNCTION</th></tr>
<tr><td>Photoresistor (also known as a light-dependent resistor – LDR):
Light-sensitive device used to indicate the presence or absence of light or to measure light intensity

</td><td>• Controls flow of current by changing its resistance when light shines on it (refer to resistors above)</td></tr>
<tr><th colspan="2">DIODES</th></tr>
<tr><td colspan="2">Two-terminal electronic device, e.g. LED and Zener</td></tr>
<tr><td>Diode</td><td>Allow flow of current in one direction only</td></tr>
<tr><td>LED (light-emitting diode)


RGB (red, blue and green) LED

</td><td>When connected in a specific way, an LED emits light when current flows through it
DID YOU KNOW
RGB LEDs combine red, blue and green to produce over 16 million colours.</td></tr>
<tr><th colspan="2">SWITCHES</th></tr>
<tr><td>Push button switch
</td><td>Turns the control circuit on and off</td></tr>
<tr><th colspan="2">SENSORS</th></tr>
<tr><td colspan="2">Three-terminal devices</td></tr>
<tr><td>Passive/Pyroelectric infrared (PIR) sensor
• Work in pairs connected side by side
• When signal difference between the two changes, the sensor will engage
</td><td>To detect heat energy/infrared radiation/ motion in the surrounding environment</td></tr>
</table>

COMPONENT	FUNCTION
Temperature sensor 	Measure or monitor temperature in circuits which control a variety of equipment
RELAYS	
Electrical switches that use **electromagnetism** to open and close circuits – they control one electrical circuit by opening and closing contacts to another circuit	
Relay 	• Opens and closes circuits • Converts a small electrical input to a much larger current
CONVERTERS	
Analogue-to-digital converter (ADC) 	Convert analogue signals to digital signals
AUDIO DEVICE	
Buzzer Uses small, magnetic coil to vibrate a metal disk inside the plastic housing 	Produces different rates and frequencies (pitches) of sound when electrical current moves through it

VOCABULARY

Electromagnetism – *magnetism developed by a current of electricity*

4.2.6 Using a breadboard to present a typical circuit

Even though your circuit does not have to match the diagrams exactly, it is probably best when you are first using breadboards.

Using a breadboard diagram

A *breadboard diagram* is a computer-generated drawing of a circuit on a breadboard. Unlike a circuit diagram or a schematic, which uses symbols to represent electronic components, breadboard diagrams make it easy for beginners to follow instructions to build a circuit because they are designed to look like the 'real thing'.

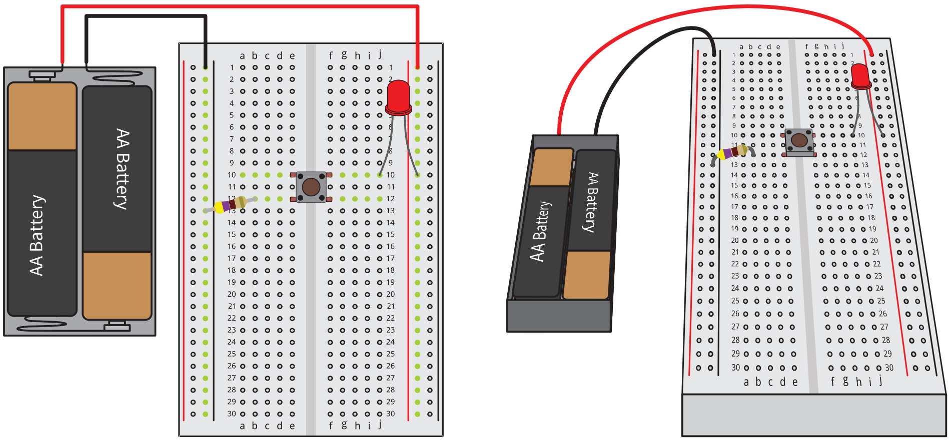


Figure 4.24: Example of a breadboard diagram (left) and an actual breadboard circuit (right)

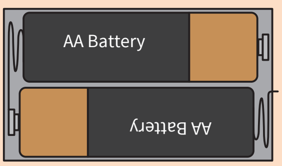
The diagram above (made with a free program called *Fritzing*) shows a basic circuit with a battery pack, an LED, a resistor and a push button. Note that it looks very similar to the physical circuit.

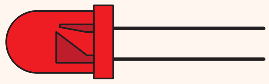
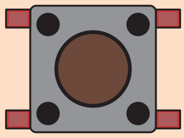
Sometimes, breadboard diagrams might be accompanied by or replaced with written directions that tell you where to connect each component on the breadboard.

Example

1. Connect the red lead of the battery pack to the power rail.
2. Connect black lead of the battery pack to the ground rail.
3. Connect the resistor from hole B12 to the ground rail.
4. Insert the four pins of the push button into holes E10, F10, E12, and F12.
5. Insert the long lead of the LED into the power rail, and the short lead into hole J10.

This information can also be formatted as a table:

COMPONENT	IMAGE	LOCATION
Battery pack		Red lead to + rail Black lead – rail

COMPONENT	IMAGE	LOCATION
LED		Long lead to + rail Short lead to J10
Push button		Holes E10, F10, E12, F12
Resistor		Hole B12 to – rail

There are different ways to change the physical layout of a circuit on a breadboard without changing the electrical connections. For example, the two circuits shown below are electrically identical; even though the leads of the LED have moved, there is still a complete path (called a *closed circuit*) for electricity to flow through the LED (indicated with yellow arrows). So, even if the directions say, 'Insert the long lead of the LEDs into hole F10', the circuit will still work if you use hole H10 instead. However, if you connect the lead to hole F9 or F11, it will NOT work because the different rows are not connected. (Revise the configuration of the breadboard, specifically how the rows are electrically connected.)

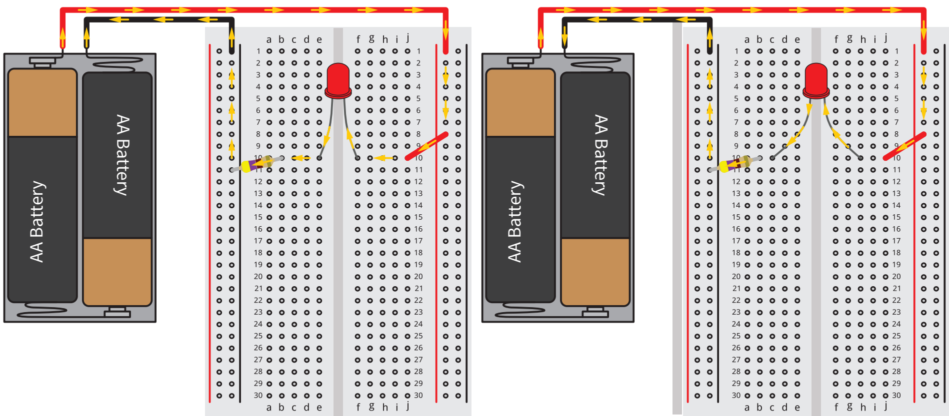


Figure 4.25: The same circuit connected in two different ways

Points to remember when using a breadboard

- Only use solid-core wire of the right gauge or male-to-male jumper wires (±0,6 mm in diameter) as connecting wires for the circuits.
- Use the power rails to connect the power supply and ground (+ and –).
- Make double sure that components or connecting wires are connected to the correct holes.

How to build a circuit

- Follow the breadboard diagram for the circuit, connecting one component at a time.
- Always connect the batteries or power supply to your circuit *last*. This will give you a chance to double-check all the connections before you turn on the circuit for the first time.

Testing the circuit

How you test your circuit will depend on the specific circuit you are building. In general, you should follow these steps:

- Double-check your circuit by comparing it to the breadboard diagram to make sure all your components are in the right place.
- Check the project directions – What is the purpose of the circuit? Is the circuit designed to flash lights, sound a buzzer or respond to a sensor (like a motion or light sensor), for example?
- Turn on the power supply to your circuit. If you see or smell smoke, turn off or disconnect the power supply *immediately*. This means there is a short circuit.
- Follow the project directions for using the circuit; for example, press the push button switch to sound the buzzer or wave your hand in front of a motion sensor to turn on the LED.
- If the circuit does not work correctly, you need to **troubleshoot** or **debug** the circuit. (Refer the common mistakes discussed below.)

VOCABULARY

Troubleshoot – *analyse and solve problems by tracing and correcting faults in the system*

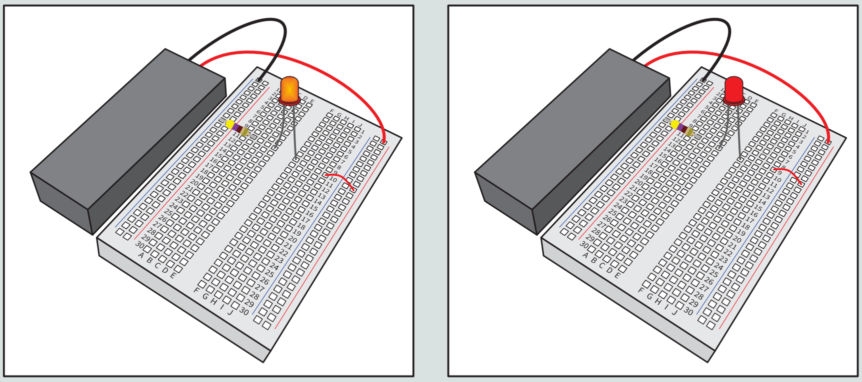
Debug – *identify and remove faults from a system*

Common mistakes

Connecting incorrect rows

EXAMPLE 4.1

Can you spot the difference between the two circuits shown below?



A B

Figure 4.26: Breadboard connections

At first glance, they might look the same. However, when we turn on the battery packs, only the LED on breadboard A will light up. To identify the problem, we need to go back to the breadboard diagram for the circuit. The circuit should match the diagram given alongside.

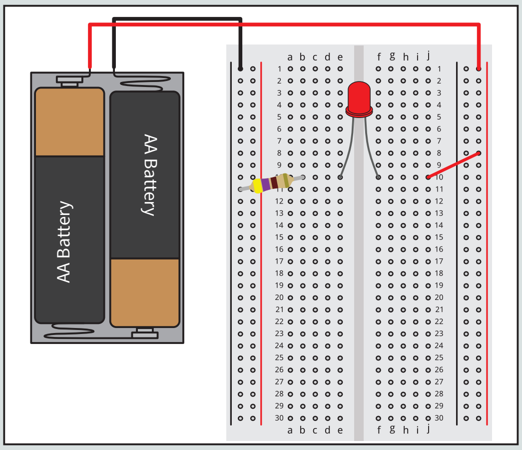


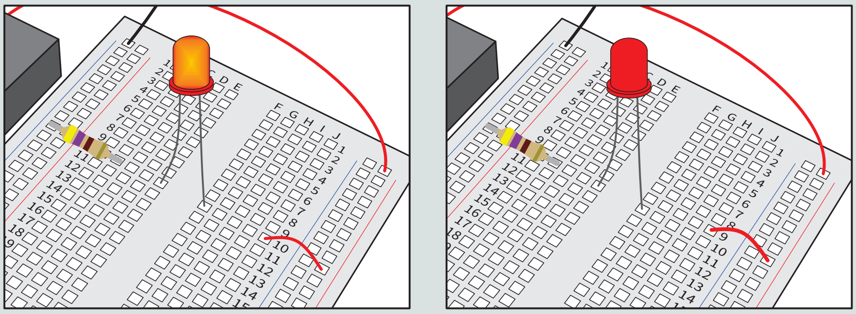
Figure 4.27: Breadboard diagram

REMEMBER

The red lead is always positive (live/power) and the black lead is negative (ground).

EXAMPLE 4.1 (CONTINUED)

If you take a closer look at the two circuits and compare each to the breadboard diagram, you should spot the error. Here is a clue: Check the jumper wire.



A B

Figure 4.28: Breadboard circuits

In circuit A, the red jumper wire is connected to hole J10 and the positive power rail, which matches the breadboard diagram. However, in circuit B, the wire is connected to hole J9 and the positive power rail. Remember, holes in different rows are not electrically connected to each other. So, if the jumper wire in row 9 and the LED in row 10 are not electrically connected, no current can flow to the LED.

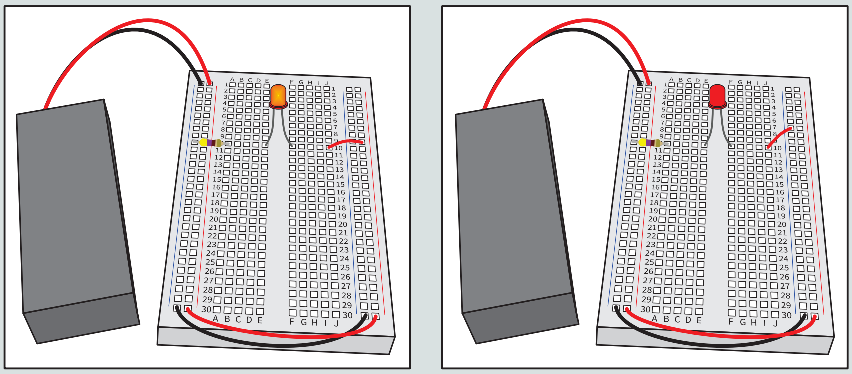
It can be difficult to spot such a tiny error! And it only takes one misplaced wire or component lead to stop a circuit from working. Therefore, you should always double-check the wiring before you test a circuit.

Confusing power and ground

e.g.

EXAMPLE 4.2

Confusing the power and ground rails is another common mistake. Can you spot the difference between the circuits below? Only the LED on breadboard A lights up.



A B

Figure 4:29: Breadboard circuits

In circuit A, the red jumper wire is connected to the positive (+) rail. In circuit B, it is connected to the negative (–) rail. According to the breadboard diagram given above, it should be connected to the positive (+) rail. (Remember that 'positive' and 'negative' can also be referred to as *power* and *ground*.)

eg

EXAMPLE 4.3

Compare the circuits given below. Find the reason why only the LED on breadboard A lights up.

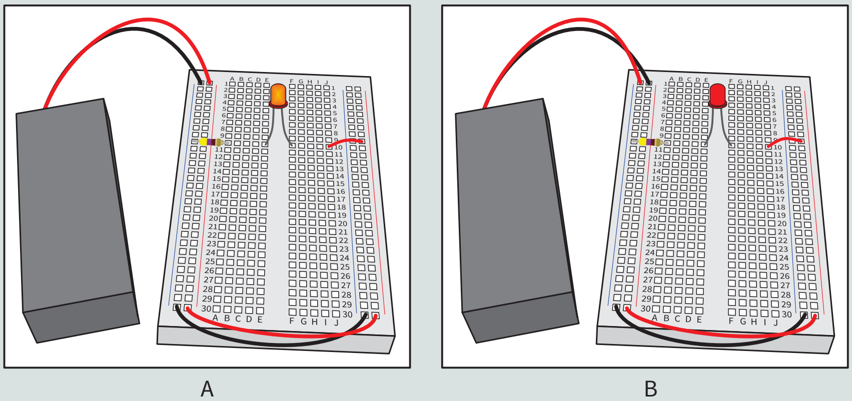


Figure 4.30: Breadboard circuits

REMEMBER

There is no physical difference between the positive and negative rails themselves. Using them only for positive or negative connections, respectively, is not a requirement.

Take a closer look to see if you can spot the problem.

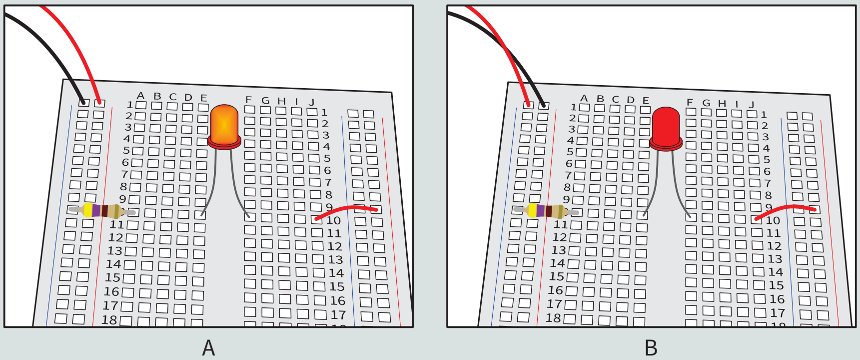


Figure 4.31: Power rail connections

Here the leads of the battery pack/power supply are reversed. The red lead is connected to the negative (–) rail and the black lead to the positive (+) rail. Remember that unlike jumper wires, the colours of battery pack leads *do* matter. Red is used for positive and black is used for negative.

Finally, remember on some breadboards the positive rail is on the left and the negative on the right. On other breadboards this is reversed. Be careful when you switch between breadboards since the left-right positions of the rails may change.

Not pushing leads and wires in all the way

Electronic components and jumper wires have leads of varying lengths. Sometimes the lead is only pushed in partially instead of firmly all the way (until they cannot go any farther). This can result in loose connections that lead to strange circuit behaviour, like an LED flickering on and off.

NOTE

Some components, like LEDs, have very long leads that do not fit all the way into the breadboard. Other components, like pre-cut jumper wires, typically have leads of the right length, so they fit flush against the breadboard.

Compare the figures below. The leads on breadboard A have clearly not been pushed in all the way, as is the case in breadboard B.

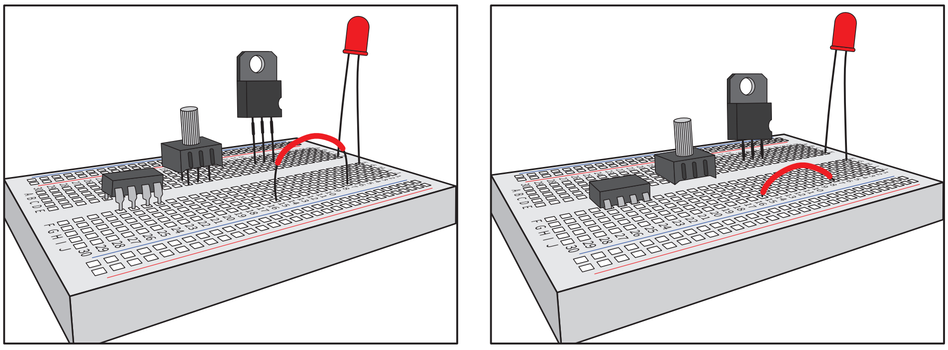


Figure 4.32: Lead connections

Putting components in backwards

As explained in section 4.2.5, some electronic components must be connected in a specific direction. Such components have polarity, i.e. they have a positive side and a negative side that must be connected into the circuit correctly. Other components have multiple pins that all serve different functions. Putting these components into the circuit backwards or facing the wrong way will prevent your circuit from functioning properly.

If your circuit is not working and it contains any of the following components, check to make sure they are inserted the right way:

- Batteries
- ICs
- Polarised capacitors
- Diodes (allow current to flow in one direction only).

VOCABULARY

Short circuit – *an electrical circuit that allows a current to travel along an unintended path, especially one resulting from the unintended contact of components and resulting diversion of the current*

Short circuits

If you ever see or smell smoke when building a circuit, it is probably due to a **short circuit**. You must immediately disconnect the battery pack. Short circuits occur when two components that are not supposed to be connected accidentally make contact. This can happen when:

- components are inserted into the wrong rows or rails; or
- exposed metal parts of components make contact.

For example, resistors and LEDs have long metal leads. If you are not careful, these leads could make contact and cause a short circuit. Always make sure the leads cannot make contact with those of another component.

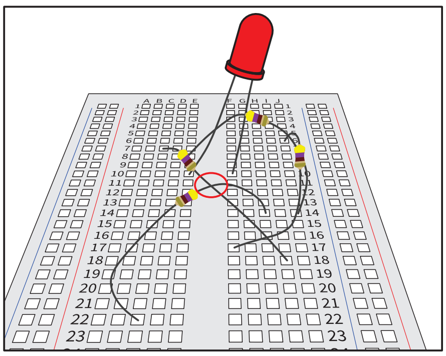


Figure 4.33: Connection that can cause a short circuit

Sometimes short circuits are harmless. They may just prevent the circuit from functioning properly until they are located and fixed. However, sometimes short circuits can melt or burn out components and cause permanent damage.

IMPORTANT

A short circuit between the power and ground rails must be avoided because it can get hot enough to burn you and even melt the plastic breadboard and wire insulation!

INTEGRATED CIRCUITS (ICS)

Enrichment

As you know, ICs or chips are specialised circuits that serve a variety of purposes. Many ICs have what is called a *dual in-line package* (or DIP) which means they have two parallel rows of pins. The gap in the middle of a breadboard (between columns E and F) is just the right width for an IC to fit, straddling the gap, with one set of pins in column E and the other in column F. Projects that use ICs will always tell you to connect them to the breadboard in this manner.

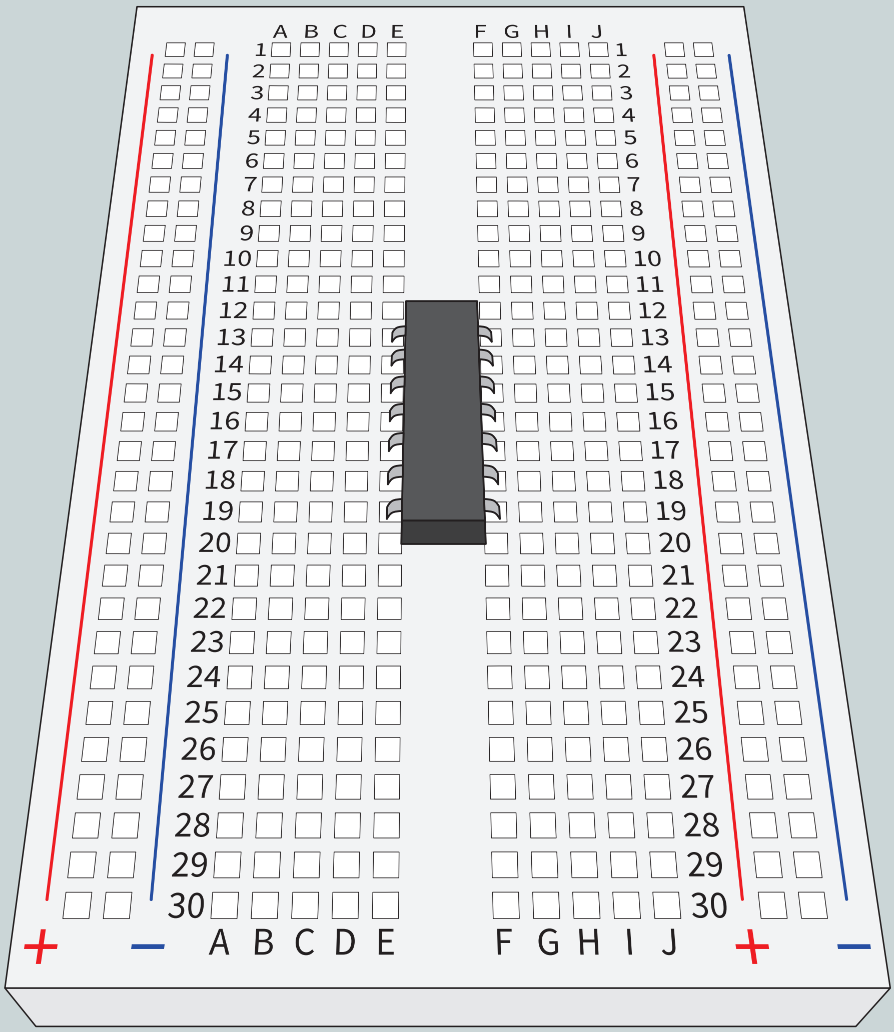


Figure 4.34: Connecting an IC to a breadboard

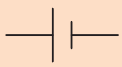
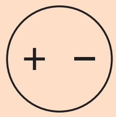
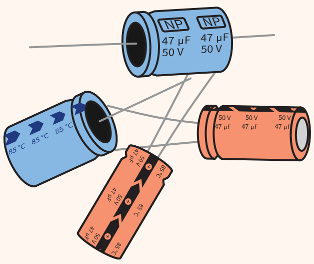
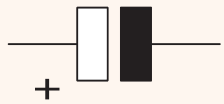
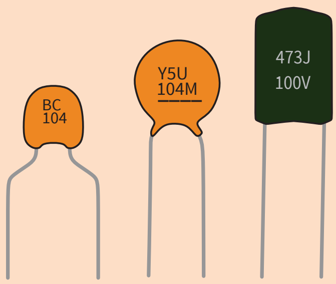
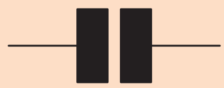
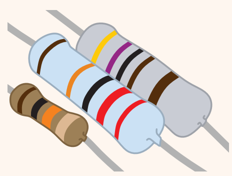
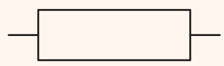
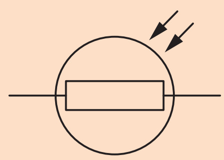
4.2.7 Circuit diagrams and basic electrical component symbols

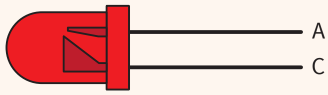
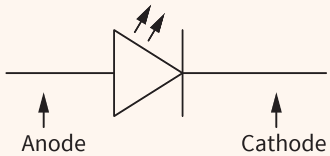
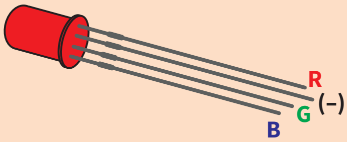
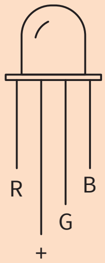
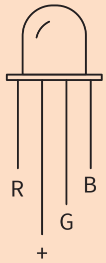
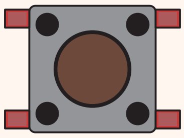
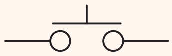
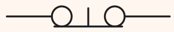
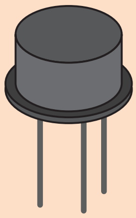
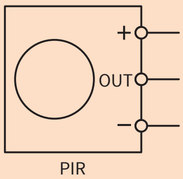
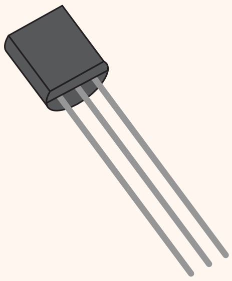
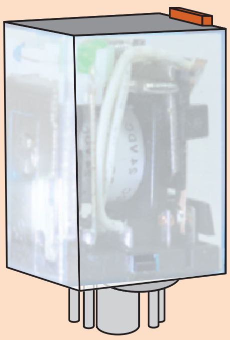
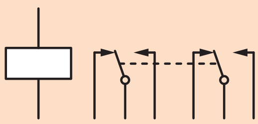
VOCABULARY

Circuit diagram – *a simplified representation of the components of an electrical circuit using either the images of distinct parts or standard symbols*

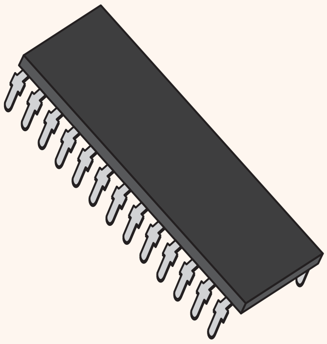
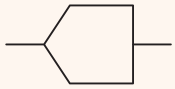
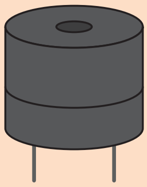
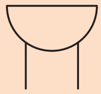
Circuit diagrams, also called *schematics*, are a way to represent a circuit using symbols for each component. Circuit diagrams, as opposed to breadboard diagrams, are used by professionals in the electrical/electronic field when they design circuits, and they are much more convenient for more complicated circuits.

For easy identification of electrical/electronic components, each component has its own unique symbol. The table below lists some of the components used in electrical circuits and the symbol used for the easy identification of each component. Knowing these symbols will help you understand an electrical circuit.

COMPONENT	SYMBOL			
Power sources	 Battery – single cell	 Battery – multicell	 AC supply	 DC supply
Capacitor (polarised) 				
Capacitor (non-polarised) 				
Resistors 				
Photoresistor (also known as a *light-dependent resistor – LDR*)				

COMPONENT	SYMBOL	
LED  A – anode C – cathode		
RGB LED 	 Common anode	 Common Cathode
Push button switch 	 Normally open (on)	 Normally closed (off)
Passive/Pyroelectric infrared (PIR) sensor 		
Temperature sensor 	?????	
Relay 		

???

COMPONENT	SYMBOL
ADC 	
Buzzer 	

DID YOU KNOW

The International Electrotechnical Commission (IEC) is an organisation that prepares and publishes international standards for all electrical, electronic and related technologies. The Institute of Electrical and Electronics Engineers (IEEE) is a global organisation of professionals working towards the development, implementation and maintenance of technology-centred services. Both these organisations have published a list of prescribed electrical component symbols. However, we use symbols recognised and prescribed by the IEC.

Examples of electrical circuit drawings

Unlike breadboard diagrams, circuit diagrams only show electrical connections between components. They do not necessarily correspond with the physical layout of the components on a breadboard.

The circuit illustrated below consists of a battery, resistor and an LED connected in series (explained in section 4.2.8). Note the difference between the circuit drawing and the breadboard diagram.

PLEASE CHECK! This component is not indicated in circuit diagram!?]

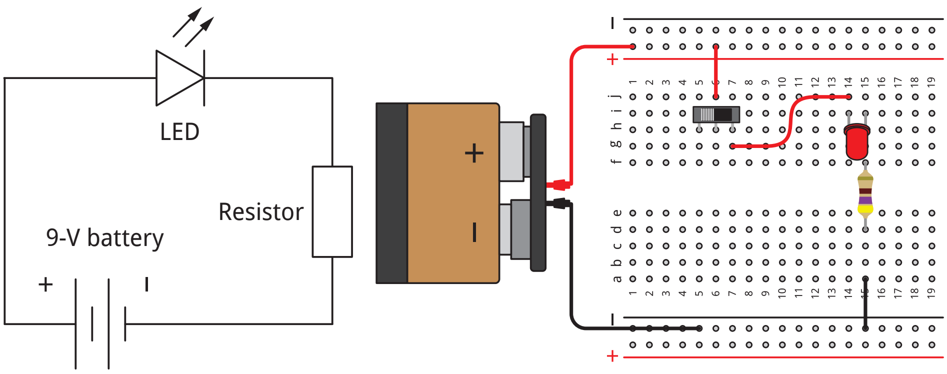


Figure 4.35: Circuit drawings

If you have a hard time understanding this, try using your figure to trace the wire (the red/black line) through the circuit, starting at the positive terminal of the battery. Notice how your finger still moves 'through' each component in the same order, even though they may have been physically rearranged.

Here is another example of a circuit drawing. Here we have used a battery, switch (open), an LED and a resistor.

PLEASE CHECK – I haven't come across circuit diagrams like this one?

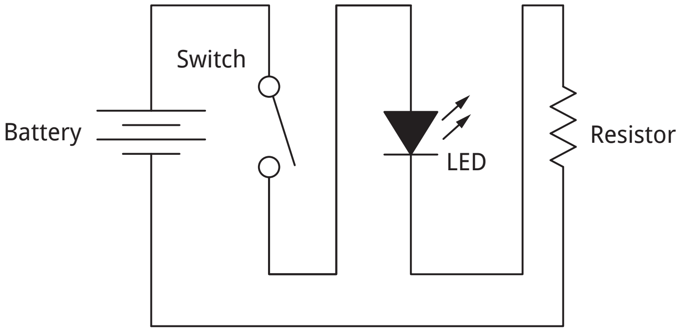


Figure 4.36: Circuit diagram

NOTE: MORE examples needed here, according to syllabus – " ... examples of circuits that include all the components listed in 4.2.4" which also includes:
Capacitors, push button, photoresistor, PIR, temperature sensor, relay, ADC and buzzer]

NOTE

It takes some practice to learn how to read and interpret circuit diagrams. Most beginner electronics projects – especially those on the Science Buddies *website – will provide breadboard diagrams that you can follow to build a circuit.*

4.2.8 Breadboard connection to GPIO

[NOTE: Syllabus requirements not met]

Before we look at how to set up a breadboard connected to a GPIO as power source, let us revise the difference between a *series circuit* and a *parallel circuit* (refer to Module 3 section 3.1.4).

Series and parallel circuits

A *series circuit* consists of several devices, each of them linked up one after another in a single loop. Even though different devices have different voltages across them, the same current flows through every device in the series circuit. If any one of the devices in a series circuit breaks down, the whole circuit will fail. What is more, the more lamps that one connects in series, the dimmer they shine. In the circuits below, two lamps are connected in series.

REMEMBER

An electric circuit *is a closed path in which electrons move to produce electric currents. The circuit is a network consisting of a closed loop, giving a return path for the current. It is made up of interconnected electrical components. The circuit directs and controls electric current to perform various functions.*

The figure below shows two bulbs connected in series.

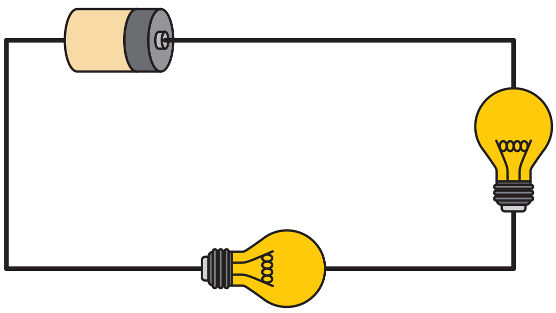


Figure 4.37: Series circuit

Please check if I changed this right.

The circuit drawing below shows a three LEDs connected in series.

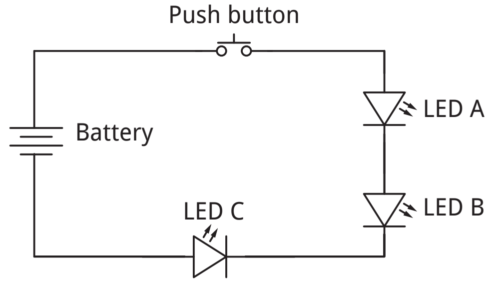


Figure 4.38: Circuit drawing of a series circuit

In parallel circuits, different devices are arranged so that a single source supplies voltage to separate loops of wire. The voltage across every device in the circuit is the same. If one device fails, the others will continue to work. For instance, if two light bulbs are linked up in parallel and one is unscrewed, the other one will work. In addition, all the lights in these circuits burn equally brightly, irrespective of how many are connected.

The figure below shows two bulbs connected in parallel.

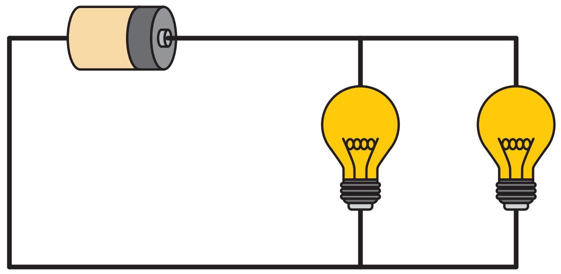


Figure 4.39: Parallel circuit

The circuit drawing below shows three LEDs connected in parallel.

Please check if I changed this right.

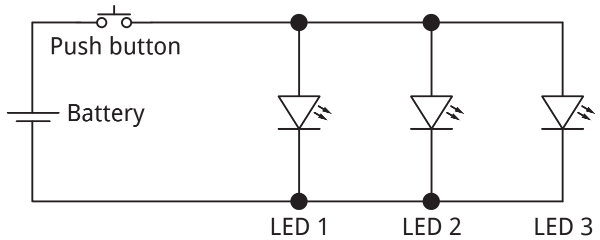


Figure 4.40: Circuit drawing of a parallel circuit

Breadboard connected to GPIO

[Syllabus: Set up a breadboard connected to a GPIO as power source with applicable components to illustrate the difference between a series and a parallel circuit

- Use resistors, push buttons and LED in:
- Series circuit
- Parallel circuit

Series connection

[provide missing info]

Parallel connection

[provide missing info]

4.2.9 Analogies for parallel and series circuits

[PLEASE CHECK!]

Parallel and series circuits are revised in the previous section. In this section, analogies are used to explain the flow of current. These stories/comparisons make it easier to understand concepts because we cannot see electrons and how they flow through the circuit.

Parallel circuit

Let us return to the water analogy used in Module 3 to explain resistance and current flow. A parallel circuit is like a river breaking into three streams which later re-join to from one river again. The amount of water flowing in the river is equal to the sum of the amounts of water flowing in the individual streams (voltage is the same). If one of the streams is blocked, the other two will continue to flow as strongly as before. In the same way, if there is a break in one part of the circuit, the rest of the circuit will still work. The components that are connected in parallel experience the same voltage drop across all the components.

Series circuit

In a single closed loop of pipe (circuit), the pressure of the pump (power generated by the battery) drives the water (electrons) through the loop (circuit) at a certain flowrate (current). If there is no resistance in the loop, the water flows fast and freely, depending on the pressure of the pump (the voltage supplied by the battery). However, if there is a narrowing in the pipe (resistor), the flow of water slows down – as the resistance to flow is increased, the volume flowrate of the water decreases proportionately. The same happens as the resistance is increased in a circuit.

This circuit can also be compared to a big loop of rope. One person acts as the battery and pulls the loop through his/her hand. Another person acts as the resistor. He/she hold onto the rope to resist the pull. The friction that is felt represents the energy transferred as heat.

eLINKS

To learn more about the analogies that explain series and parallel currents, visit this link:

futman.pub/Analogies

4.2.10 Passive and active electronic components

[MUST BE CHECKED!]

Electronic components are used to control current or voltage, to amplify electrical signals and to bring about mechanical changes. In this section, the focus is on passive and active components.

Passive components

A *passive component* does not need an external power source to function.

These components have the following characteristics:

- Usually have two terminals
- Use the power or energy from the circuit
- Can store energy in the circuit in the form of current or voltage
- Cannot control the flow of the current.

Active components

An *active component* relies on an external power source to control or change electrical signals.

The characteristics of active components:

- Rely on an external power source (battery) to control or adjust electrical signals
- Can produce or deliver energy to the circuit in the form of voltage or current
- Can control the flow of current.

The table below summarises the differences between active and passive electronic components:

ACTIVE COMPONENTS	PASSIVE COMPONENTS
Transform/Inject power/energy into a circuit	Use power/energy in a circuit
Produce energy in the form of voltage/ current	Store energy in the form of voltage/current
Have function and provide gain – amplify	Do not have function to provide gain
Control flow of current	Cannot control flow of current
Require external and conditional source to operate in the circuit	Do not require any external source to operate in the circuit
Energy donors	Energy receivers
Examples: diodes, transistors, ICs	Examples: resistors, capacitors, motors, etc.

4.2.11 Analogue and digital components

Before we look at the different components, let us examine the terms. You may think you know the difference between concepts *analogue* and *digital* based on the type of watch you wear or the multimeters that were used in previous modules. However, everything electrical can be divided into these broad categories. Working in an electrical or electronic environment, you will deal with both analogue and digital signals, inputs and outputs. Data can be displayed by either of these means.

Analogue

In an electrical field, **analogue** refers to data or signals that gradually change from one level to the next. Information or data can be sent as electronic signals of changing frequency or **amplitude** on a carrier wave. Analogue signals are made up of a continuous range of values, where each point can have a slightly different value as time varies. These signals are represented as **sine waves**.

VOCABULARY

Analogue – *signals that can vary or change*

Amplitude – *maximum difference of an alternating electrical current or potential from the average value*

Sine wave – *a continuous, uniform wave with a constant frequency and amplitude*

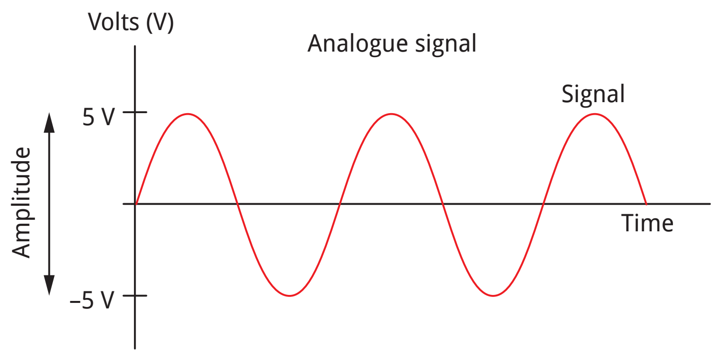


Figure 4.41: Analogue signal

Digital

In **digital** electronics, quantities are counted rather than measured. When you count something, you get an exact result. When you measure something, you get an approximate result.

Digital signals and objects deal with what we call the **discreet** or **finite** – they have a limited set of values. Electronic signals are actually time-varying 'quantities' – these are quantities that change over time. In an electrical field, the quantity that is varies depending on the instance (time) is voltage or current.

Information can be classified as digital when it is represented by a string of zeros (0) and ones (1). The digital signals vary only between two logic states: high (which equals 0 volts) and low (which equals 5 volts).

VOCABULARY

Digital – *signals or data expressed as series of the digits 0 and 1, typically represented by values of a physical quantity such as voltage*

Discreet – *a signal that exists in one of two states (either on or off, true or false)*

Finite – *limited in size or extent*

Example

- On = logic 1 — This is the signal to *turn on* certain circuits, e.g. a transistor.
- Off = logic 0 — This is the signal to *turn off* certain circuits, e.g. a light.

Digital signals are represented as square waves, as illustrated in the figure below. Because the values in a digital signal must be either 0 or 1, the signal can easily be corrected to eliminate (get rid of) noise. They can also be received clearly even after travelling awfully long distances.

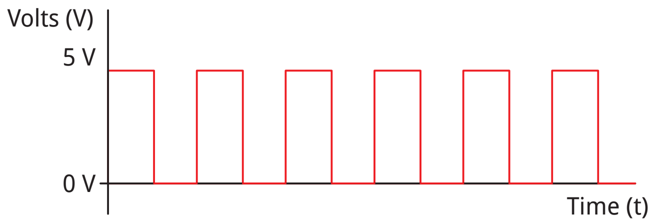


Figure 4.42: Digital signals

The difference between analogue and digital can be summarised as follows:

DIGITAL	ANALOGUE
Signals are discrete and contain only distinct values (0 or 1)	Signals are continuous as time varies
Makes fault finding easier	Fault finding is more difficult
Signal can easily be corrected to remove noise/interference	More prone to noise interference
Represented by a square wave	Represented by a sine wave or a continually varying wave
Digital signals use less power	Analogue signals use more power
High volume of information can be stored or transmitted	Less information can be stored or transmitted

Analogue and digital components

Most of the important electronic components are inherently *analogue*. This includes resistors, capacitors, diodes and transistors, to name but a few. Circuits built with a combination of only these components are usually analogue circuits. Analogue circuits are much more difficult to design than digital circuits. These circuits are more inclined to be disrupted by noise – small, undesired variations in voltage. These small changes in voltage levels of an analogue signal can produce huge errors.

Digital components, on the other hand, operate using discreet signals (explained above). They are made up of a combination of transistors and **logic gates**. Microcontrollers or other computing chips are examples of digital components. As mentioned above, digital circuits use the binary scheme for digital signalling.

VOCABULARY

Logic gate – *an electronic circuit having one or more than one input and only one output; acts as a building block for digital circuits*

4.2.12 Defining *sensor*

VOCABULARY

Sensor – *device that detects a physical parameter of interest (e.g. heat, light, sound)*

A **sensor** can be defined as a device which detects a change in environmental conditions (e.g. light, sound, temperature, etc.) and presents the output in an easy-to-read format. For example, in a mercury thermometer the mercury simply expands when the temperature rises to give the user a reading. There are no electrical implications or changes.

Sensors can sense a wide range of energy forms. A thermistor is an example of a sensor that converts a change that is non-electrical (heat) to a change in electrical quantity (resistance).

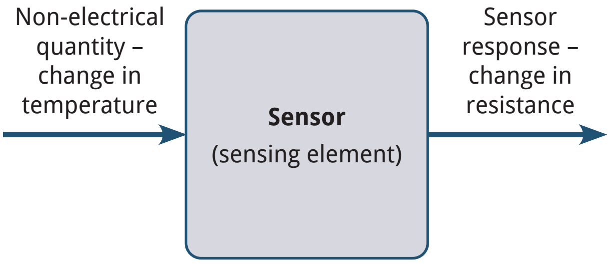


Figure 4.43: How sensors work

To summarise

SENSOR
Detects a change in a physical environment
A component in itself
Requires an additional circuit to process its output signal into a readable form
Output is analogue
Output cannot be directly applied to any other system
Does not require external power to operate

4.2.13 Use of sensors as part of electrical circuits

In our everyday lives, we often use different types of sensors in our power systems without even realising it. They are used in electronic appliances, load-control systems, home automation, etc.

Sensors can be divided into analogue and digital sensors as well as into the following types:

- Temperature sensors
- IR sensors
- Ultrasonic sensors
- Touch sensors
- Proximity sensors
- Pressure sensors
- Smoke and gas sensors.

In this module, we are only focusing on the following three types:

- the PIR motion sensor;
- pressure-pad or force-sensing resistors (FSR); and
- the temperature sensor.

PIR motion sensor

PIR is the abbreviation used for *passive* or *pyroelectric infrared* sensors. It is more complicated than other sensors because there are many variables that affect its input and output. Everything gives off some level of infrared radiation (referred to as *IR*). The hotter something is, the more radiation is given off or *emitted*. A PIR motion sensor can detect levels of IR caused by movement, e.g. when a human has moved in or out of the sensor's range.

It consists of a pyroelectric sensor housed in a sealed metal can. There is a window made of IR-transmissive material to protect the sensing element. Behind the window, there are two balanced sensors. Think of the sensor as having two halves. They are wired so they cancel each other out. Each half is made of special material that is sensitive to IR. The lens allows the two halves to 'see' for some distance. If one half detects more IR than the other, it causes a positive differential change between the two halves. As the source of radiation leaves the sensing area, the reverse happens. The other half generates a negative differential change.

These sensors are commonly used in alarm systems and other gadgets and appliances.

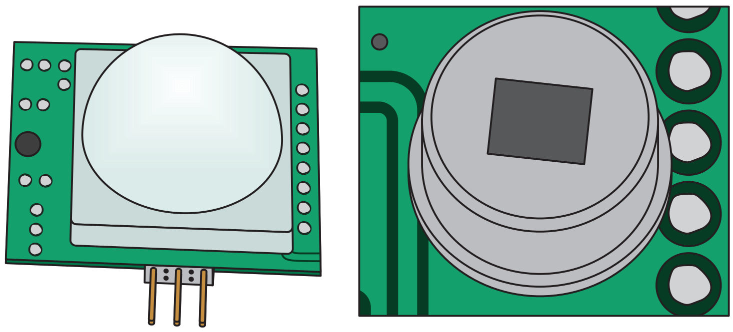


Figure 4.44: PIR motion sensor

Pressure pads (force-sensing resistor – FSR)

Force-sensing resistors, also called *printed-force sensors*, *pressure pads* or *force-sensitive resistors*) are used in embedded components. An FSR is a material whose resistance changes when a force, pressure or mechanical stress is applied. It consists of a semiconductor material contained between two thin layers or films, as shown below.

The sensor detects or measures the following:

- The rate of change in force
- The relative change in force
- Contact and/or touch
- Force levels that trigger an action of sorts.

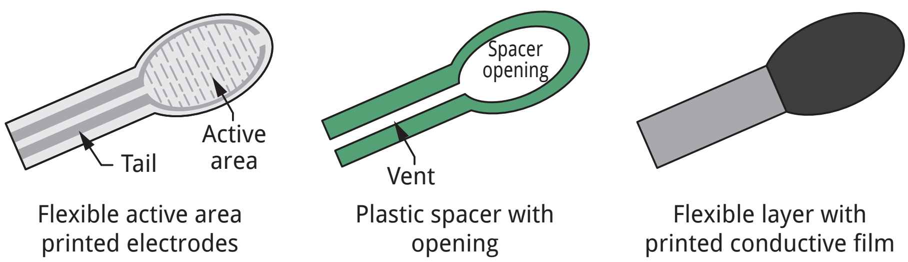


Figure 4.45: Parts of an FSR

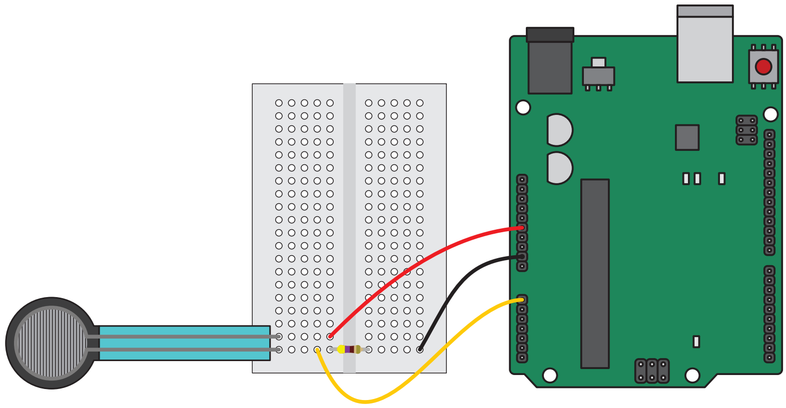


Figure 4.46: FSR connected to a breadboard and GPIO

Temperature sensor

This is one of the most used sensors. A temperature sensor is an electronic device that detects thermal parameters and provides signals to the inputs of control and display devices. It relies on a resistance temperature detector (RTD) or a thermistor to measure the temperature and covert it to an output voltage.

They can be used:

- for switching the load in a circuit on or off at a specific temperature detected by the sensor; or
- to measure the thermal characteristics of gases, liquids and solids.

NOTE

An RTD *is a sensor whose resistance changes as its temperature changes. The resistance increases as the temperature of the sensor increases.*

A thermistor *is an electrical resistor whose resistance is greatly reduced by heating. It is used for measurement and control in circuits.*

4.2.14 Purpose of analogue-to-digital converter

As the name suggests, an analogue-to-digital converter converts an analogue signal such as voltage to a digital or binary form so that it can be read and processed by a microcontroller. (Refer to section 4.2.11, where analogue and digital components are discussed.) Today, most microcontrollers have built-in ADC converters.

[Subject specialist to check content as well as illustration below, e.g. Vdd?]

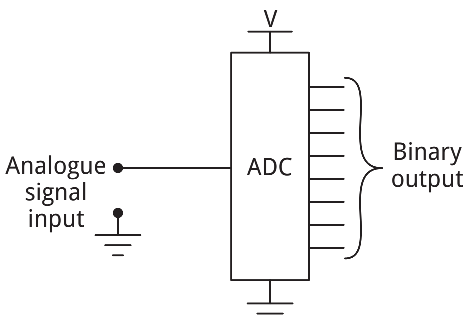


Figure 4.47: Diagram of an ADC

Activity 4.2 **Work on your own**

1. Differentiate between a *breadboard* and a *breakout board*.
2. State whether the following statements are TRUE or FALSE.
   - **2.1** If the leads of an LED is inserted into row B1 and D3, it will be electrically connected to the breadboard.
   - **2.2** Power rails are metal, conductive strips that typically run along the entire length of the breadboard.
   - **2.3** A battery snap can be used to connect power rails on opposite sides of the breadboard.
   - **2.4** Polarised components like capacitors can be connected in any direction in a circuit.
   - **2.5** Relays can be used like switches to open and close circuits.
3. Name the TWO most important considerations when you buy jumper wires.
4. List THREE important points to remember when you connect components to a breadboard.
5. Make neat drawings of the symbols used for each of the following circuit components. Provide labels where necessary.
   - **5.1** LED
   - **5.2** Resistor
   - **5.3** Normally closed push button switch
   - **5.4** ADC
   - **5.5** Buzzer
6. Use neat, labelled drawings to illustrate the difference between a series circuit and a parallel circuit. Each circuit must contain the following components: Battery, switch and three LEDs.
7. What is the MAIN difference between *active components* and *passive components*.
8. Give THREE examples of analogue circuit components.
9. Define *sensor* as used in an electrical context.
10. What does each of the following abbreviations mean?
    - **10.1** PIR
    - **10.2** FSR

4.2.15 Incorporating various components as part of design and creation of an artefact or prototype solution

[More info/guidelines needed here AND in LG. What is provided here is not helpful at all.]

Circuit and code design

A project is usually inspired by a specific idea or a need that has been identified. There are several websites that offer projects ideas, schematics and code samples. If you have a specific project in mind, using a search engine such as Google might produce a variety of results.

Dedicated sites like those maintained by Arduino, the Raspberry Pi foundation, Hackster and Instructables are also good sources of both circuit and code samples. Manufacturers' data sheets can be used as specification sheets and are great resources. They often include both schematics and code samples. And of course, there is GitHub. There are other code repositories as well, but GitHub is certainly the most well-known. You can find code and documentation for almost anything here, and as most of the code is open source you can start using it right away.

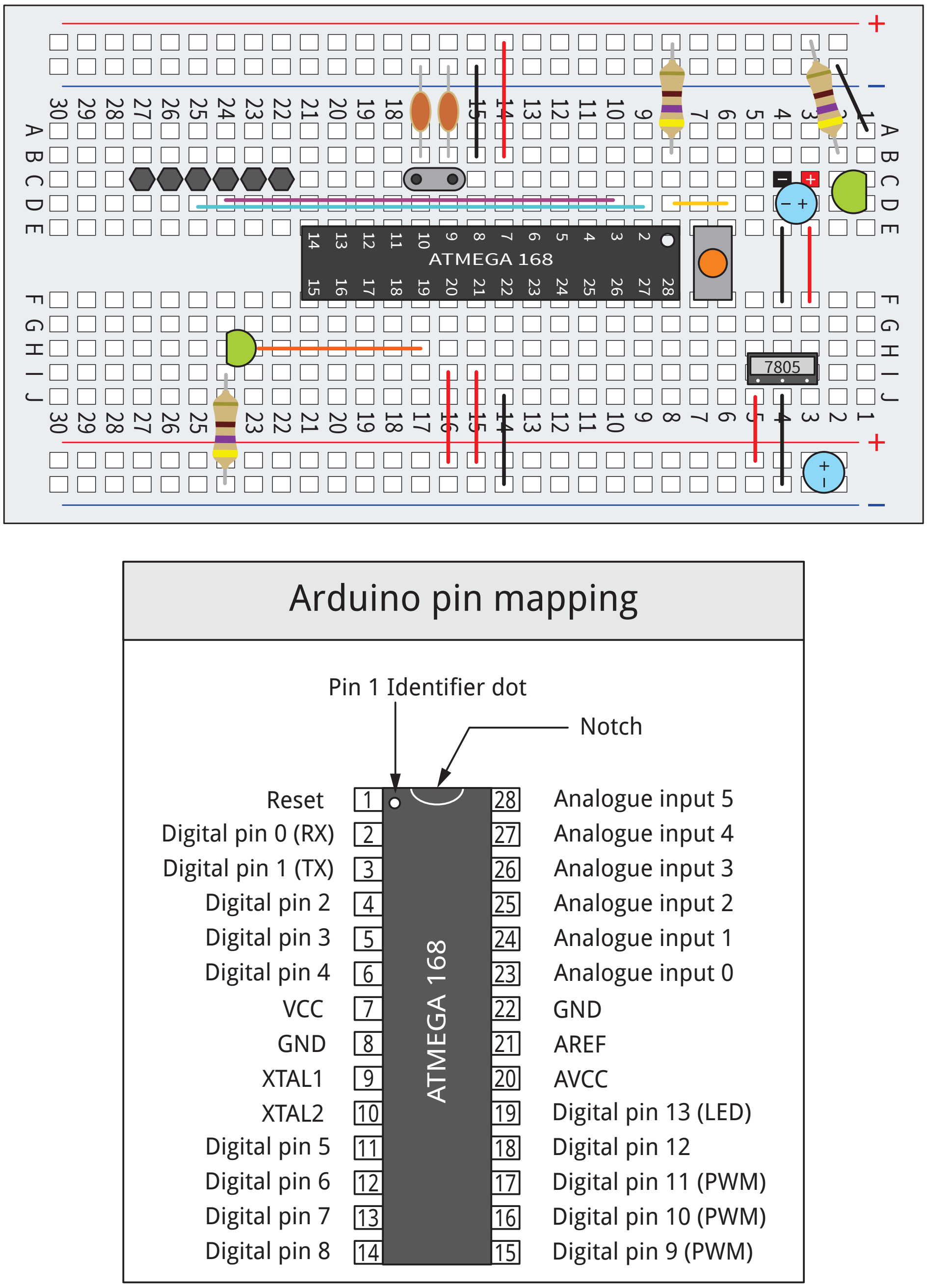


Figure 4.48: Example of breadboard and pin setup

Solderless breadboarding

Creating a temporary but functional version of the circuit you have designed on a breadboard is an essential part of the design process for most projects. For small projects you might be tempted to skip this step and move straight to the construction phase, but that is never a very good idea!

You will also need a source of power for your design. A bench power supply is ideal, but a USB power supply will work for most logic circuits. Batteries are always an good option, although the need to recharge or replace them periodically can be a nuisance.

Testing and troubleshooting

??? [no suggestion re. how to get here?]

Once you have everything wired up and (if necessary) programmed, it is time to test it.

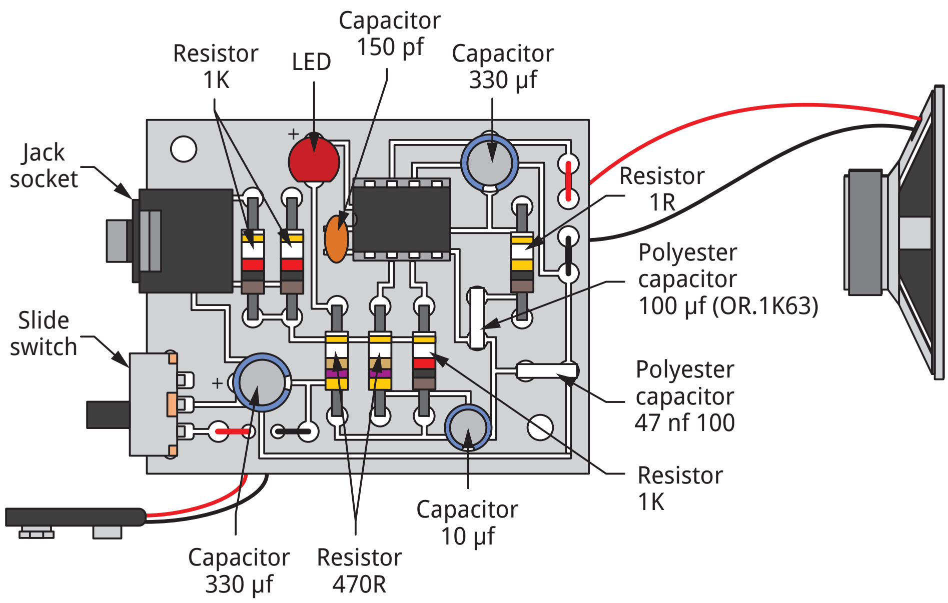
Depending on the nature of the product there will be different tests to perform. Aside from the obvious – making sure that it works – you might want to see how much current your invention is consuming or whether the components are getting too hot.

When you discover a fault, it is time to troubleshoot. If your project involves the use of only hardware, you need to determine if the connections are all good, check the polarity components and whether the voltages are correct. If you have used software, you need to determine whether the problem is being caused by a code or a hardware error. A good way to do this is to write simple diagnostic programs to check specific functions.

Prototyping or building

The final stage involves putting everything together. You might be building a single device and with no intention of creating any copies. On the other hand, you may want to reproduce your design, commercially or because you need several units for yourself. At this stage, you can use a PCB or a *perfboard* – a thin, rigid sheet used for prototyping electronic circuits; it has pre-drilled holes, ringed by round/square coper pads, spaced at standard 2,54-mm intervals across a grid.

[I'm not sure whether these images are relevant? I have deleted the rest as they do not serve to explain the content at all.]



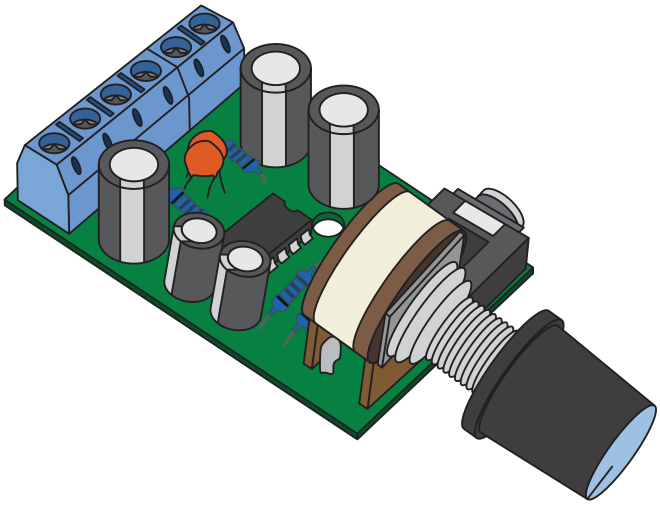


Figure 4.49: Amplifier printed circuit board parts

Practical activity

Practical activity needed with clear instruction and a memorandum in LG

Summative assessment

[To be set once content has been checked and finalised]