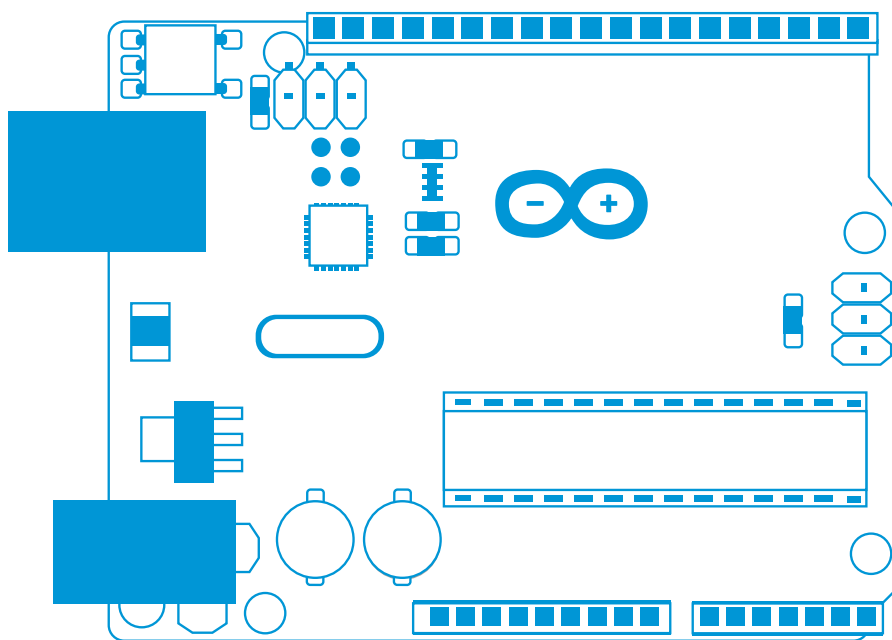
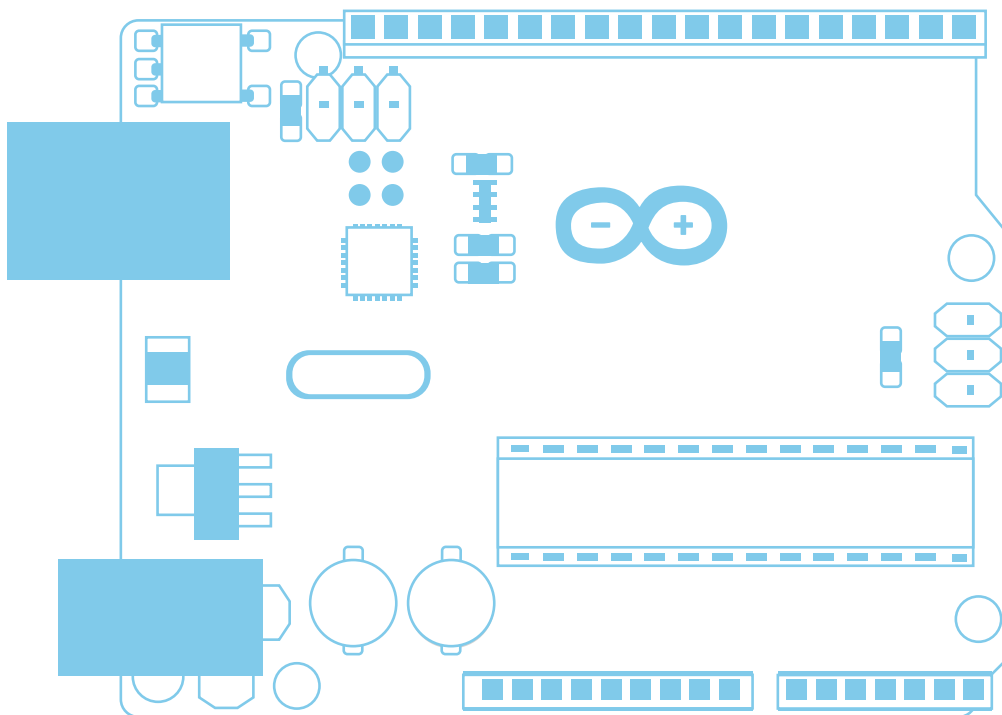


teach with space

→ MEET ARDUINO!

Introduction to Arduino computing using C++





Fast facts	page 3
Summary of activities	page 4

Activity 0: Getting started	page 5
Activity 1: Make me blink!	page 8
Activity 2: Blink an S.O.S.!	page 11
Activity 3: Measuring temperature	page 12
Activity 4: Measuring pressure	page 16
Activity 5: Measuring altitude	page 18

Student worksheets	page 19
--------------------	---------

Links	page 34
-------	---------

teach with space – meet arduino | T04.1
www.esa.int/education

The ESA Education Office welcomes feedback and comments
education@esa.int

An ESA Education production

Copyright 2018 © European Space Agency

→ MEET ARDUINO!

Introduction to Arduino computing using C++

Fast facts

Age range: 14-20 years old

Curriculum links: programming, electronics

Complexity: Medium

Lesson time required: 90-120 minutes

Location: Indoor

Includes the use of: Arduino software and hardware

Keywords: Arduino, Sensor, Code

Outline

Students will explore technology used in space through the Arduino tool. They will build circuits to blink an LED and to measure temperature, pressure and altitude. The basics of programming in C++ will be introduced using the Arduino IDE (Integrated Development Environment) software. This set of activities could be the starting point for future participation in the [CanSat competition](#).

Learning objectives

- Improve analytical skills
- Develop understanding of a programming language
- Develop understanding of building circuits
- Understand how sensors can be used to collect data
- Improve teamwork skills

Activities - Content and outcome

	Title	Description	Outcome	Requirements	Time
1	Getting started	An introduction to the components used throughout the activity	Students will become familiar with required components and their functions	Basic understanding of electronic components	10 minutes
1	Make me blink!	Students will build their first Arduino circuit	Students will be able to control an LED with code they have written	Completion of previous activity	15 minutes
2	Blink an S.O.S	Students will learn how to send more complex messages using an LED	Students will be able to write the code and program the LED to send an S.O.S	Completion of previous activities	15 minutes
3	Measuring temperature	Students will use a thermistor to measure the temperature of their surroundings	Students will be able to build more complex circuits and use them to take measurements of the environment	Completion of previous activities	20 minutes
4	Measuring pressure	Students are shown how to use a pressure sensor to measure the local air pressure	Students will be able to build more complex circuits and use them to take measurements of the environment	Completion of previous activities	20 minutes
5	Measuring altitude	Students are shown how to determine altitude from local air pressure measurements	Students will be able to confidently build circuits and manipulate code to take measurements of their surroundings	Completion of previous activities	20 minutes

→ Activity 0: Getting started

Introduction

Arduino is an open-source platform used to build a wide variety of electronic projects, including the possibility to simulate real space missions like ExoMars (<http://exploration.esa.int/mars/>). Arduino incorporates both a physical, programmable circuit board (hardware) and a piece of software or IDE (Integrated Development Environment) that runs on a computer.

The Arduino hardware and software was designed for artists, designers, hobbyists, hackers and people interested in creating interactive objects and environments. Arduino can be used to interact with buttons, LEDs, sensors, motors, speakers, GPS units, cameras, the internet, and even your smartphone or TV! This flexibility, combined with the facts that the Arduino software is free, the hardware boards and sensors are relatively cheap, and both the software and hardware are intuitive, has led to a large community of users (www.arduino.cc).

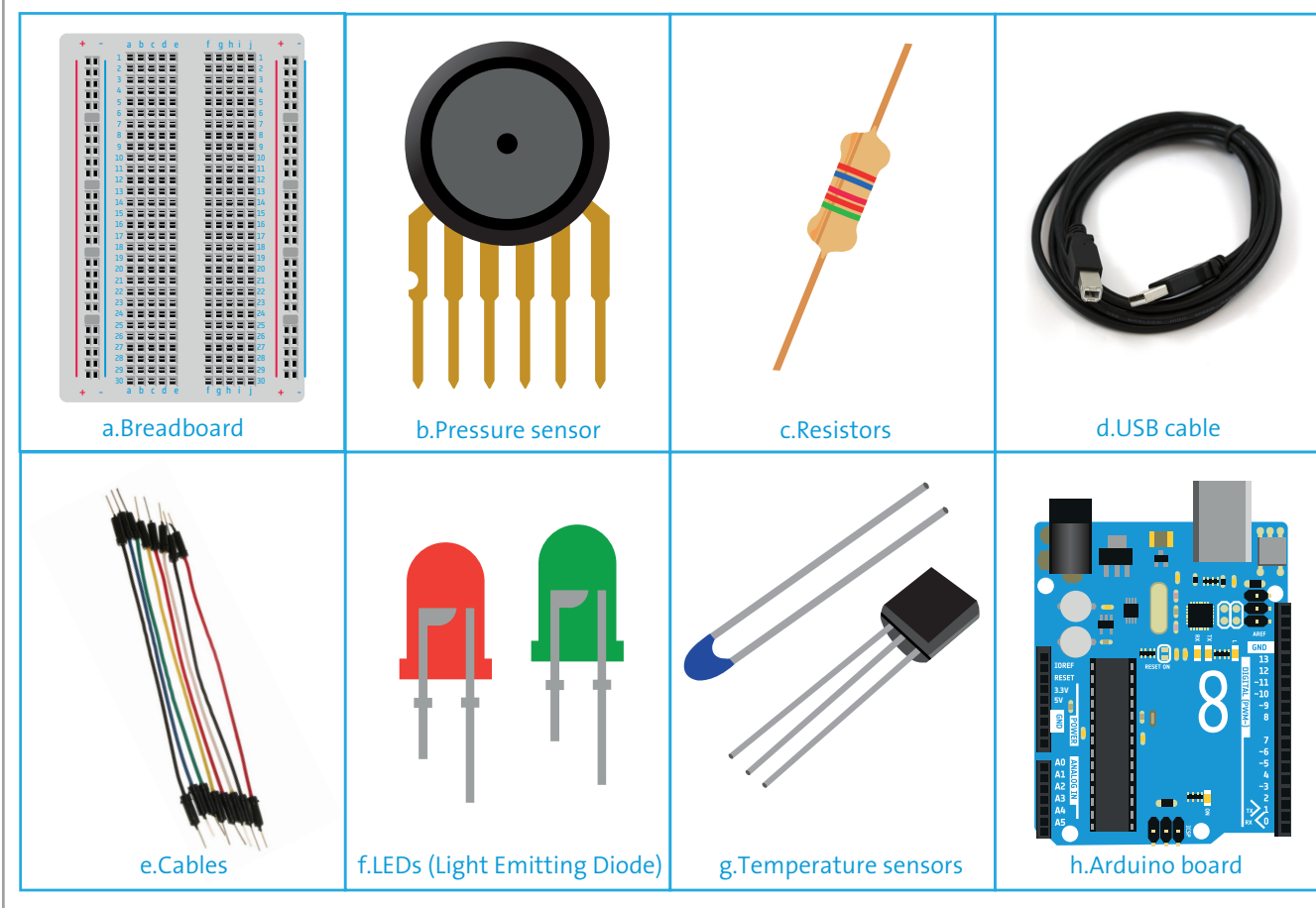
Hardware

For this activity we are going to use the Arduino Uno (Figure A1, box number h). The Arduino can provide voltage to its pins (metallic legs) and read the voltage from them. Notice the number printed next to each pin, with **GND** (ground) acting as the negative pole and **Vin** or **5V** acting as positive poles (like the positive and negative sides of a battery).

We are going to use the following components shown in Figure A1:

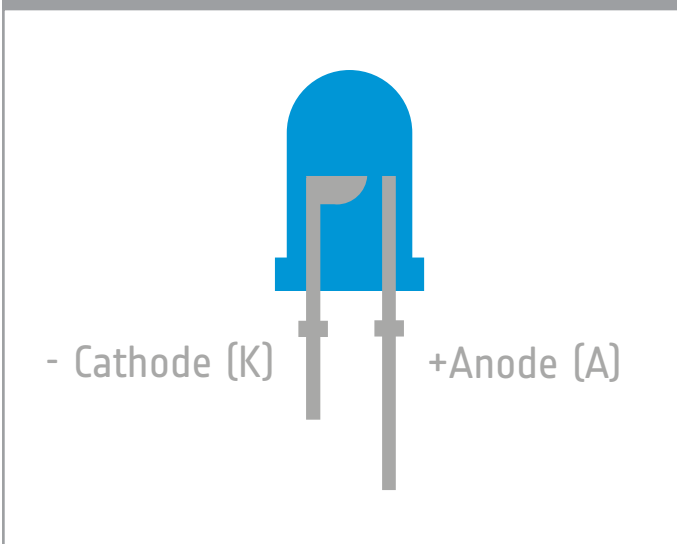
- a. **Breadboard:** A support structure to easily connect electronic components.
- b. **Pressure sensor MPX4115A:** A component that measures the absolute ambient pressure.
- c. **Resistors of 220 Ω and 10k Ω :** To control the current flow. Resistance is measured in ohms (Ω). Resistors have coloured bands on them to indicate their value.
- d. **USB cable:** Used to connect the Arduino to the computer and/or power source.
- e. **Cables:** Used to electrically connect components.
- f. **LEDs (Light Emitting Diode):** Electronic components that generate light when electricity is passed through them.
- g. **Temperature sensors:** An analog sensor (such as a LM35) and a thermistor (a resistor that is especially sensitive to temperature).
- h. **Arduino board:** Works like a simple computer that contains a processor, memory and some input/output pins.

Figure A1



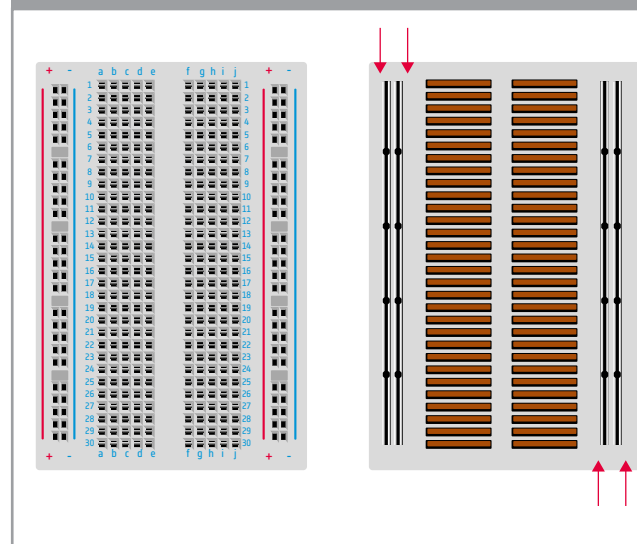
↑ Arduino basic kit components

Figure A2



↑ LED: Note that the LED has a long and a short leg. The long leg is called the anode and the short the cathode.

Figure A3



↑ Breadboard: Note that all the pins of each row are connected.

When you connect the Arduino, your computer should recognise it and initiate the driver installation process. If you have problems installing drivers you can find further instructions here: www.arduino.cc/en/Guide/ArduinoUno

*Please be sure to select the right board (try Genuino Uno) and the correct serial port (COM) in the Tools menu!

Software

The Arduino software can be easily downloaded and installed here:
www.arduino.cc/en/Main/Software

Programs written using the Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the .ino file extension. The editor has features for cutting/pasting and for searching/replacing text. The message area at the bottom (Figure A4) gives feedback while saving and exporting and also displays errors. The selected board and serial port are displayed at the bottom right corner of the window. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches and open the serial monitor (to view data collected from sensors). Figure A4 is a screenshot of the IDE Arduino graphical interface.

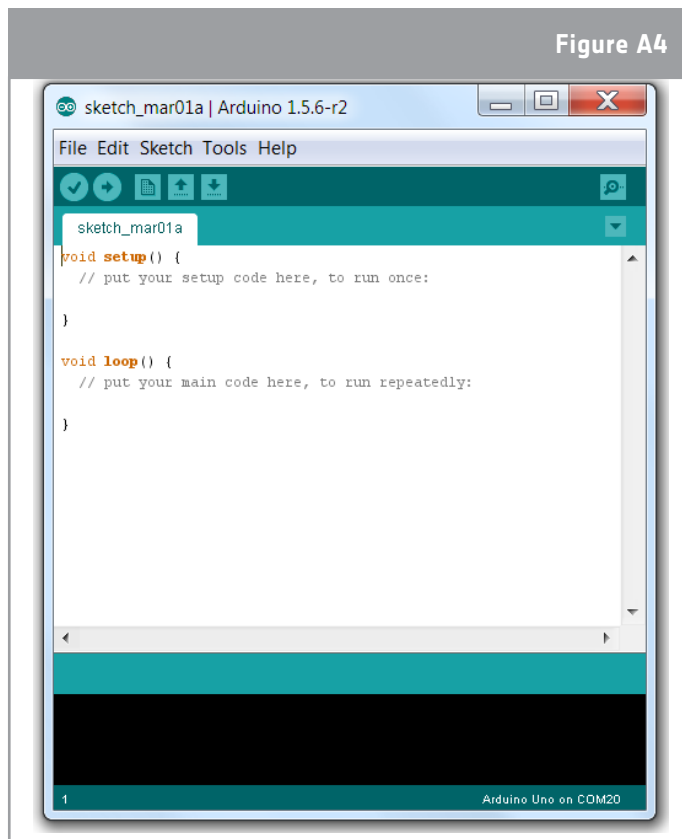


Figure A4

↑ Arduino IDE interface

Arduino commands - Table A1

	Verify: Checks your code for errors when compiling it.
	Upload: Compiles your code and uploads it to the configured board.
	New: Creates a new sketch.
	Open: Presents a menu of all the sketches in your sketchbook.
	Save: Saves your sketch.
	Serial Monitor: Opens the serial monitor, allowing you to see data collected from sensors.

Additional commands are found within the five menus: File, Edit, Sketch, Tools and Help.

→ Activity 1: Make me blink!

In this activity students will control an LED with the Arduino. The purpose is to control the blinking of the LED and to understand how the code is written/used to control each LED. More information can be found through a quick Google search: Arduino LED blink.

Equipment

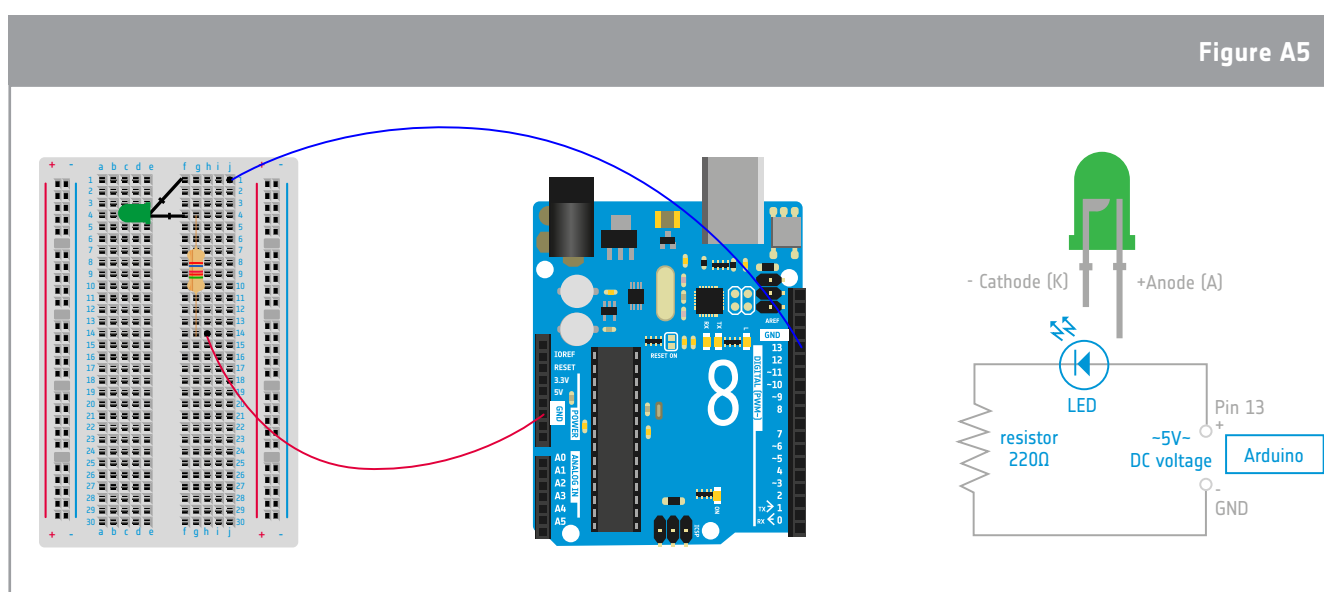
- 1 Arduino Uno
- 1 breadboard
- 1 LED (green)
- 1 220 Ω resistor (red/red/black/black/brown)
- 2 wires

Exercise

Setup

The circuit is set up as follows:

- Connect the green LED on the breadboard as shown in Figure A5.
- Connect the 220 Ω resistor (red/red/black/black/brown) to the short leg of the LED as shown in Figure 5.
- Using a wire, connect the long leg of the LED to pin number **13** of the Arduino Uno
- Using a wire, connect the resistor to the **GND** pin of the Arduino Uno.
- Finally, connect the Arduino UNO to the computer using the USB cable .



↑ Arduino circuit ready to blink the green LED

Code

Students will now upload a program to control the LED. They will use an example code that can be found by opening the Arduino Software (IDE) and clicking on:

File → Examples → Basics → Blink

The Blink code will appear on the screen and is shown in the image below. Comments are displayed in grey after the slashes and explain each line of the code.

Upload the program to the Arduino

```

1  /*
2   Blink
3   Turns on an LED on for one second, then off for one second, repeatedly.
4
5   Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6   it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
7   of use the correct LED pin whatever is the board used.
8   If you want to know what pin the on-board LED is connected to on your Arduino model, check
9   the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
10
11  This example code is in the public domain.
12
13  modified 8 May 2014
14  by Scott Fitzgerald
15
16  modified 2 Sep 2016
17  by Arturo Guadalupi
18  */
19
20
21 // the setup function runs once when you press reset or power the board
22 void setup() {
23   // initialize digital pin LED_BUILTIN as an output.
24   pinMode(LED_BUILTIN, OUTPUT);
25 }
26
27 // the loop function runs over and over again forever
28 void loop() {
29   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
30   delay(1000); // wait for a second
31   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
32   delay(1000); // wait for a second
33 }
34
35

```

The grey text is comments, it explains what the code does

LED_BUILTIN is the name/number of the pin where the LED is connected

HIGH = LED is on
LOW = LED is off

1000ms is the delay between turning the LED on

The main elements of the above code are explained below:

```
void setup()
```

This command is always written at the beginning when a sketch starts; it initializes the variables that are going to be used in the rest of the sketch. It will only run once, after each power up or reset of the Arduino board.

```
pinMode(LED_BUILTIN, OUTPUT) -> pinMode(13, OUTPUT)
```

Configures the specified pin to behave either as an input or an output. In the blink example, replace the **LED_BUILTIN** by the number of the pin where the LED is connected, i.e. **13**.

```
void loop()
```

This command does precisely what its name suggests and consecutively loops all the commands that it contains, allowing your program to do multiple things successively.

```
digitalWrite(13, HIGH)
```

Turns the LED on. HIGH sets the voltage to around 5V to pin 13.

```
delay(1000)
```

Pauses the program for the amount of time indicated (in milliseconds).

```
digitalWrite(13, LOW)
```

Turn the LED off by making the voltage LOW. Low means giving a 0V voltage to pin 13.

A complete list of all possible commands can be found through a quick Google search: Arduino commands.

→ Activity 2: Blink an S.O.S.!

In this activity students use the Arduino circuit built in activity 1. They will control the green LED to send a message in Morse to demonstrate how they could communicate with a rover on Mars. To send an O, students should change the delay in the code to increase the time of the lighted green LED. In this case we changed 1000 ms to 5000 ms, but it could be any number greater than 1000 ms. The code is shown in Figure A6:

Figure A6

```
void loop() {  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
  
  digitalWrite(13, HIGH);  
  delay(5000);  
  digitalWrite(13, LOW);  
  delay(1000);  
  
}
```

[↑ Code to send an O in Morse](#)

→ Activity 3: Measuring temperature

In this activity, students measure temperature using the Arduino. The purpose is to read temperature from the temperature sensor (thermistor*) and to understand how the code is written/used to control the sensor. More information can be found through a quick Google search: Arduino thermistor sensor.

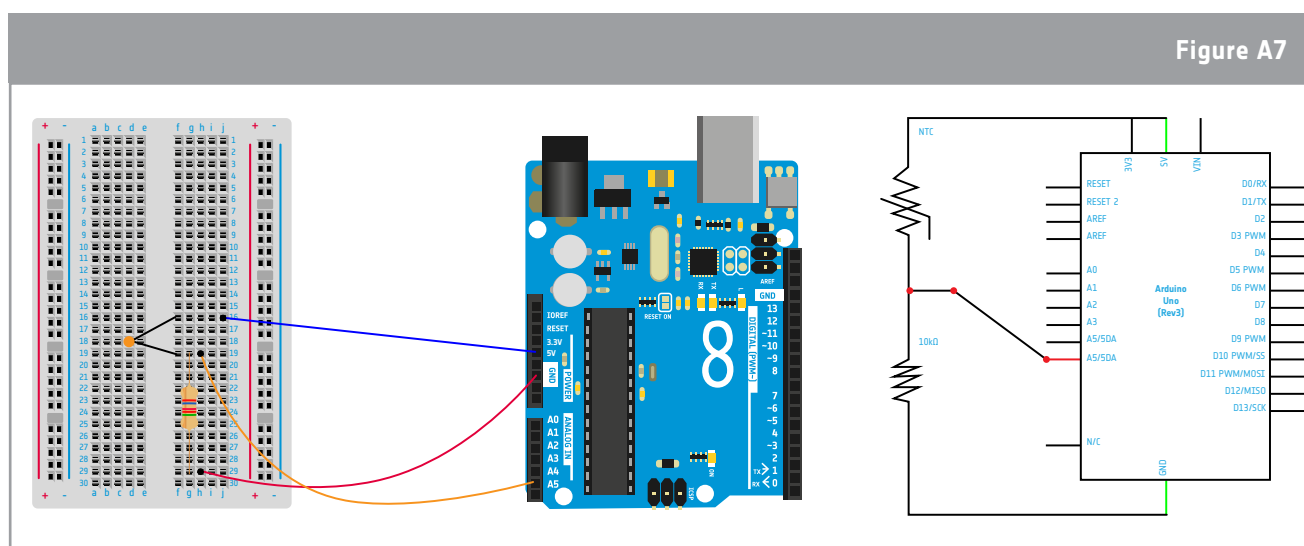
Equipment

- 1 Arduino Uno
- 1 breadboard
- 1 temperature sensor (thermistor*)
- 1 10kΩ resistor (brown/black/black/red/brown)
- 3 wires

Exercise

Setup

- Connect the thermistor on the breadboard as shown in Figure A7.
- Connect the 10kΩ resistor (brown/black/black/red/brown) to the thermistor as shown in Figure A7.
- Using a wire, connect the thermistor to the **5V** pin of the Arduino Uno
- Using a wire, connect the resistor to the **GND** pin of the Arduino Uno.
- Using a wire, connect the thermistor and the resistor to the **A5** pin of the Arduino.
- Finally connect the Arduino UNO to the computer using the USB cable .



↑ Arduino circuit ready to measure the temperature

* thermistor: an electrical resistor whose resistance is greatly reduced by heating

Code

Students will now write their own code. A complete list of all possible commands can be found through a quick Google search: Arduino commands. Figure A8 illustrates an example of a code that measures the temperature. In this example, the temperature that is read from the sensor in the circuit is called **AnalogT**, and is in units of volts because it is an electric signal; it does not have a physical meaning of temperature. The temperature that is converted to degrees Celsius (the unit of temperature used in Europe) is called **AnalogTf**. The name of the pin on the Arduino Uno is A5. Once the code is uploaded and begins running, open the Serial Monitor* window in order to see the data being collected.

Figure A8

```

1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   float AnalogT;
9   float AnalogTf;
10  AnalogT= float(analogRead(A5));
11  AnalogTf = (-40000/AnalogT) + 100;
12  Serial.println();
13  Serial.print("Temperature: ");
14  Serial.print(AnalogTf);
15  Serial.print("C");
16  delay(1000);
17
18 }
```

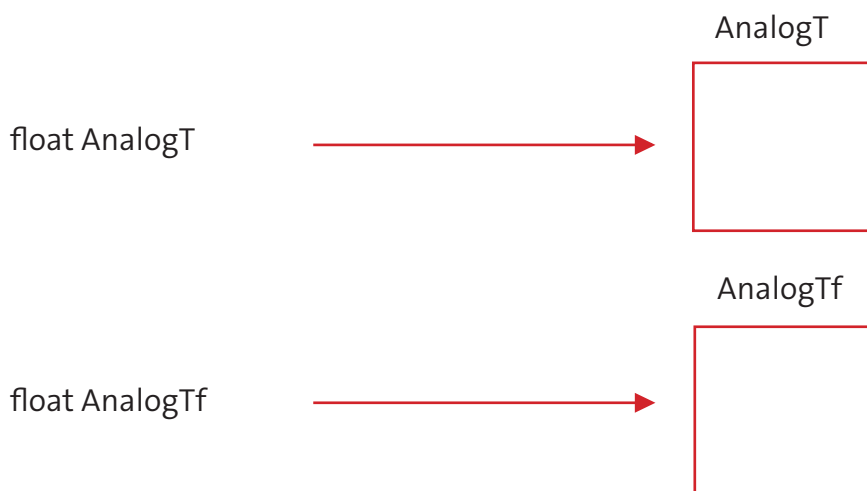
↑ [Arduino code for measuring the temperature](#)

*see Table A1: Arduino commands.

For explanations of the void setup and void loop, please refer to Activity 1 Exercise 3.

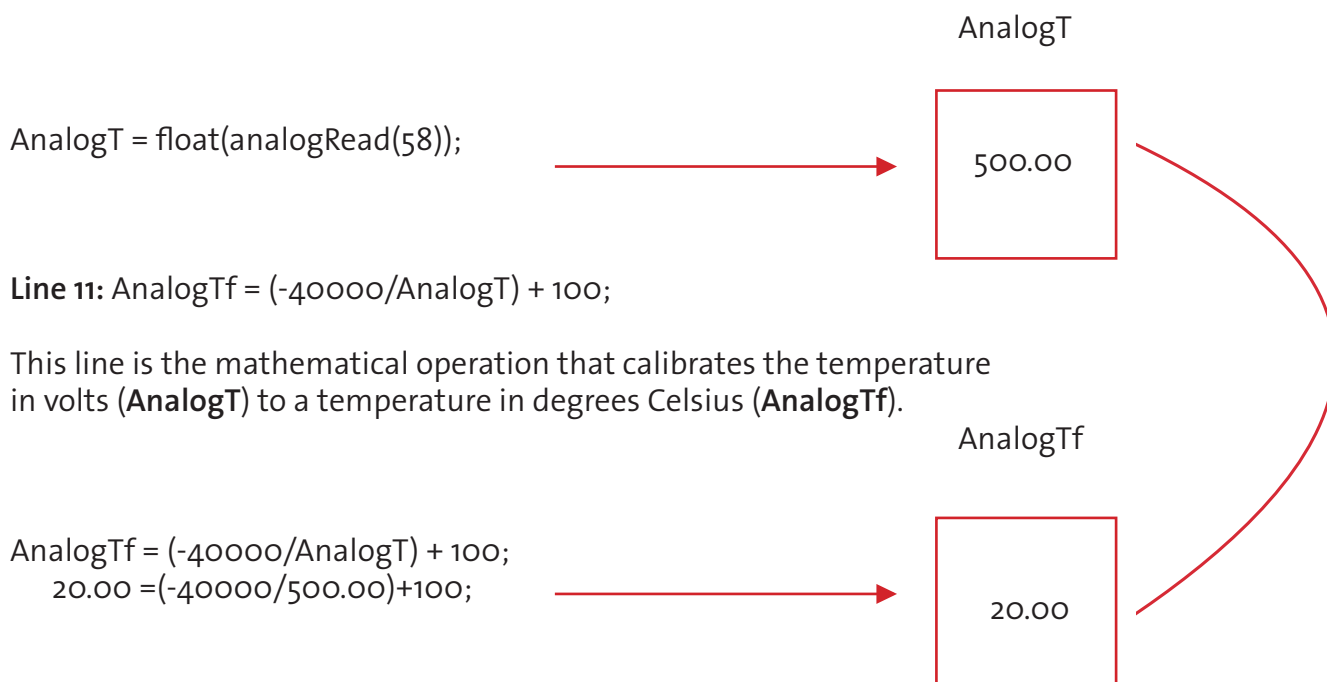
Line 3: As the temperature will be printed on the screen of the computer, it is necessary to predefine the data rate (9600 bits/second) for the data transmission between the Arduino and the computer.

Line 8-9: These two lines of code are used to define variables called AnalogT and AnalogTf. These names are chosen by the programmer. To model this process, we will use a graphical explanation. When you write float AnalogT, inside the memory of the computer you create a box or an empty space called AnalogT that has a specific size indicated by 'float'. Float means that the number inside the box will have a decimal point.



Line 10: `AnalogT = float(analogRead(A5));`

The computer will write the number read from the A5 pin on the Arduino Uno (e.g. 500.00) inside the AnalogT box. We use the function **analogRead()** to read any pin from the Arduino.



Lines 12-15: To display the measure of the temperature on the screen, you need to use the print function. This function will recover the number inside any boxes and print it in the Serial monitor. In this example, the function will print the number inside the box called AnalogTf. All the words inside the “...” are directly printed on the serial monitor.

Line 16: The delay function is used to wait 1000 milliseconds to display the temperature on the serial monitor.

Discussion

Students will probably have to calibrate their own temperature sensor. They can use another thermometer to measure the real temperature as a reference for calibrating their sensor. To roughly calibrate, they run the code as currently written and if their data output is not the same as the real reference temperature, they try changing the last number (+ 100) of line 11 (AnalogTf) of the code in Figure A8.

Practice

Students are asked to calibrate the temperature sensor as if they were on the surface of Mars. As the average temperature of the Martian surface is considered to be around -60°C , and $+14^{\circ}\text{C}$ for Earth, students should calculate a shift of -74 . The code should be adapted like this:

```
AnalogTf = (-40000/AnalogT) + 100 - 74;
```

→ Activity 4: Measuring pressure

In this activity, students measure the ambient pressure with the Arduino. The purpose is to read the pressure from the sensor (MPX4115A) and to understand how the code is written/used to control the sensor. More information can be found through a quick Google search: Arduino MPX4115A.

Equipment

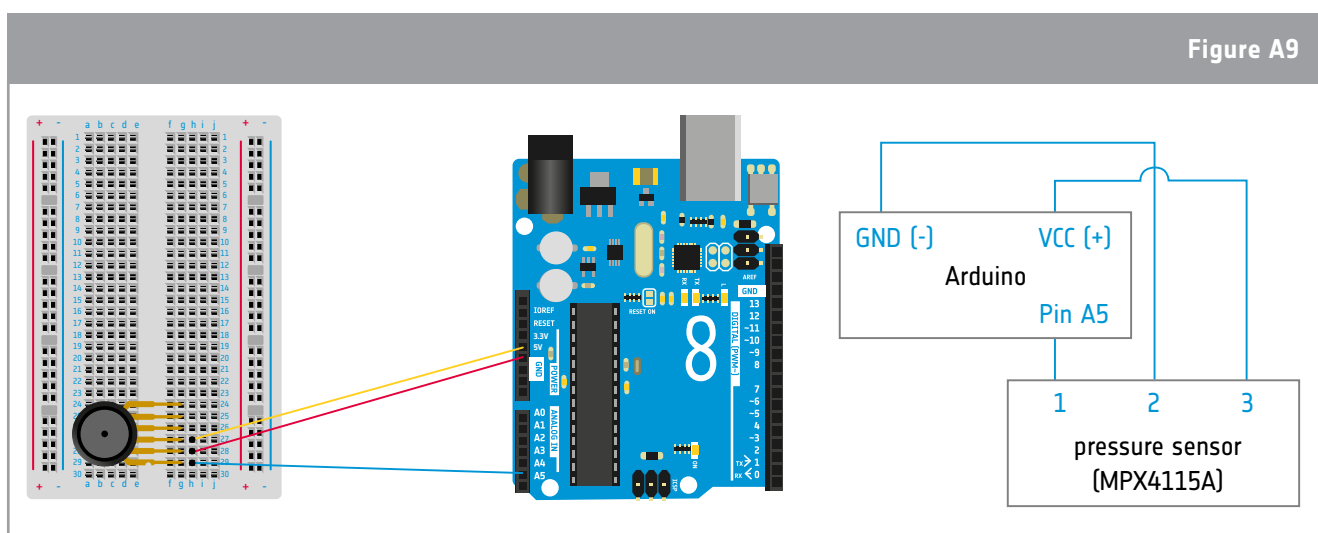
- 1 Arduino Uno
- 1 breadboard
- 1 pressure sensor
- 3 wires

Exercise

Setup

- Connect the pressure sensor to the breadboard.
- Using a wire, connect leg 1 of the pressure sensor (marked with a notch) to pin **A5**.
- Using a wire, connect leg 2 of the pressure sensor in the **GND** row of the Arduino.
- Using a wire, connect leg 3 of the pressure sensor to pin **5V** of the Arduino.
- Finally connect the Arduino UNO to the computer using the USB cable .

Figure A9



↑ Arduino circuit ready to measure the pressure

Code

In this case, students use the same commands as in the previous temperature activity; again they will focus on calibrating the sensor. An important and highly useful step in using sensors is to look at their datasheets, which contain detailed information about how the sensor works. In this case, we can easily find useful information with a quick Google search: MPX4115A datasheet.

The pressure sensor converts the measured pressure into a voltage. This is an analogue signal that will be translated to a digital signal. To translate the measured voltages to ambient pressure values (in units of Pa or pascals in the International System of Units), the transfer function of the sensor is needed. This function describes the mathematical relation between the voltage output of the sensor and the equivalent pressure. This function can be found in the datasheets of the sensor. This function can vary from sensor to sensor.

If we look at the datasheet, we will find the following transfer function: $V_{\text{out}} = V_s (P \times 0.009 - 0.095)$. We only need to isolate the value of P (Pressure) and take into account that $V_s = 1024$ (the voltage read is divided in 1024 steps). $V_{\text{out}} = \text{Analogue value that we will read from pin A5}$.

Below is an example of the code that prints the values of pressure. Use it as a reference.

Figure A10

```

1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   float AnalogP;
9   float AnalogPf;
10  AnalogP= float(analogRead(A5));
11  AnalogPf = ((AnalogP/1024)+0.095)/0.009;
12  Serial.println();
13  Serial.print("Pressure: ");
14  Serial.print(AnalogPf);
15  Serial.print("kPa");
16  delay(1000);
17
18 }
```

[↑ Arduino code for measuring the pressure](#)

The name given for the pressure in volts is AnalogP and for the pressure in pascal is AnalogPf. The name of the pin where the pressure sensor is connected is A9.

To calibrate their pressure sensor students can check the ambient pressure of their location on the internet. To roughly calibrate, they run the code as currently written and if their data output is not the same as the real reference pressure, they can adapt the line 11 (**AnalogPf**) of the code.

→ Activity 5: Measuring altitude

Pressure varies with altitude, and so pressure can be used to measure altitude. To measure altitude, students need to add this part of the code inside the loop of the code measuring the pressure.

Below is an example of the code that prints the altitude in metres. Use it as a reference.

Figure A11

```
float Altitude;
float Altitudef;
  Altitude = pow(AnalogPf/101.325,0.1903);
  Altitudef = (1-Altitude)*44300 + 100;
Serial.print("Altitude: ");
Serial.print(Altitudef); Serial.print ("meters");
```

[↑ Arduino code for measuring the pressure](#)

The name given to the altitude in metres is Altitudef. The pressure in units of kPa is then given by AnalogPf, in this example.

From the hints given in the 'Did you know' box, the students should realise that as altitude increases, the temperature of the atmosphere decreases, and this relationship should be seen in their graph of altitude and temperature.

How to use Arduino in the classroom

Arduino is an excellent tool to introduce students to basic programming and the wide range of applications it has. It can also be used to teach and learn STEM related subjects. Some examples are:

- **Introduction to circuit basics** - Activity 1 can be used to introduce students to a simple circuit.
- **Introduction to data analysis** - Students collect data from the real world (temperature and/or pressure) and analyse data using graphs. Students read information and draw conclusions.
- **Introduction to weather elements and how data is collected** - Students can develop their own weather station and predict sunny/cloudy days.

→ MEET ARDUINO!

Introduction to Arduino computing using C++

Did you know?



The 2020 mission of the ExoMars programme will deliver a European rover and a Russian science payload to the surface of Mars to establish if life ever existed there!

→ Activity 0: Getting started

Everyone, every day, uses technology! In space missions, engineers use technology to command robots and communicate with them to extend our scientific knowledge. In this activity, you will become the engineer to investigate Mars, the Red Planet, using a technology tool called Arduino. Let's find out how to identify components used in Arduino!

Equipment

- The Arduino basic kit

Ground Station components:

1. Work in pairs. Open your Arduino kit and match the components with the descriptions on the next page.

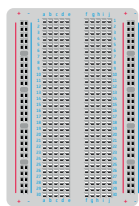
Figure A1

<p>a. Breadboard</p>	<p>b. Pressure sensor</p>	<p>c. Resistors</p>	<p>d. USB cable</p>
<p>e. Cables</p>	<p>f. LEDs (Light Emitting Diode)</p>	<p>g. Temperature sensors</p>	<p>h. Arduino board</p>

↑ Arduino basic kit components

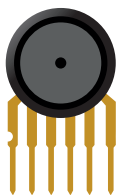
COMPONENT

DESCRIPTIONS



Breadboard

has a circular shape and measures ambient pressure



Pressure sensor

has input and output pins and works like a simple computer



Resistors

is a white support used to easily connect electronic components



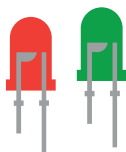
USB cable

has two long legs connected to a blue head and measures the temperature



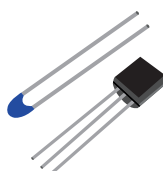
Cables

cylinder shape with coloured bands, they reduce the current flowing in a circuit



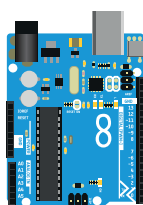
LEDs (Light Emitting Diode)

are red and green and emit light when electricity passes through them



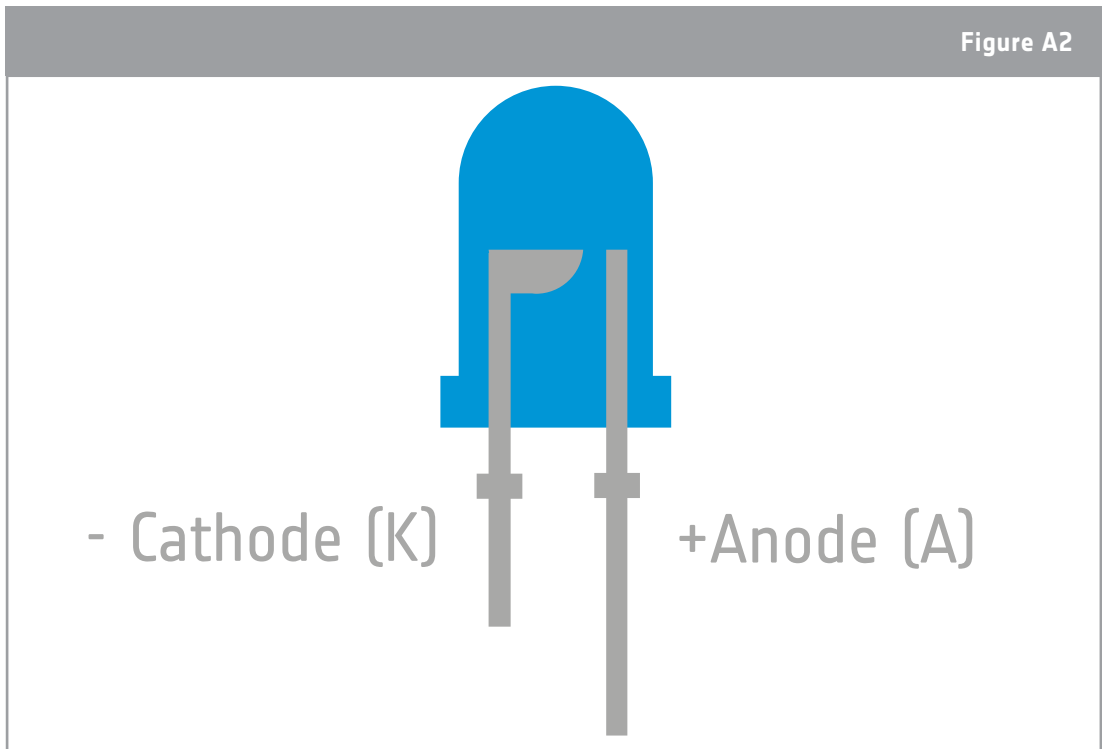
Temperature sensors

have various colours and are used to connect components to conduct electricity

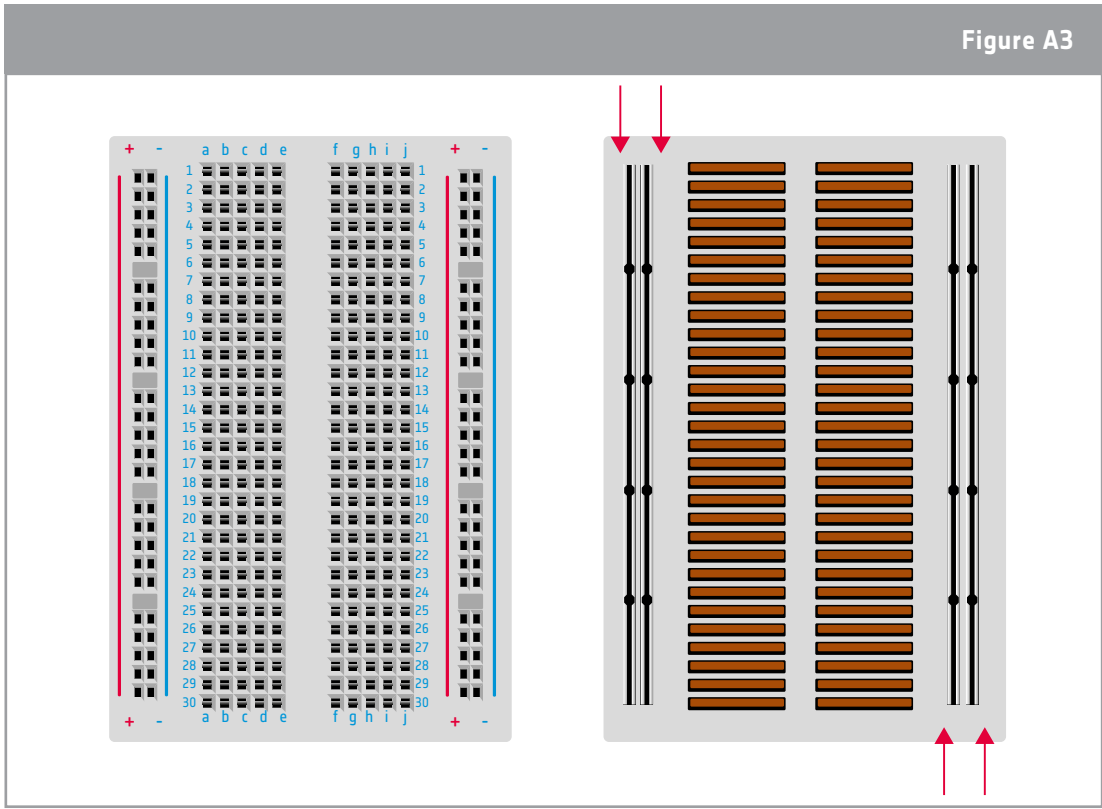


Arduino board

a long black cable used to connect the Arduino Uno to the computer



↑ LED: Note that the LED has a long and a short leg. The long leg is called the anode and the short the cathode.



↑ Breadboard: Note that all the pins of each row are connected.

→ Activity 1: Make me blink!

To command interactive objects, you need to use a tool such as an Arduino, made up of several electronic components. As you are the engineer in charge of studying the Red Planet, you need to communicate with the rover to command its experiments and collect its data. In this activity you will learn the C++ programming language in order to communicate with your Arduino Uno, and you will learn how to control light!

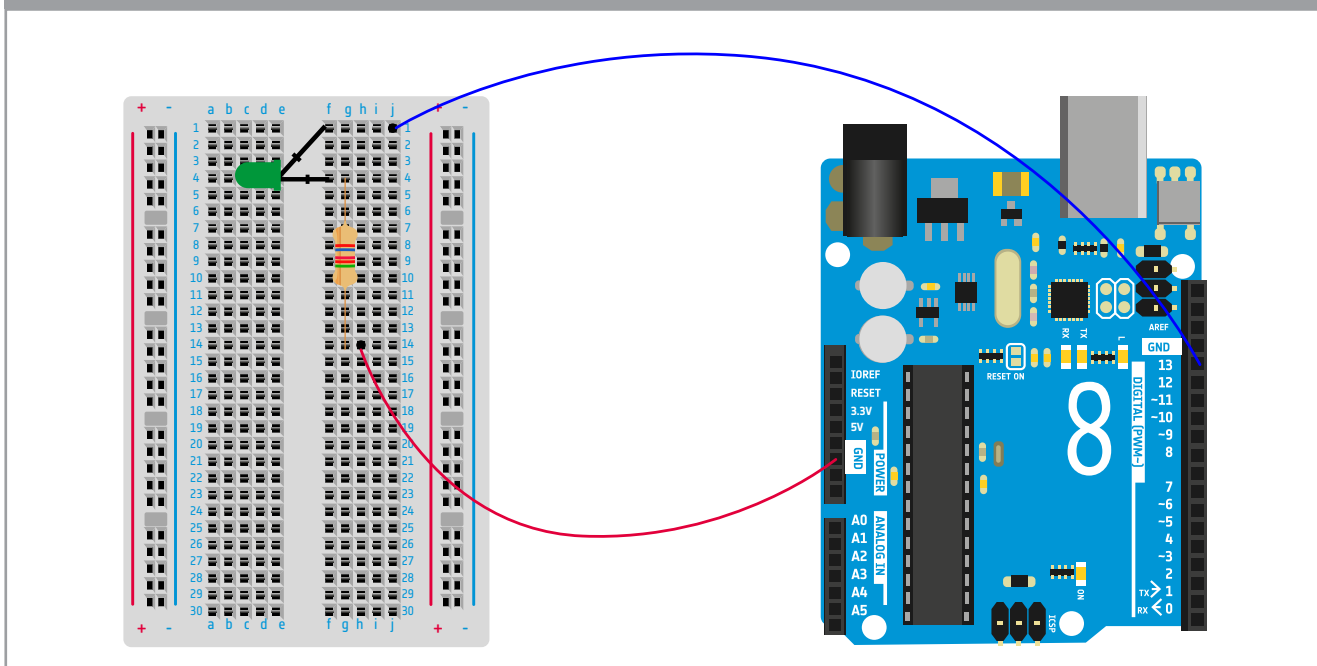
Equipment

- 1 Arduino Uno
- 1 breadboard
- 1 green LED
- 1 resistor 220Ω (red/red/black/black/brown)
- 2 wires

Exercise

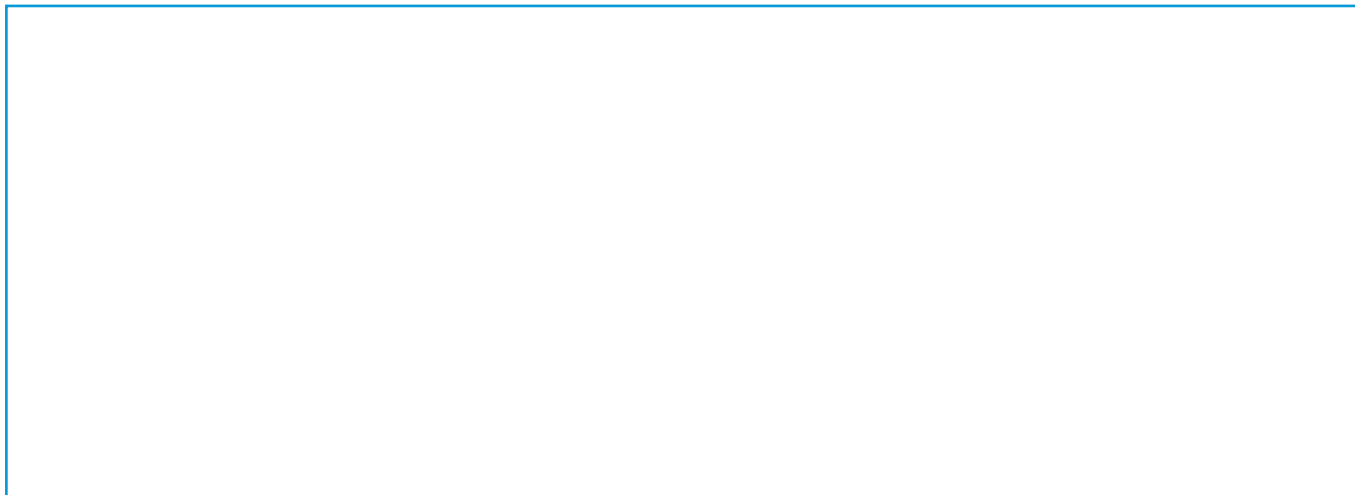
- Carefully follow these instructions to set up the breadboard and control the green LED:
 - Connect the green LED (clear plastic) on the breadboard as shown in Figure A4.
 - Connect the 220Ω resistor (red/red/black/black/brown) to the short leg of the LED as shown in Figure A4.
 - Using a wire, connect the long leg of the LED to pin number **13** of the Arduino Uno.
 - Using a wire, connect the resistor to the **GND**(ground) pin of the Arduino Uno.
 - Connect the Arduino Uno to the computer using the USB cable.

Figure A4



↑ Arduino circuit ready to blink the green LED

2. In the box below sketch the electrical circuit corresponding to the electrical connexions of Figure A4. Use these symbols:



3. Now that the breadboard is ready, you need to send instructions to the Arduino Uno to blink the LED.

Open the Arduino software (IDE) on the computer and click on File → Examples → Basics → Blink.

The Blink code will appear on the screen and is shown in the figure below. Carefully read the explanation to fully understand:

Upload the program to the Arduino

The grey text is comments, it explains what the code does

LED_BUILTIN is the name/number of the pin where the LED is connected

HIGH = LED is on
LOW = LED is off

1000ms is the delay between turning the LED on

```

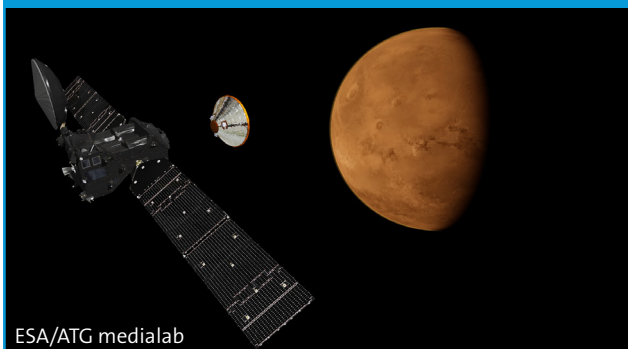
1  /*
2  3  Blink
4  5  Turns on an LED on for one second, then off for one second, repeatedly.
6  7
7  8  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
9  10 it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
10 11 of use the correct LED pin whatever is the board used.
11 12 If you want to know what pin the on-board LED is connected to on your Arduino model, check
12 13 the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
13 14
14 15 This example code is in the public domain.
15 16
16 17 modified 8 May 2014
17 18 by Scott Fitzgerald
18 19
19 20 modified 2 Sep 2016
20 21 by Arturo Guadalupi
21 22
22 23 */
23 24
24 25 // the setup function runs once when you press reset or power the board
25 26 void setup() {
26 27   // initialize digital pin LED_BUILTIN as an output.
27 28   pinMode(LED_BUILTIN, OUTPUT);
28 29 }
29 30
30 31 // the loop function runs over and over again forever
31 32 void loop() {
32 33   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
33 34   delay(1000); // wait for a second
34 35   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
35 36   delay(1000); // wait for a second
36 37 }
37 38
38 39
39 40
40 41
41 42
42 43
43 44
44 45
45 46
46 47
47 48
48 49
49 50
49 51
51 52
52 53
53 54
54 55
55 56
56 57
57 58
58 59
59 60
59 61
61 62
62 63
63 64
64 65
65 66
66 67
67 68
68 69
69 70
70 71
71 72
72 73
73 74
74 75
75 76
76 77
77 78
78 79
79 80
80 81
81 82
82 83
83 84
84 85
85 86
86 87
87 88
88 89
89 90
90 91
91 92
92 93
93 94
94 95
95 96
96 97
97 98
98 99
99 100
100 101
101 102
102 103
103 104
104 105
105 106
106 107
107 108
108 109
109 110
110 111
111 112
112 113
113 114
114 115
115 116
116 117
117 118
118 119
119 120
120 121
121 122
122 123
123 124
124 125
125 126
126 127
127 128
128 129
129 130
130 131
131 132
132 133
133 134
134 135
135 136
136 137
137 138
138 139
139 140
140 141
141 142
142 143
143 144
144 145
145 146
146 147
147 148
148 149
149 150
150 151
151 152
152 153
153 154
154 155
155 156
156 157
157 158
158 159
159 160
160 161
161 162
162 163
163 164
164 165
165 166
166 167
167 168
168 169
169 170
170 171
171 172
172 173
173 174
174 175
175 176
176 177
177 178
178 179
179 180
180 181
181 182
182 183
183 184
184 185
185 186
186 187
187 188
188 189
189 190
190 191
191 192
192 193
193 194
194 195
195 196
196 197
197 198
198 199
199 200
200 201
201 202
202 203
203 204
204 205
205 206
206 207
207 208
208 209
209 210
210 211
211 212
212 213
213 214
214 215
215 216
216 217
217 218
218 219
219 220
220 221
221 222
222 223
223 224
224 225
225 226
226 227
227 228
228 229
229 230
230 231
231 232
232 233
233 234
234 235
235 236
236 237
237 238
238 239
239 240
240 241
241 242
242 243
243 244
244 245
245 246
246 247
247 248
248 249
249 250
250 251
251 252
252 253
253 254
254 255
255 256
256 257
257 258
258 259
259 260
260 261
261 262
262 263
263 264
264 265
265 266
266 267
267 268
268 269
269 270
270 271
271 272
272 273
273 274
274 275
275 276
276 277
277 278
278 279
279 280
280 281
281 282
282 283
283 284
284 285
285 286
286 287
287 288
288 289
289 290
290 291
291 292
292 293
293 294
294 295
295 296
296 297
297 298
298 299
299 300
300 301
301 302
302 303
303 304
304 305
305 306
306 307
307 308
308 309
309 310
310 311
311 312
312 313
313 314
314 315
315 316
316 317
317 318
318 319
319 320
320 321
321 322
322 323
323 324
324 325
325 326
326 327
327 328
328 329
329 330
330 331
331 332
332 333
333 334
334 335
335 336
336 337
337 338
338 339
339 340
340 341
341 342
342 343
343 344
344 345
345 346
346 347
347 348
348 349
349 350
350 351
351 352
352 353
353 354
354 355
355 356
356 357
357 358
358 359
359 360
360 361
361 362
362 363
363 364
364 365
365 366
366 367
367 368
368 369
369 370
370 371
371 372
372 373
373 374
374 375
375 376
376 377
377 378
378 379
379 380
380 381
381 382
382 383
383 384
384 385
385 386
386 387
387 388
388 389
389 390
390 391
391 392
392 393
393 394
394 395
395 396
396 397
397 398
398 399
399 400
400 401
401 402
402 403
403 404
404 405
405 406
406 407
407 408
408 409
409 410
410 411
411 412
412 413
413 414
414 415
415 416
416 417
417 418
418 419
419 420
420 421
421 422
422 423
423 424
424 425
425 426
426 427
427 428
428 429
429 430
430 431
431 432
432 433
433 434
434 435
435 436
436 437
437 438
438 439
439 440
440 441
441 442
442 443
443 444
444 445
445 446
446 447
447 448
448 449
449 450
450 451
451 452
452 453
453 454
454 455
455 456
456 457
457 458
458 459
459 460
460 461
461 462
462 463
463 464
464 465
465 466
466 467
467 468
468 469
469 470
470 471
471 472
472 473
473 474
474 475
475 476
476 477
477 478
478 479
479 480
480 481
481 482
482 483
483 484
484 485
485 486
486 487
487 488
488 489
489 490
490 491
491 492
492 493
493 494
494 495
495 496
496 497
497 498
498 499
499 500
500 501
501 502
502 503
503 504
504 505
505 506
506 507
507 508
508 509
509 510
510 511
511 512
512 513
513 514
514 515
515 516
516 517
517 518
518 519
519 520
520 521
521 522
522 523
523 524
524 525
525 526
526 527
527 528
528 529
529 530
530 531
531 532
532 533
533 534
534 535
535 536
536 537
537 538
538 539
539 540
540 541
541 542
542 543
543 544
544 545
545 546
546 547
547 548
548 549
549 550
550 551
551 552
552 553
553 554
554 555
555 556
556 557
557 558
558 559
559 560
560 561
561 562
562 563
563 564
564 565
565 566
566 567
567 568
568 569
569 570
570 571
571 572
572 573
573 574
574 575
575 576
576 577
577 578
578 579
579 580
580 581
581 582
582 583
583 584
584 585
585 586
586 587
587 588
588 589
589 590
590 591
591 592
592 593
593 594
594 595
595 596
596 597
597 598
598 599
599 600
600 601
601 602
602 603
603 604
604 605
605 606
606 607
607 608
608 609
609 610
610 611
611 612
612 613
613 614
614 615
615 616
616 617
617 618
618 619
619 620
620 621
621 622
622 623
623 624
624 625
625 626
626 627
627 628
628 629
629 630
630 631
631 632
632 633
633 634
634 635
635 636
636 637
637 638
638 639
639 640
640 641
641 642
642 643
643 644
644 645
645 646
646 647
647 648
648 649
649 650
650 651
651 652
652 653
653 654
654 655
655 656
656 657
657 658
658 659
659 660
660 661
661 662
662 663
663 664
664 665
665 666
666 667
667 668
668 669
669 670
670 671
671 672
672 673
673 674
674 675
675 676
676 677
677 678
678 679
679 680
680 681
681 682
682 683
683 684
684 685
685 686
686 687
687 688
688 689
689 690
690 691
691 692
692 693
693 694
694 695
695 696
696 697
697 698
698 699
699 700
700 701
701 702
702 703
703 704
704 705
705 706
706 707
707 708
708 709
709 710
710 711
711 712
712 713
713 714
714 715
715 716
716 717
717 718
718 719
719 720
720 721
721 722
722 723
723 724
724 725
725 726
726 727
727 728
728 729
729 730
730 731
731 732
732 733
733 734
734 735
735 736
736 737
737 738
738 739
739 740
740 741
741 742
742 743
743 744
744 745
745 746
746 747
747 748
748 749
749 750
750 751
751 752
752 753
753 754
754 755
755 756
756 757
757 758
758 759
759 760
760 761
761 762
762 763
763 764
764 765
765 766
766 767
767 768
768 769
769 770
770 771
771 772
772 773
773 774
774 775
775 776
776 777
777 778
778 779
779 780
780 781
781 782
782 783
783 784
784 785
785 786
786 787
787 788
788 789
789 790
790 791
791 792
792 793
793 794
794 795
795 796
796 797
797 798
798 799
799 800
800 801
801 802
802 803
803 804
804 805
805 806
806 807
807 808
808 809
809 810
810 811
811 812
812 813
813 814
814 815
815 816
816 817
817 818
818 819
819 820
820 821
821 822
822 823
823 824
824 825
825 826
826 827
827 828
828 829
829 830
830 831
831 832
832 833
833 834
834 835
835 836
836 837
837 838
838 839
839 840
840 841
841 842
842 843
843 844
844 845
845 846
846 847
847 848
848 849
849 850
850 851
851 852
852 853
853 854
854 855
855 856
856 857
857 858
858 859
859 860
860 861
861 862
862 863
863 864
864 865
865 866
866 867
867 868
868 869
869 870
870 871
871 872
872 873
873 874
874 875
875 876
876 877
877 878
878 879
879 880
880 881
881 882
882 883
883 884
884 885
885 886
886 887
887 888
888 889
889 890
890 891
891 892
892 893
893 894
894 895
895 896
896 897
897 898
898 899
899 900
900 901
901 902
902 903
903 904
904 905
905 906
906 907
907 908
908 909
909 910
910 911
911 912
912 913
913 914
914 915
915 916
916 917
917 918
918 919
919 920
920 921
921 922
922 923
923 924
924 925
925 926
926 927
927 928
928 929
929 930
930 931
931 932
932 933
933 934
934 935
935 936
936 937
937 938
938 939
939 940
940 941
941 942
942 943
943 944
944 945
945 946
946 947
947 948
948 949
949 950
950 951
951 952
952 953
953 954
954 955
955 956
956 957
957 958
958 959
959 960
960 961
961 962
962 963
963 964
964 965
965 966
966 967
967 968
968 969
969 970
970 971
971 972
972 973
973 974
974 975
975 976
976 977
977 978
978 979
979 980
980 981
981 982
982 983
983 984
984 985
985 986
986 987
987 988
988 989
989 990
990 991
991 992
992 993
993 994
994 995
995 996
996 997
997 998
998 999
999 1000
1000 1001
1001 1002
1002 1003
1003 1004
1004 1005
1005 1006
1006 1007
1007 1008
1008 1009
1009 1010
1010 1011
1011 1012
1012 1013
1013 1014
1014 1015
1015 1016
1016 1017
1017 1018
1018 1019
1019 1020
1020 1021
1021 1022
1022 1023
1023 1024
1024 1025
1025 1026
1026 1027
1027 1028
1028 1029
1029 1030
1030 1031
1031 1032
1032 1033
1033 1034
1034 1035
1035 1036
1036 1037
1037 1038
1038 1039
1039 1040
1040 1041
1041 1042
1042 1043
1043 1044
1044 1045
1045 1046
1046 1047
1047 1048
1048 1049
1049 1050
1050 1051
1051 1052
1052 1053
1053 1054
1054 1055
1055 1056
1056 1057
1057 1058
1058 1059
1059 1060
1060 1061
1061 1062
1062 1063
1063 1064
1064 1065
1065 1066
1066 1067
1067 1068
1068 1069
1069 1070
1070 1071
1071 1072
1072 1073
1073 1074
1074 1075
1075 1076
1076 1077
1077 1078
1078 1079
1079 1080
1080 1081
1081 1082
1082 1083
1083 1084
1084 1085
1085 1086
1086 1087
1087 1088
1088 1089
1089 1090
1090 1091
1091 1092
1092 1093
1093 1094
1094 1095
1095 1096
1096 1097
1097 1098
1098 1099
1099 1100
1100 1101
1101 1102
1102 1103
1103 1104
1104 1105
1105 1106
1106 1107
1107 1108
1108 1109
1109 1110
1110 1111
1111 1112
1112 1113
1113 1114
1114 1115
1115 1116
1116 1117
1117 1118
1118 1119
1119 1120
1120 1121
1121 1122
1122 1123
1123 1124
1124 1125
1125 1126
1126 1127
1127 1128
1128 1129
1129 1130
1130 1131
1131 1132
1132 1133
1133 1134
1134 1135
1135 1136
1136 1137
1137 1138
1138 1139
1139 1140
1140 1141
1141 1142
1142 1143
1143 1144
1144 1145
1145 1146
1146 1147
1147 1148
1148 1149
1149 1150
1150 1151
1151 1152
1152 1153
1153 1154
1154 1155
1155 1156
1156 1157
1157 1158
1158 1159
1159 1160
1160 1161
1161 1162
1162 1163
1163 1164
1164 1165
1165 1166
1166 1167
1167 1168
1168 1169
1169 1170
1170 1171
1171 1172
1172 1173
1173 1174
1174 1175
1175 1176
1176 1177
1177 1178
1178 1179
1179 1180
1180 1181
1181 1182
1182 1183
1183 1184
1184 1185
1185 1186
1186 1187
1187 1188
1188 1189
1189 1190
1190 1191
1191 1192
1192 1193
1193 1194
1194 1195
1195 1196
1196 1197
1197 1198
1198 1199
1199 1200
1200 1201
1201 1202
1202 1203
1203 1204
1204 1205
1205 1206
1206 1207
1207 1208
1208 1209
1209 1210
1210 1211
1211 1212
1212 1213
1213 1214
1214 1215
1215 1216
1216 1217
1217 1218
1218 1219
1219 1220
1220 1221
1221 1222
1222 1223
1223 1224
1224 1225
1225 1226
1226 1227
1227 1228
1228 1229
1229 1230
1230 1231
1231 1232
1232 1233
1233 1234
1234 1235
1235 1236
1236 1237
1237 1238
1238 1239
1239 1240
1240 1241
1241 1242
1242 1243
1243 1244
1244 1245
1245 1246
1246 1247
1247 1248
1248 1249
1249 1250
1250 1251
1251 1252
1252 1253
1253 1254
1254 1255
1255 1256
1256 1257
1257 1258
1258 1259
1259 1260
1260 1261
1261 1262
1262 1263
1263 1264
1264 1265
1265 1266
1266 1267
1267 1268
1268 1269
1269 1270
1270 1271
1271 1272
1272 1273
1273 1274
1274 1275
1275 1276
1276 1277
1277 1278
1278 1279
1279 1280
1280 1281
1281 1282
1282 1283
1283 1284
1284 1285
1285 1286
1286 1287
1287 1288
1288 1289
1289 1290
1290 1291
1291 1292
1292 1293
1293 1294
1294 1295
1295 1296
1296 1297
1297 1298
1298 1299
1299 1300
1300 1301
1301 1302
1302 1303
1303 1304
1304 1305
1305 1306
1306 1307
1307 1308
1308 1309
1309 1310
1310 1311
1311 1312
1312 1313
1313 1314
1314 1315
1315 1316
1316 1317
1317 1318
1318 1319
1319 1320
1320 1321
1321 1322
1322 1323
1323 1324
1324 1325
1325 1326
1326 1327
1327 1328
1328 1329
1329 1330
1330 1331
1331 1332
1332 1333
1333 1334
1334 1335
1335 1336
1336 1337
1337 1338
1338 1339
1339 1340
1340 1341
1341 1342
1342 1343
1343 1344
1344 1345
1345 1346
1346 1347
1347 1348
1348 1349
1349 1350
1350 1351
1351 1352
1352 1353
1353 1354
1354 1355
1355 1356
1356 1357
1357 1358
1358 1359
1359 1360
1360 1361
1361 1362
1362 1363
1363 1364
1364 1365
1365 1366
1366 1367
1367 1368
1368 1369
1369 1370
1370 1371
1371 1372
1372 1373
1373 1374
1374 1375
1375 1376
1376 1377
1377 1378
1378 13
```

- a. Before uploading the code to the Arduino Uno, you need to match the name of the pin in the code with the number of the pin where the green LED is connected to the Arduino Uno. Explain what you changed in the code to control the green LED:

- b. Upload the code to the Arduino Uno by clicking on the  at the top left of the screen.

Congratulations, the green LED is blinking!

Did you know?

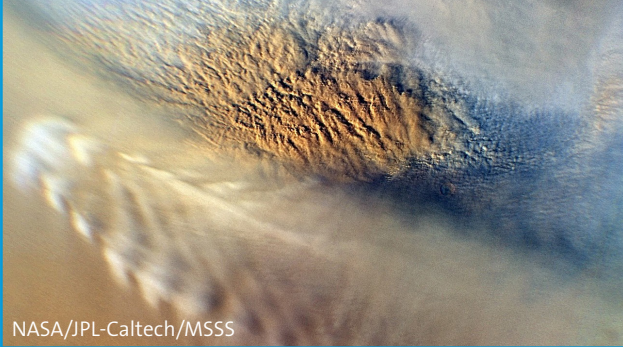


Mars is so far away that signals take quite a long time to travel from the spacecraft or rover back to Earth. This delay in communication makes it challenging to have a conversation with the rover, or react quickly if anything unexpected occurs.

→ Activity 2: Blink a S.O.S.!

Now that you know how to blink an LED using Arduino code, this activity will teach you how to communicate using an LED and to send a message to a rover on Mars. In fact, a very strong sandy storm is planning to pass the rover! To prevent damage to the rover, make it send an S.O.S.!

Did you know?



NASA/JPL-Caltech/MSSS

This image of a dust storm on Mars shows the clouds that result from changes in atmospheric pressure, temperature and height because of vertical displacement, such as when wind blows over a mountain or crater wall.

Equipment

- Circuit built in Activity 1

Exercise

Use Morse code to blink the green LED. In Figure A5, a point means a short signal, which will light up the LED for a short time, and a dash means a long signal, which will light up the LED for a longer time. The code to send an S is given as an example in Figure A6.

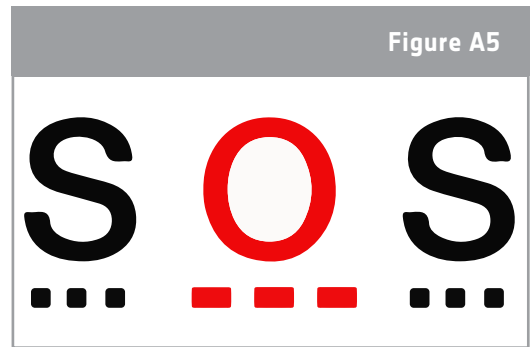


Figure A5

↑ SOS in Morse code

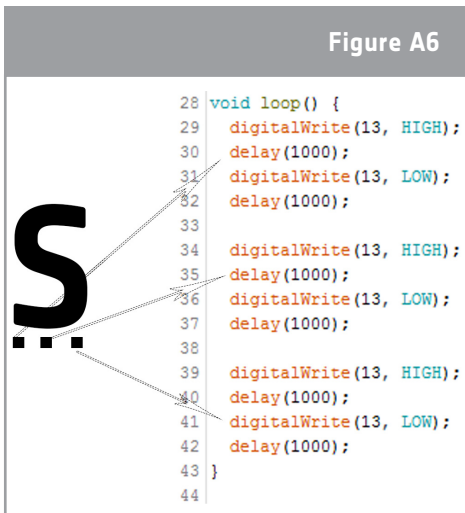


Figure A6

↑ Code to send a S in Morse

Explain what you need to change in the code to send an O:

Adapt the code for sending an S.O.S., then upload it to the Arduino and test it!

→ Activity 3: Measuring temperature

Arduino technology allows you to control an LED connected to an Arduino Uno using a code written on a laptop. When you upload the code, it sends the instructions to the Arduino Uno and activates the electronic components. In the ExoMars mission, technology will be used to investigate the environment of Mars thanks to some sensors. In this activity, you will discover how to measure the temperature of the classroom and simulate the temperature on Mars!

Equipment

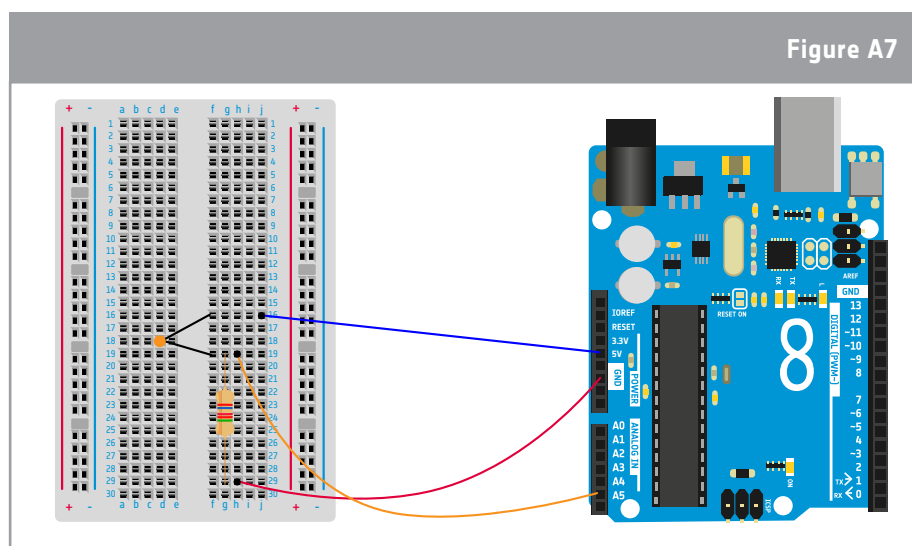
- 1 Arduino Uno
- 1 breadboard
- 1 temperature sensor (thermistor)
- 1 10kΩ resistor (brown/black/black/red/brown)
- 3 wires

Exercise

1. To measure the temperature in the classroom, you need to build the circuit on the breadboard as shown below.

Carefully follow these instructions:

- a. Connect the thermistor on the breadboard as shown in Figure A7.
- b. Connect the 10kΩ resistor (brown/black/black/red/brown) to the thermistor as shown in Figure A7.
- c. Using a wire, connect the thermistor to the **5V** pin of the Arduino Uno.
- d. Using a wire, connect the resistor to the **GND** pin of the Arduino Uno.
- e. Using a wire, connect the thermistor and the resistor to the pin **A5** of the Arduino.
- f. Connect the Arduino Uno to the computer using the USB cable.



↑ Arduino circuit ready to measure the temperature

2. Figure A8 illustrates a partly missing code to measure the temperature. Three values, shown as boxes (red, green and blue) need to be filled in to give correct instructions to the Arduino Uno in order to measure the temperature. Let's find out how to fill in this code.

Figure A8

```

1
2 void setup() {
3   Serial.begin(9600);
4 }
5
6 void loop() {
7
8   float ;
9   float ;
10
11    = float(analogRead());
12    = (-40000/) + 100;
13
14   Serial.println();
15   Serial.print("Temperature in classroom: ");
16   Serial.print();
17   Serial.print("C");
18   delay(1000);
19
20 }

```

↑ [Arduino code for measuring the temperature](#)

- a. What unit is used to determine temperature in Europe?

The temperature sensor connected to the Arduino Uno will first read the temperature in another unit, the volt. To transform this number from volts to degrees Celsius, you will use a formula written on line 12 in Figure A8. Finally, you will display the temperature on the screen using the Serial.print function in the code.

- b. Define the variable containing the temperature in volts:

Define the variable containing the temperature in degrees Celsius:

Look for the pin of the Arduino where the temperature sensor is connected:

- c. You are ready to write the code on your laptop. Open the Arduino software and write the code in Figure A8, replacing the boxes with the variables you have defined for the red, green and blue boxes. Be careful to replace every colour box with its respective variable!

Upload the code to the Arduino Uno by clicking on the button.

- d. To see the temperature measurements, you need to open a specific window in the Arduino software called Serial monitor. Look for this symbol on the toolbar to access it.

Discussion

1. Discuss the relevance of the temperature measurements with your peers. How can you verify that the temperature sensor accurately measures the temperature of the classroom?

2. Which line of the code in Figure A8 can you adapt to calibrate the sensor?

3. To practice a little more with the Arduino, calibrate the temperature sensor as if it were on the surface of Mars. Consider the average temperature of Mars' surface to be around -60°C .

- a. Average temperature of Earth =

- b. Write the new calibration formula of the temperature sensor here:

→ Activity 4: Measuring pressure

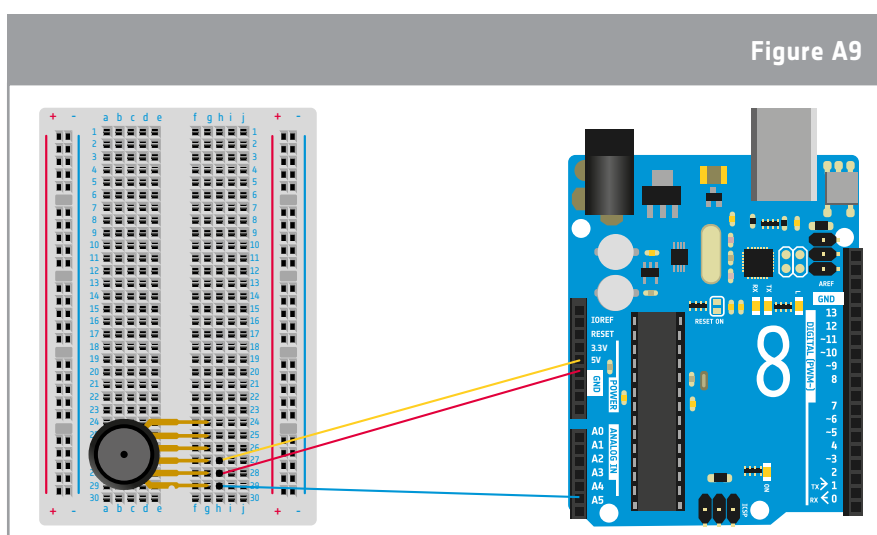
In this activity, you will learn how to measure the atmospheric pressure using the Arduino and a pressure sensor.

Equipment

- 1 Arduino Uno
- 1 breadboard
- 1 pressure sensor
- 3 wires

Exercise

1. To measure the pressure in the classroom, you first need to build the circuit on the breadboard (Figure A9). Carefully follow these instructions (Note: Each pin in the pressure sensor behaves differently, so make sure it is in the same orientation as in the figure):



↑ Arduino circuit ready to measure the pressure

2. Figure A10 illustrates a partly missing code to measure the pressure. Three variables, shown as boxes (red, green and blue), need to be filled in to give correct instructions to the Arduino Uno in order for it to measure the pressure. Let's find out how to fill in this code.

Figure A10

```

1 void setup() {
2     Serial.begin(9600);
3 }
4
5 void loop() {
6
7     float [red box] ;
8     float [green box] ;
9
10    [red box] = float(analogRead([blue box]));
11    [green box] = (([red box] / 1024) + 0.095) / 0.009;
12    Serial.println();
13    Serial.print("Pressure in classroom: ");
14    Serial.print([green box]);
15    Serial.print("kPa");
16    delay(1000);
17 }
18 }
    
```

↑ Arduino code for measuring the pressure

- a. What is the physical unit used to determine the pressure in the International System of Units?

- b. Define the variable containing the pressure in volts:

Define the variable containing the pressure in kilopascal:

Look for the pin of the Arduino where the pressure sensor is connected:

The pressure sensor connected to the Arduino Uno will first read the pressure in another unit, the volt, which is the unit for electricity. To transform this number from volts to pascals, we use this formula from the datasheet of the sensor $V_{out} = V_s (P \times 0.009 - 0.095)$.

Finally, display the measurement of the pressure on the screen using the Serial.print function in the code.

- c. Using the formula $V_{out} = V_s (P \times 0.009 - 0.095)$; calculate the calibration formula to complete line 12 in Figure A10. Because the voltage read is divided in 1024 steps, consider $V_s = 1024$.

- d. You are ready to write the code on your laptop. Open the Arduino software and write the code in Figure A10, complete with the variables you have defined for the red, green and blue boxes. Be careful to replace every coloured box with its respective values!

Upload the code to the Arduino Uno by clicking on the



- e. To see the pressure measurements, you need to open a specific window in the Arduino software called Serial monitor. Look for this symbol on the toolbar to access it:



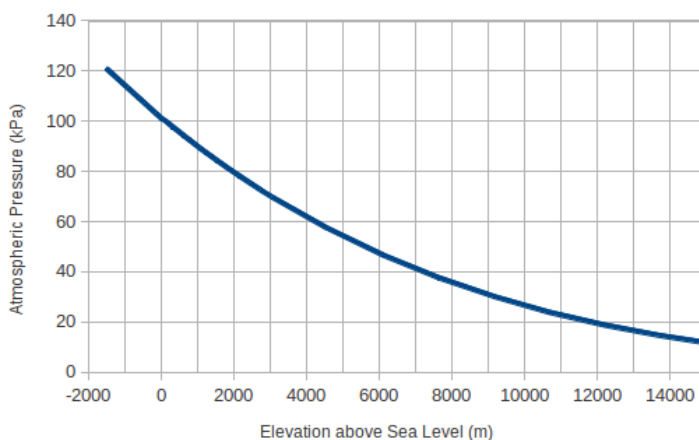
Discussion

Discuss the relevance of the pressure measurements with your peers. How can you verify that the pressure sensor accurately measures the pressure in the classroom?

Which line of the code in Figure A10 should you adapt to calibrate the sensor?

→ Activity 5: Measuring altitude

The pressure of the air varies with altitude. For example, the air pressure at sea is higher than the air pressure in the mountains. You can see how it varies in the graph. With a bit of maths and some understanding of the physics underpinning gas laws, we can calculate exactly what the relationship between pressure and altitude is. Right now, we don't need to worry about this. In this activity you will use your pressure measurements to determine your altitude using a simplified version of the pressure-altitude relationship formula known as the barometric formula.



Equipment

- Arduino circuit built in Activity 4
- Code to measure pressure in Activity 4

Exercise

1. Open the pressure code you have written in Activity 4 in the Arduino software. Add this partly missing code inside the void loop:

Figure A11

```

1 float [ ] ;
2 float [ ] ;
3
4 [ ] = pow( [ ] / 101.325, 0.1903) ;
5 [ ] = (1 - [ ]) * 44300 + 100 ;
6
7 Serial.println();
8 Serial.print("Altitude in classroom: ");
9 Serial.print([ ]);
10 Serial.print("meters");
11 delay(1000);

```

↑ [Arduino code for measuring the altitude](#)

2. To determine the altitude, you will use the measurements of the pressure sensor in a formula written on line 4 in Figure A11. As you discovered in the previous exercises, a sensor connected to the Arduino Uno will first read the measurement in volts. To transform this number from volts to metres, you will use a formula written on line 5 in Figure A11. Finally, altitude data can be displayed using the Serial.print function in the code. If the reading is not calibrated, it can be adjusted by changing line 5 of the code.

3. a. Define the variable containing the altitude in volts:

Define the variable containing the altitude in metres:

Look for the name you gave to the green box containing the pressure in pascal (see Activity 4):

b. Complete the code with the variables you defined for the yellow, purple and green boxes. Be careful to replace every coloured box with its respective value! Then upload the code.

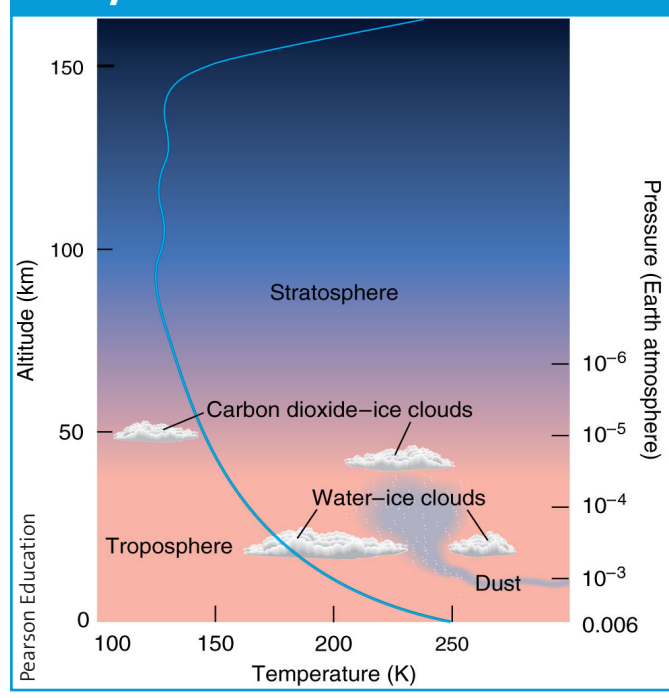
c. To display the altitude data, you need to open a specific window in the Arduino software called Serial monitor.

4. Combine both temperature and altitude code into a single sketch and upload it to the Arduino Uno.

Collect the temperature and altitude data from the floor to the ceiling of your classroom and then draw a graph displaying the results.

Can you explain the shape of the graph?

Did you know?



The Martian atmosphere is an extremely thin sheet of gas, principally carbon dioxide, that extends from the surface of Mars to the edge of space. The sun heats the surface of Mars and some of this heat goes into warming the gas near the surface. The heated gas then diffuses or convects up through the atmosphere. Thus, the gas temperature is highest near the surface and decreases as we increase altitude.

→ LINKS

Arduino blog, useful for finding the latest Arduino information:

<https://blog.arduino.cc/>

Guides to using various Arduino components:

<https://quarkstream.wordpress.com/>

Many Arduino based projects you can try at home can be found on Instructables:

<https://www.instructables.com/teachers/projects/?subjects=arduino>

HackADay hosts many other Arduino projects:

<https://hackaday.io/list/3611-arduino-projects>