Creative Design and Innovation

G10 Teacher's Guide

CREATIVE DESIGN
INNOVATION

Term 3 2017-18

Table of Content:

Contents

Instructional Planner:

Week	Period	Unit	Lesson	Task	Page	Key skills	Assessment focus
1	1	Unit 1: Architecture	**Lesson 1** **History of Architecture**	• Introduce the structure of the term based around a combination of preparation units and a final practical project. • Introduce key words and learning outcomes. • Explain what architecture is. • Introduce students to the history of architecture: timeline and the different design styles. • Facilitate as students complete activities 1.1-1.7. • Explain what a mood board is. • Facilitate as students complete activity 1.8. • Student reflection.	Pg. 14 Pgs. 15-18 Pg. 18 Pgs. 18-31 Pgs. 19-31 Pg. 32 Pgs. 34-35	• Describe and apply different styles in architecture • Define the roles and responsibilities of an architect	
	2		**Lesson 2** **Islamic architecture, architecture in the UAE and sustainable architecture**	• Introduce key words and learning outcomes. • Introduce students to Islamic architecture including the timeline. • Facilitate as students complete activity 2.1.	Pgs. 36-37 Pgs. 38-39 Pg. 40	• Examine Islamic architecture in comparison to	

				• **Introduce students to architecture in the UAE:** • Facilitate students to complete entrepreneurial activity 2.2. • Students to recognise some of the famous buildings in the UAE. • Facilitate as students complete activity 2.3. • Explain the influence of biomimicry in design. • Facilitate as students complete activity 2.4 and 2.5. • Introduce sustainable architecture. • Facilitate as students complete activities 2.6 and 2.7. • Student reflection.	Pgs. 40-54 Pg. 41 Pgs. 43-45 Pgs. 46-48 Pgs. 49-50 Pgs. 51 and 53 Pg. 54 Pgs. 56-57 Pgs. 59-60	architecture in the UAE • Create a research page or mood board on your chosen sustainable project or development	
2	1 & 2		**Lesson 3** **Architectural technical drawings**	• Introduce key words and learning outcomes. • Explain how and why bubble diagrams are used as part of the design process in architecture. • Facilitate as students complete activity 3.1. • Introduce the standard views used in architectural drawings. • Facilitate as students complete activities 3.2-3.6. • Student reflection.	Pg. 61 Pg. 62 Pgs. 62-63 Pgs. 63-68 Pgs. 64-71 Pgs. 72-73	• Identify architectural floor plans, sections and elevations • Sketch a plan, section and elevation	

3	1	**Unit 2: Powering a smart city**	**Lesson 1** **Essentials in Arduino**	• Introduce key words and learning outcomes for lesson 1. • Introduce students to the role of Arduino microcontroller in electronics. • State the importance of the Arduino Leonardo board and Arduino IDE software. • Facilitate as pupils complete activities 1.1-1.3. • Introduce students to the first program (On-Board LED) • Facilitate as student's complete activity 1.4. • Facilitate and provide feedback as student's go through the Arduino code syntax in activity 1.5. • Complete the final activity and student reflection. • State the importance of Arduino programming which must be included in the final project.	Pgs. 107-108 Pg. 109 Pgs. 110-115 Pgs. 110-114 Pg. 116 Pg. 117 Pgs. 118-119 Pg. 133	• Define key words • Explain the role of the Arduino microcontroller in electronics. • Identify the main parts of the Arduino board. • Recognise the layout of the Arduino IDE programming interface. • Explain the Arduino programming structure.	

	2	**Unit 2: Powering a smart city**	**Lesson 2 Analogue and digital signals**	• Introduce key words and learning outcomes for lesson 2. • Introduce students to digital and analogue signals. • Facilitate as pupils complete activity 2.1. • State the importance of Arduino functions. • Introduce students to Arduino’s digital function. • Facilitate as student’s complete digital output-practical work, and activities 2.2-2.5. • Facilitate and provide feedback as students complete the final activity. • Complete the student reflection. • State the importance of Arduino programming which must be included in the final project.	Pgs. 135-136 Pgs. 137-138 Pgs.139 Pgs. 140 Pgs. 141-149 Pgs. 142-149 Pg. 161 Pg. 162	• Define key words • Differentiate between digital and analogue signals. • Interpret how to generate a digital signal.	
4	1&2	**Unit 2: Powering a smart city**	**Lesson 3: Control structures**	• Introduce key words and learning outcomes for lesson 3. • Introduce students to the different types of control structures. • Facilitate as pupils complete activity 3.1. • Introduce students to sequence control structures. • Facilitate as student’s complete activity 3.2-3.3.	Pgs. 163-164 Pg. 164 Pg.165 Pgs. 166-167 Pgs. 166, 169	• Define key words • Implement sequence statements in Arduino sketches. • Implement ‘if’ and “if-	

				• Introduce students to selection control structures. • Facilitate as student's complete activity 3.4-3.5. • Facilitate and provide feedback as students complete the final activity. • Complete the student reflection. • State the importance of Arduino programming which must be included in the final project.	Pgs. 171-176 Pgs. 172, 177 Pgs. 191-192 Pg. 193	else" selection statements in Arduino sketches	
5&6	Week 5 + Week 6 (Period 1 ONLY)	**Unit 1: Architecture**	**Lesson 4** **Architecture project**	• Introduce key words and learning outcomes. • Introduce the design brief. • Introduce the different stages (1-4) of the design process. • Facilitate as students complete activities 4.1 and 4.6. • Introduce stage 5 (make it) of the design process. • Facilitate as students make a prototype architectural model. • Facilitate as students create their final model. • Facilitate as students evaluate their projects (stage 6). • Student reflection.	Pgs. 75-76 Pg. 77 Pgs. 78-95 Pgs. 78-95 Pg. 96 Pg. 97 Pgs. 98-102 Pg. 103 Pgs. 104-105	• Write a design specification for an architectural project • Generate and develop design ideas that meet the design brief • Develop a 3D architectural model using Fusion 360 • Produce the architectura	

						l model using 3D printing • Assemble all parts of the smart city • Write an evaluation for the completed city.	
6&7	Week 6 (Period 2 ONLY) + Week 7	**Unit 2: Powering a smart city**	**Project: Smart city features**	• Introduce key words and learning outcomes. • Introduce the students to the concept of a smart city. • Facilitate as students complete activities 4.1 and 4.2. • Introduce the students to the 8 possible features that can be included in the smart city. • Teachers will direct students to choose **only one** feature per group. • Divide students into groups and make sure each group has the required components. • Teacher will facilitate students as they build their electronic circuit. • Teacher will facilitate students as they test the function of their electronic circuit.	Pgs. 195-196 Pg. 196-200 Pgs. 198, 199	• Write a design specification for an architectural project • Generate and develop design ideas that meet the design brief • Develop a 3D architectural model using Fusion 360 • Produce the architectural model	

				- Students are encouraged to combine their model of the city that was designed in Week 6 with the electronic circuit. - Students should present their work. - Student reflection.		using 3D printing - Assemble all parts of the smart city - Write an evaluation for the completed city.	
8	1&2	**Assessment week**				-	

9	1-2	-	***Project improvement***	Teachers use this week to allow students finalise and improve their project.			

Lesson Plans:

Unit 1: Architecture - Lesson 1: History of architecture

Aim:

In this lesson students shall study the history of architecture along with the key dates for each design style. Students shall understand the importance of how the designs of the past still influence the designs of today. Students shall also differentiate the main design features for each style and then create a research page or mood board for a style they like the most.

Student Learning Outcomes:

Learning outcomes refer to what the student can expect from the lesson, teachers must share these outcomes with all students. Teachers are to tick the box when the outcome is achieved. Outcomes in **red** are essential.

Students should be able to:

- □ **Describe and apply different styles in architecture.**
- □ **Define the roles and responsibilities of an architect.**
- □ Explain and sketch the different design features of each design style.
- □ Compare the main design features of the old design styles to today's designs.
- □ Create a research page or mood board for one specific architectural style.
- □ **Apply entrepreneurial attributes and recognise them in the field of architecture.**

Keywords	**What are the keywords the students must learn?** • architecture • architect • archaeology • symmetry • mosaic • decorative • industrial revolution • entrepreneurship • social entrepreneurship • teamwork • creativity • passion • determination • risk taking • project management • leadership
Resources	**What resources are required?** • textbooks • projector • drawing equipment (pencil, ruler, eraser, etc.)
Prior Knowledge	• Create a sketch • Create a sketch in projected views • Research using the internet and other sources • Use ACCESS FM to analyse buildings

Possible teaching method(s) or approach for this lesson

- □ Collaborative Teaching (student centred)
- □ Instructional / Demonstrative Teaching (teacher centred)
- □ Inquiry-based Teaching (student centred)
- □ Lecture Style Teaching (teacher centred)
- □ Coach Style Teaching (teacher centred)
- □ Facilitator Style Teaching (student centred)

Essential and non-essential sections:

In some lessons it may not be possible to cover every section of the book due to time constraints or lesson variables. Below is a guideline to essential sections for examination and project knowledge.

Topic	Page	
	Essential	Non-essential/Self Study
What is architecture?	Pg. 18	
The role of an architect.	Pg. 21	
History of architecture: timeline and different styles.		Pgs. 18-31

Note about differentiation: *All lessons can be different depending on ability and success of the previous lesson. Place additional notes or activities to cater for differentiation where necessary throughout the lesson.*

Development (phases or chunks of learning):	**Assessment opportunities:**
Phase 1 of lesson (Connect) **Starter** *Teacher to give students an overview of the unit and to go through the learning outcomes that must be achieved.*	
Teacher to assess the prior knowledge of the students in relation to those described in the book.	**Questioning**
Teacher to discuss the importance of architecture in the UAE and how the buildings of today are connected to the economy of the country.	**Questioning**
Teacher Tip: Teacher to set high expectations which inspire, motivate and challenge pupils.	
Phase 2 of lesson (Activate) *Teacher to go through the key words required within the lesson and make sure the students understand the importance of these key words.*	

Phase 3 of lesson (Engage and Demonstrate) *Teacher to go through the various architectural styles and discuss the important features of each.* *Teachers to show the videos/website provided as QR codes in the book (links are provided under* ***resources*** *on the next page).* *Teachers must also ensure the students understand that the dates provided are approximate dates as there is no specific date an architectural style started and ended (ie. Modernism did not start on 1st January 1900).*	
Teacher to support the students as they complete the activities in the book. Some activities may be set as homework. **Teacher Tip:** Teacher to promote good progress and outcomes by students.	**Formal assessment**
Plenary (Consolidate) *Teacher to ensure that students have completed the activities and their individual mood boards.* **Teacher Tip:** Teacher to make accurate and productive use of assessment.	**Formal assessment**

Answer Key/Resources

QR code links:		
Page	**Topic**	**Link**
22	Stonehenge	https://www.youtube.com/watch?v=wf7xwHFuH2o
25	360° view of the Roman Colosseum	https://www.airpano.ru/files/Italy-Rome-Colosseum/2-2
31	Modernism	https://www.youtube.com/watch?v=Q4tqFabGgyc

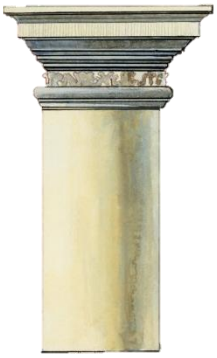
Activity 1.1 & 1.2

Answers will vary. Teachers should verify if the architect matches the building for activity 1.1.

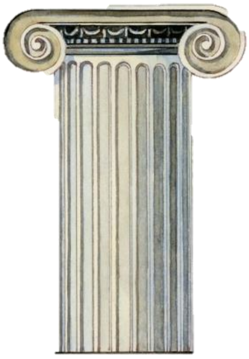
Activity 1.3

Answers may vary. Examples of answers could be the following; Project management, designing, customer service, site visits, marketing, visualizing, cost estimation.

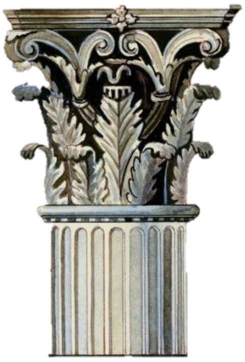
Activity 1.4



Greek Doric



Roman Ionic



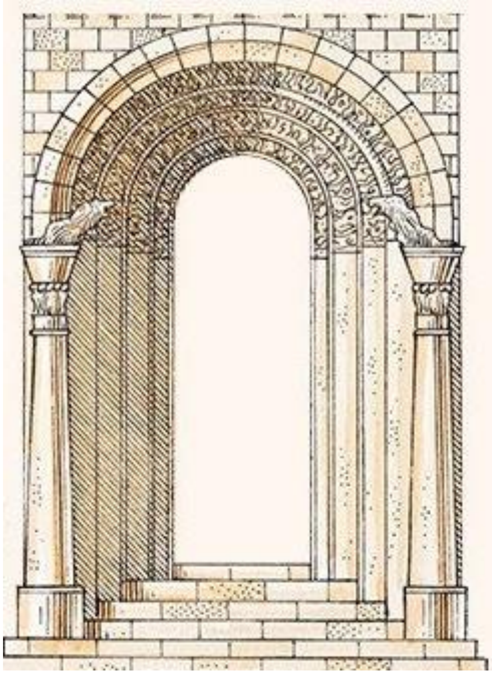
Roman Corinthian

Activity 1.5

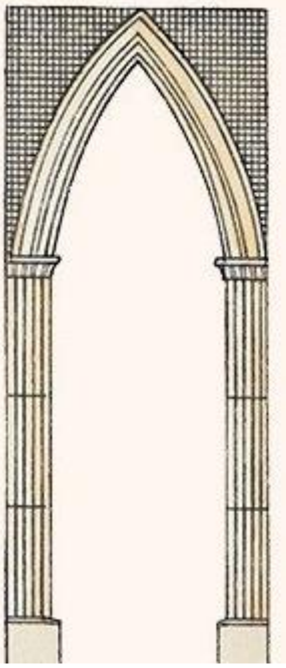
Classical era: Symmetry and proportion, use of columns, highly decorated.

Byzantine era: Large complex domes, brick and plaster were used along with stone, mosaics replaced carved decorations.

Activity 1.6

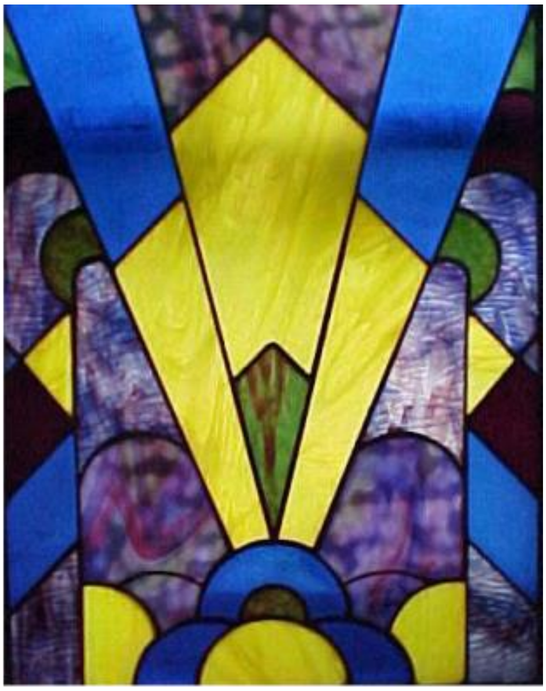


Romanesque arch



Gothic arch

Activity 1.7



Activity 1.8

Mood board or research page will vary depending on which architectural style has been chosen by the student.

Lesson 2: Islamic architecture, architecture in the UAE and sustainable architecture

Aim:

In this lesson students shall go through the history of Islamic architecture and architecture in the UAE and see if there are any connections between them. Students shall then investigate the key architectural features of some of the most famous buildings in the UAE. Biomimicry and designing for a hot climate also feature in this lesson. The lesson ends by students creating a research page/mood board in regards to one of the sustainable projects/developments they decide to research.

Student Learning Outcomes:

Learning outcomes refer to what the student can expect from the lesson, teachers must share these outcomes with all students. Teachers are to tick the box when the outcome is achieved. Outcomes in **red** are essential.

Students should be able to:

- □ Identify the key features of Islamic architecture.
- □ **Examine Islamic architecture in comparison to architecture in the UAE.**
- □ **Create a research page or mood board on your chosen sustainable project/development.**
- □ List and explain the responsibilities of a social entrepreneur and relate this to sustainable design.

Keywords	**What are the keywords the students must learn?** • courtyard • hypostyle • mihrab • minaret • marquetry • calligraphy • biomimicry
Resources	**What resources are required?** • textbooks • projector • drawing equipment (pencil, ruler, eraser, etc.)
Prior Knowledge	• Create a sketch • Create a sketch in projected views • Research using the internet and other sources • Use ACCESS FM to analyse buildings • Understand the fundamentals of 3D printing

Possible teaching method(s) or approach for this lesson

- □ Collaborative Teaching (student centred)
- □ Instructional / Demonstrative Teaching (teacher centred)
- □ Inquiry-based Teaching (student centred)
- □ Lecture Style Teaching (teacher centred)
- □ Coach Style Teaching (teacher centred)
- □ Facilitator Style Teaching (student centred)

Essential and non-essential sections:

In some lessons it may not be possible to cover every section of the book due to time constraints or lesson variables. Below is a guideline to essential sections for examination and project knowledge.

Topic	**Page**	
	Essential	**Non-essential/Self Study**
Islamic architecture.	Pgs. 38-39	
Architecture in the UAE.	Pgs. 40-48	
Biomimicry.		Pgs. 49-51
Architecture and 3D printing in the UAE.	Pg. 52	
Design objectives when designing for a hot climate.		Pgs. 53-54
Sustainable architecture.		Pg. 54
Passive and active design.		Pg. 55
Renewable and non-renewable resources.		Pgs. 56-57

***Note about differentiation:** All lessons can be different depending on ability and success of the previous lesson. Place additional notes or activities to cater for differentiation where necessary throughout the lesson.*

Development (phases or chunks of learning):	**Assessment opportunities:**
Phase 1 of lesson (Connect) **Starter** *Teacher to go through the learning outcomes of the lesson.*	
Teacher to assess the prior knowledge of the students in relation to the ones described in the book.	**Questioning**
Teacher to ask students if they know any famous buildings in the Islamic world and if any of these designs have influenced designs of buildings in the UAE.	**Oral assessment**
Teacher Tip: Teacher to set high expectations which inspire, motivate and challenge pupils.	

Teacher to demonstrate good subject and curriculum knowledge. **Phase 2 of lesson (Activate)** *Teacher to go through the key words required within the lesson and make sure the students understand the importance of these key words.* *Teacher to go through the Islamic architectural timeline with the students and discuss the key features of these buildings.* **Phase 3 of lesson (Engage and Demonstrate)** *Teacher to discuss the key architectural features that are common in Islamic architecture and how they have developed over time.* *Teacher to go through the history of architecture in the UAE and the main design considerations for designing in this kind of environment. Students must understand what materials were used to create buildings before the introduction of cement and how the introduction of oil has influenced the design of buildings and the infrastructure in the country.* *Students must recognise and write facts about some of the famous buildings in the UAE.* *Teacher to explain what biomimicry is and how important it is in the design world by giving examples shown in the book.* *Teacher to discuss the intentions of the UAE in regards to 3D buildings in the future.* *Teacher to explain how the weather has a major influence on the designs of buildings in the UAE.* *Teacher to discuss sustainable architecture and renewable and non-renewable sources.* *Teacher to support the students as they complete the activities in the book. Some activities may be set as homework.* **Teacher Tip:** Teacher to promote good progress and outcomes by students. **Plenary (Consolidate)** *Teacher to ensure that students have completed the activities and their individual mood boards.* **Teacher Tip:** Teacher to make accurate and productive use of assessment.	**Formal assessment** **Summative assessment**

Answer Key/ Resources

QR code links:		
Page	***Topic***	***Link***
44	Al Bahr Towers	https://www.youtube.com/watch?v=GSbJIlpv4Dc
50	Biomimicry	https://www.youtube.com/watch?v=iMtXqTmfta0
50	Termite mounds	https://www.youtube.com/watch?v=Brl_a4I6qqc
52	Office of the future	https://www.youtube.com/watch?v=APTt2UoXRTI

Activity 2.1

1. Answers may vary. For example, students might design a minaret, mihrab or dome.

2. C – a tall tower used to call Muslims for prayer.

Activity 2.2

Answers may vary. For example; construction, real estate agencies, travel and tourism.

Activity 2.3

1. Coral stone and mud bricks.

2. Hot climate, social life and religious beliefs.

3. Note: Answers may vary for the facts given by students, examples are given below.

Bee'ah Headquarters

Architect: Zaha Hadid
Year: 2014-2019
Fact(s): The headquarters is also being constructed using recycled materials.

Louvre Abu Dhabi

Architect: Jean Nouvel
Year: 2017
Fact(s): The dome is composed of eight superimposed layers: four outer layers in stainless steel and four inner layers separated by a steel structure that is five metres high.

Masdar City

Architect: Foster and Partners
Year: 2010
Fact(s): The city relies on solar energy and other renewable energy sources.

Dubai Frame

Architect: Fernando Donis
Year: 2018
Fact(s): The frame measures 150 metres high and 93 metres wide.

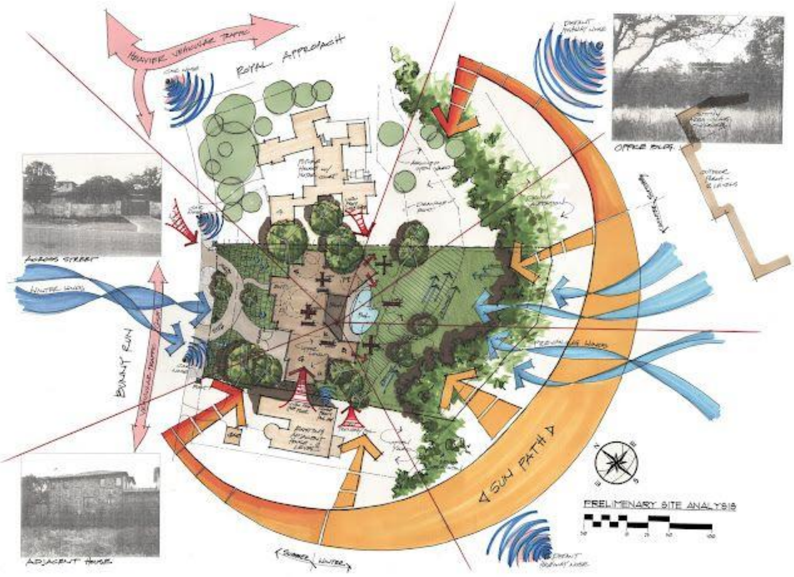
Activity 2.4

a. Lotus flower and paint. The lotus flower can naturally repel dirt and dust, which allows the petals to look clean at all times. The paint is made with similar properties.

b. Bur and velcro. Bur has hooks or teeth that can stick to fur or clothing. Velcro has the same features and was invented by a Swiss engineer.

Activity 2.5

To promote heat loss.



Activity 2.6

1. Answers may vary. Examples for recyclable could be plastic bottles and drink cans. Examples for non-recyclable could be motor oil.

2. Answers may vary. Examples could be creativity, knowledge of environmental issues, social awareness and team spirit.

Activity 2.7

Mood boards will vary depending on the building or project chosen by the students.

Unit 1: Architecture

Lesson 3: Architectural technical drawings

Aim:

This lesson aims to explore how different views of a building can reveal their internal and external structures.

Student Learning Outcomes:

Learning outcomes refer to what the student can expect from the lesson, teachers must share these outcomes with all students. Teachers are to tick the box when the outcome is achieved. Outcomes in **red** are essential.

Students should be able to:

- □ **Identify architectural floor plans, sections and elevations.**
- □ Sketch a bubble diagram.
- □ **Sketch a plan, section and elevation.**

Keywords	**What are the keywords the students must learn?** • bubble diagram • floor plan • section • elevation
Resources	**What resources are required?** • textbooks • projector • drawing equipment (pencil, ruler, eraser, etc.)
Prior Knowledge	• Create a sketch • Create a sketch in projected views • Research using the internet and other sources

Possible teaching method(s) or approach for this lesson

- □ Collaborative Teaching (student centred)
- □ Instructional / Demonstrative Teaching (teacher centred)
- □ Inquiry-based Teaching (student centred)
- □ Lecture Style Teaching (teacher centred)
- □ Coach Style Teaching (teacher centred)
- □ Facilitator Style Teaching (student centred)

Essential and non-essential sections:

In some lessons it may not be possible to cover every section of the book due to time constraints or lesson variables. Below is a guideline to essential sections for examination and project knowledge.

Topic	Page	
	Essential	Non-essential/Self Study
Bubble diagram.		Pgs. 62-63
Standard views used in architectural drawings.	Pgs. 63-68	

Note about differentiation: *All lessons can be different depending on ability and success of the previous lesson. Place additional notes or activities to cater for differentiation where necessary throughout the lesson.*

Development (phases or chunks of learning):	Assessment opportunities:
Phase 1 of lesson (Connect) **Starter** *Teacher to go through the learning outcomes of the lesson.* *Teacher to assess the prior knowledge of the students in relation to the ones described in the book.* **Teacher Tip:** Teacher to plan and teach a well-structured lesson.	
Phase 2 of lesson (Activate) *Teacher to go through the key words required within the lesson and make sure the students understand the importance of these key words.*	**Oral assessment**
Phase 3 of lesson (Engage and Demonstrate) *Teacher to discuss how and why bubble diagrams are important at the beginning of the design process for architects.* *Teacher to explain the three main standard views in architectural technical drawings: floor plan, section and elevation.* *Teacher to support the students as they complete the activities in the book. Some activities may be set as homework.* **Teacher Tip:** Teacher to adapt teaching to respond to the strengths and needs of all students.	**Formal assessment**
Plenary (Consolidate) *Teacher to ensure that students have completed all the activities.* **Teacher Tip:**	**Summative assessment**

Teacher to make accurate and productive use of assessment.	

Answer Key

Activity 3.1

Answers will vary. An example is shown on page 62 in the book.

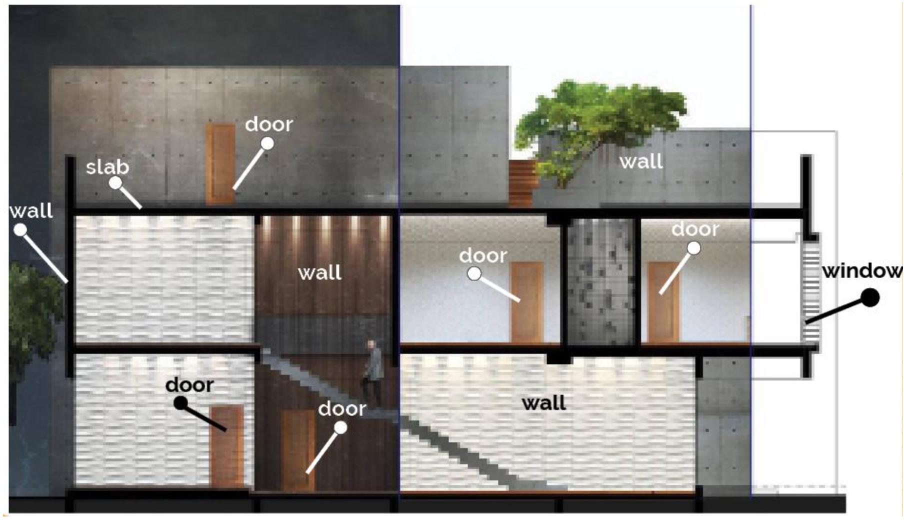
Activity 3.2

1. Bedroom 2. Balcony 3. Bedroom 4. Bathroom 5. Dining room 6. Living room
7. Balcony 8. Kitchen

Activity 3.3 (Extension)

Drawings of floor plans will vary.

Activity 3.4



Activity 3.5 (Extension)

Drawings of sections will vary.

Activity 3.6

1. b – site plan   *2. b – sections*   *3. a – elevations*   *4. a – floor plan*
5. c – interior view   *6. c – elevations*   *7. b – section*

Unit 2 - Lesson 1: Essentials of Arduino

Aim:

This lesson aims to introduce you to Arduino and explains the basic features of Arduino IDE software. The session starts with a brief introduction to Arduino, and the Arduino board, introducing its various ports and hardware components. This is followed by an explanation of the Arduino IDE software and its key features. A step by step procedure on how to upload a test sketch to the Arduino board will be the final task in this lesson.

Student Learning Outcomes: Learning outcomes refer to what the student can expect from the lesson, Teachers must share these outcomes with all students. Teachers are to tick the box when the outcome is achieved. Learning outcomes can be assessed using oral questioning and the written activities.

Students should be able to:

1. Explain the role of the Arduino microcontroller in electronics.
2. Identify the main parts of the Arduino board.
3. Recognise the layout of the Arduino IDE programming interface.
4. Explain the Arduino programming structure.
5. Explain how to control the LCD screen.
6. Explain how to use Arduino's serial monitor.
7. Configure the Arduino IDE software to work with the Arduino board.
8. Use pseudocode and flowcharts to understand how a program works.

Keywords	**What are the keywords the students must learn?** 1. microcontroller 2. Arduino board 3. pins 4. power supply 5. GND 6. IDE 7. sketch 8. serial monitor 9. LCD 10. variables
Resources	**What resources are required?** 11. textbooks 12. projector 13. calculator 14. Arduino board
Prior Knowledge	15. Use breadboards for building electronic circuits. 16. Identify the basic electronic components.

Possible Teaching Method(s) or Approach for this lesson

17. Collaborative Teaching (student centred)
18. Instructional / Demonstrative Teaching (teacher centred)
19. Inquiry-based Teaching (student centred)
20. Lecture Style Teaching (teacher centred)
21. Coach Style Teaching (teacher centred)
22. Facilitator Style Teaching (student centred)

Essential and non-essential Sections:

In some lessons it may not be possible to cover every section of the book due to time constraints or lesson variables. Below is a guideline to essential sections for examination and project knowledge.

Topic	Page	
	Essential	**Non-essential/Self Study**
What is Arduino?	Pg. 107-113	
Arduino IDE software	Pg. 113-114	
Arduino code structure	Pg. 115	
Example – On-board LED	Pg. 116-117	
Arduino code syntax	Pg. 118-119	
Arduino- LCD and Serial monitor		Pg. 120-128
Pseudocode and flowcharts		Pg. 129-130
Arduino code using pseudocode and flowcharts		Pg. 131-132

Notes for differentiation: All lessons can be different depending on ability and success of previous lesson. Place additional notes or activities to cater for differentiation where necessary throughout the lesson.

Development [Phases or chunks of learning]:	**Assessment Opportunities:**
Note: *All lessons start with Phase 1, Lessons can move back and forth between phases 2 and 3 as content is covered and then students engage. All lessons must finish with phase 4 to evaluate learning.*	
Phase 1 of lesson (Connect) **Starter** Teacher to introduce students to the lesson aim. Teacher to place all student learning outcomes on the board and ensure student understanding of aims and outcomes. Discuss prior knowledge of breadboards and basic electronic components. Show motivational videos / models to outline the end goal of the term.	**Questioning**
Teacher Tip: *Teacher to set high expectations which inspire, motivate and challenge pupils.*	
Phase 2 of lesson (Activate) Teacher to introduce all key words, discuss meaning and ensure understanding before progressing. Teacher to introduce the role of Arduino microcontroller in electronics, while students research the topic.	**Questioning**
Question students on what aspects are new to them when compared to prior knowledge discussion. Teacher to lead the class discussion about parts of the Arduino board and the IDE software layout. Teacher to introduce the Arduino code structure and allow students to explore their first Arduino program. Teacher to monitor the students' progress throughout the lesson by using the different assessment opportunities.	
Teacher Tip: *Teacher to demonstrate good subject and curriculum knowledge*	
Phase 3 of lesson (Engage and Demonstrate) Task 1: Ask students to find a partner and make a mind map about the advantages, disadvantages and uses of the Arduino microcontroller. Students to complete Activity 1.1.	**Mind Map**

Teacher to facilitate as peer teaching takes place. Task 2: Divide students into groups and assign each group an Arduino board. Allow students to label the Arduino board as a group. After they finish students will complete Activity 1.2. Task 3: Students to be given an IDE software layout, each group to fill out one section. The answers will be given through a class discussion. Students demonstrate learning by completing activity 1.3. Task 4: Students to upload and test their first On-board LED program individually. Students demonstrate learning by completing activity 1.4 in groups. Task 5: Students to study the Arduino code syntax and complete activity 1.5.	
Teacher to facilitate as collaborative learning takes place.	**Peer teaching**
Students demonstrate learning by completing activities 1.1-1.5	**Written Activities 1.1-1.3**
Teacher Tip: *Use group work as appropriate, get to know your class and organise groups to support mixed ability's.* **Phase 4 Plenary (Consolidate)** Teacher to facilitate as students evaluate learning. Question pupils on what they have learned. Have learning outcomes been met? Has the lesson aim been achieved?	
All students must complete the final activity and the student evaluation.	**Written Activity 1.4-1.5**
Students to complete activities 1.1-1.5 and finish for homework if incomplete.	**Oral Assessment** **Student evaluation**

Answer Key/ Resources

QR code links:		
Page	**Topic**	**Link**
110	Various Arduino projects	https://www.youtube.com/watch?v=B7dtdBgOHWM
117	Fritzing	http://fritzing.org/download/?donation=0
117	TinkerCad	https://www.tinkercad.com/#/

Activity 1.1

. Take out your Arduino board and place it in front of you. Can you identify the board model?

The board is an Arduino Leonardo model.

Activity 1.2

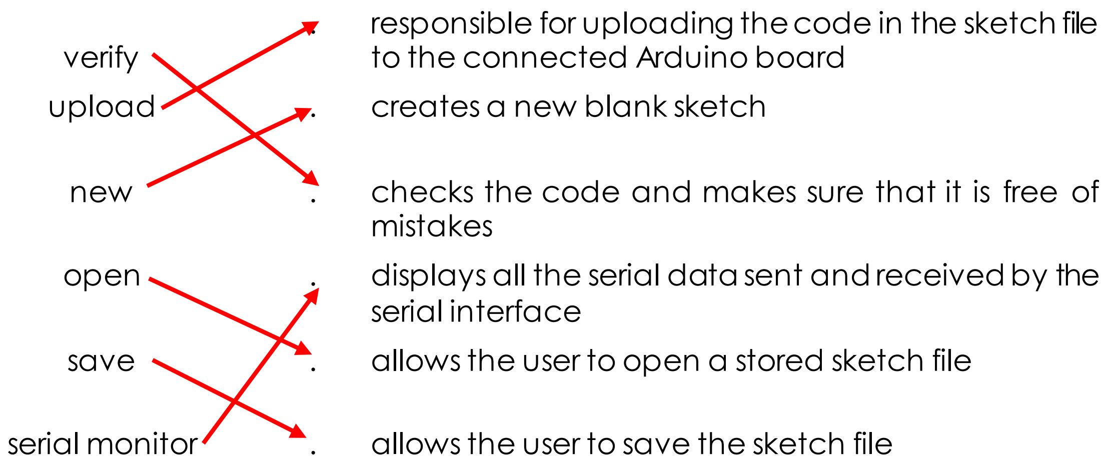
. Identify the four power socket pins and write down their names.
1. 3.3V pin
2. 5V
3. GND pins
4. VIN pins

. How many analogue pins are there in the Analog IN socket? Name them.
6 analogue pins. A0, A1, A2, A3, A4, A5.

. How many digital pins are there in the digital socket?
14 digital pins.

Activity 1.3

Match the toolbar button to its description.



. responsible for uploading the code in the sketch file to the connected Arduino board

. creates a new blank sketch

. checks the code and makes sure that it is free of mistakes

. displays all the serial data sent and received by the serial interface

. allows the user to open a stored sketch file

. allows the user to save the sketch file

Activity 1.4

Check the status of the ‘L LED’. What do you observe?
The LED will turn on and off (blink) with a delay of 1 second.

Activity 1.5

Use the information provided in **Error! Reference source not found.** to answer the following questions:

33. Find the syntax errors in the lines of code below, and then correct them.

```
int variable_$ = 3;
setup(){
pinMode(3, INPUT)
```

Error	Correction	Justify
int variable_$ = 3;	int variable= 3;	the variable name must be meaningful, include no spaces or special characters
pinMode(3, INPUT)	pinMode(3, INPUT);	semicolon is used to end a statement

setup(){
pinMode(3, INPUT)
	setup(){
	pinMode(3, INPUT)
	}
		unbalanced braces will result in a compiler error

34. Mark the lines below as a comment using two different methods.

```
 Blink
 Turns on an LED on for one second, then off for one second, repeatedly.
```

Method 1:

```
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
*/
```

Method 2:

```
//Blink
//Turns on an LED on for one second, then off for one second, repeatedly.
```

35. Declare the variables below.

declare ‘y’ as an integer, and set its initial value to 0

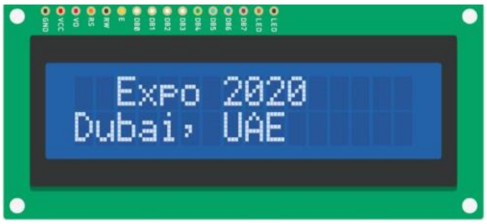
```
int y= 0;
```

declare ‘age’ as a character

	char age;	

Activity 1.6

Write the Arduino code to print the following message on the LCD screen:



Arduino code:

```
void setup() {
  lcd.begin(16, 2);

  lcd.setCursor(2, 0);
  lcd.print("Expo 2020");

  lcd.setCursor(0, 1);
  lcd.print("Dubai, UAE");
}
```

Activity 1.7

Modify the Arduino example to print your name. Take a picture of the LCD screen and paste it below.

Arduino code:

```
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print(“NAME");
}

void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);
```

}

Activity 1.8

1. Try pressing the Arduino's Reset button a few times. What happens?

   Whenever the reset button is pressed the message is printed once on the serial monitor.

2. Now, write the print function inside the loop() function. Verify and upload the code and then observe what happens.

   The message will be displayed on the serial monitor repeatedly.

Activity 1.9

Write the proper statements for the following statements:

36. Define an integer variable, name it 'count' and give it a value of zero.

```
int count=0;
```

37. Print out an explanation message.

```
Serial.println("Counting integer numbers starting from 0");
```

38. Display the current count.

```
Serial.print("Count = ");
Serial.println(count);
```

39. Implement a procedure to increase the counts.

```
count=count +1;
```

40. Add 100 milliseconds of delay time.

```
delay(100);
```

41. Which one of those steps needs to be done only once?

   The counter initialisation and the explanation message.

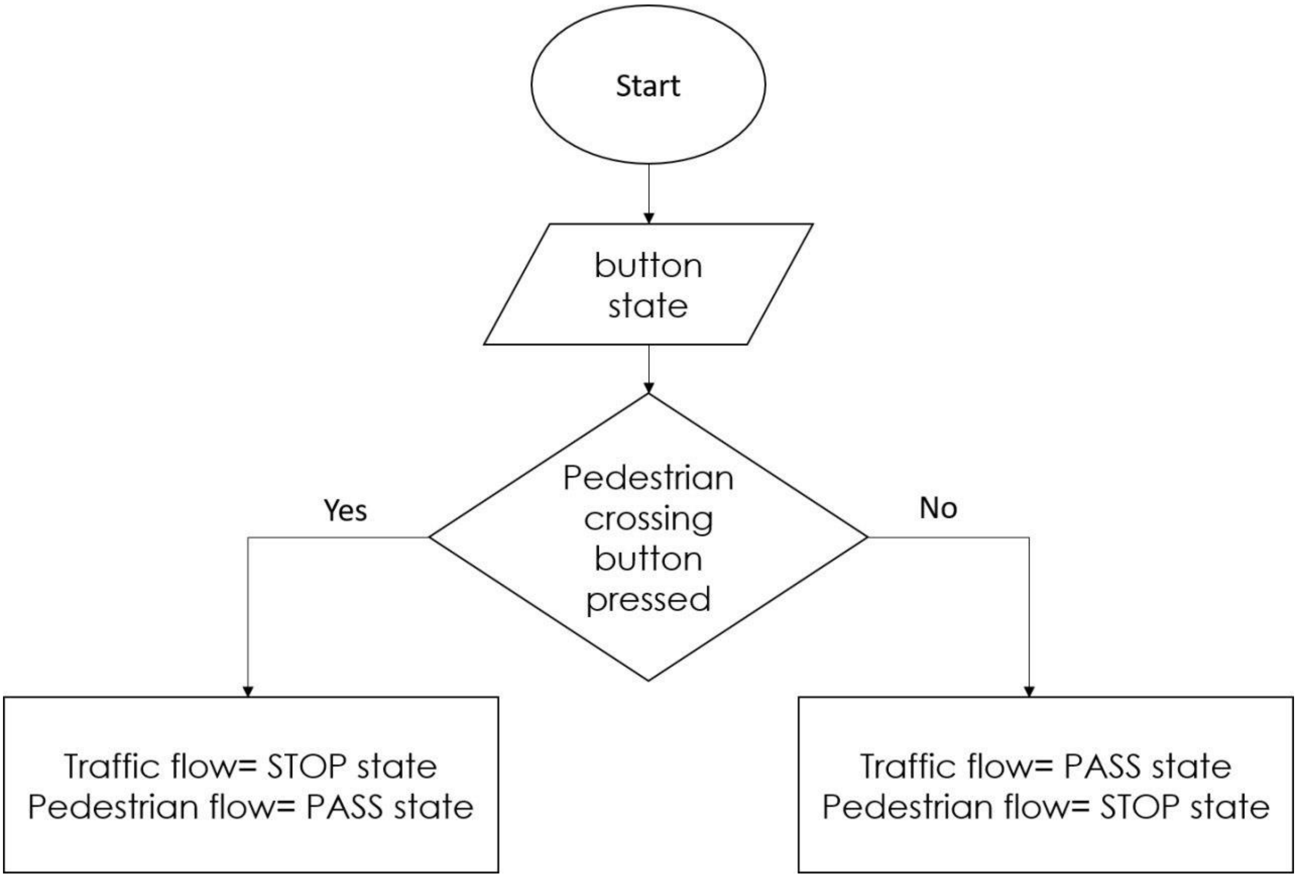
42. Which one of those steps needs to be run continuously in a loop?

   Updating the counter and displaying its value.

Activity 1.10

Using the flowchart diagrams shown in **Error! Reference source not found.**, draw the flowchart for the following traffic system:

43. If the pedestrian crossing button is pressed → change to the **Stop state** for traffic flow, and **Pass state** for pedestrian flow.
44. Otherwise, keep the Pass state for traffic flow and the Stop state for pedestrian flow.


Start
button state
Pedestrian crossing button pressed
Yes
No
Traffic flow= STOP state
Pedestrian flow= PASS state
Traffic flow= PASS state
Pedestrian flow= STOP state

Activity 1.11

1. What is a flowchart? Why do I need it?

A flowchart is a visual representation of the sequence of the process. It shows how the code is executed. It also helps you identify the different elements of the process and understand how the different steps are linked together.

2. What is a pseudocode? Why do I need it?

Pseudocode is a method to communicate the design problem using English-like statements. It is used to outline the structure of the code, making the process of writing the actual code much easier.

Final activity

Write an Arduino program to calculate the voltage value in Ohm's law, knowing that the current is 2 mA, and the resistance is 100 Ω.

Hint: Ohm's law V = I× R

45. Define the variables.

```
float current=0.002;
int resistance=100;
float voltage;
```

46. Print out an explanation message (comment).

```
Serial.println("Calculating the voltage using Ohm's law");
```

47. Implement a procedure to calculate the voltage.

```
voltage= current*resistance;
```

48. Print out the voltage value on the serial monitor.

```
Serial.println("Voltage is:");
Serial.println(voltage);
```

49. Print out the voltage value on the LCD.

```
void setup() {
 lcd.begin(16, 2);

 lcd.setCursor(0, 0);
 lcd.print("Voltage is:");

 lcd.setCursor(0, 1);
 lcd.print(voltage);
}
```

50. Write the complete Arduino code and run the program.

```
#include <LiquidCrystal.h>

float current=0.002;
int resistance=100;
float voltage;

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
```

```
void setup() {

Serial.begin(9600);
Serial.println("Calculating the voltage using Ohm's law");

}

void loop() {
voltage=current*resistance;

Serial.println("Voltage is:");
Serial.println(voltage);

delay(1000);

  // set up the LCD's number of columns and rows:
lcd.begin(16, 2);

lcd.setCursor(0, 0);
lcd.print("Voltage is:");
lcd.setCursor(0, 1);
lcd.print(voltage);

delay(1000);
}
```

Lesson 2: Analogue and digital signals

Aim:

This lesson aims to introduce you to the world of signals. You will learn how to read the digital signal on Arduino using digital input statements. The content explains in detail the procedure for generating and reading digital signals.

Student Learning Outcomes: Learning outcomes refer to what the student can expect from the lesson, Teachers must share these outcomes with all students. Teachers are to tick the box when the outcome is achieved. Learning outcomes can be assessed using oral questioning and the written activities.

Students should be able to:

1. Differentiate between digital and analogue signals.
2. Identify how to read digital signals in Arduino.
3. Interpret how to generate a digital signal.
4. Identify how to read analogue signals in Arduino.
5. Describe the function of PWM signals in electric circuits.

Keywords	**What are the keywords the students must learn?** 1. digital signal 2. floating state 3. analogue signal 4. PWM 5. time period 6. frequency 7. duty cycle
Resources	**What resources are required?** 8. textbooks 9. projector 10. calculator 11. Arduino board
Prior Knowledge	• Identify the fundamentals of Arduino programming. • Recognise the structure of LEDs, push-buttons, and potentiometers. • Use breadboards for building electronic circuits. • Identify the basic electronic components.

Possible Teaching Method(s) or Approach for this lesson

12. Collaborative Teaching (student centred)
13. Instructional / Demonstrative Teaching (teacher centred)
14. Inquiry-based Teaching (student centred)
15. Lecture Style Teaching (teacher centred)
16. Coach Style Teaching (teacher centred)
17. Facilitator Style Teaching (student centred)

Essential and non-essential Sections:

In some lessons it may not be possible to cover every section of the book due to time constraints or lesson variables. Below is a guideline to essential sections for examination and project knowledge.

Topic	Page	
	Essential	Non-essential/Self Study
What are signals?	Pg. 137-139	
Getting started	Pg. 140	
Digital input	Pg. 141	
Read the digital input	Pg. 141-142	
Digital input- practical work	Pg. 143- 145	
Digital output	Pg. 146	
Digital output- practical work	Pg. 147-149	
Analogue input		Pg.150
Analogue input- practical work		Pg.150-154
Analogue output		Pg. 155-156
Analogue output- practical work		Pg. 157-159

Notes for differentiation:

All lessons can be different depending on ability and success of previous lesson. Place additional notes or activities to cater for differentiation where necessary throughout the lesson.

Development [Phases or chunks of learning]:	**Assessment Opportunities:**
Note: *All lessons start with Phase 1, Lessons can move back and forth between phases 2 and 3 as content is covered and then students engage. All lessons must finish with phase 4 to evaluate learning.*	
Phase 1 of lesson (Connect) **Starter** Teacher to introduce students to the lesson aim. Teacher to place all student learning outcomes on the board and ensure student understanding of aims and outcomes. Discuss prior knowledge of Arduino programming and basic electronic components. Show motivational videos / models to outline the end goal of the term. **Teacher Tip:** *Teacher to set high expectations which inspire, motivate and challenge pupils.*	**Questioning.**
Phase 2 of lesson (Activate) Teacher to introduce all key words, discuss meaning and ensure understanding before progressing. Teacher to introduce the topic of digital and analogue signals, while students think-pair-share their thoughts. Question students on what aspects are new to them when compared to prior knowledge discussion. Teacher to introduce Arduino functions while students explore these functions through various activities. Teacher to monitor the students' progress throughout the lesson by using the different assessment opportunities. **Teacher Tip:** *Teacher to demonstrate good subject and curriculum knowledge.*	**Questioning / Written Activity 2.1**
Phase 3 of lesson (Engage and Demonstrate) Task 1: Students to study the first most important Arduino function (digitalRead). They will go through digital input-practical work to explore and test this function as groups. Students demonstrate learning by completing activity 2.2-2.3.	**Written Activity 2.2-2.3**

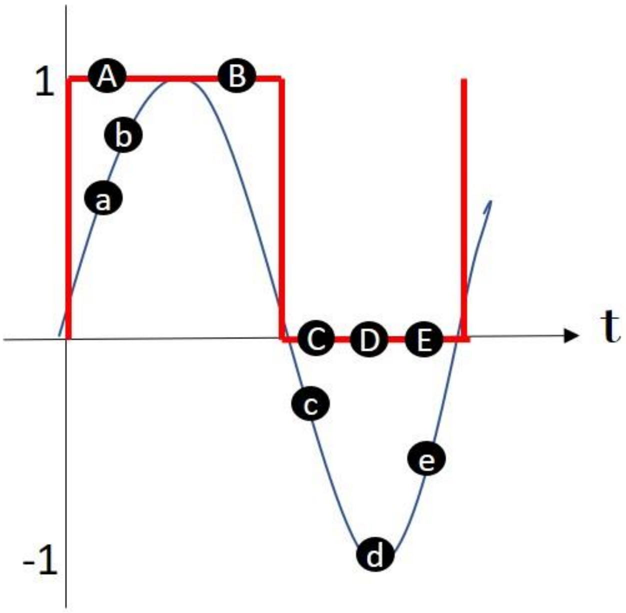
Task 2: Students to study the second most important Arduino function (digitalWrite). They will go through digital output- practical work to explore and test this function as groups. Students demonstrate learning by completing activity 2.4-2.5. Task 3: Divide students into two groups assigning each group an Arduino function. Use an "Each one teach one" approach to have groups show their understanding and improve other groups understanding of Arduino functions. Teacher to facilitate as peer teaching takes place. **Teacher Tip:** *Use groupwork as appropriate, get to know your class and organise groups to support mixed ability's.* **Phase 4 Plenary (Consolidate)** Teacher to facilitate as students evaluate learning. Question pupils on what they have learned. Have learning outcomes been met? Has the lesson aim been achieved? All students must complete student reflection. Students to complete activities 2.1-2.5 and the final activity for homework if incomplete. *This lesson should be conducted in one lesson (1 period – 45min)*	**Peer teaching** **Written Activity 2.4-2.5** **Oral Assessment** **Student evaluation**

Answer Key/ Resources

QR code links:		
Page	***Topic***	***Link***
138	Differences between analogue and digital signals	http://qrs.ly/c3690tu
143	Push-button tutorial	https://www.youtube.com/watch?v=wxjerCHCEMg

Activity 2.1

Using the figure below, find the corresponding values of the labelled letters, and then record their values in the table.



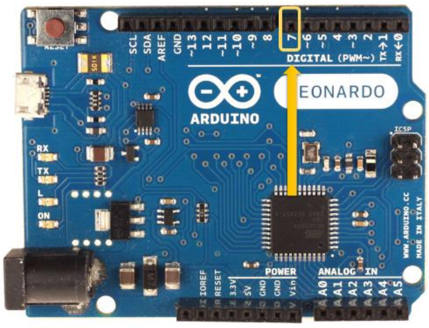
Analogue signal	Digital signal
a = 0.5	A = 1
b = 0.7	B = 1
c = -0.2	C = 0
d = -1	D = 0
e = -0.5	E = 0

Using the recorded values, what is the difference between the analogue and digital signals?

Digital signals have a finite number of values, either 0 or 1.
Analogue signals have an infinite number of values.

Activity 2.2

The mode for an electronic component connected to Arduino digital pin number **7** is **input**. Write the initialisation statement for this pin using the pinMode function.



```
pinMode(7, INPUT);
```

Activity 2.3

Write a code that prints the status of a push-button on the serial monitor.

When the push-button is pressed, the serial monitor should display the value 1. When the push-button is released, the serial monitor should display the value 0. Follow the instructions below.

21. Open the Arduino IDE software, and then click file → new.
22. In the setup() function, define pin 7 (push-button) as an INPUT.
23. In the loop() function, print a sentence on the serial monitor to display the signal value using the function → Serial.println(); .
24. Wait 500 milliseconds before the next loop using the function, delay(500); .
25. Verify and upload the code to read the value from pin 7.

Arduino code:

```
void setup()
{
Serial.begin(9600);
while (!Serial);
pinMode(7,INPUT);
}

void loop()
{
// print a sentence on the serial monitor
Serial.print("The value of the signal is = ");
Serial.println(digitalRead(7));     // print the signal value
delay(500);       // wait for 500 milliseconds before the next loop

}
```

Activity 2.4

Write a code that defines a digital component as an output pin, and write it within the setup() function.

```
Void setup(){
Serial.begin(9600);
pinMode(7,OUTPUT);

}
```

Where do you put the command to turn the digital components ON and OFF for a specific amount of time?

In the loop() function, using the delay command.

Activity 2.5

Write a full code to flash the LED ON and OFF. Follow the instructions below.

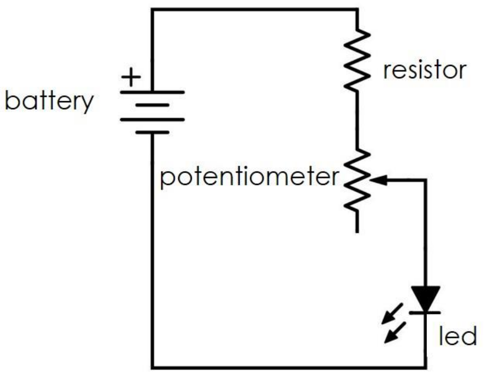
29. Turn the LED ON.
30. Apply a delay of 1 second.
31. Turn the LED OFF.
32. Apply a delay of 1 second.
33. Repeat again.

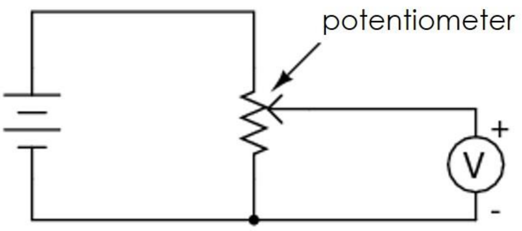
Arduino code:

```
void setup()
{
Serial.begin(9600);
while (!Serial);
pinMode(7,OUTPUT);
}

void loop()
{
digitalWrite(7.HIGH);
delay(1000);
digitalWrite(7,LOW);     // print the signal value
delay(1000);        // wait for 1000 milliseconds before the next loop

}
```

Activity 2.6

Identify whether the potentiometer in the schematic diagrams below was used as a voltage divider or as a variable resistor.





-- --

variable resistor

35. When only two legs of the potentiometer are connected, the wiper and one of the end terminals, then it acts as a variable resistor.

voltage divider

36. When all three legs of the potentiometer are connected, and a voltage is applied across the end terminals, then it acts as a voltage divider.

Activity 2.7

Write a code to read the value of the potentiometer and display it on the serial monitor.

38. Write a statement to define and initialise the variable x as an integer.

```
int x=0;
```

39. Write a statement to display the value of variable x on the serial monitor.

```
Serial.println("The potentiometer value is:");
Serial.println(x);
```

40. Write a statement to execute a delay of 100 milliseconds.

```
delay(100);
```

41. Finalise your code and run it on your Arduino.

```
int x = 0;

void setup() {
Serial.begin(9600);
}
void loop() {
x=analogRead(2);
Serial.println("The potentiometer value is:");
Serial.println(x);
delay(100);
}
```

Activity 2.8

Write a code to manipulate the brightness of the LED as detailed below.

43. Define a variable to store the PWM value.
44. Initialise an output pin.
45. Generate the PWM signal using the PWM value.
46. Apply a delay to observe the brightness.
47. Increase the PWM value by 10.
48. Repeat from step 3.

Arduino code:

```
int Brightness = 0;

void setup() {
pinMode(9, OUTPUT); // Define the pin #9 as an output pin.
}

void loop() {
analogWrite(9, Brightness); // Generate the PWM signal at pin #9
delay(100);     // apply a delay of 100 milliseconds

Brightness = Brightness + 10; // increase the Brightness value by 10

}
```

Final activity

49. Find the errors in the Arduino code below. Then, correct the mistakes.

Hint: There are 10 syntax errors in the code.

```
50. potentiometer=4;
51. int LED = 14;
52. int pushbutton=3;

53. void setup() {
54. pinMode(potentiometer, OUTPUT);
55. pinmode(pushbutton, INTPUT);
56. }

57. void loop() {
58. int val = analogRead(potentiometer);
59. if(pushbutton ==LOW){ //if sensor value is one
60. digitalWrite(LED,HIGH)
61. delay(val); //wait for 2 seconds
62. digitalwrite(LED,LOW);
63. }
64. else{
65. digitalWrite(LED,low);
66. }
```

Line No.	Syntax error	Correction
1	potentiometer=4;	int potentiometer=4;
2	int LED =14;	int LED =10; /*digital I/O pins: 2-13 (pins 0 and 1 are reserved for Tx and Rx)*/
5	pinMode(potentiometer, OUTPUT);	pinMode(LED, OUTPUT); /*the potentiometer is an analogue electronic device, hence, it doesn't need to be defined using the pinMode function. Unlike the push-button and the LED, where both of them are digital components and their pinMode must be defined.*/ /*if you were to define the potentiometer, then it's an input device not an output*/
6	pinmode(pushbutton, INTPUT);	pinMode(pushbutton, INPUT);
10	//if sensor value is one if(push-button ==LOW){	//if sensor value is one if(pushbutton ==HIGH){
11	digitalWrite(LED,HIGH)	digitalWrite(LED,HIGH);

12	//wait for 2 seconds
	delay(val);
		//wait for 2 seconds
		delay(2000);

13	digitalwrite(LED,LOW);	digitalWrite(LED,LOW);
16	digitalWrite(LED,low);	digitalWrite(LED,LOW);

18	curly bracket to close the void loop() function is missing
	Void loop(){
	//commands
	} →the closing bracket is missing
		}

Lesson 3: Control structures

Aim:

This lesson aims to introduce you to the different types of control structures, namely ‘if’, ‘if-else’, ‘for’ and ‘while’ statements. The aim of using control statements is to help you write a well-structured program.

Student Learning Outcomes: Learning outcomes refer to what the student can expect from the lesson, Teachers must share these outcomes with all students. Teachers are to tick the box when the outcome is achieved. Learning outcomes can be assessed using oral questioning and the written activities.

Students should be able to:

1. Implement sequence statements in Arduino sketches.
2. Implement ‘if’ selection statements in Arduino sketches.
3. Implement ‘if-else’ selection statements in Arduino sketches.
4. Implement ‘while-loop’ repetition statements in Arduino sketches.
5. Implement ‘for-loop’ repetition in Arduino sketches.

Keywords	**What are the keywords the students must learn?** 1. control structures 2. sequence structure 3. selection structure 4. loop structure
Resources	**What resources are required?** 5. textbooks 6. projector 7. Arduino board
Prior Knowledge	• Use breadboards for building electronic circuits. • Identify the basic electronic components. • Identify the resistance of colour-coded resistors. • Interpret schematic diagrams.

Possible Teaching Method(s) or Approach for this lesson

8. Collaborative Teaching (student centred)
9. Instructional / Demonstrative Teaching (teacher centred)
10. Inquiry-based Teaching (student centred)
11. Lecture Style Teaching (teacher centred)
12. Coach Style Teaching (teacher centred)
13. Facilitator Style Teaching (student centred)

Essential and non-essential Sections:

In some lessons it may not be possible to cover every section of the book due to time constraints or lesson variables. Below is a guideline to essential sections for examination and project knowledge.

Topic	**Page**	
	Essential	**Non-essential/Self Study**
Control structures	Pg. 164-165	
Sequence control structure	Pg. 166-169	
Selection control structure	Pg. 171-177	
Repetition control structure		Pg.178-190

Notes for differentiation:

All lessons can be different depending on ability and success of previous lesson. Place additional notes or activities to cater for differentiation where necessary throughout the lesson.

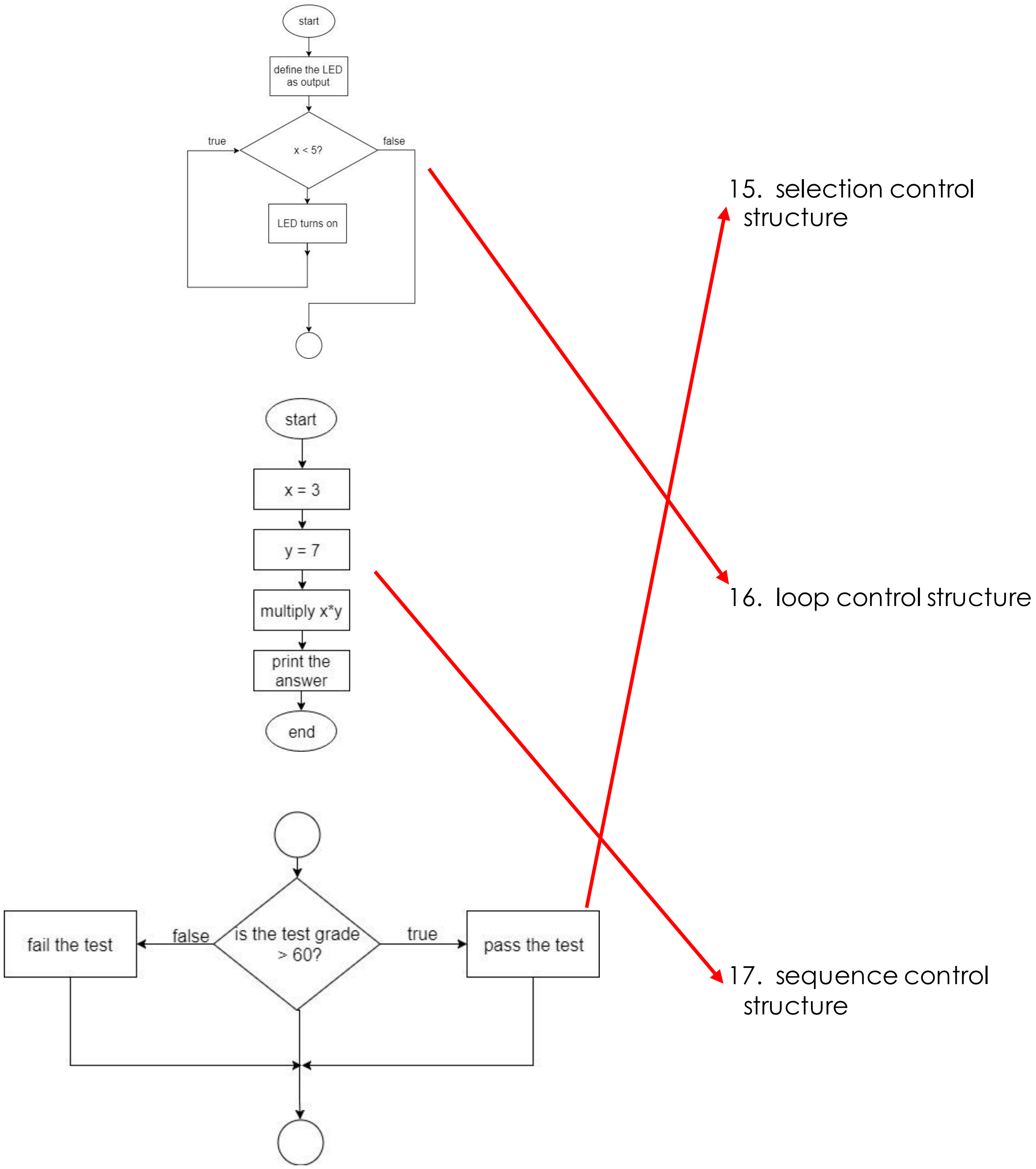
Development [Phases or chunks of learning]:	**Assessment Opportunities:**
Note: *All lessons start with Phase 1, Lessons can move back and forth between phases 2 and 3 as content is covered and then students engage. All lessons must finish with phase 4 to evaluate learning.*	
Phase 1 of lesson (Connect) **Starter** Teacher to introduce students to the lesson aim. Teacher to place all student learning outcomes on the board and ensure student understanding of aims and outcomes. Discuss prior knowledge of Arduino programming and basic electronic components. Show motivational videos / models to outline the end goal of the term. **Teacher Tip:** *Teacher to set high expectations which inspire, motivate and challenge pupils.*	**Questioning.**
Phase 2 of lesson (Activate) Teacher to introduce all key words, discuss meaning and ensure understanding before progressing. Teacher to introduce the topic of control structures, while students think-pair-share their thoughts. Question students on what aspects are new to them when compared to prior knowledge discussion. Teacher to introduce the types of control structures while students explore these structures through various activities. Teacher to monitor the students' progress throughout the lesson by using the different assessment opportunities. **Teacher Tip:** *Teacher to demonstrate good subject and curriculum knowledge.*	**Questioning / Written Activity 3.1**
Phase 3 of lesson (Engage and Demonstrate) Task 1: Students to study the first control structure, the sequence control structure. They will go through Arduino practical example to explore and test this structure as groups.	**Written Activity 3.2-3.3**

Students demonstrate learning by completing activity 3.2-3.3. Task 2: Students to study the second control structure, the selection control structure. They will go through Arduino practical example to explore and test this structure as groups. Students demonstrate learning by completing activity 3.4-3.5. Task 3: Divide students into two groups assigning each group a control structure. Use an "Each one teach one" approach to have groups show their understanding and improve other groups understanding of control structures. Teacher to facilitate as peer teaching takes place. **Teacher Tip:** *Use groupwork as appropriate, get to know your class and organise groups to support mixed ability's.* **Phase 4 Plenary (Consolidate)** Teacher to facilitate as students evaluate learning. Question pupils on what they have learned. Have learning outcomes been met? Has the lesson aim been achieved? All students must complete student reflection. Students to complete activities 3.1-3.5 and the final activity for homework if incomplete. *This lesson should be conducted in one lesson (2 periods – 90min)*	**Peer teaching** **Written Activity 3.4-3.5** **Oral Assessment** **Student evaluation**

Answer Key/ Resources

Activity 3.1

14. Match the program flowchart to its corresponding control structure.



15. selection control structure

16. loop control structure

17. sequence control structure

Activity 3.2

18. Give an example of a program that runs using sequence control structure.

Example1:

```
Start
Enter a number
Calculate the square root of the number
Display the result
End
```

Example2:

```
Start
Enter the length of the rectangle
Enter the width of the rectangle
Calculate the area
Display the result
End
```

Example3:

```
Start
Wake up
Eat breakfast
Go to school
Come back home
Eat lunch
Do homework
Eat dinner
Watch TV
Sleep
End
```

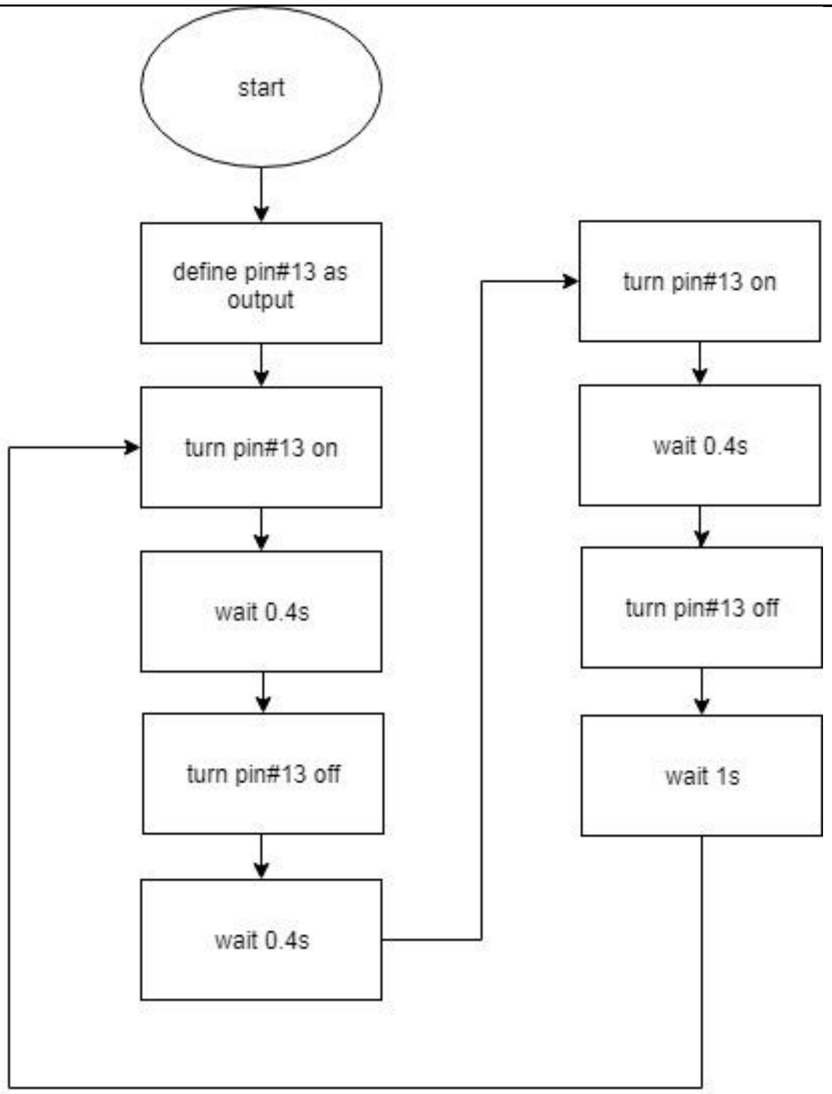
Answers may vary, any example of sequence of events that run one after the other is accepted.

Activity 3.3

Design an Arduino program that behaves like a heartbeat. Where an LED blinks twice, waits for one second, then blinks twice again.

Create a flowchart representing the design problem and write the program's pseudocode. Then, write the actual code for the program and implement it using Arduino.

Flowchart



Pseudocode

```
setup() {
set pin#13 as output
}
loop() {
//1st blink
turn LED ON
wait 400ms
turn LED OFF
wait 400ms
//2nd blink
turn LED ON
wait 400ms
turn LED OFF
//wait for 1 second
wait 1000ms
}
```

Arduino code

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(400);
  digitalWrite(13, LOW);
  delay(400);
  digitalWrite(13, HIGH);
  delay(400);
  digitalWrite(13, LOW);
  delay(1000);
}
```

Activity 3.4

Give an example of a program that runs using selection control structure. You can use the rational operators displayed in Table 2.14.

Example1:
Start
Enter a number
Is number is greater than 5?
If yes, blink the LED
Else
End

Example2:
Start
Enter age
Is age > 18?
If yes, you can vote
If no, display message “you can’t vote”
End

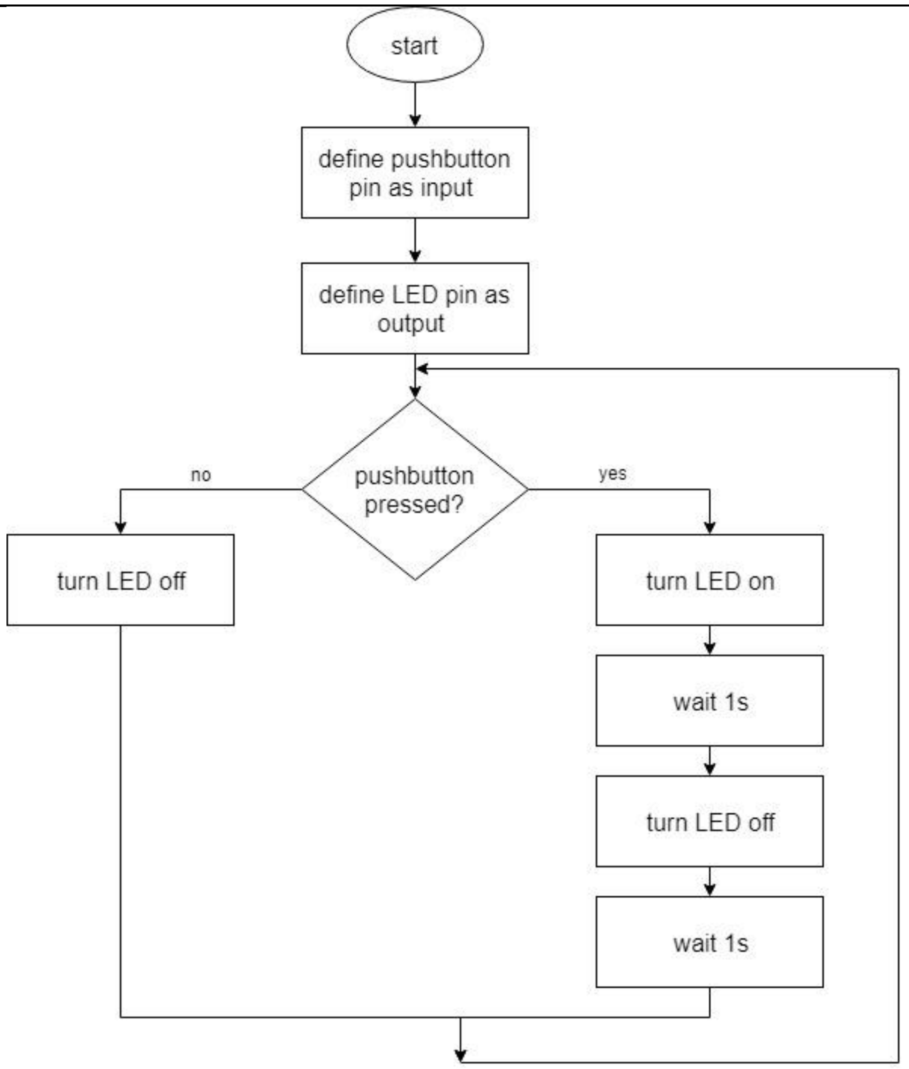
Answers may vary, any example of a selection process is accepted.

Activity 3.5

Modify the previous Arduino program so that when you press the button, the LED will start blinking (turns ON for one second, and then turns OFF for one second), and when you release it, the LED turns OFF.

Start by drawing the program’s flowchart, and then write its pseudocode. Finally, write the program’s actual code and implement it using Arduino.

Flowchart



Pseudocode

```
setup() {
set pin#12 as output //LED
set pin#8 as input //push-button
}
loop() {

    If (you press the pushbutton)
    {
    LED turns on;
    wait 1000ms;
    LED turns off;
    wait 1000ms;
    }
    Else
    {
    LED turns off;
    }
}
```

Arduino code

```
int buttonState = 0;
void setup() {
  pinMode(12, OUTPUT);
  pinMode(8, INPUT);
}
void loop() {
  // read the state of the pushbutton value
  buttonState = digitalRead(8);
  // check if pushbutton is pressed
  if (buttonState == HIGH) {
    digitalWrite(12, HIGH);
    delay(1000);
    digitalWrite(12, LOW);
    delay(1000);
  } else {
    // turn LED off
    digitalWrite(12, LOW);
  }
}
```

Activity 3.6

Give an example of a program that runs using the repetition control structure.

Example1:

```
Start
define variable 'x'
for (0<x<20)
x=x+5
display the value of x
End
```

Example2:

```
Start
define variable 'x', where x time in minutes
for (0<x<60)
do cardiovascular exercises
End
```

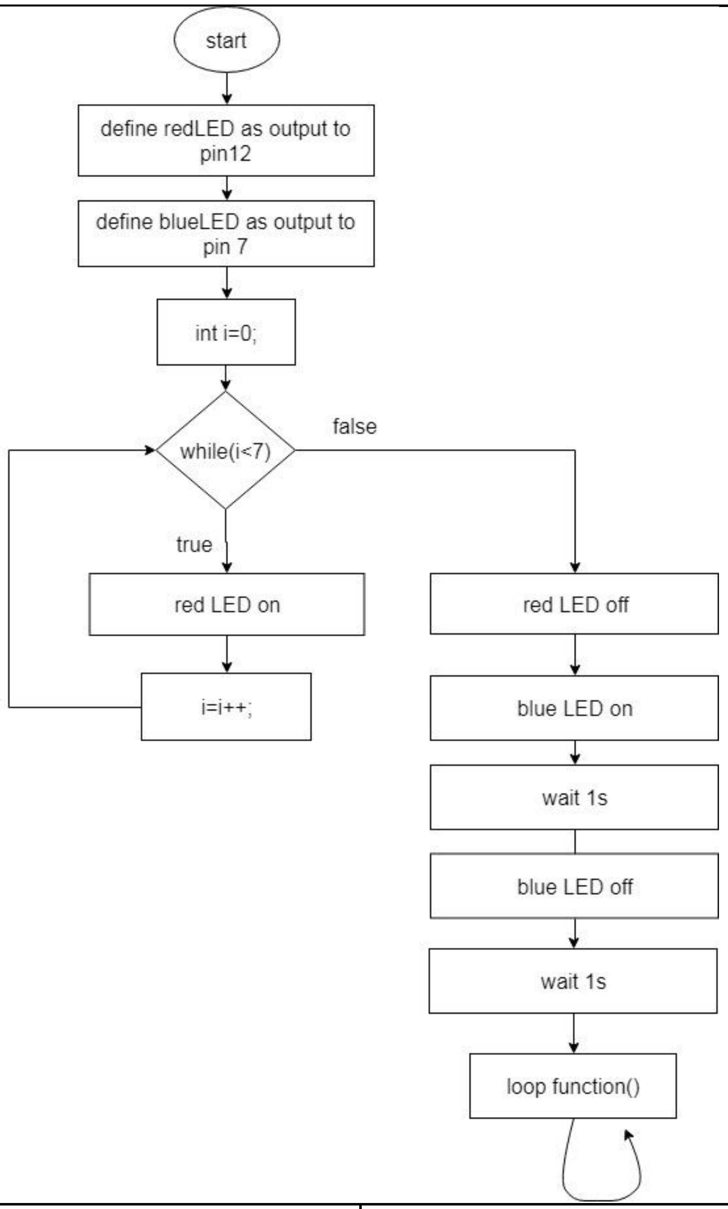
Answers may vary, any example of a repetition process is accepted.

Activity 3.7

Modify the previous Arduino program so that the control variable condition stops at 7. While the condition is true, the red LED will turn ON. Otherwise, the red LED will turn OFF, and the blue LED will start blinking (turning ON for one second and then OFF for one second).

Start by drawing the program's flowchart, and then write its pseudocode. Finally, write the program's actual code and implement it using Arduino.

Flowchart



Pseudocode

```
initialise the control variable
setup() {
set pin#12 as output //red LED
set pin#7 as output //blue LED
  while (control variable)
  {
  turn red LED on
  i=i+1
  }
  turn red LED off
  turn blue LED on
  wait 1s (1000ms)
  turn blue LED off
  wait 1s (1000ms)
}
loop() {
}
```

Arduino code

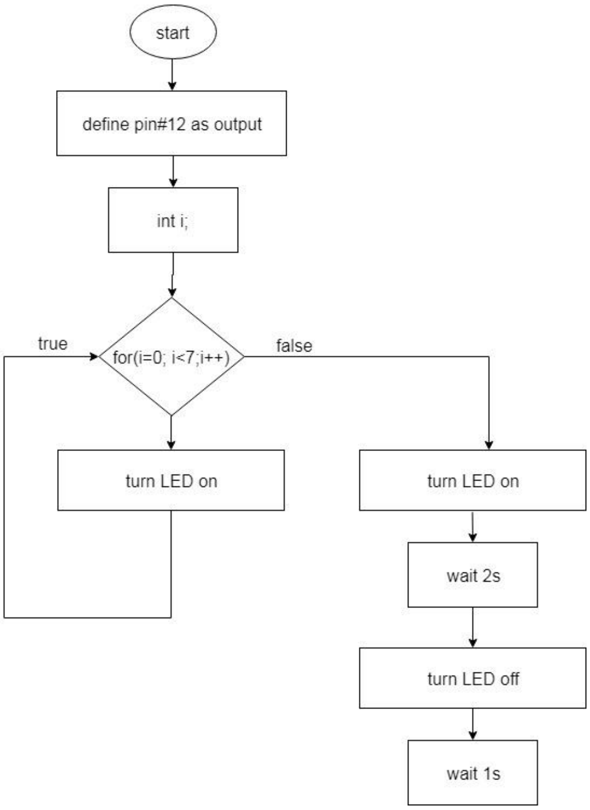
```
int i=0;
void setup()
{
  pinMode(12, OUTPUT);
  pinMode(7, OUTPUT);
}
void loop()
{
  while(i<7){
  digitalWrite(12, HIGH);
  i++;
  }
  digitalWrite(12, LOW);
  digitalWrite(7, HIGH);
  delay(1000);
   digitalWrite(7, LOW);
  delay(1000);
}
```

Activity 3.8

Modify the previous Arduino program so that the control variable condition stops at 7. If the condition is true, the LED will turn ON. Otherwise, the LED will start blinking (turning ON for two second, then OFF for one second).

Start by drawing the program's flowchart, and then write its pseudocode. Finally, write the program's actual code and implement it using Arduino.

Flowchart



Pseudocode

```
setup() {
      set pin#12 as output // LED
}

loop() {
   for (initialise the control variable i; i<7;
   increment i )
   {
      turn LED on
   }
   turn LED on
   wait 2s (2000ms)
   turn LED off
   wait 1s (1000ms)
}
```

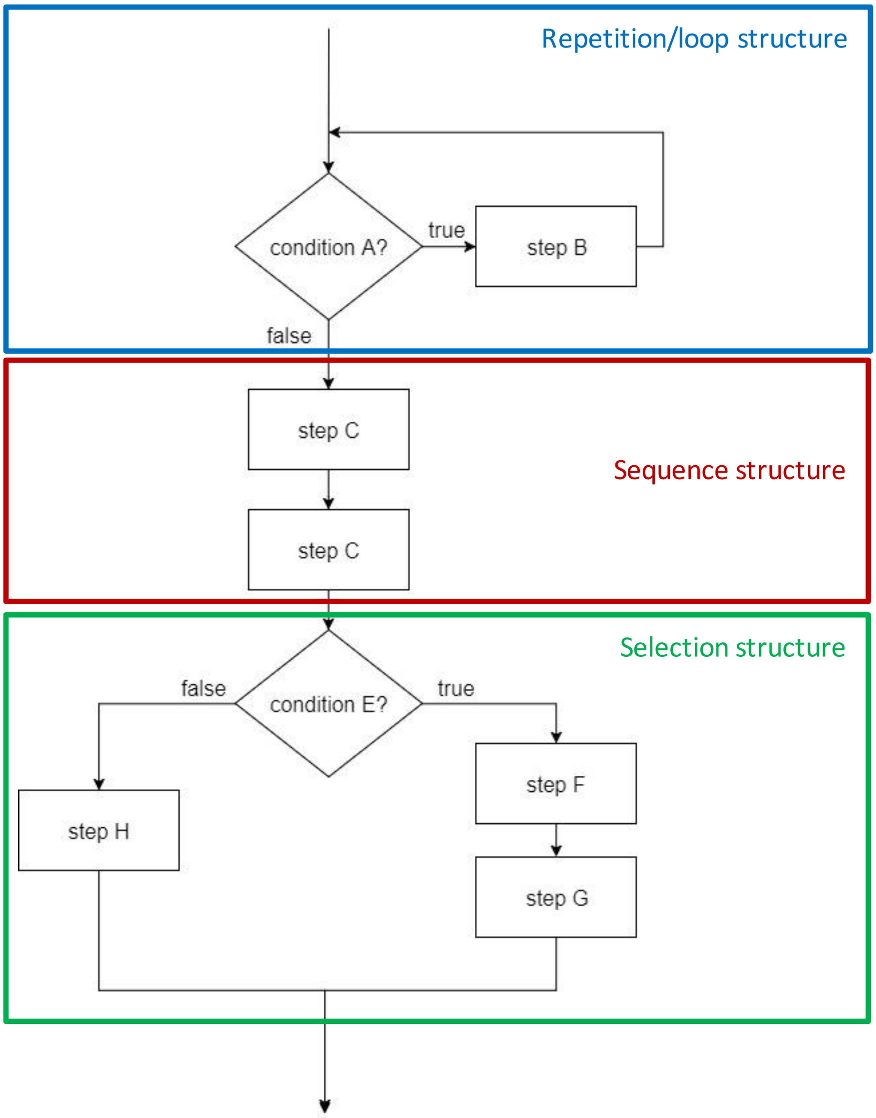
Arduino code

```
int i=0;
void setup()
{
  pinMode(12, OUTPUT);
}
void loop()
{
  for(i=0;i<7;i++){
  digitalWrite(12, HIGH);
  }
  digitalWrite(12, HIGH);
  delay(2000);
  digitalWrite(12,LOW);
  delay(1000);
}
```

Final activity

The following flowcharts combines all three structures, can you match the different parts of the flowchart to their corresponding structure?

The first one is done for you as an example.



Please note that the following figures were misprinted in the book:

Figure 3.20: While-loop example program Arduino code, pg.180

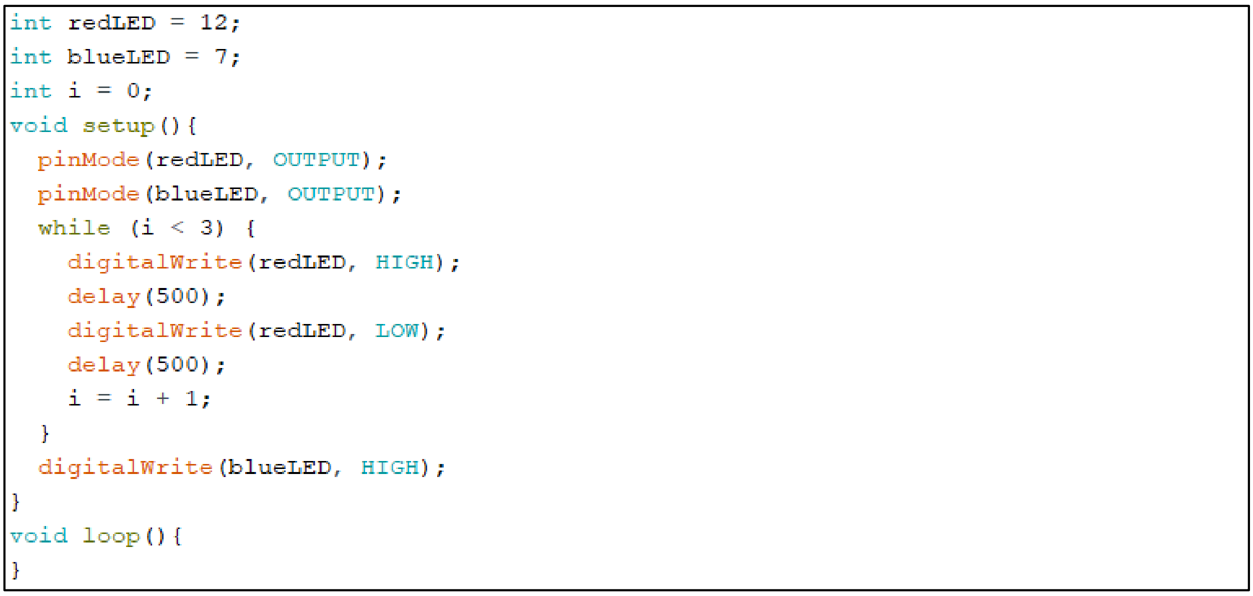
Figure 3.21: Matching flowchart/pseudocode to Arduino's code, pg.181

Figure 3.26: Arduino code for the for-loop example program, pg.186

Figure 3.27: Matching flowchart/pseudocode to Arduino's code, pg.187

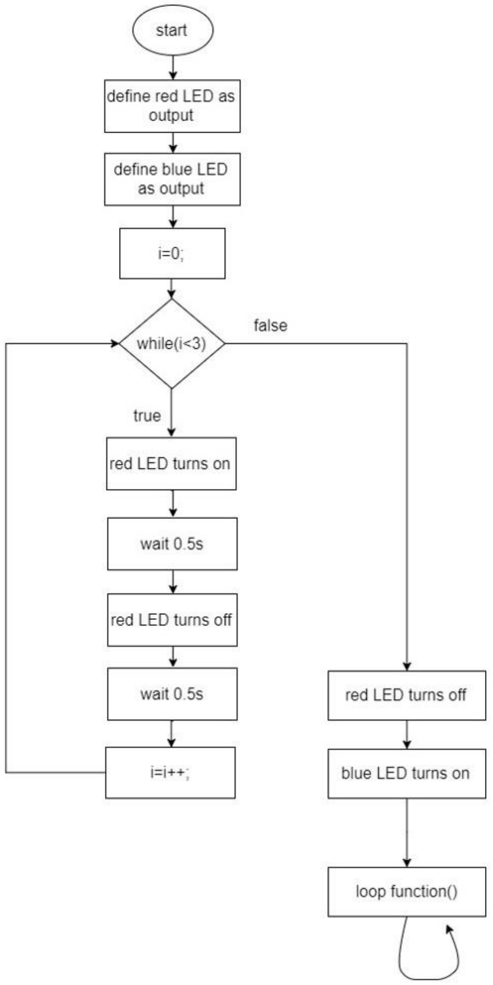
The correct figures are as follow:

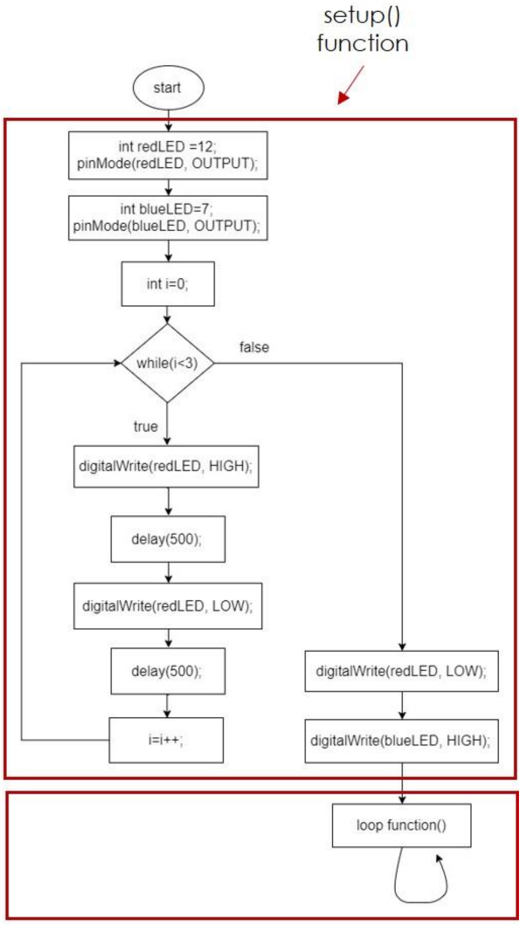
Figure 3.20: While-loop example program Arduino code, pg.180



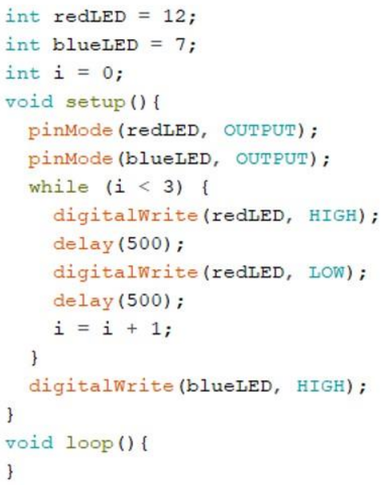
```
int redLED = 12;
int blueLED = 7;
int i = 0;
void setup(){
  pinMode(redLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  while (i < 3) {
    digitalWrite(redLED, HIGH);
    delay(500);
    digitalWrite(redLED, LOW);
    delay(500);
    i = i + 1;
  }
  digitalWrite(blueLED, HIGH);
}
void loop(){
}
```

Figure 3.21: Matching flowchart/pseudocode to Arduino's code, pg.181



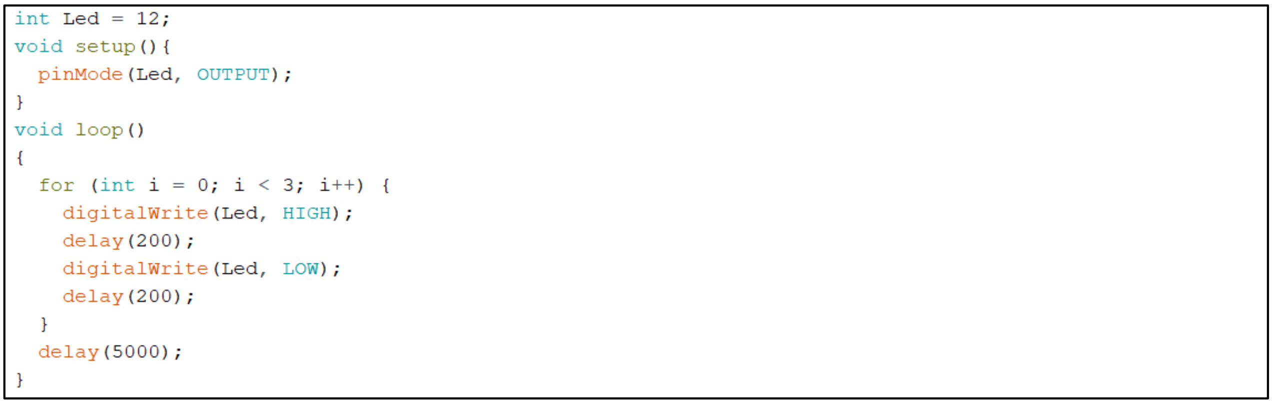


```
initialize the control variable
setup() {
set pin#12 as output //red LED
set pin#7 as output //blue LED
 while (control variable)
{
turn red LED on
wait 0.5s (500ms)
turn red LED off
wait 0.5s (500ms)
i=i+1
}
turn red LED off
turn blue LED on
}
loop() {
}
```



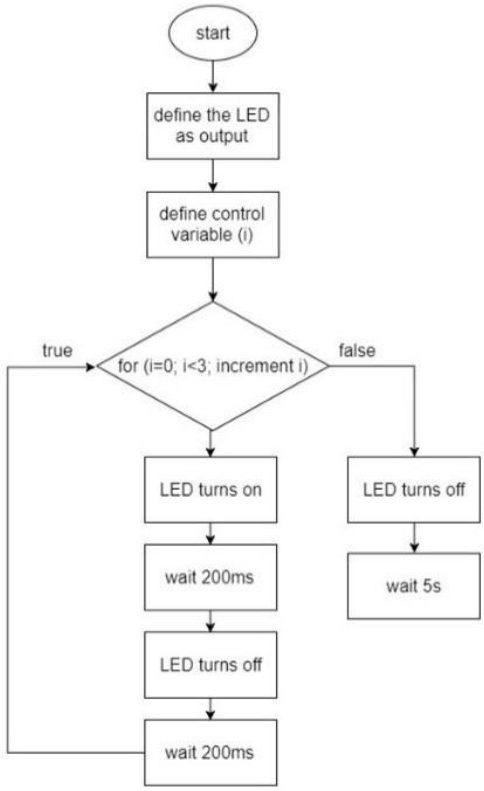
```
int redLED = 12;
int blueLED = 7;
int i = 0;
void setup(){
  pinMode(redLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  while (i < 3) {
    digitalWrite(redLED, HIGH);
    delay(500);
    digitalWrite(redLED, LOW);
    delay(500);
    i = i + 1;
  }
  digitalWrite(blueLED, HIGH);
}
void loop(){
}
```

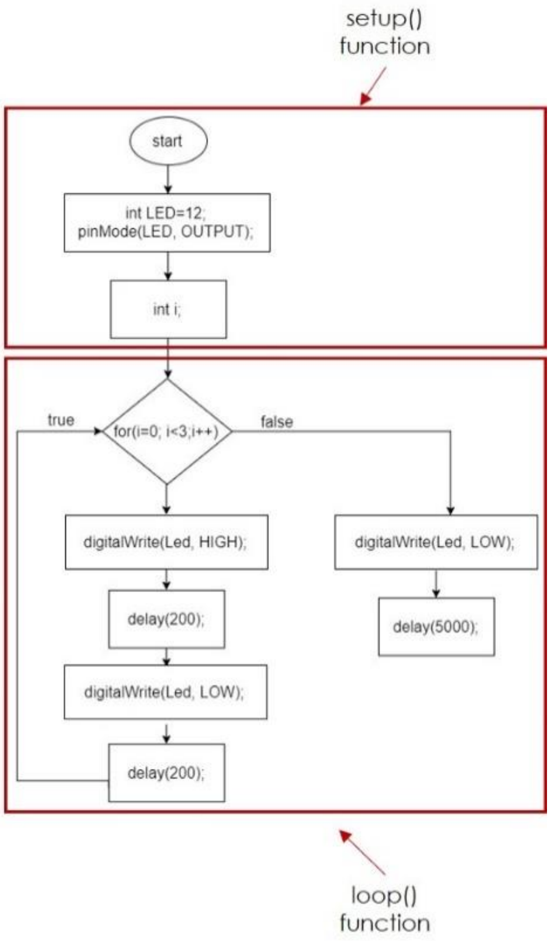
Figure 3.26: Arduino code for the for-loop example program, pg.186

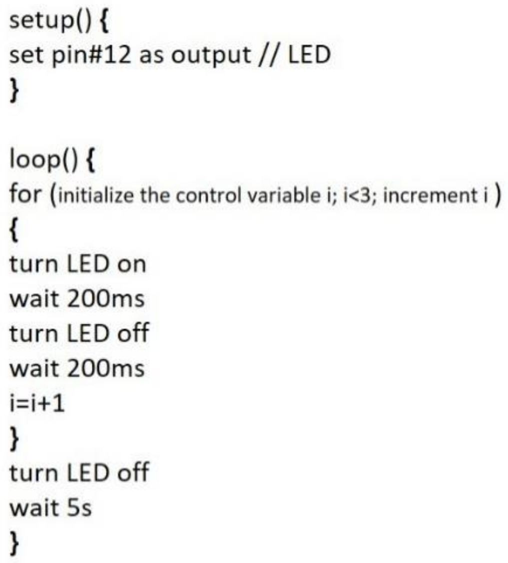


```
int Led = 12;
void setup(){
  pinMode(Led, OUTPUT);
}
void loop()
{
  for (int i = 0; i < 3; i++) {
    digitalWrite(Led, HIGH);
    delay(200);
    digitalWrite(Led, LOW);
    delay(200);
  }
  delay(5000);
}
```

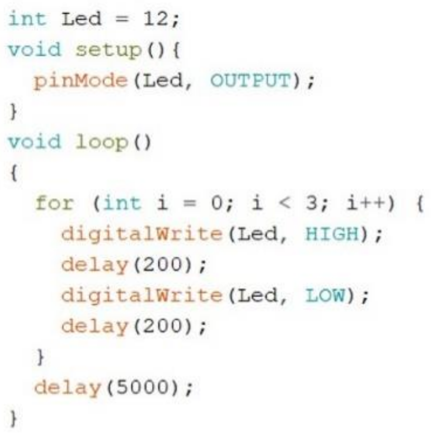
Figure 3.27: Matching flowchart/pseudocode to Arduino's code, pg.187







```
setup() {
set pin#12 as output // LED
}

loop() {
for (initialize the control variable i; i<3; increment i )
{
turn LED on
wait 200ms
turn LED off
wait 200ms
i=i+1
}
turn LED off
wait 5s
}
```



```
int Led = 12;
void setup(){
  pinMode(Led, OUTPUT);
}
void loop()
{
  for (int i = 0; i < 3; i++) {
    digitalWrite(Led, HIGH);
    delay(200);
    digitalWrite(Led, LOW);
    delay(200);
  }
  delay(5000);
}
```

Unit 1: Architecture

Lesson 4: Architecture project

Aim:

The aim of this project is to design, analyse and build a smart city in line with the UAE's futuristic vision for 2030 to exploit 3D technology and serve humanity.

Student Learning Outcomes:

Learning outcomes refer to what the student can expect from the lesson, teachers must share these outcomes with all students. Teachers are to tick the box when the outcome is achieved. Outcomes in **red** are essential.

Students should be able to:

- □ Demonstrate an understanding of the key stages in the design process.
- □ Analyse a design brief.
- □ **Write a design specification for an architectural project.**
- □ **Generate and develop design ideas that meet the design brief.**
- □ **Develop a 3D architectural model using Fusion 360.**
- □ Plan the manufacturing of an architectural physical model.
- □ **Produce the architectural model using 3D printing.**
- □ **Assemble all parts of the smart city.**
- □ **Write an evaluation for the completed city.**

Keywords	**What are the keywords the students must learn?** • architectural model • integral • residential • skyscraper • microclimate • ventilate • self-shading • proportions • infrastructure • façade
Resources	**What resources are required?** • textbooks • projector • drawing equipment (pencil, ruler, etc.) • model making equipment (staples, card, anything students bring themselves) • Fusion 360 • 3D printer
Prior Knowledge	• Create sketches in projected views • Research using the internet and other sources • Use Autodesk Fusion 360 in 3D modelling. • Render a 3D modelled design. • Print a 3D model using a 3D printer.

Possible teaching method(s) or approach for this lesson

- □ Collaborative Teaching (student centred)
- □ Instructional / Demonstrative Teaching (teacher centred)
- □ Inquiry-based Teaching (student centred)
- □ Lecture Style Teaching (teacher centred)
- □ Coach Style Teaching (teacher centred)
- □ Facilitator Style Teaching (student centred)

Essential and non-essential sections:

In some lessons it may not be possible to cover every section of the book due to time constraints or lesson variables. Below is a guideline to essential sections for examination and project knowledge.

Topic	Page	
	Essential	Non-essential/Self Study
Design brief.		Pgs. 77-79
Design specifications.	Pgs. 80-83	
Generation of ideas.	Pgs. 84-91	
Development of ideas.	Pgs. 92-93	
Project analysis.		Pg. 94
Planning to manufacture.		Pg. 95
Make it.	Pgs. 96-102	
Evaluation.	Pg. 103	

Note about differentiation: *All lessons can be different depending on ability and success of the previous lesson. Place additional notes or activities to cater for differentiation where necessary throughout the lesson.*

Development (phases or chunks of learning):	**Assessment opportunities:**
Phase 1 of lesson (Connect) **Starter** *Teacher to go through the learning outcomes of the lesson.* *Teacher to assess the prior knowledge of the students in relation to the ones described in the book.* **Teacher Tip:** Teacher to set high expectations which inspire, motivate and challenge students. Teacher to plan and teach a well-structured lesson. **Phase 2 of lesson (Activate)** *Teacher to go through the key words required within the lesson and make sure the students understand the importance of these key words.* *Teacher to go through the design process with the students.*	**Questioning**

Phase 3 of lesson (Engage and Demonstrate) *Teacher to ensure the students understand the requirements in the design brief.*	**Formal assessment**
Teacher to split the class in groups and make sure all groups have an equal level of ability to complete all tasks and create a successful architectural project. Teachers to make sure that each group creates a different type of building.	
Teacher to support the students/groups as they complete the activities in the book. Some activities may be set as homework.	**Formal assessment**
Teacher to support students/groups in regards to printing their designs using the 3D printer.	
Teacher Tip: Teacher to promote good progress and outcomes by students. Teacher to manage behaviour effectively to ensure a good and safe learning environment. Teacher to demonstrate good subject and curriculum knowledge.	
Plenary (Consolidate) *Teacher to ensure that students/groups have created an architectural model which meets the requirements set in the brief.*	**Summative assessment**
Teacher to ensure all models are assembled together to create a smart city.	**Summative assessment**
Teacher Tip: Teacher to make accurate and productive use of assessment.	

Answer Key/Resources

Stage 1: Analysis of brief:

This section should show evidence of understanding of the given brief. Areas which require solutions should be identified and problems which will be encountered should be recognised.

What is a Design Brief?

A design brief is a written description that outlines the design or engineering problem being posed to the student. It also highlights the requirements of the student's final solution.

Design brief

His Highness Sheikh Mohammed bin Rashid Al Maktoum, Vice President and Prime Minister of the UAE and Ruler of Dubai, said that 25% of Dubai's buildings will be 3D printed by 2030. In line with this vision, a leading architectural company in the UAE has asked you to design a 3D architectural model to be built in the UAE. Your model will form an integral part of a smart city and must be as creative and innovative as possible. The building type can be one of the following:

A residential building like a house, apartment block or villa

A commercial building like an office, restaurant, shopping mall or skyscraper.

A mosque.

An educational building like a school, college or university.

A medical building like a hospital or clinic.

A governmental building like the RTA, DEW A, SEW A. Abu Dhabi municipality or Dubai municipality.

An industrial building.

A transport building like an airport, metro station, bus stations or hyperloop station.

Remember to form a city you need to fulfil the users' needs; therefore, the city should consist of different building types. Each group should choose one building type from the ones above to be modelled, then combine it with the other groups to make the city

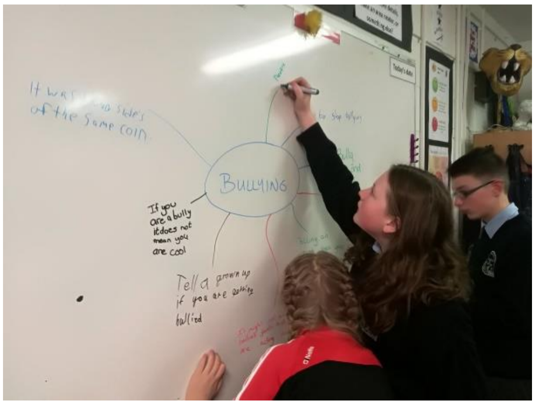
Where do I start?

Explore the brief carefully yourself before you introduce to your students. A good exercise to start is reading the brief with your students and getting them to highlight what they feel are the keywords in the brief. This will help to break down the design one step at a time. The students will fill this into their books, some examples of keywords are shown below:

How will I encourage my student's creativity?

A very useful group exercise at this stage is brainstorming. Brainstorming is a group creativity technique, designed to generate a large number of ideas for solving a problem. Students or the teacher can write the discussed ideas on the board. Throughout the session you should:

- Focus on quantity.
- No criticism is allowed.
- Unusual ideas are welcome.
- There are no wrong answers at this stage.
- Combine and improve ideas


It was two side's of the same coin
BULLYING
If you are a bully it does not mean you are cool
Tell a grown up if you are getting bullied

Teacher Tip: The group could be seated in a U direction to encourage discussion. The teacher will act in a facilitating role and can guide the discussion where necessary.

Activity 4.1

Highlight or circle keywords and phrases in the brief. This will help to break down the design one step at a time. **List five of these keywords below and describe their meaning**.

Keyword	*Meaning*
architectural model	a physical representation of a structure to communicate design ideas
integral	having all the parts that are necessary to be complete
residential	designed for people to live in
apartment	suite of rooms forming one residence; a flat
skyscraper	a very tall building of many storeys

Activity 4.2

In the space below, create your own unique mind map detailing all the requirements of the smart city (building): *You may use Figure 4.3 as a starting point for some ideas to discuss.*

Encourage students to explore every aspect of the brief. Use as many branches as necessary to demonstrate the brief on a mind map. Use colour to improve the presentation of your mind map.

Please refer to the example on page 79

Where can I encourage my students to look for inspiration?

You could do up a PowerPoint presentation addressing some of the points below or you could even ask a guest speaker to come into school to speak about design.

When designing a product, designers often look at various areas for inspiration. Some of these include:

- **nature -** The natural patterns and forms found in nature are often used as a starting point for fresh ideas.
- **architecture -** Common shapes or forms can provide inspiration when thinking of creative ideas.
- **design movements -** Design movements such as Art Nouveau, Modernism, Bauhaus, Art Deco, etc. can provide inspiration for new innovative ideas.
- **past and future solutions -** Looking at previous designs of can really help. Most modern-day inventions or designs are an improvement on, or inspired by, an existing product. Futuristic or concept designs can really get creativity flowing.
- **internet and social media –** Google images, Pinterest, YouTube and Instagram can be great assets in gaining inspiration for a design.

Teacher Tip: Put up various images of successful designers work or quotes on your walls to help develop the interest of your students in design.

Stage 2: Design specification:

Using the requirements identified in the brief, encourage the students to list design specifications for their smart city. The students should transform it into a detailed system which is creative, smart, feasible and sustainable. Remember, the smart city should enhance the users' quality of life.

What is required of my students in this section?

There are nine elements to this stage, each one of them will form the smart city. To list the design specifications, students need to answer the following questions linked to each element:

Target market

Who are the people you are designing for?

Materials

What materials are suitable for your building?

Size

What size should your building be?

Environmental issues

How can the building be sustainable under the climate in the UAE?
What are the recommendations for building in a hot climate?

Cost

What would the cost of your design be?

Aesthetics

What shapes, styles and colours are best for your building design?

Safety

What safety procedures will you need to consider whilst building your smart city?

Function

What is the function of your building?

Manufacturing

How would your building be made?
What are the maximum dimensions that can be printed by the 3D printer you have?

An example of the design specifications can be found on page 81.

QR code links:		
Page	***Topic***	***Link***
81	Louvre Abu Dhabi	https://www.louvreabudhabi.ae/

Activity 4.4

What would be the building design limitations, and how would these affect your design?

Availability of materials will affect the final design appearance

Size limitations will affect the dimensions of your 3D printed model

How would you design your building (future city) to be thermally comfortable? You may use sketches to explain your answer.

Answers may vary, some examples are listed below:

Windows and lighting: Use of properly designed and installed high efficiency windows and lighting.

Passive houses: using the sun's energy for heating and cooling living spaces, the building itself, or some element of it, taking advantage of the natural energy potential of materials and air that have been exposed to the sun.

Using more greenery.

Building orientation: the buildings to be oriented in a way that allows wind ventilation, avoiding direct sun light.

Using shading devices.

Stage 3: Generation of ideas

What is required of my students for generation of ideas?

Using the information gathered, students should sketch at least two possible solutions. They can use a number of sketching techniques.

Generation of ideas must:

- explain the operation of the design solution.
- state advantages and disadvantages of each solution.
- Show how it meets the brief

What choice do your students have in this section?

1. Students may wish to generate ideas by redesigning the existing designs they researched.

OR

2. Communicate their own new design ideas graphically using sketches.

This means that students have the choice to create possible proposals based on an existing design or they can come up with their own completely new design.

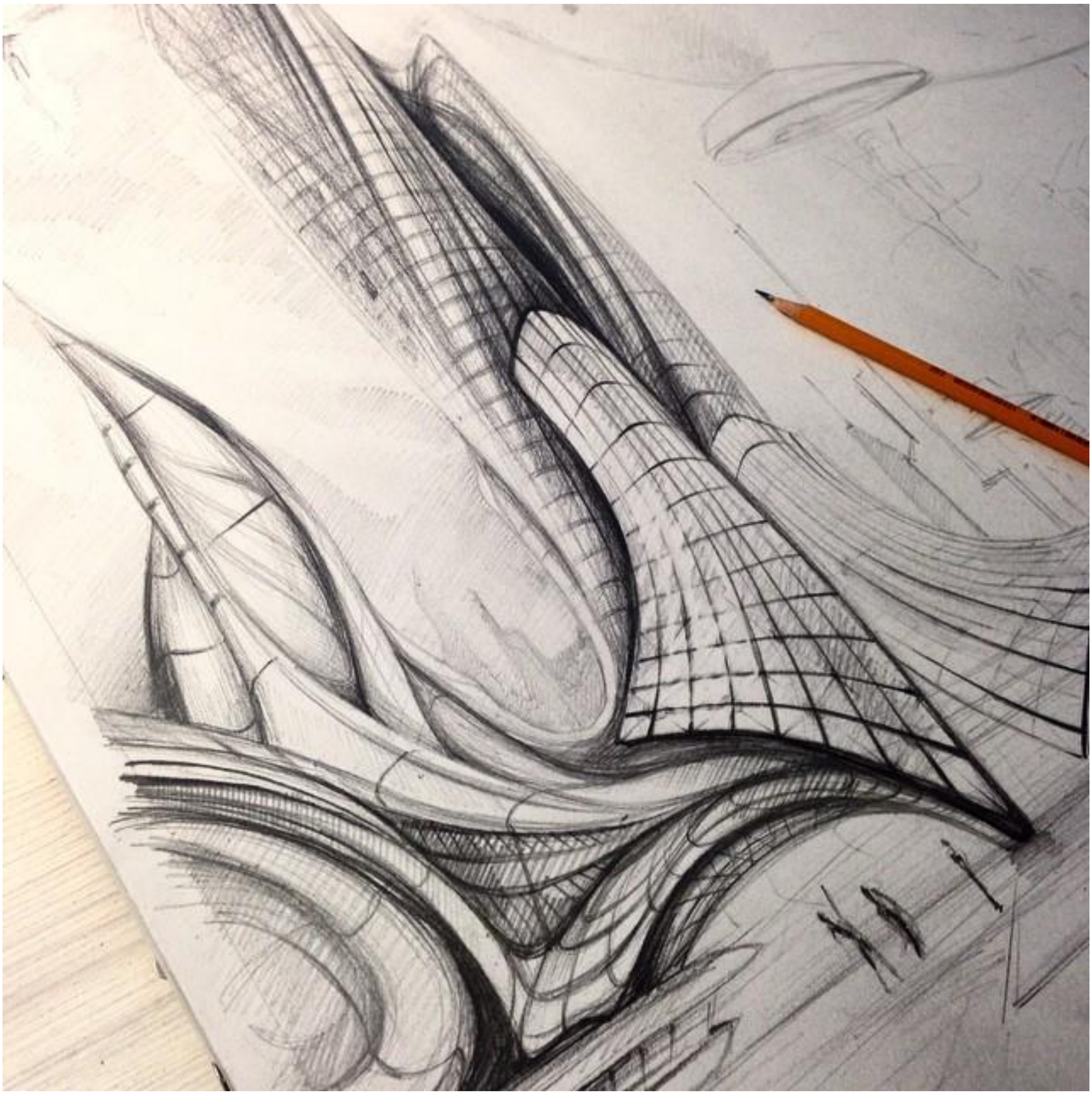
Do my students have to use freehand sketching in this section?

Yes, the student's design ideas must be represented using freehand sketches. Students should practice sketching their possible designs before they transfer it into their books.

Teacher Tip: It would be a good idea to photocopy this section out of book for students to practice on and show you their attempts before they transfer it into their books.

Sketching- practice makes perfect

When it comes to sketching, the more practice our students get the more they improve as designers. Students should be encouraged to practice their sketching techniques if they get any spare time in school or at home. YouTube is full of videos on sketching techniques if students wish to practice at home.

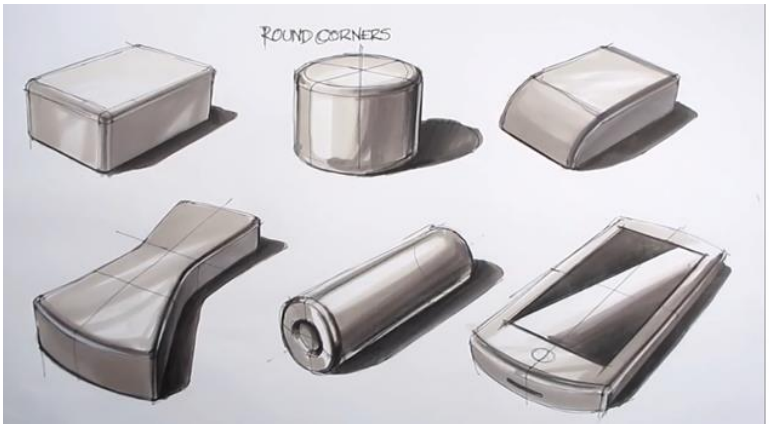


Teacher Tip: You could encourage students that like sketching or who are particularly passionate about design to get themselves a sketch pad to store their sketches.

Sketching exercise

You could get students practising how to sketch basic shapes using the video exercise below.

https://www.youtube.com/watch?v=6ZU-ryDOtLw&t=22s



How should my students present their work?

The layout of the sketches is up to the student, they can have a single sketch to present their ideas or a number of sketches. Students should be encouraged to consider the following:

- All sketches should be very neat and tidy. The use of colour and shading is encouraged.
- Students can use a variety of sketching techniques.
- 2D and 3D format are both acceptable, but a combination of both will be a better representation.
- Neat annotations or notes would help in explaining the operation of student designs.
- Students should give at least two advantages and two disadvantages to each design.
- Students may include extra pages to accompany their book for this section.

What guidance can I give my students to generate ideas?

- pay attention to colour, shape, texture
- what materials are used in my design and what is their pattern

Note: Does the student's design ideas meet the criteria of brief? If not give them feedback on areas to change or to come up with another idea

Students should complete at least 2 possible designs. State the advantages and disadvantages of each. Encourage rendering or color to enhance designs.

Activity 4.5

How do your design ideas meet the brief?

Answers may vary an example is listed below

The design should form an integral part of a smart city and must be as creative and innovative as possible. The building type should be one of the following:

A residential building like a house, apartment block or villa

A commercial building like an office, restaurant, shopping mall or skyscraper.

A mosque.

An educational building like a school, college or university.

A medical building like a hospital or clinic.

A governmental building like the RTA, DEWA, SEWA. Abu Dhabi municipality or Dubai municipality.

An industrial building.

A transport building like an airport, metro station, bus stations or Hyperloop station.

Stage 4: Development of ideas

What is required of my students for this selection?

In this stage, one final building design that fits the brief must be chosen. This final design may be based on one of the design ideas or a mixture of both. It is important to show the reasons for choosing one design over another.
Students will present one final design for their building design using sketches, or present this using 3D software like Fusion 360. An example is shown on page 92.

What guidance can I give my students in this selection?

First and foremost, the final design must meet the criteria of the brief.

How should my students present their work?

The layout of the sketches is up to the student, they can have a single sketch to present their ideas or a number of sketches. Students should be encouraged to consider the following:

- All sketches should be very neat and tidy. The use of colour and shading is encouraged.
- Students can use a variety of sketching techniques.
- 2D and 3D format are both acceptable, but a combination of orthographic 2D and 3D sketches will be a better representation.
- Neat annotations or notes would help in explaining the operation of student design
- State reasons for choosing this design.
- Select suitable materials for manufacturing and give reasons.

- Students can get creative with how they present their final design.
- Students can present their designs using 3D software like Fusion 360

Note: Each student design should be unique and innovative. Encourage students to think outside the box and try to add features that will make their designs unique and stand out against their classmates.

Project analysis:

In this section students should analyse their projects to help them plan the location of their building block in the smart city

Accessibility and circulation

Main road:

The students should plan the building circulation and where it is going to be located in the smart city. (Note that the building's main entrance is usually located in front of the main road)

Students can name their smart city roads to be able to identify the surrounding roads in this activity.

Sub-main road:

Students should plan the smart city infrastructure with the teacher's assistance, each group should choose the buildings location in the smart city.

Number of surrounding roads:

After planning the building's location in the city each group should be able to identify how many surrounding roads they have around their building.

Building accessibility: easy/complicated

The building accessibility must be easy in general, however, students must know if their building can be accessed easily or not.

Climate

Average temperature in daylight:

Students should search for the average temperature in daylight in the United Arab Emirates.

The average temperature in the summer is **41°C** during the day, while it is **23°C** during the winter.

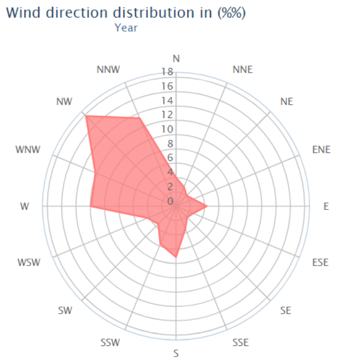
Average temperature at night:

Students should search for the average temperature at night in the United Arab Emirates.

The average temperature in the summer is **30°C** overnight, while it is **14°C** during the winter.

Wind direction:

North Western in the UAE



Average humidity in the area:

Summer: 50-60% in the UAE

Winter: 56% in the UAE

Vegetation:

Students should plan their vegetation (trees locations) in the city with the teacher's assistance.

Teachers should encourage students to use vegetation in their cities as a passive design solution that filters the air around the buildings or maybe in the building itself by using green roofs or green walls.

Site in city

City:

The students should choose one city in the UAE.

Location:

The students must choose a location in the city they have chosen earlier.

Dimensions:

Students should decide their building plot dimensions taking into consideration the total scale of the city and the printing size limitations.

Total area:

Students should calculate the total area of their plot.

Building type:

Each group should choose one building type from the ones mentioned earlier in the brief.

Surrounding buildings:

Each group in the class will design a building in the city, therefore, each group should identify the buildings types next to it, which was made by the other groups.

Ideal place for building:

Encourage student's critical thinking to figure out the most suitable place in the city for their building.

Best view is to the: North/South/East/West

The students should choose the best building view from the following.

Most shaded spot:

The students must choose the most shaded spot around the site.

Advantages:

Students must list the advantages of the building and the design, this could be the social and economic impact of the building. This could also be described as the regeneration of an area or even the improvement of living spaces.

Disadvantages:

Students must list the disadvantages of the building and the design, this could be the impact of the population in the area due to the increase of services via the buildings or it could result in environmental issues of litter and pollution.

How to improve the design:

Encourage the students evaluate the positive and negative aspects of their designs, against the design brief and list the modifications they would make.

Planning to manufacture (Gantt chart):

A gantt chart is used regularly in the design industry. It allows the project manager to plan a schedule for the work that needs to be carried out and in what time these tasks need to be completed.

Important characteristics of a gantt chart:

- the bar in each row identifies a certain task
- the length of the bar identifies how long each task should take
- the beginning of the bar shows the start time of the task
- the end of the bar shows the last time of the task

Activity 4.6

Below is an incomplete gantt chart, using the example above, plan your weekly activities based on the tasks given, start by writing the weeks at the top.

Students should plan their weekly activities based on the tasks given by writing the weeks at the top and deciding the time required to finish each activity, refer to the example on page 95. Each student will allocate the appropriate timings for each of the stages. Emphasise that this activity should allow them to complete the project on time.

Stage 5: Make it

Now that the students have designed their architectural model to be added to the city, the students will model a prototype of their design using the 3D printers or any available materials. Once students have completed the 3D model, they must assemble all the parts of their smart city along with their classmates.

An example of a model made from staples is shown on pages 96 and 97. Students have to use staples to make a similar model. Before they start with their own final design, students should decide which aspects of their model will be 3D printed and which aspects will be made from staples or other model-making materials like foam board or Styrofoam.

Teachers have to prepare modelling materials in advance. Teachers have to highlight Sheikh Mohammed bin Rashid's 3D Printing Strategy which aims to have 25% of the buildings in Dubai 3D printed by 2030. Students have to 3D print **25% of their model for the smart city.**

Refer to the examples given on pages 98, 99, 100 and 101.

Students have to paste a picture of the completed smart city on page 102.

Stage 6: Evaluation

Now that the students have completed their project, an evaluation is needed. The evaluation will allow the students to reflect on the completed project. They must state what went well but also what improvements could be made to their design.
Students should use the following questions to guide their evaluation:

How well does the model meet the brief?
How well does the model function?
What material(s) and appearance(s) did you use? Are they suitable?
Is your model aesthetically pleasing? How could it be improved?
State two things that went well.
State two things that could be improved.

All activities in this lesson will have different types of answers depending on the building the students/groups decide they want to design and make. Teachers are expected to support and advise students/groups while completing the architectural project. All stages of the design process must be completed.

Lesson 4: Project – Smart city features

Aim:

The aim of this project is to identify the challenges facing cities around the world and to find innovative solutions to overcome them. Eight smart city solutions are proposed, but, only one solution per group must be implemented.

Student Learning Outcomes: Learning outcomes refer to what the student can expect from the lesson, Teachers must share these outcomes with all students. Teachers are to tick the box when the outcome is achieved. Learning outcomes can be assessed using oral questioning and the written activities.

Students should be able to:

35. Understand the challenges that cities face today due to rapid urbanisation.
36. Identify the engineering design process steps.
37. Apply the engineering design process to solve the cities' problems using technology.
38. Apply critical and creative thinking in order to find effective solutions.
39. Apply different solutions to the city's model designed in Unit 1.
40. Develop team management and teamwork skills.
41. Develop presentation skills to demonstrate the city's features.

Keywords	**What are the keywords the students must learn?** 42. smart city 43. engineering design process 44. urbanisation 45. sustainable energy 46. critical thinking
Resources	**What resources are required?** 47. textbooks 48. projector 49. Arduino board
Prior Knowledge	• Use breadboards for building electronic circuits. • Identify the basic electronic components. • Identify the fundamentals of the Arduino programming. • Interpret schematic diagrams.

Possible Teaching Method(s) or Approach for this lesson

50. Collaborative Teaching (student centred)
51. Instructional / Demonstrative Teaching (teacher centred)
52. Inquiry-based Teaching (student centred)
53. Lecture Style Teaching (teacher centred)
54. Coach Style Teaching (teacher centred)
55. Facilitator Style Teaching (student centred)

Essential and non-essential Sections:

In some lessons it may not be possible to cover every section of the book due to time constraints or lesson variables. Below is a guideline to essential sections for examination and project knowledge.

Topic	**Page**	
	Essential	**Non-essential/Self Study**
What is a smart city?	Pg. 196-200	
Engineering design process		Pg.200-203
8 smart city features (project)	One feature is essential Pg. 204-326	7 features are non-essential Pg. 204-326

Notes for differentiation:

All lessons can be different depending on ability and success of previous lesson. Place additional notes or activities to cater for differentiation where necessary throughout the lesson.

Development [Phases or chunks of learning]:	**Assessment Opportunities:**
Note: *All lessons start with Phase 1, Lessons can move back and forth between phases 2 and 3 as content is covered and then students engage. All lessons must finish with phase 4 to evaluate learning.*	
Phase 1 of lesson (Connect) **Starter** Teacher to introduce students to the lesson aim. Teacher to place all student learning outcomes on the board and ensure student understanding of aims and outcomes. Discuss prior knowledge of Arduino programming and basic electronic components. Show motivational videos / models to outline the end goal of the term. **Teacher Tip:** *Teacher to set high expectations which inspire, motivate and challenge pupils.*	**Questioning.**
Phase 2 of lesson (Activate) Teacher to introduce all key words, discuss meaning and ensure understanding before progressing. Teacher to go through the design process with the students.	
Phase 3 of lesson (Engage and Demonstrate)	
Task 1: Students to learn what a smart city is. Students demonstrate learning by completing activity 4.1-4.2. Teacher to introduce the eight smart features.	**Questioning / Written Activity 4.1-4.2**
Task 2: Teacher to split the class in groups and make sure all groups have an equal level of ability to complete all tasks and create a successful project. Teachers to make sure that each group choose a different smart city feature to implement, and to make sure all circuits are built properly. Teacher to support the students/groups as they complete the activities in the book. Some activities may be set as homework. Teacher to support students/groups in regards to building their electronic circuits and writing the Arduino codes. **Teacher Tip:** Teacher to promote good progress and outcomes by students. Teacher to manage behaviour effectively to ensure a good and safe learning environment.	**Formal assessment**

Teacher to demonstrate good subject and curriculum knowledge. **Phase 4 Plenary (Consolidate)** Teacher to ensure that students/groups have implemented at least one of the smart city features and that their circuit functions correctly. Teacher to facilitate as students evaluate learning. Question pupils on what they have learned. Have learning outcomes been met? Has the lesson aim been achieved? All students must complete student reflection. *This lesson should be conducted in one lesson (2 periods – 90min)*	**Summative assessment** **Oral Assessment** **Student evaluation**

Answer Key/ Resources

QR code links:		
Page	***Topic***	***Link***
197	Smart city	https://www.youtube.com/watch?v=Br5aJa6MkBc
307	Kinetic façade	https://www.youtube.com/watch?v=w0D0qfBgbgU
310	DC motor	https://www.renesas.com/en-eu/media/support/technical-resources/engineer-school/brushless-dc-motor-01-overview/img01_dc_motor.gif

Activity 4.1

What are the advantages of creating a smart city?

1- Makes the city more responsive to its citizens, thus, making people happier and more connected to their community.
2- Smart cities would be cost-effective and safer, providing a better quality of life for the citizens
3- Environmental advantages, since smart cities can reduce energy consumption and improve waste management.

Answers may vary.

Activity 4.2

Download the 'Dubai Now' app on your smartphone and name eight more services that the application provides

1- Utilities and bills – DEWA, NOL, salik…etc
2- Public transportation – NOL, metro, flight information…etc
3- Security and justice – emergency call, police stations…etc
4- Health – pharmacy, doctor and clinic…etc
5- Driving – MyCar, mParcking, traffic fines…etc
6- Residency visa – check visa status…etc
7- Islam – prayer times and mosques
8- Education – schools, univercities…etc

Answers may vary.

Activity 4.3

‘Build a mechanical wall clock using cardboard and without using a compass’

Objective:

To build a mechanical wall clock
The clock is made using cardboard material

Constrain(s):

Compass can’t be used as part of the clock design

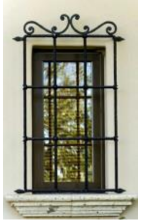
Activity 4.4

Why do you think research is important?

- Helps you learn new information
- Improves your writing skills
- Improves your communication skills
- Teaches you how to think, to question
- Helps you solve problems

Activity 4.5

The design in the picture on the right was suggested to reduce the risk of children falling out of windows. Evaluate the design in term of advantages and disadvantages.



Advantages:

The window security bars will protect children from falling out of the window.

Disadvantages:

In case of fire, children might be trapped in sides the house. A better solution might be a window opening control devices or locks.

Activity 4.6

Compare and contrast the following images using ACCESS FMM.

ACCESS FMM	**old TV**	**new TV**
aesthetics	Brown, grey, black Box shaped	Black Flat screen
cost	Cheap, 300 dirhams	Expensive, 1300 – 5000 dirhams
customer	No longer used	everyone
environment	-	-
size	Bulky, heavy Small screen size	Light weight Large screen size
safety	Safe to use	Safe to use
function	Cathode-ray tube (CRT) TV sets project an image by taking an incoming signal and splitting its audio and video components.	Liquid crystal display (LCD) televisions, on the other hand, function by shining light through a thin layer of liquid crystals.
material	Wood, glass	Fiber glass, plastic
manufacture	-	-

Project 1: accident detection and alert system

Activity 4.7

Following the design process steps, ‘define’ the project requirements.

Objective:
Build an accident detection and alert system using GPS and GSM modules.

Constrain(s):
You should use the components in the resources list only.

Activity 4.8

Copy the $GPGGA string displayed on your serial monitor in the box below:

```
$GPGGA,104534.000,7791.0381,N,06727.4434,E,1,08,0.9,510.4,M,43.9,M,,*47
```

Then, extract the longitude and latitude coordinates of your location.
Longitude: **7791.0381**
Latitude : **06727.4434**

Activity 4.9

Modify the GSM code to send a different message to two different mobile numbers.

Edit the void SendMessage function by adding the following lines of code

```
mySerial.println(‘AT+CMGS=\’+971xxx\’\r’); // Replace x with mobile number
delay(1000);
mySerial.println("msg to the 2nd number");//the content of the message
delay(1000);
mySerial.println((char)26);//the message stopping character
delay(1000);
```

Testing – accident detection and alert system

Indoor testing

Use the Google Maps link you received in the alert message. What are your findings? Explain

You’ll receive a text message with the wrong coordinates, for example
Latitude: 0.05
Longitude: 0.00

This is because the GPS module needs a clear line of sight to the sky to work properly. But, inside the building there will be no direct line from the satellite signals to your device.

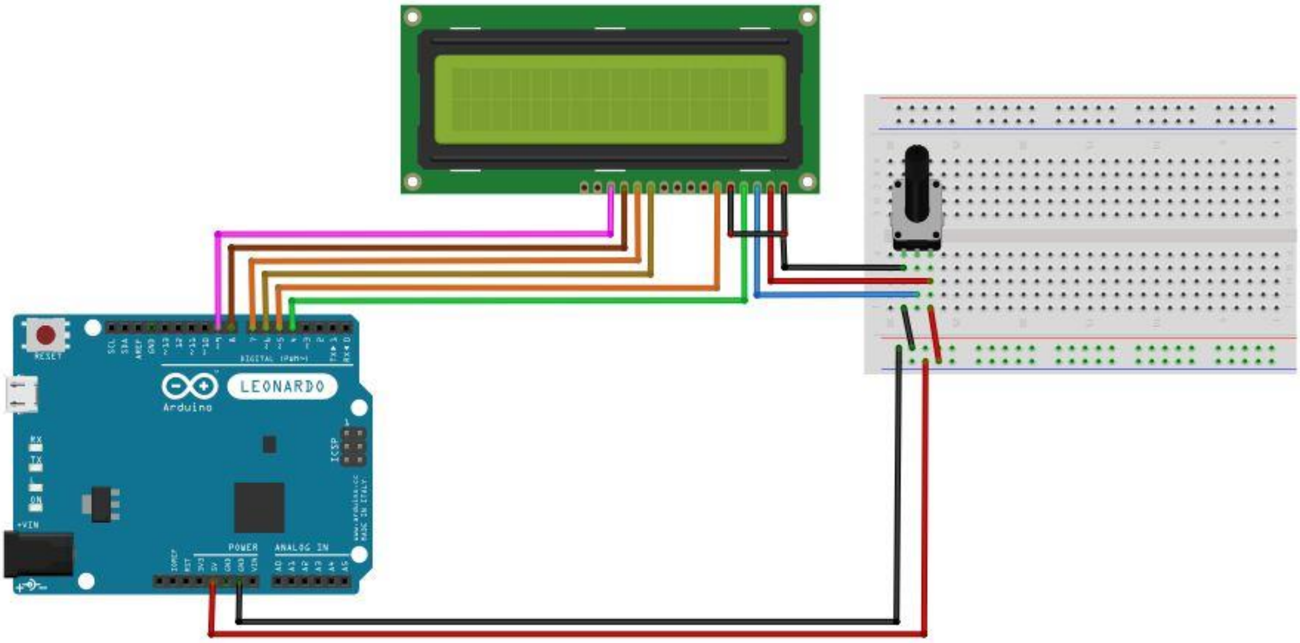
Outdoor testing

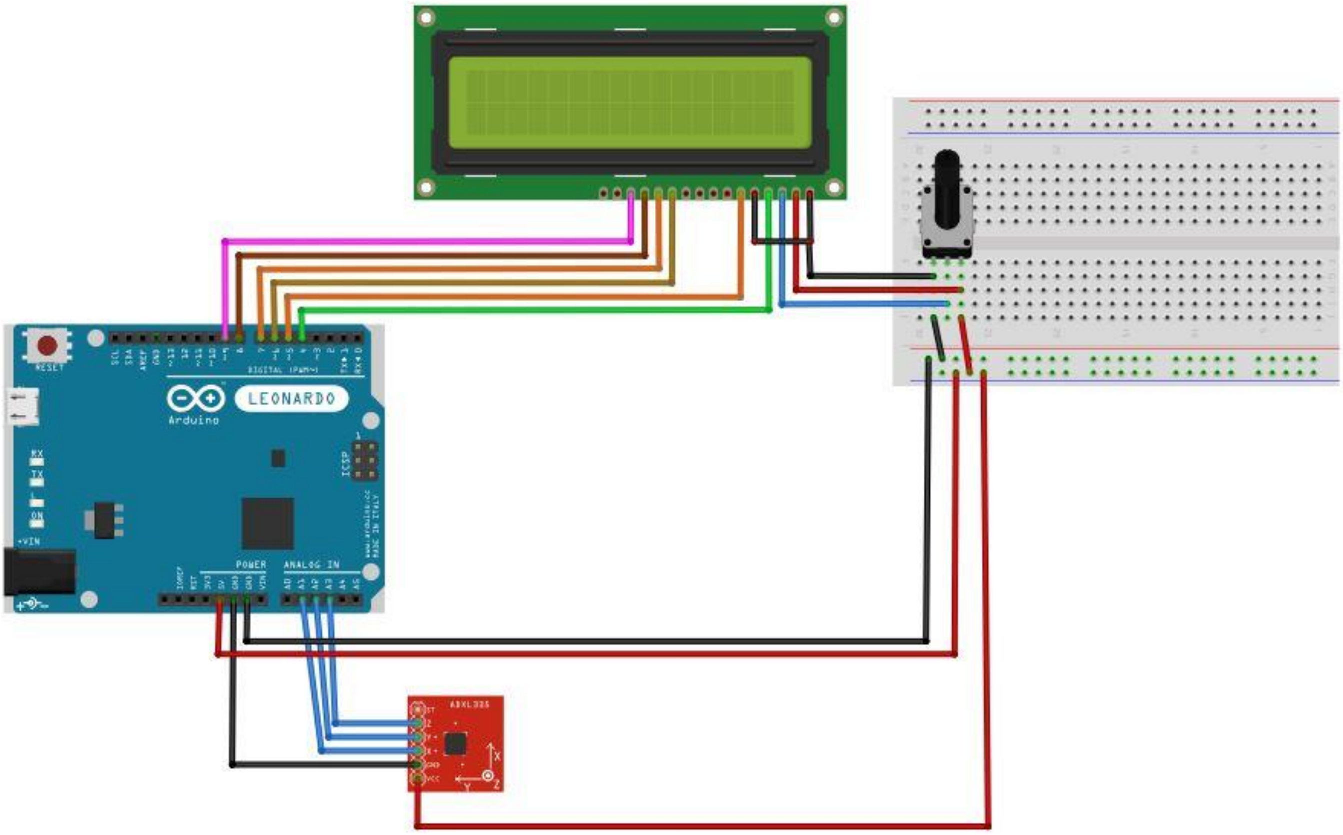
Use the Google Maps link you received in the alert message. What are your findings? Explain

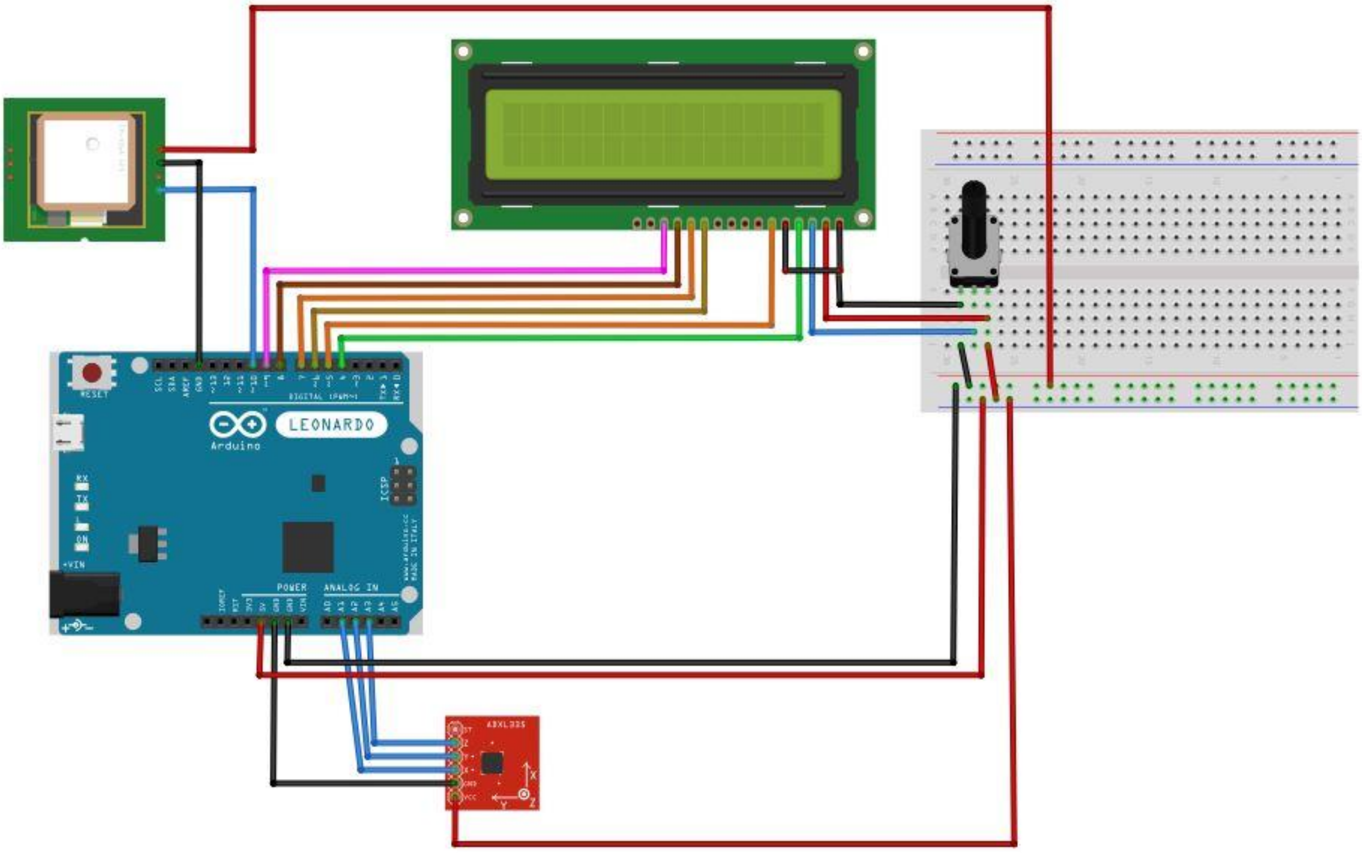
You'll receive a text message with your location coordinates, for example
Latitude: 25.47
Longitude: 55.54

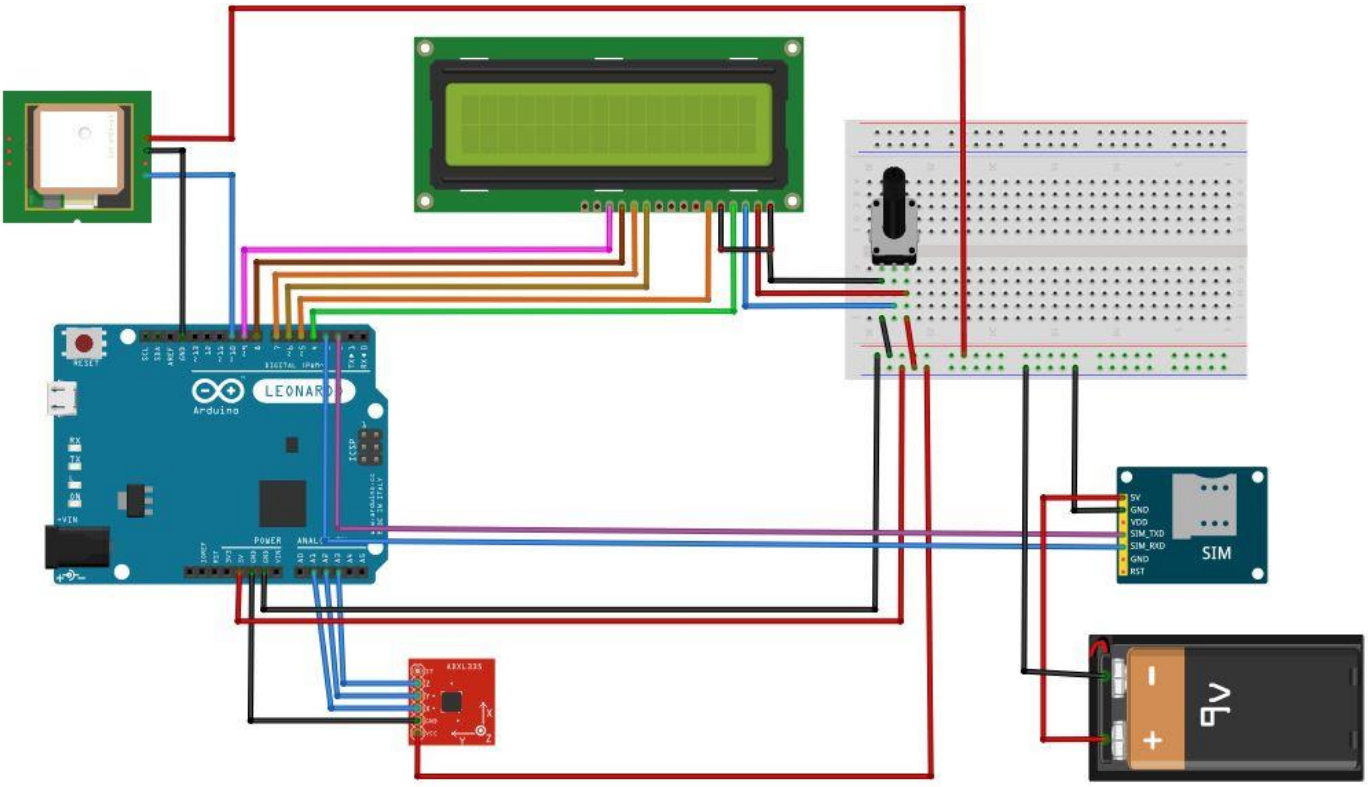
With a clear line of sight to the satellite, the GPS module was able to locate your position up to 7 minutes accuracy away from your actual position.

Step by step guide to build the circuit




RESET
DIGITAL (PWM~)
LEONARDO
Arduino
ICSP
RX
TX
L
ON
POWER
ANALOG IN
ADXL335


RESET
DIGITAL (PWM~)
LEONARDO
Arduino
ICSP
RX
TX
L
ON
POWER
ANALOG IN
ADXL335


LEONARDO
Arduino
RESET
DIGITAL (PWM~)
POWER
ANALOG IN
ICSP
5V
GND
VDD
SIM_TXD
SIM_RXD
GND
RST
SIM
ADXL335
X
Y
Z
−
+
9V

Project 2: collision detection system

Activity 4.10

Following the design process steps, ‘define’ the project requirements.

Objective:

Build a collision detection system using an ultrasonic sensor.

Constrain(s):

You should use the components in the resources list only.

Writing the code – testing the ultrasonic sensor

1- Define the variables

- The trig pin of the ultrasonic module is connected to pin 9 of the Arduino board.
- The echo pin of the ultrasonic module is connected to pin 10 of the Arduino board.

```
#define trigPin 9
#define echoPin 4
```

2- Void setup

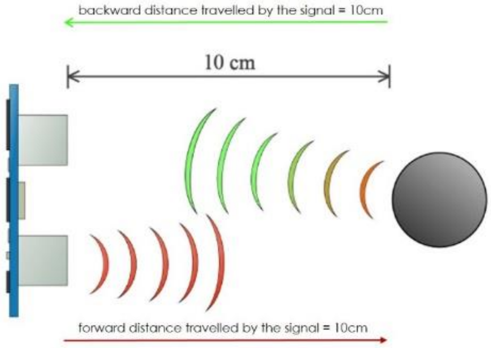
- Start a serial communication to be able to use the serial monitor.
- Define the trig pin as an output because it sends out an object detection signal.
- Define the echo pin as an input because it receives a signal when an object is detected.

```
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

3- Void loop

- Define two variables to hold the values of the duration and distance.
- Set the trig pin on a LOW state for 2 µs to make sure that the trig pin is clear.
- Set the trig pin on a HIGH state for 10 µs, and then LOW again to generate the ultrasound wave.
- Duration is measured using the ‘pulseIn’ function, which reads the signal’s travel time.

  - The ‘pulseIn’ function takes two variables, the signal pin and its status as high or low.
  - When it’s set to pulseIn(Echo_pin, HIGH), the echo_pin goes HIGH (a reflected signal is detected) and the timing starts.
- Calculate the distance where the speed of sound is equal to 340 m/s (0.034 cm/µs). The distance calculated should be divided by 2 as it represents the forward and backward distances that the signal has travelled. An example is shown in the figure below.

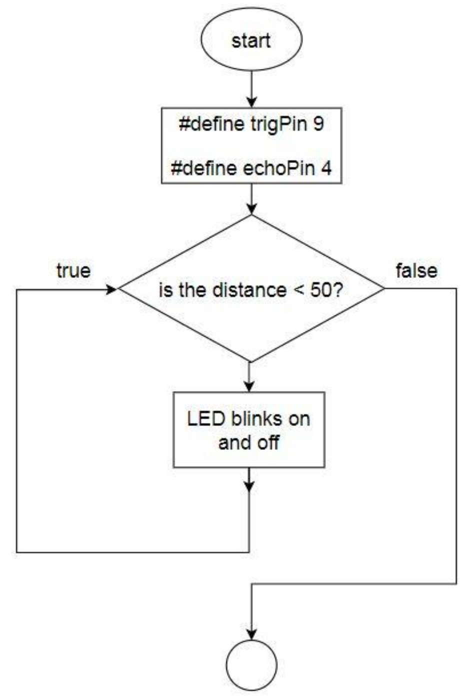

backward distance travelled by the signal = 10cm
10 cm
forward distance travelled by the signal = 10cm

- Print the distance value on the serial monitor.

```
void loop() {
  //variables
  long duration, distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(2);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  //print the distance value on the serial monitor
  Serial.print("Distance =");
  Serial.println(distance);
  delay(500);
}
```

Activity 4.11

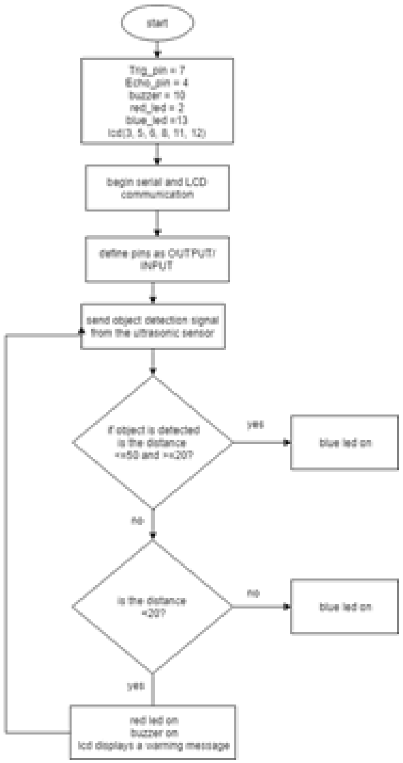
Create a simple collision detection system flowchart. If distance is less than 50cm, then an LED should start blinking ON and OFF to alert the driver.



Activity 4.12

Once you have built the circuit, modify the code you wrote in the previous task when testing the ultrasonic sensor to function as shown in the below flowchart.

Hint: to generate sound from the buzzer use the ‘tone function’. Example, tone(pin#, 2000).


start
Trig_pin = 7
Echo_pin = 4
buzzer = 10
red_led = 2
blue_led =13
lcd(3, 5, 6, 8, 11, 12)
begin serial and LCD communication
define pins as OUTPUT/ INPUT
send object detection signal from the ultrasonic sensor
if object is detected is the distance <=50 and >=20?
yes
blue led on
no
is the distance <20?
no
blue led on
yes
red led on
buzzer on
lcd displays a warning message

Arduino code

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(3, 5, 6, 8, 11, 12);
```

```
//variable initialisation

const int trigPin = 7;
const int echoPin = 4;
int buzz = 10;
int red_led=2;
int blue_led=13
long duration;
int distance;

void setup() {
  //serial communication
  Serial.begin(9600);
  Serial.println("collision detection system");

  //lcd communication
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("collision detec-");
  lcd.setCursor(0, 1);
  lcd.print("tion system");

  // initialize digital pins as output or input
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(blue_led, OUTPUT);
  pinMode(red_led, OUTPUT);
}

void loop(){
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // calculate the distance using the ultrasonic sensor
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2;

  if (distance <= 50 && distance >= 20)  {
    digitalWrite(blue_led, HIGH);
  }

  else  {
    digitalWrite(blue_led, LOW);
  }

  if (distance <= 20)  {
   Serial.println("Attention! Too close");
```

```
    lcd.clear();

    lcd.setCursor(3, 0);
    lcd.print("Attention! ");
    lcd.setCursor(3, 1);
    lcd.print("Too close");

    digitalWrite(red_led, HIGH);

    tone(buzz, 2000);
    delay(100);
    noTone(buzz);
    delay(100);

    tone(buzz, 2000);
    delay(100);
    noTone(buzz);
    delay(100);

    tone(buzz, 2000);
    delay(100);
    noTone(buzz);

    tone(buzz, 2000);
    delay(100);
    noTone(buzz);
    delay(100);
     }
  else  {
    digitalWrite(red_led, LOW);    // turn the LED off by making the voltage LOW
    lcd.clear();
    lcd.print("Have a good day");
    Serial.println("Have a good day");
  }
}
```

Testing – collision detection

Testing

- When the object is far from the sensor:

LEDs: off
Buzzer: off
LCD screen: collision detection system / Have a good day

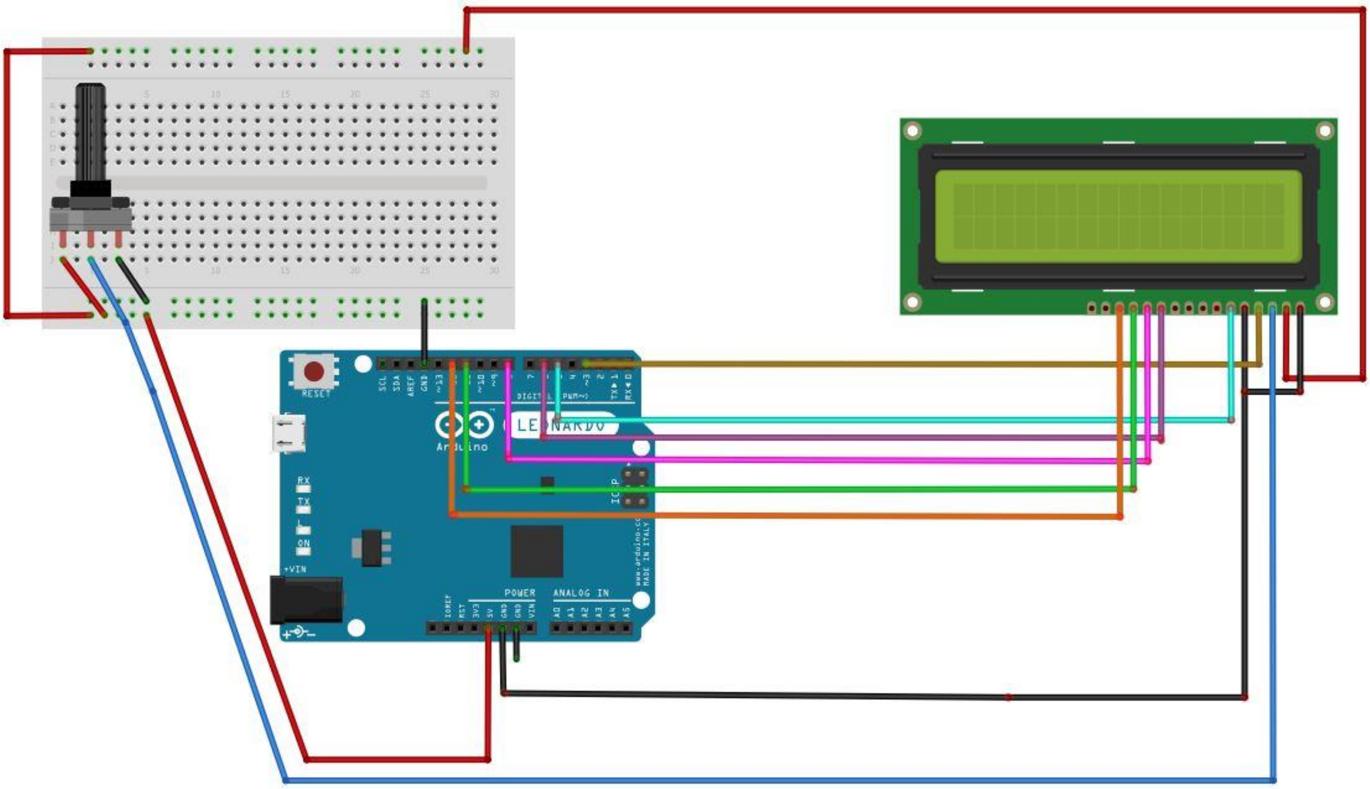
- When the object is close to the sensor:

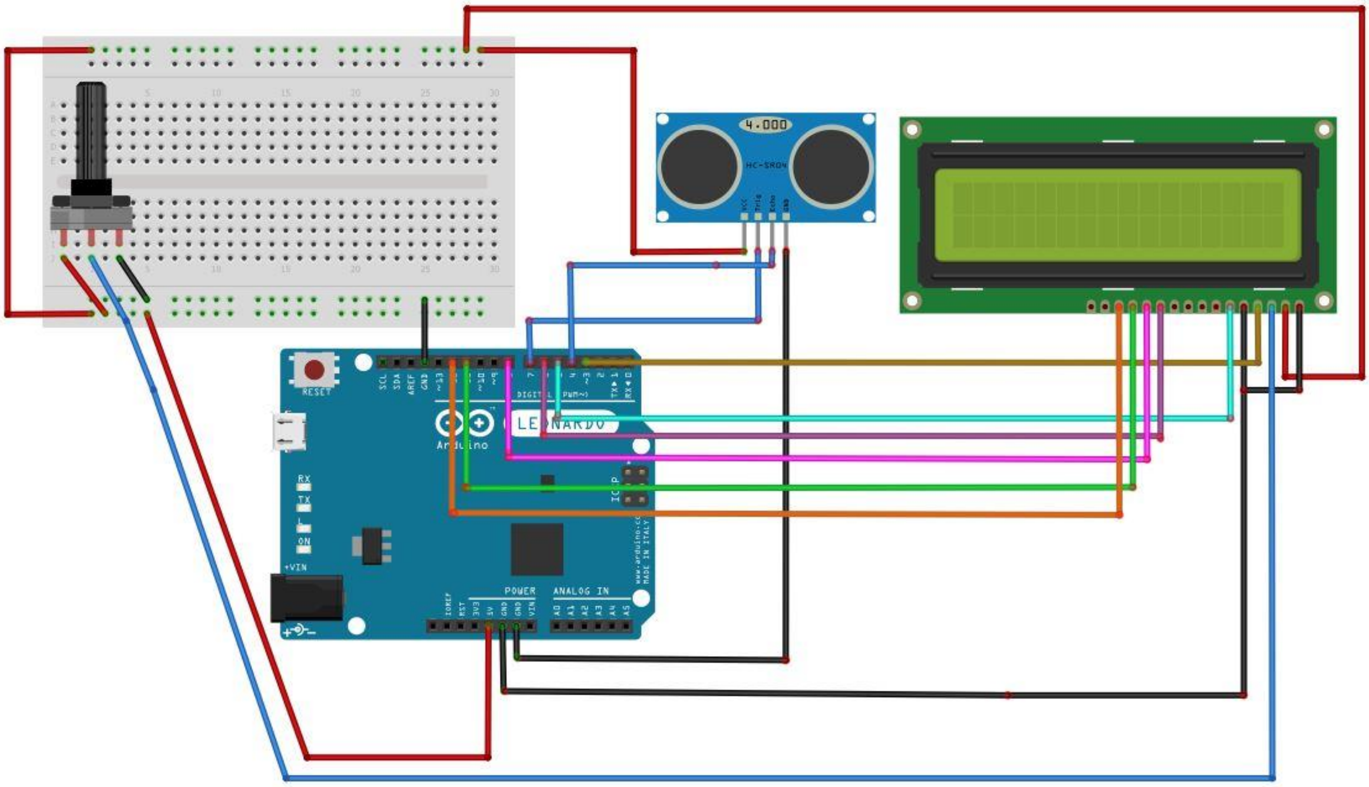
LEDs: blue LED on
Buzzer: off
LCD screen: collision detection system / Have a good day

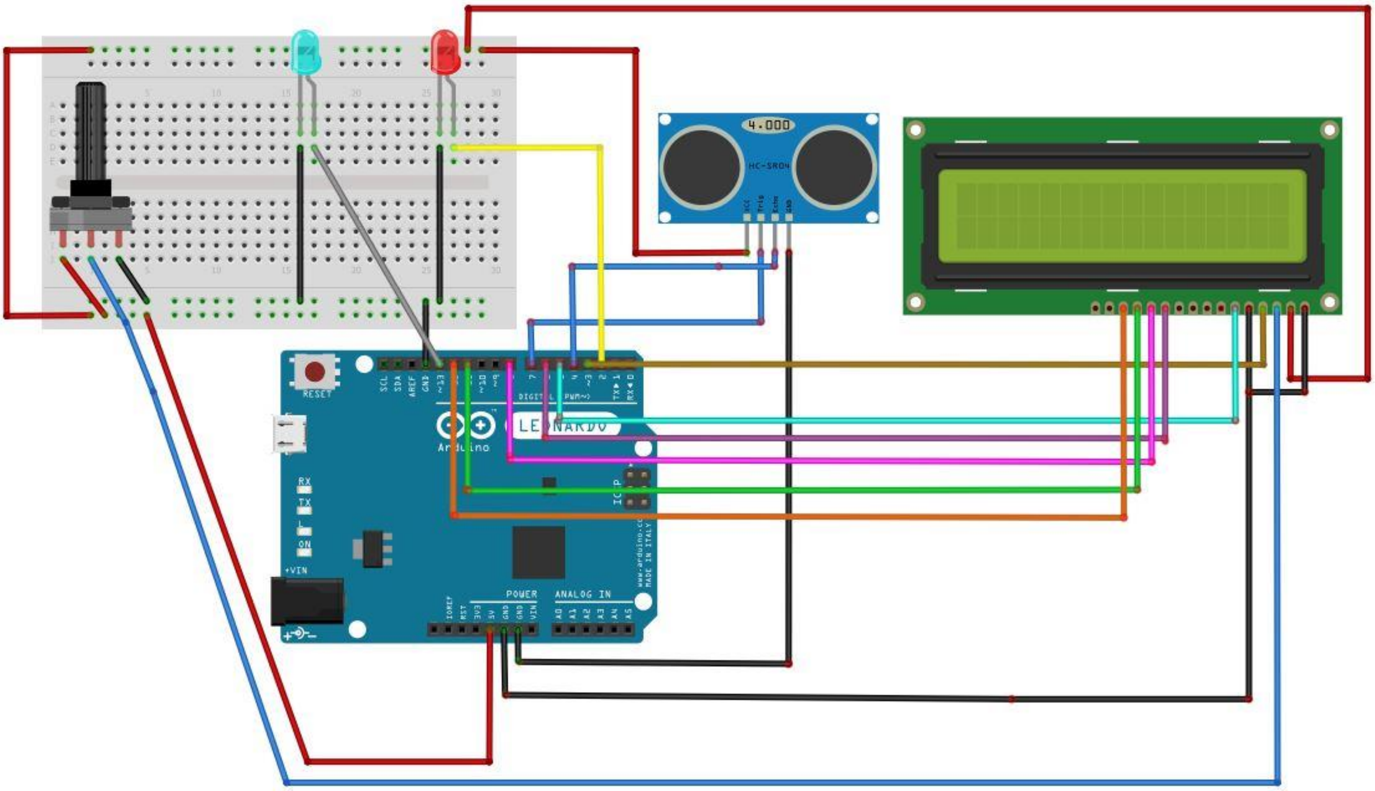
- When the object is very close to the sensor:

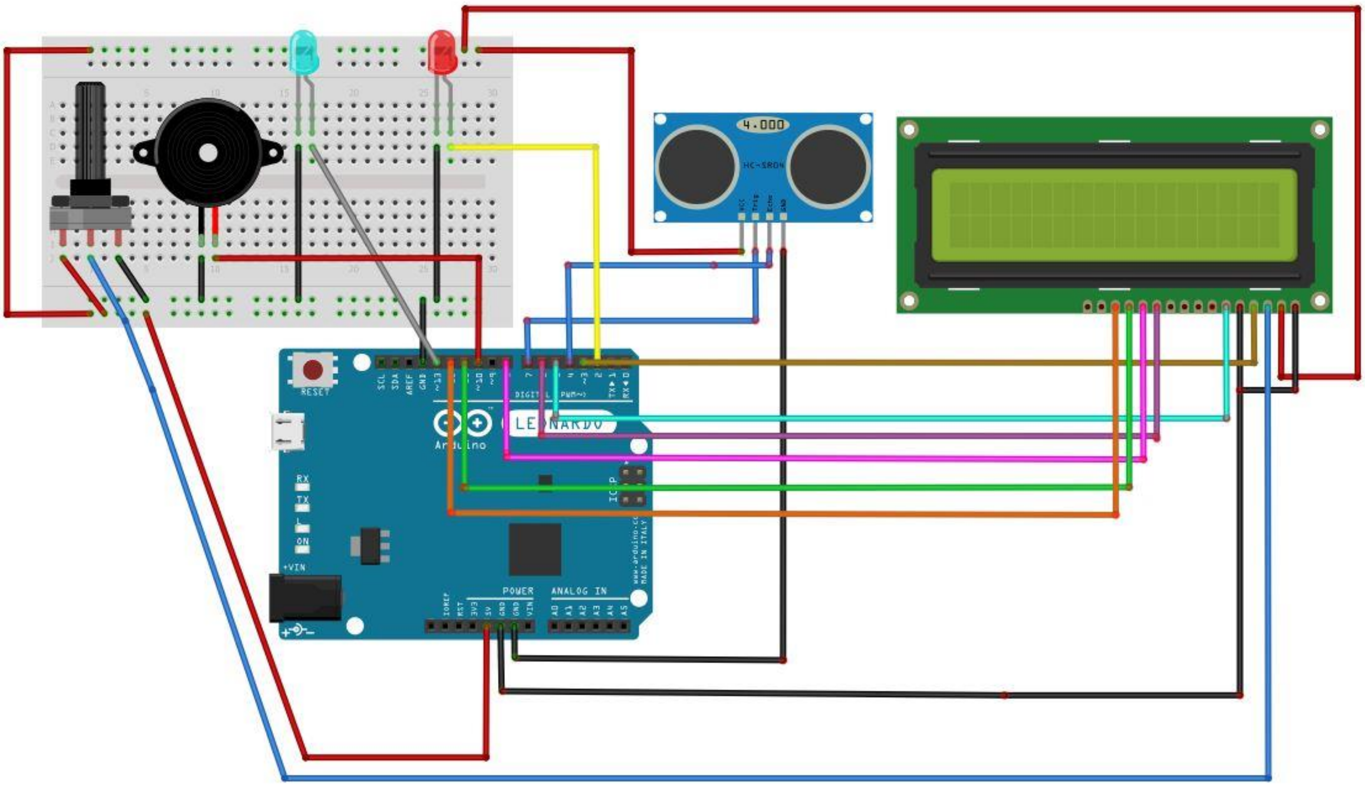
LEDs: red LED on
Buzzer: on
LCD screen: Attention! Too close

Step by step guide to build the circuit






4.000
HC-SR04
RESET
LEONARDO
Arduino
DIGITAL (PWM~)
POWER
ANALOG IN
ICSP
RX
TX
ON
+VIN


4.000
HC-SR04
RESET
LEONARDO
Arduino
DIGITAL (PWM~)
POWER
ANALOG IN
ICSP
RX
TX
ON
+VIN

Project 3: smart traffic lights

Activity 4.13

Following the design process steps, 'define' the project requirements.

Objective:

Build a smart traffic lights system using an ultrasonic sensor.

Constrain(s):

You should use the components in the resources list only.

Writing the code – testing the ultrasonic sensor

1- Define the variables

- The trig pin of the ultrasonic module is connected to pin 9 of the Arduino board.
- The echo pin of the ultrasonic module is connected to pin 10 of the Arduino board.

```
#define trigPin 9
#define echoPin 4
```

2- Void setup

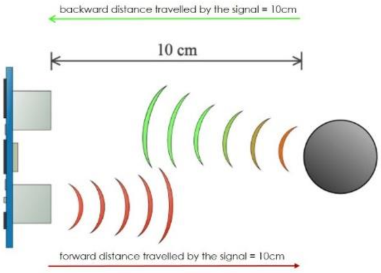
- Start a serial communication to be able to use the serial monitor.
- Define the trig pin as an output because it sends out an object detection signal.
- Define the echo pin as an input because it receives a signal when an object is detected.

```
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

3- Void loop

- Define two variables to hold the values of the duration and distance.
- Set the trig pin on a LOW state for 2 µs to make sure that the trig pin is clear.
- Set the trig pin on a HIGH state for 10 µs, and then LOW again to generate the ultrasound wave.
- Duration is measured using the 'pulseIn' function, which reads the signal's travel time.
    - The 'pulseIn' function takes two variables, the signal pin and its status as high or low.
    - When it's set to pulseIn(Echo_pin, HIGH), the echo_pin goes HIGH (a reflected signal is detected) and the timing starts.

- Calculate the distance where the speed of sound is equal to 340 m/s (0.034 cm/µs). The distance calculated should be divided by 2 as it represents the forward and backward distances that the signal has travelled. An example is shown in the figure below.

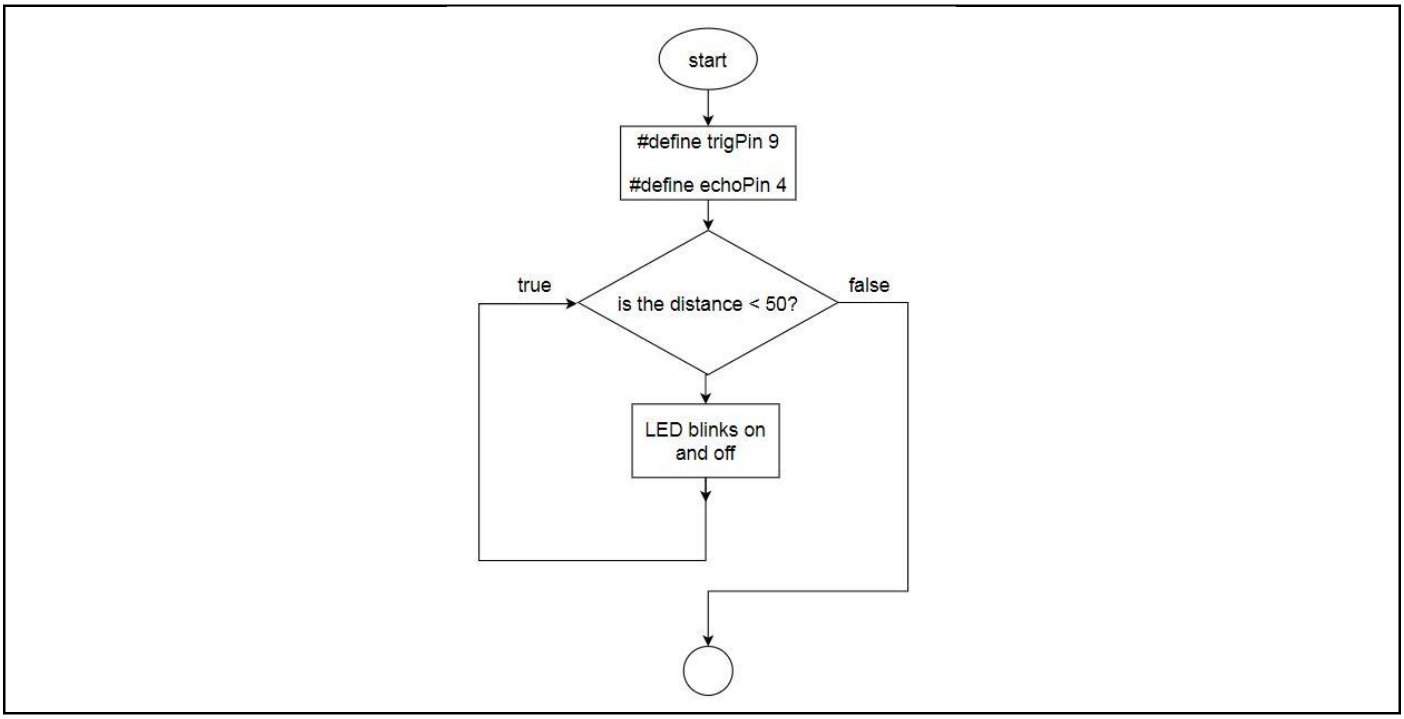

backward distance travelled by the signal = 10cm
10 cm
forward distance travelled by the signal = 10cm

- Print the distance value on the serial monitor.

```
void loop() {
  //variables
  long duration, distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(2);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  //print the distance value on the serial monitor
  Serial.print("Distance =");
  Serial.println(distance);
  delay(500);
}
```

Activity 4.14

Create a simple collision detection system flowchart. If distance is less than 50cm, then an LED should start blinking ON and OFF to alert the driver.



Testing – smart traffic lights

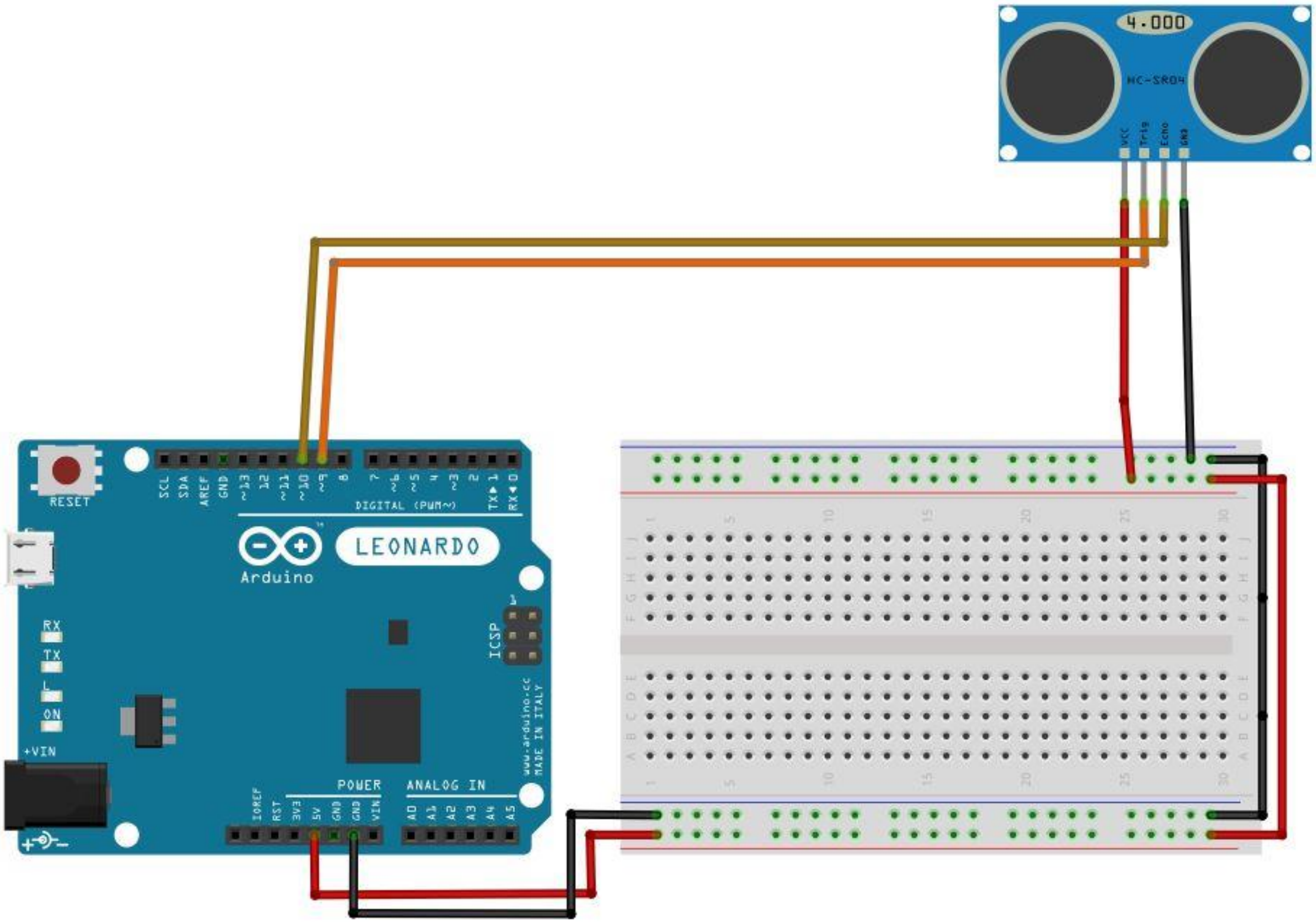
Testing

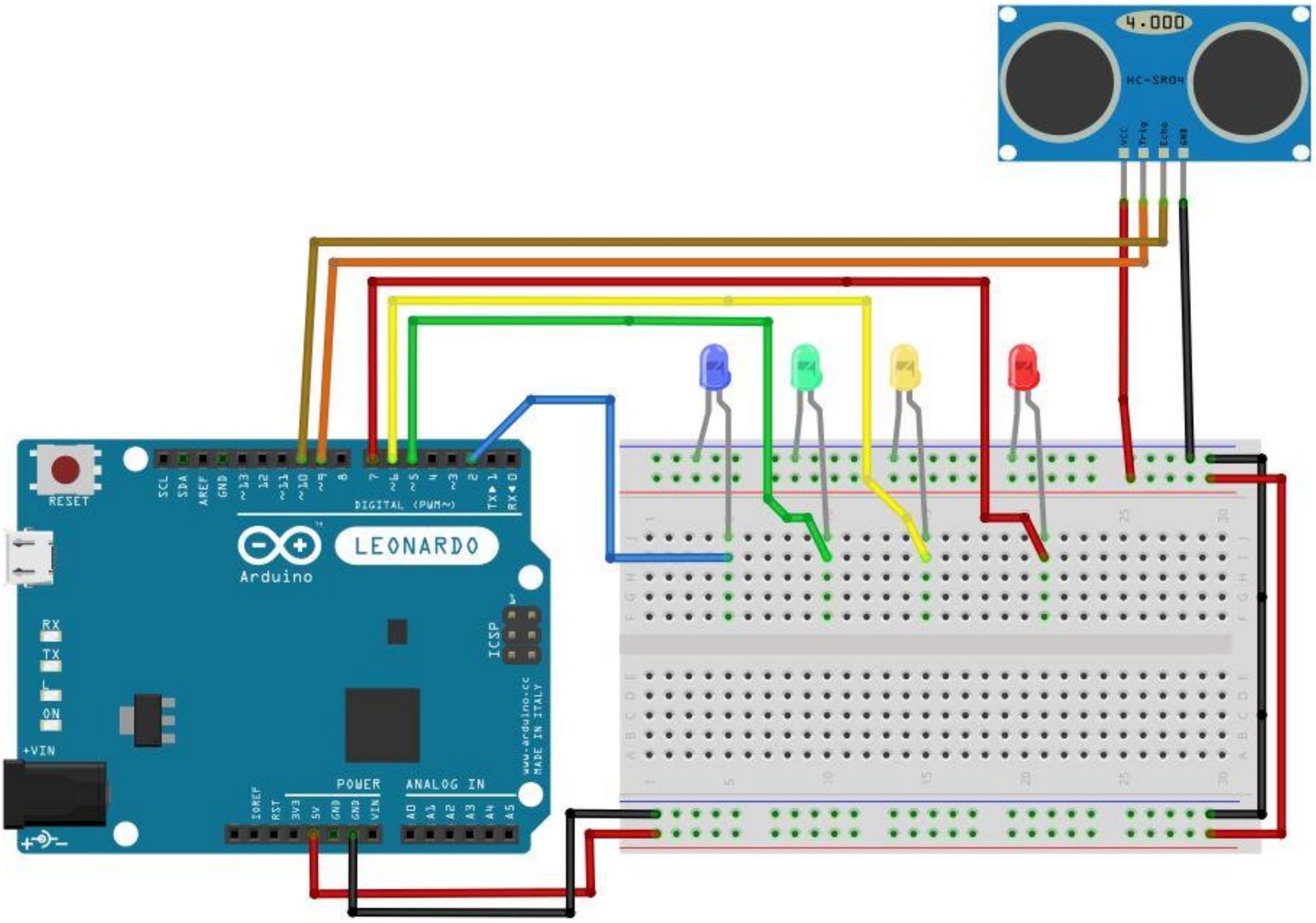
- When there is traffic on lane1:
  The traffic light signal on lane 2 will turn red, and the signal on lane 1 will turn green as long as the lane is congested with cars.
  Note, when the distance measured by the ultrasonic sensor is greater than 150 the lane is congested.

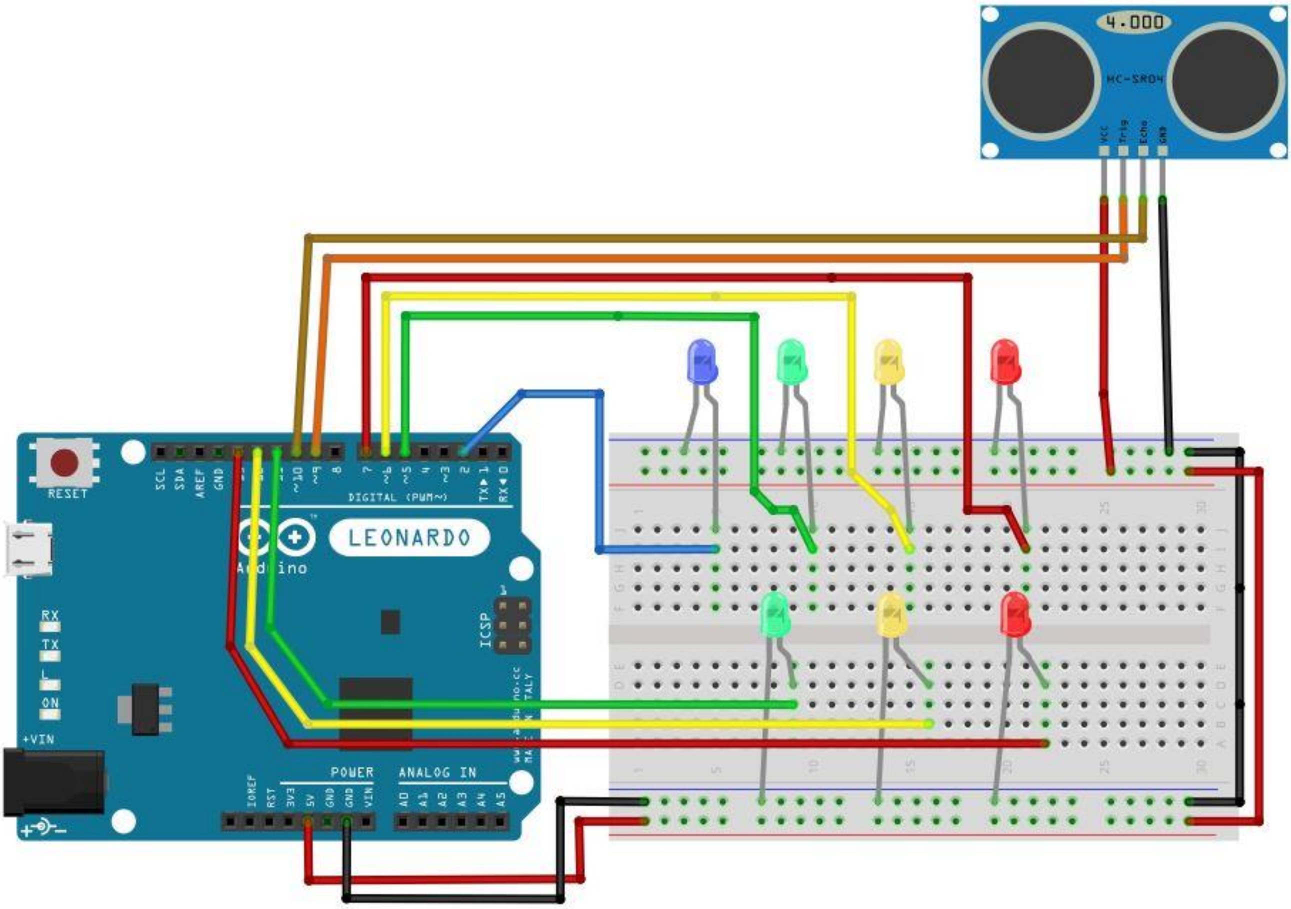
- When there is no traffic on lane1:
  The traffic light system runs normally, where the traffic lights on lanes 1 and 2 change in congestion with each other.

- From testing the system, what does the blue LED indicate?
  Indicator when the system runs under traffic condition.

Step by step guide to build the circuit


4.000
HC-SR04
VCC
Trig
Echo
GND
RESET
LEONARDO
Arduino
DIGITAL (PWM~)
ICSP
POWER
ANALOG IN
RX
TX
L
ON
+VIN
www.arduino.cc
MADE IN ITALY


4.000
HC-SR04
VCC
Trig
Echo
GND
RESET
LEONARDO
Arduino
DIGITAL (PWM~)
ICSP
POWER
ANALOG IN
RX
TX
L
ON
+VIN
www.arduino.cc
MADE IN ITALY


4.000
HC-SR04
VCC
Trig
Echo
GND
RESET
SCL
SDA
AREF
GND
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM~)
LEONARDO
Arduino
ICSP
RX
TX
L
ON
+VIN
IOREF
RST
3V3
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
www.arduino.cc
MADE IN ITALY

Activity 4.15

Following the design process steps, ‘define’ the project requirements.

Objective:

Build a fire detection and alert system using LM35 as the temperature sensor and the GSM module to send the alert messages.

Constrain(s):

You should use the components in the resources list only.

Activity 4.16

Modify the GSM code to send a different message to two different mobile numbers.

Edit the void SendMessage function by adding the following lines of code

```
mySerial.println(‘AT+CMGS=\’+971xxx\’\r’); // Replace x with mobile number
delay(1000);
mySerial.println("msg to the 2nd number");//the content of the message
delay(1000);
mySerial.println((char)26);//the message stopping character
delay(1000);
```

Writing the code – testing the LM35 temperature sensor

1- Define the variables

- The sensor is connected to analogue pin 1 of the Arduino board.
- Initialise a variable to hold the temperature values, and name it ‘Temp’.

```
const int sensor=A1; // Assigning analog pin A1 to variable 'sensor'
float temp;  //variable to store temperature in degree Celsius
```

2- Void setup

- Start a serial communication to be able to use the serial monitor.

```
void setup() {
  Serial.begin(9600);
}
```

3- Void loop

- Read the temperature values from the temperature sensor.

- Print the temperature values on the serial monitor.
- Add 1000ms delay.

```
void loop() {
  float vout = analogRead (sensor);
  vout=(vout*500)/1023;
  Temp = vout;

  //print the temperature value on the serial monitor
   Serial.print("in DegreeC=");
   Serial.print("\t");
   Serial.print(temp);
   Serial.println();

  //delay time
  delay(1000);
}
```

Activity 4.17

Modify the temperature sensor Arduino code so that, if the measured temperature is more than 50° C, a red LED starts blinking turning ON for 1,000 ms and then OFF for 1,000 ms.

```
const int sensor=A1; // Assigning analog pin A1 to variable 'sensor'
float temp;  //variable to store temperature in degree Celsius
float vout;  //temporary variable to hold sensor reading

int R_led =13;
void setup()
{
//pinMode(sensor,INPUT); // Configuring pin A1 as input
Serial.begin(9600);
}
void loop()
{
vout=analogRead(sensor);
vout=(vout*500)/1023;
temp=vout; // Storing value in Degree Celsius
Serial.print("in DegreeC=");
Serial.print("\t");
Serial.print(temp);
Serial.println();
  if(temp > 50){
    digitalWrite(R_led, HIGH);
    delay(1000);
    digitalWrite(R_led, LOW);
    delay(1000);
  }
delay(1000); //Delay of 1 second for ease of viewing
}
```

Testing the servo motor

```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo

void setup()
{
  myservo.attach(9);  // attaches the servo on pin# 9 of the Arduino board
}

void loop()
{
  myservo.write(0);        // for example position 0 for valve 'on'

delay(1000); // add 1000ms delay

  myservo.write(180);    // for example position 180 for valve 'off'

delay(1000); // add 1000ms delay

}
```

Testing – fire detection and alert system

Testing – Normal room temperature

Record the status of:

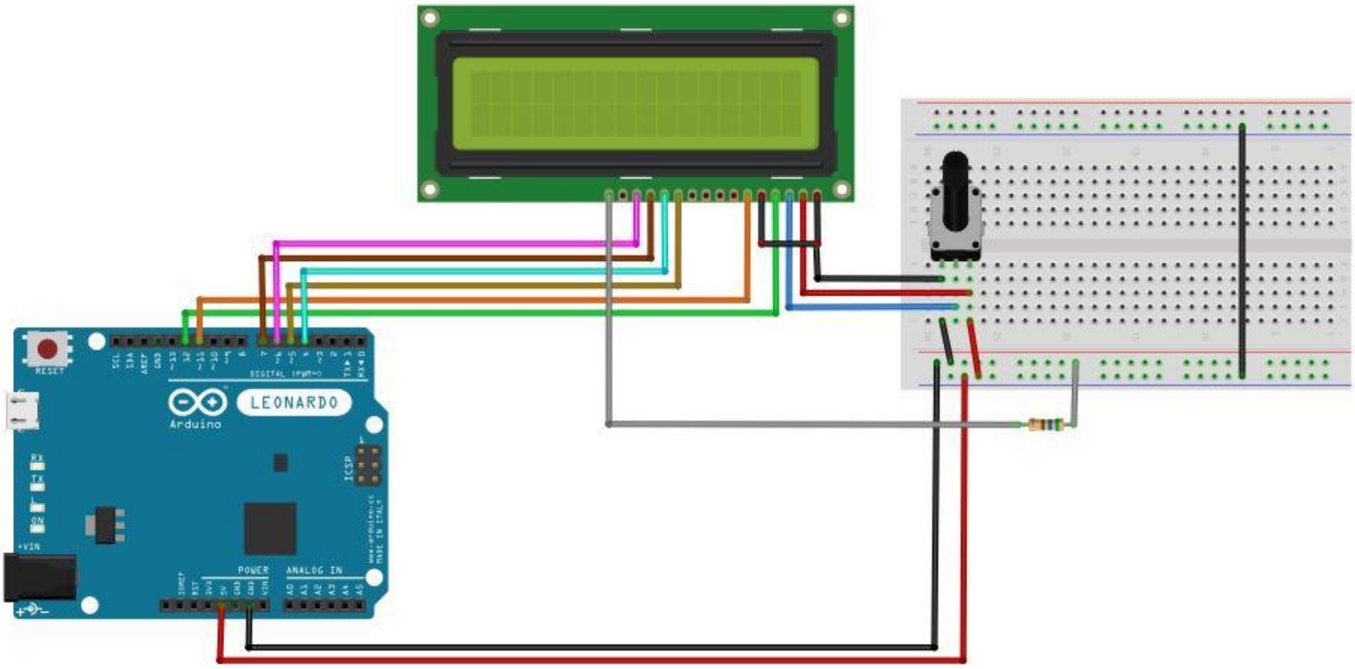
LED: off
Buzzer: off
LCD screen: no messages / Fire Scan – ON / Fire Shut! SAFE NOW
Servo motor: off
Did you receive a text message? □ yes ☒ no

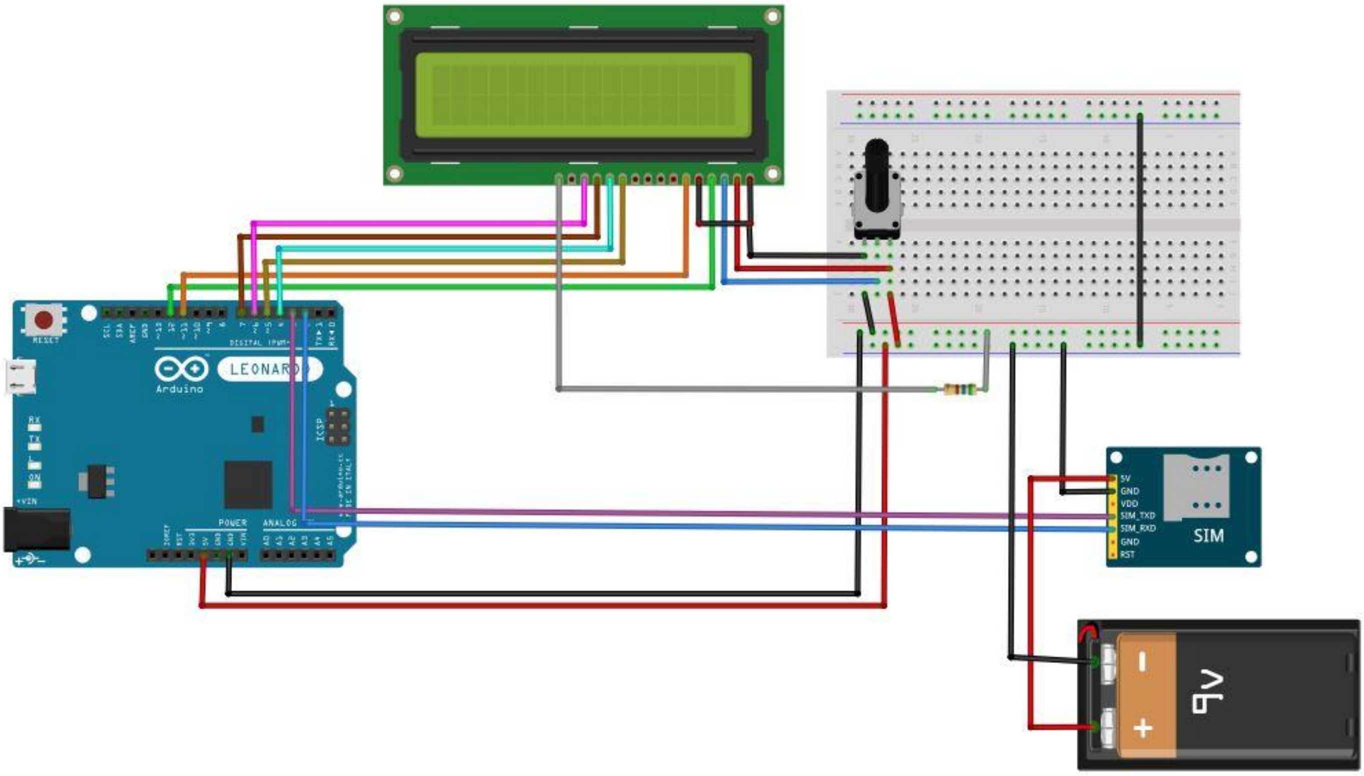
Testing – High room temperature

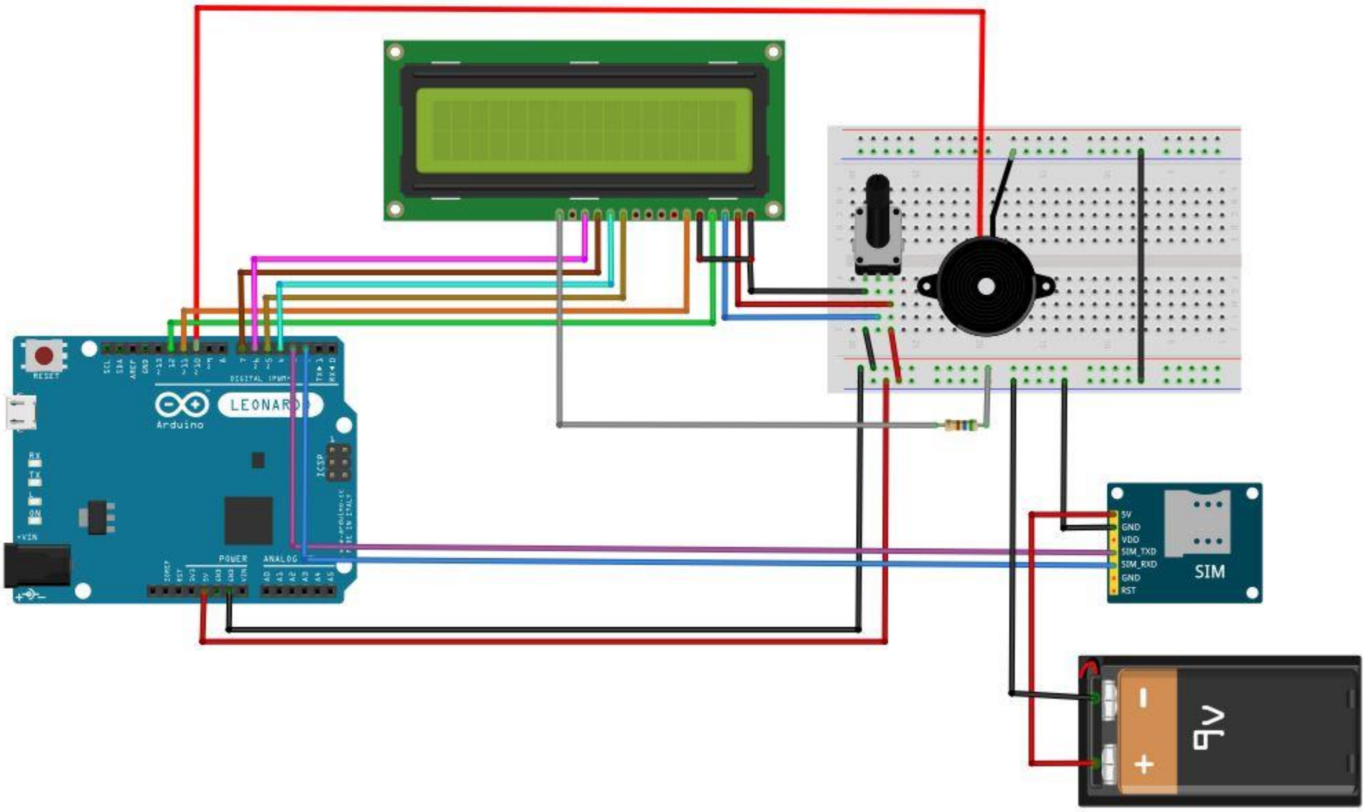
Record the status of:

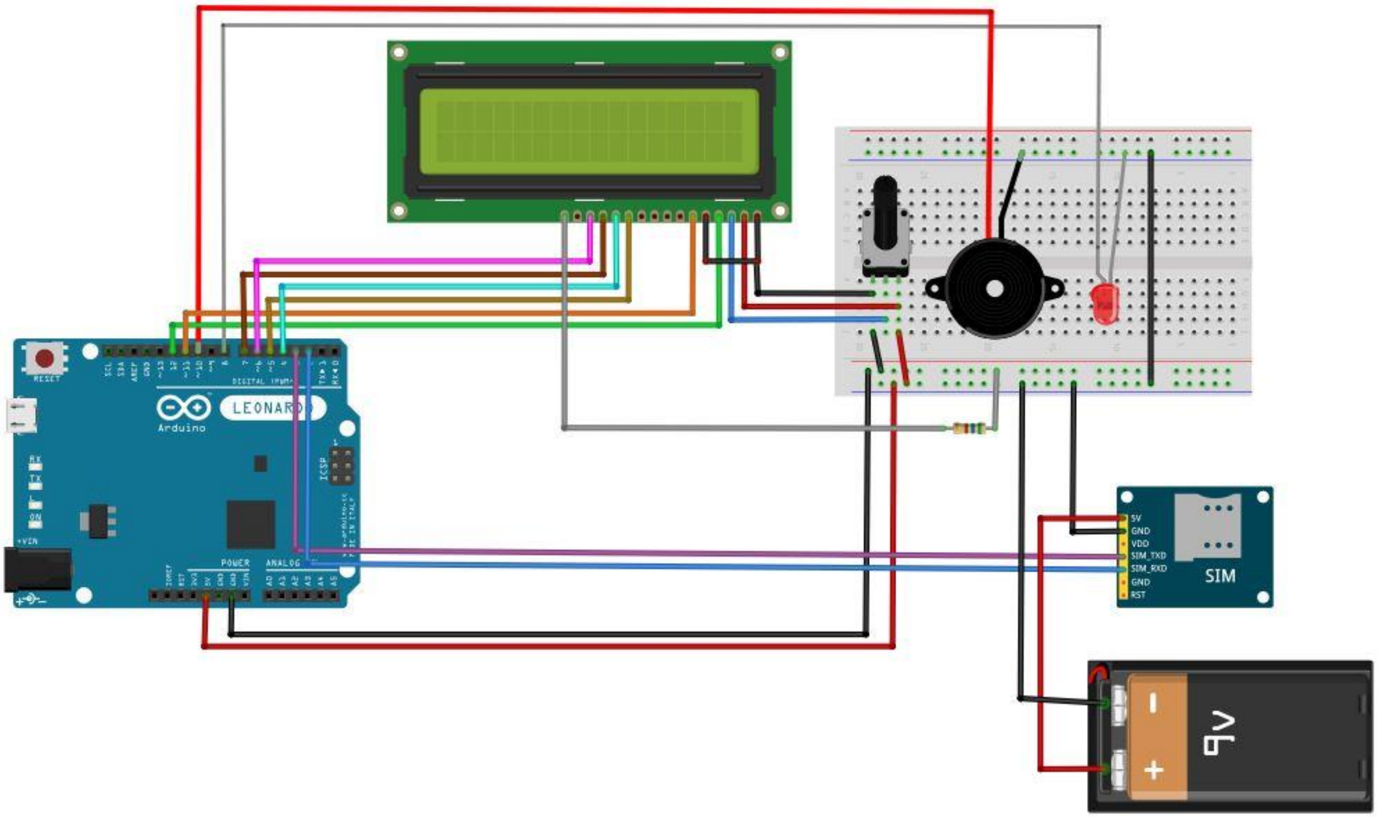
LED: blinks on and off
Buzzer: on
LCD screen: Fire Alert! SMS Sent! /
Servo motor: rotates 90°
Did you receive a text message? ☒ yes □ no

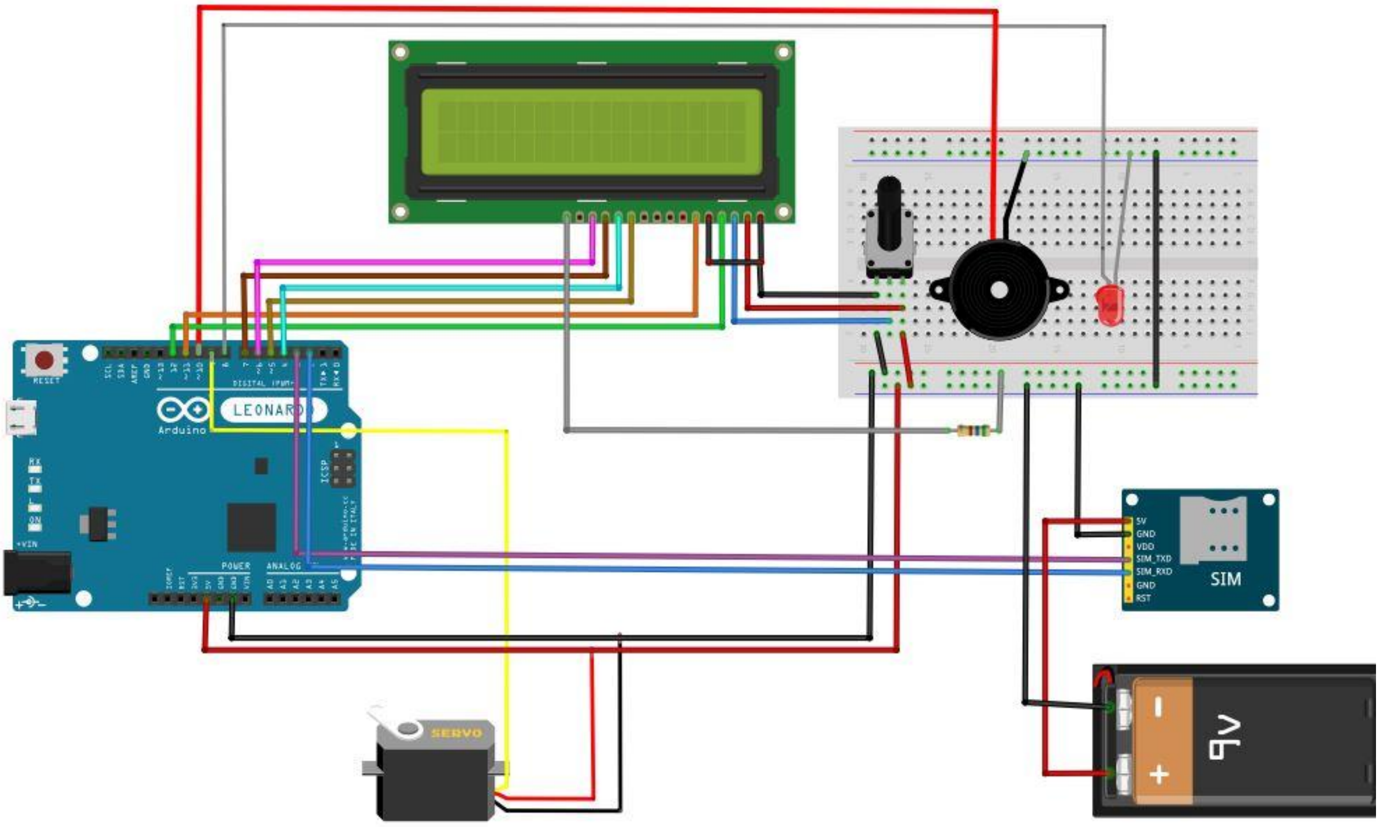
Step by step guide to build the circuit

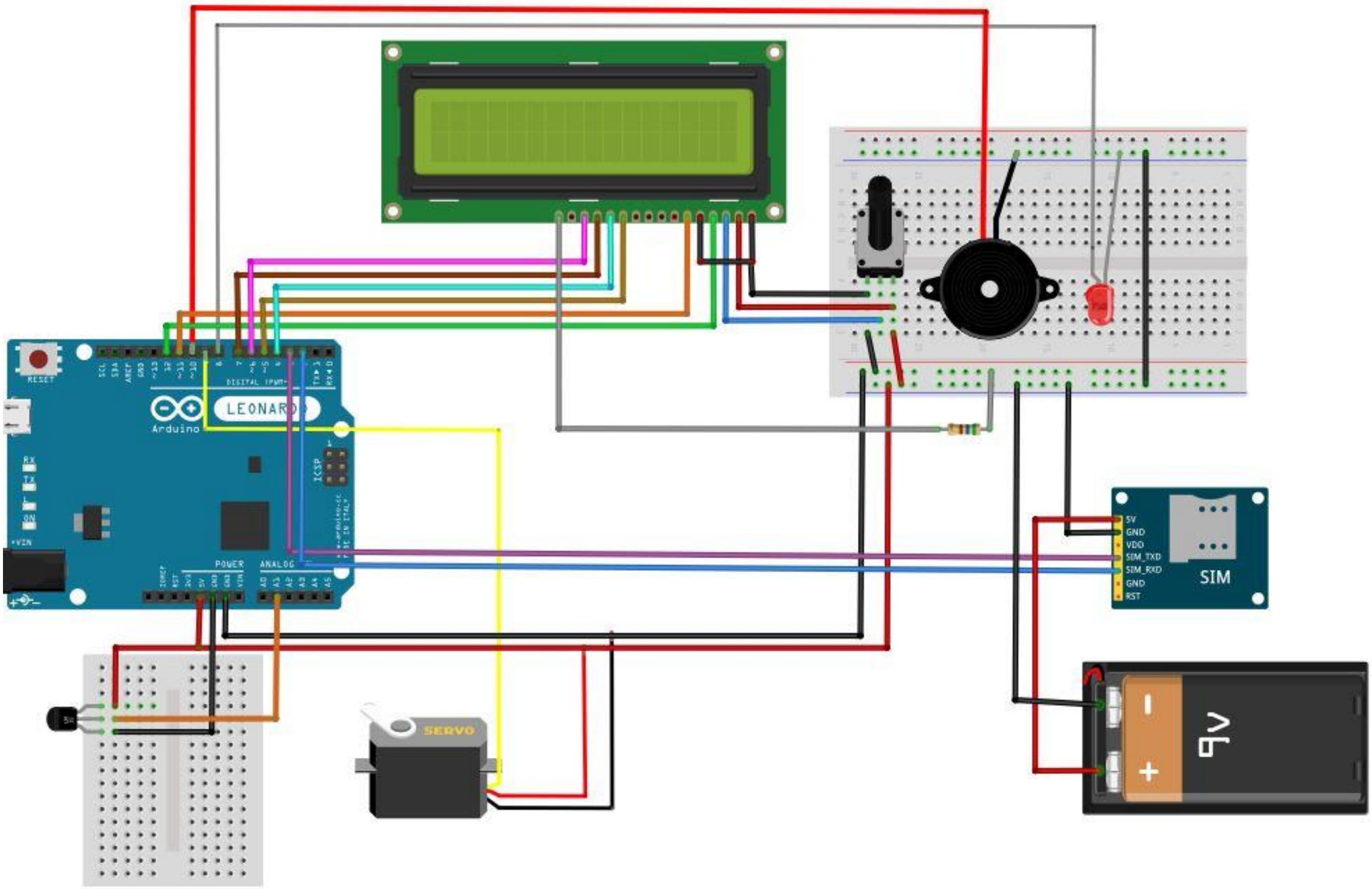





LEONARDO
Arduino
RESET
DIGITAL (PWM~)
ICSP
POWER
ANALOG
5V
GND
VDD
SIM_TXD
SIM_RXD
GND
RST
SIM
9V
-
+


LEONARDO
Arduino
RESET
DIGITAL (PWM~)
ICSP
POWER
ANALOG
5V
GND
VDD
SIM_TXD
SIM_RXD
GND
RST
SIM
9V
-
+


RESET
LEONARDO
Arduino
DIGITAL (PWM~)
POWER
ANALOG
ICSP
RX
TX
ON
SERVO
5V
GND
VDD
SIM_TXD
SIM_RXD
GND
RST
SIM
9V
-
+


RESET
DIGITAL (PWM~)
LEONARDO
Arduino
ICSP
POWER
ANALOG
SERVO
5V
GND
VDD
SIM_TXD
SIM_RXD
GND
RST
SIM
9V
-
+

Project 5: burglar alarm system

Activity 4.18

Following the design process steps, ‘define’ the project requirements.

Objective:
Build a burglar alarm system using an ultrasonic sensor and a GSM module to send alert messages.

Constrain(s):
You should use the components in the resources list only.

Activity 4.19

Modify the GSM code to send a different message to two different mobile numbers.

Edit the void SendMessage function by adding the following lines of code

```
mySerial.println(‘AT+CMGS=\’+971xxx\’\r’); // Replace x with mobile number
delay(1000);
mySerial.println("msg to the 2nd number");//the content of the message
delay(1000);
mySerial.println((char)26);//the message stopping character
delay(1000);
```

Writing the code – testing the ultrasonic sensor

1- Define the variables
- The trig pin of the ultrasonic module is connected to pin 9 of the Arduino board.
- The echo pin of the ultrasonic module is connected to pin 10 of the Arduino board.

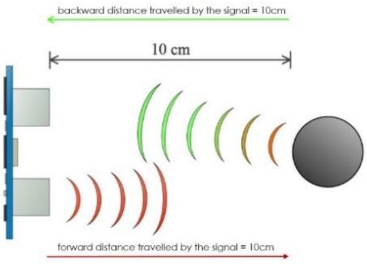
```
#define trigPin 9
#define echoPin 4
```

2- Void setup
- Start a serial communication to be able to use the serial monitor.
- Define the trig pin as an output because it sends out an object detection signal.
- Define the echo pin as an input because it receives a signal when an object is detected.

```
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

3- Void loop
- Define two variables to hold the values of the duration and distance.
- Set the trig pin on a LOW state for 2 µs to make sure that the trig pin is clear.

- Set the trig pin on a HIGH state for 10 µs, and then LOW again to generate the ultrasound wave.
- Duration is measured using the 'pulseIn' function, which reads the signal's travel time.
    - The 'pulseIn' function takes two variables, the signal pin and its status as high or low.
    - When it's set to pulseIn(Echo_pin, HIGH), the echo_pin goes HIGH (a reflected signal is detected) and the timing starts.
- Calculate the distance where the speed of sound is equal to 340 m/s (0.034 cm/µs). The distance calculated should be divided by 2 as it represents the forward and backward distances that the signal has travelled. An example is shown in the figure below.


backward distance travelled by the signal = 10cm
10 cm
forward distance travelled by the signal = 10cm

- Print the distance value on the serial monitor.

```
void loop() {
  //variables
  long duration, distance;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(2);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  //print the distance value on the serial monitor
  Serial.print("Distance =");
  Serial.println(distance);
  delay(500);
}
```

Activity 4.20

Modify the ultrasonic sensor Arduino code so that if the measured distance is less than 50cm, a red LED turns on. Otherwise, a blue LED turns on.

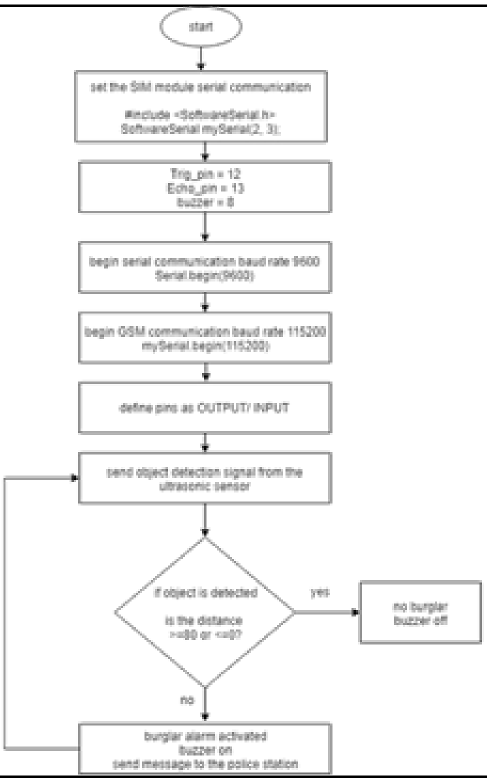
```
#define trigPin 7
#define echoPin 4

int red = 13;
int blue = 10;
```

```
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(red, OUTPUT);
  pinMode(blue, OUTPUT);
}
void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;
  Serial.print("Distance =");
  Serial.println(distance);
  delay(500);

  if (distance < 50) {
    digitalWrite(red, HIGH);
    delay(500);
    digitalWrite(blue, LOW);
  }
  else  {
    digitalWrite(red, LOW);
    delay(500);
    digitalWrite(blue, HIGH);
    delay(500);
  }
}
```

Activity 4.21

Once you have built the circuit, modify the codes you wrote in the previous tasks (when testing the GSM module and the ultrasonic sensor) to function as shown in the flowchart below.

Hint: To generate sound from the buzzer, use the 'tone function'; for example, tone(pin#, 2000).


start
set the SIM module serial communication
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
Trig_pin = 12
Echo_pin = 13
buzzer = 8
begin serial communication baud rate 9600
Serial.begin(9600)
begin GSM communication baud rate 115200
mySerial.begin(115200)
define pins as OUTPUT/ INPUT
send object detection signal from the ultrasonic sensor
if object is detected
is the distance >=80 or <=0?
yes
no burglar
buzzer off
no
burglar alarm activated
buzzer on
send message to the police station

Arduino code

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);//rx tx

//variable initialisation
#define trigPin 12
#define echoPin 13
int Buzzer = 8;

void setup() {
  //serial communication
  Serial.begin (9600);

  //GSM communication
  mySerial.begin(115200);   // Setting the baud rate of GSM Module

  // initialize digital pins as output or input
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(Buzzer, OUTPUT);
}

void loop(){
  // calculate the distance using the ultrasonic sensor
  int duration, distance;
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(1000);
  digitalWrite(trigPin, LOW );
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 74;

  if (distance >= 80 || distance <= 0) {
   //no burglar
   Serial.println("no object detected");
   digitalWrite(Buzzer, LOW);
  }
  else  {
   //activate burglar alarm
   // send message
  mySerial.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(1000);  // Delay of 1000 milli seconds or 1 second
  mySerial.println("AT+CMGS=\"+9715xxxx\"\r"); // Replace x with mobile number
  delay(1000);
  mySerial.println("Attention! Burglar alarm");// The SMS text you want to send
  delay(100);
  mySerial.println((char)26);// ASCII code of CTRL+Z
  delay(1000);

  //sound the buzzer
   Serial.println("object detected");
   tone(Buzzer, 400); // play 400 Hz tone for 500 ms
   delay(500);
   tone(Buzzer, 800); // play 800Hz tone for 500ms
   delay(500);
   tone(Buzzer, 400); // play 400 Hz tone for 500 ms
   delay(500);
   tone(Buzzer, 800); // play 800Hz tone for 500ms
   delay(500);
   tone(Buzzer, 400); // play 400 Hz tone for 500 ms
   delay(500);
   tone(Buzzer, 800); // play 800Hz tone for 500ms
   delay(500);
   noTone(Buzzer);
  }
  }
```

Testing – burglar alarm system

Testing – No burglar

Record the status of:

Buzzer: off

Did you receive a text message? □ yes ■ no

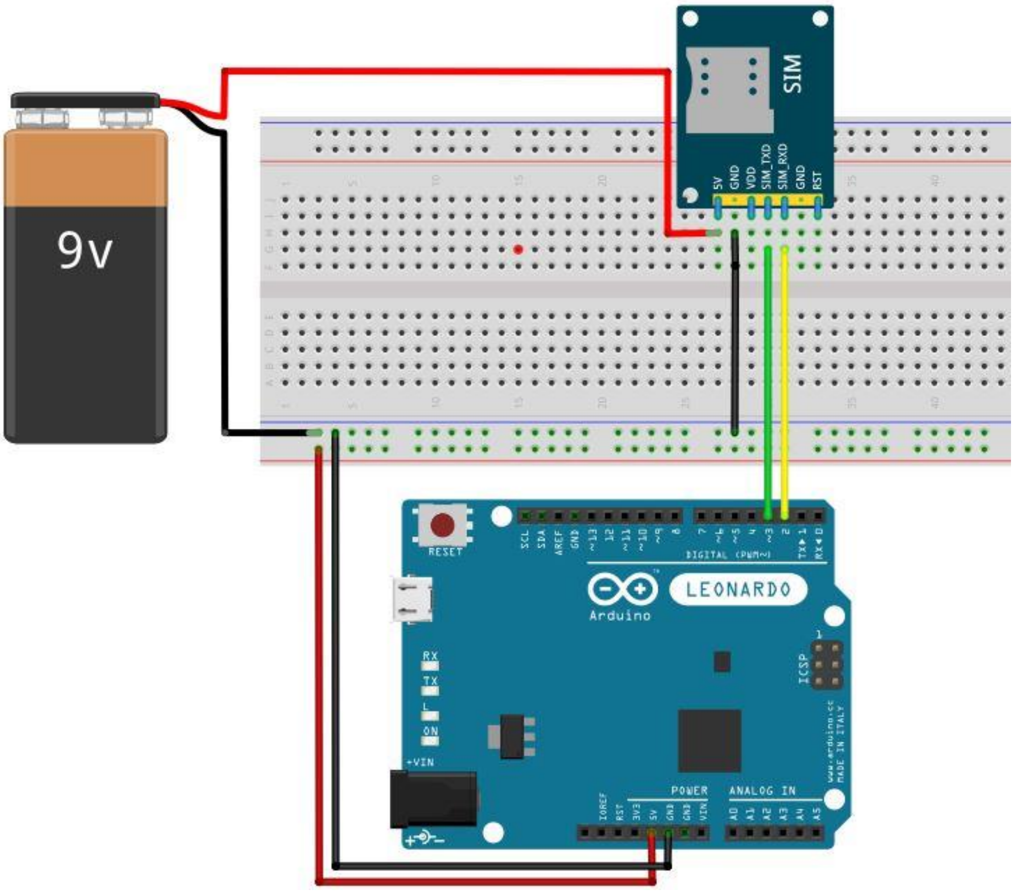
Testing – Burglar alert

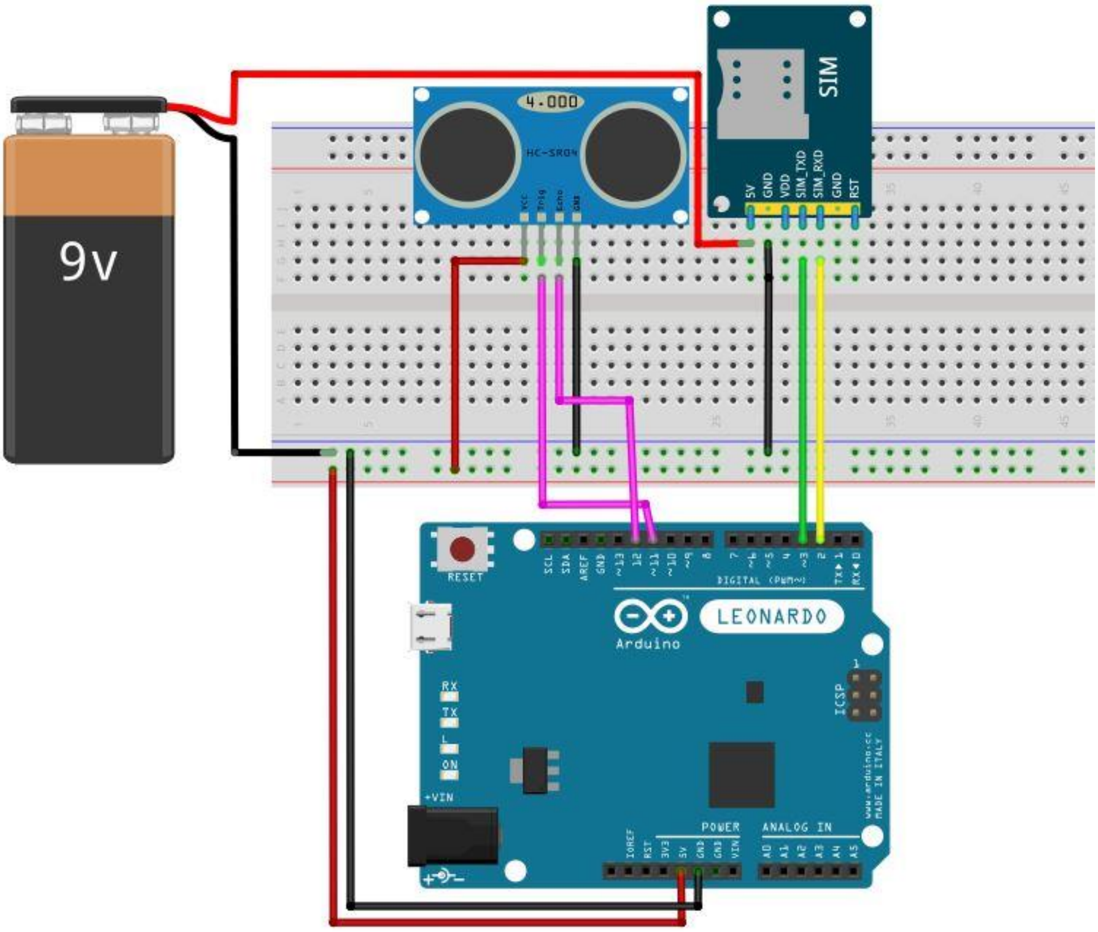
Record the status of:

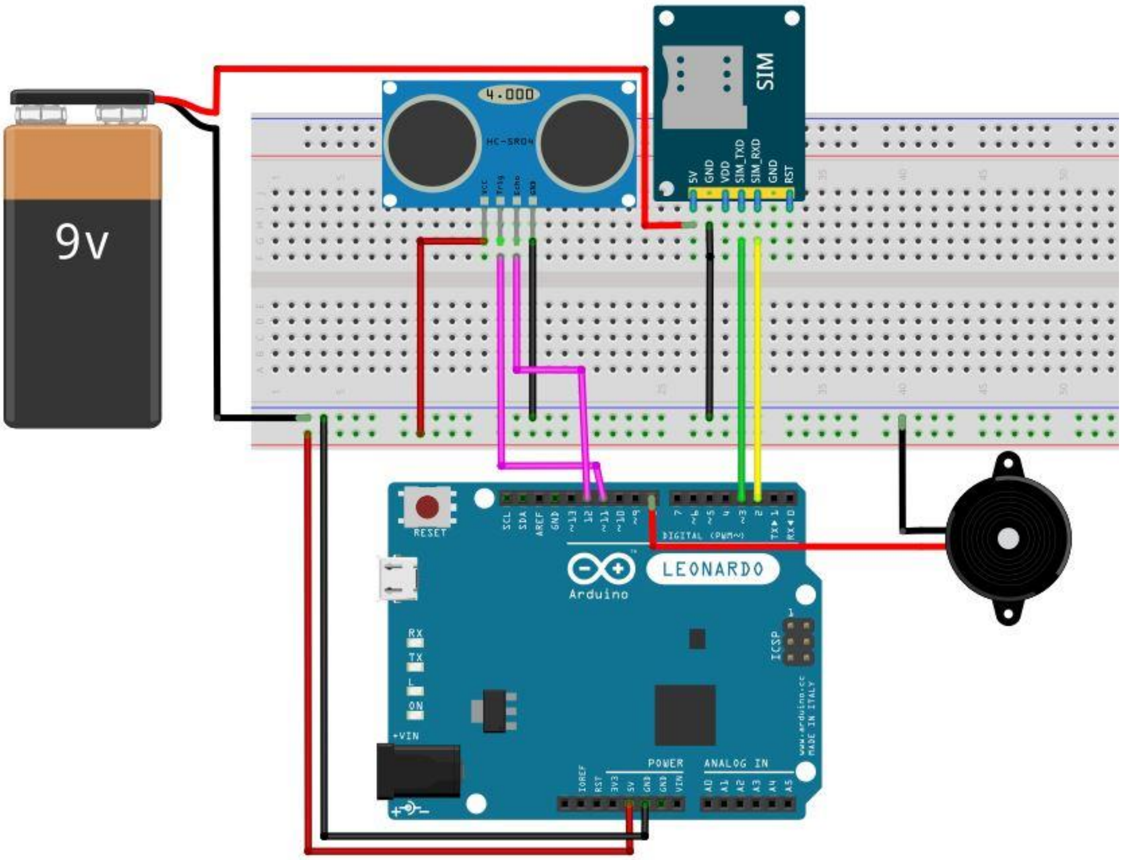
Buzzer: on

Did you receive a text message? ■ yes □ no

Step by step guide to build the circuit


9v
SIM
5V
GND
VDD
SIM_TXD
SIM_RXD
GND
RST
RESET
LEONARDO
Arduino
DIGITAL (PWM~)
RX
TX
L
ON
+VIN
ICSP
POWER
ANALOG IN
IOREF
RST
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
www.arduino.cc
MADE IN ITALY


9v
4.000
HC-SR04
Vcc
Trig
Echo
Gnd
SIM
5V
GND
VDD
SIM_TXD
SIM_RXD
GND
RST
RESET
SCL
SDA
AREF
GND
~13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM~)
LEONARDO
Arduino
ICSP
RX
TX
L
ON
+VIN
IOREF
RST
3V3
5V
GND
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
www.arduino.cc
MADE IN ITALY


9v
4.000
HC-SR04
Vcc
Trig
Echo
Gnd
SIM
5V
GND
VDD
SIM_TXD
SIM_RXD
GND
RST
RESET
SCL
SDA
AREF
GND
~13
12
~11
~10
~9
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM~)
LEONARDO
Arduino
RX
TX
L
ON
ICSP
+VIN
www.arduino.cc
MADE IN ITALY
POWER
ANALOG IN
IOREF
RST
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5

Project 6: smart parking system

Activity 4.22

Following the design process steps, ‘define’ the project requirements.

Objective:
Build a smart parking lot system using 7-segment display and a servo motor.

Constrain(s):
You should use the components in the resources list only.

Testing the servo motor

```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo

void setup()
{
  myservo.attach(9);  // attaches the servo on pin# 9 of the Arduino board
}

void loop()
{
  myservo.write(0);       // for example position 0 for valve 'on'

delay(1000); // add 1000ms delay

  myservo.write(180);    // for example position 180 for valve 'off'

delay(1000); // add 1000ms delay

}
```

Testing the 7-segment display

1- Define the variables
   - Define the pin number for each segment. The first one is done for you.

```
// define the pins number for the 7-segment

int a =2;
int b = 4;
int c = 8;
int d = 12;
int e = 13;
int f = 11;
int g = 10;
int DP = 9;
```

2- Void setup

- Define the pins as OUTPUT, using pinMode function.
- Initialise the status of the 7-segment to display the number zero.

```
void setup() {
// define the pins as output
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  pinMode(DP, OUTPUT);

  // initialise the status of the 7-segment
  digitalWrite (a, HIGH);
  digitalWrite (b, HIGH);
  digitalWrite (c, HIGH);
  digitalWrite (d, HIGH);
  digitalWrite (e, HIGH);
  digitalWrite (f, HIGH);
  digitalWrite (g, LOW);
  digitalWrite (DP, HIGH);

}
```

3- Void loop

- Display number 3 on the 7-segment display.
- Add 2000ms delay
- Display number 7 on the 7-segment display.
- Add 5000ms delay

```
void loop() {
  //display number 3
  digitalWrite (a, HIGH);
  digitalWrite (b, HIGH);
  digitalWrite (c, HIGH);
  digitalWrite (d, HIGH);
  digitalWrite (e, LOW);
  digitalWrite (f, LOW);
  digitalWrite (g, HIGH);

  //delay 2000ms
  delay(2000);

  //display number 7
  digitalWrite (a, HIGH);
  digitalWrite (b, HIGH);
  digitalWrite (c, HIGH);
  digitalWrite (d, LOW);
  digitalWrite (e, LOW);
  digitalWrite (f, LOW);
  digitalWrite (g, LOW);

 //delay 5000ms
  delay(5000);
}
```

The system's Arduino code.

Arduino code

```
void setup(){
  myservo.attach(ServoM);         // attaches the servo.

  pinMode(Exit, INPUT);           // set "EXIT" button pin to input
  pinMode(In, INPUT);             // set "IN" button pin to input
  digitalWrite(Exit, HIGH);       // Connect Pull-Up resistor.
  digitalWrite(In, HIGH);         // Connect Pull-Up resistor.

  //set the 7-segmet pins as output, the 1st one if done for you
  pinMode(segA,OUTPUT);
  pinMode(segB,OUTPUT);
  pinMode(segC,OUTPUT);
  pinMode(segD,OUTPUT);
  pinMode(segE,OUTPUT);
  pinMode(segF,OUTPUT);
  pinMode(segG,OUTPUT);
  pinMode(Bright,OUTPUT);

  analogWrite(Bright,255*INTEN/100);
  myservo.write(BarLow);          //Barrier in the low position
}
int  Available= 9;                // Number of places available.
void loop(){
Display(Available);
if(digitalRead(In)==0)
{
  if(Available != 0){
    Available--; // let a car enter then decrement the number of available spots
    myservo.write(BarUp);
    delay(3000);
    myservo.write(BarLow);
    }
  }
if(digitalRead(Exit)==0)
{
  if(Available != CAPACITY){
    // increment the number of available spots
    Available++;

    // allow a car to enter
    Available++;
    myservo.write(BarUp);
    delay(3000);
    myservo.write(BarLow);
    }
  }
}
```

Testing – smart parking system

Testing – Available parking spots

Record the status of:

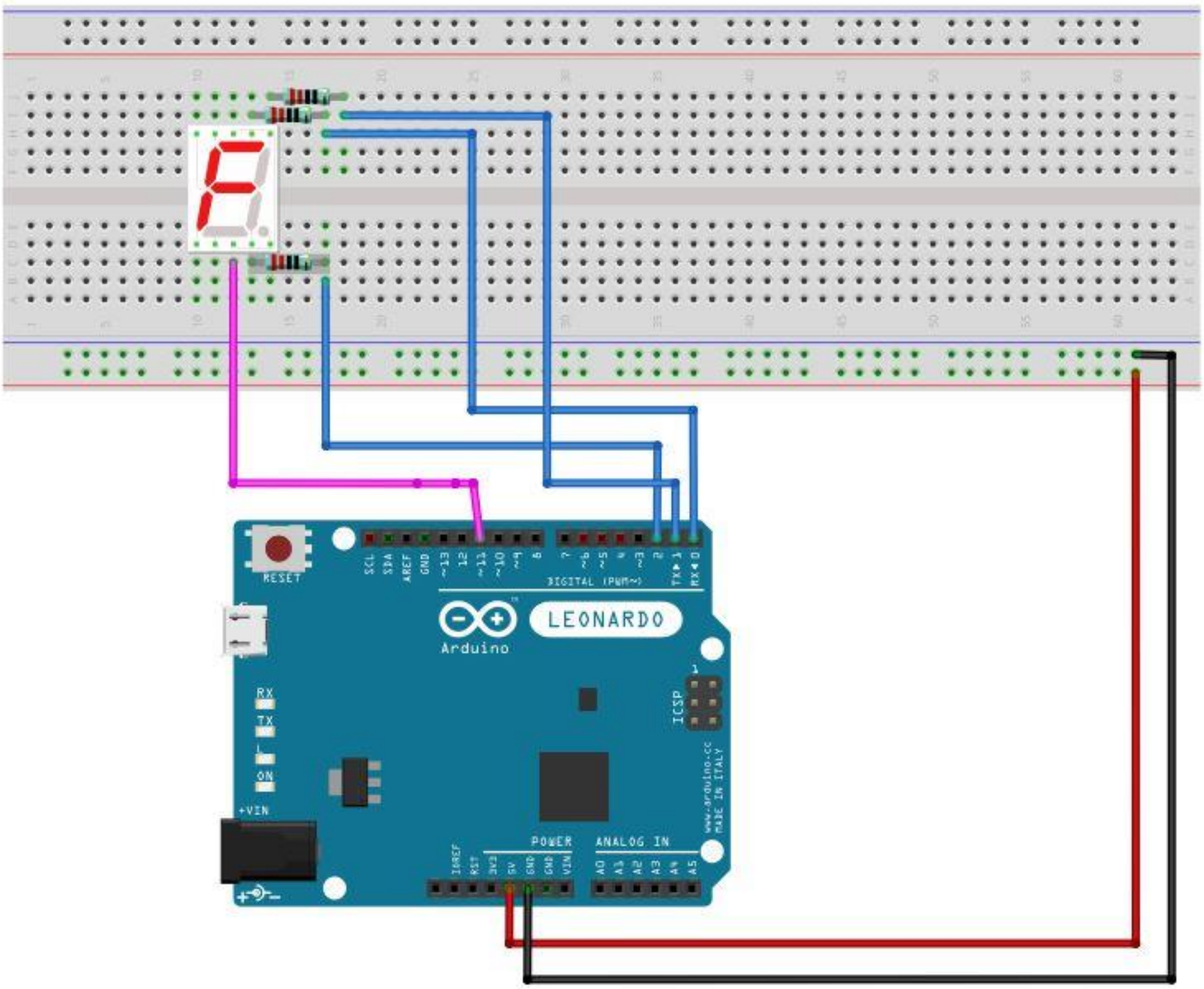
7-segment display: displays the number of available parking spaces (1-9)
Did parking allow the car to enter? 🟥 yes □ no

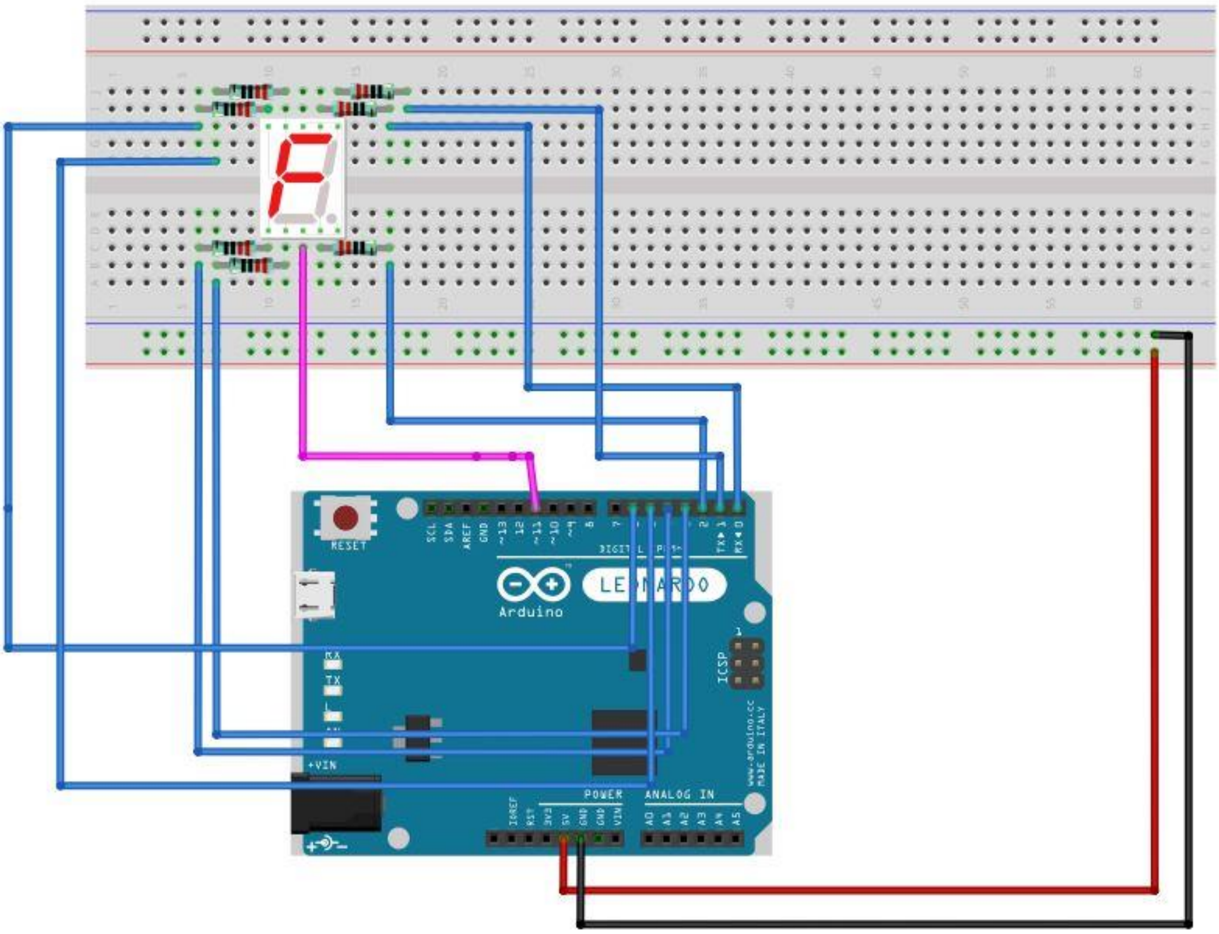
Testing – Fully occupied parking lot

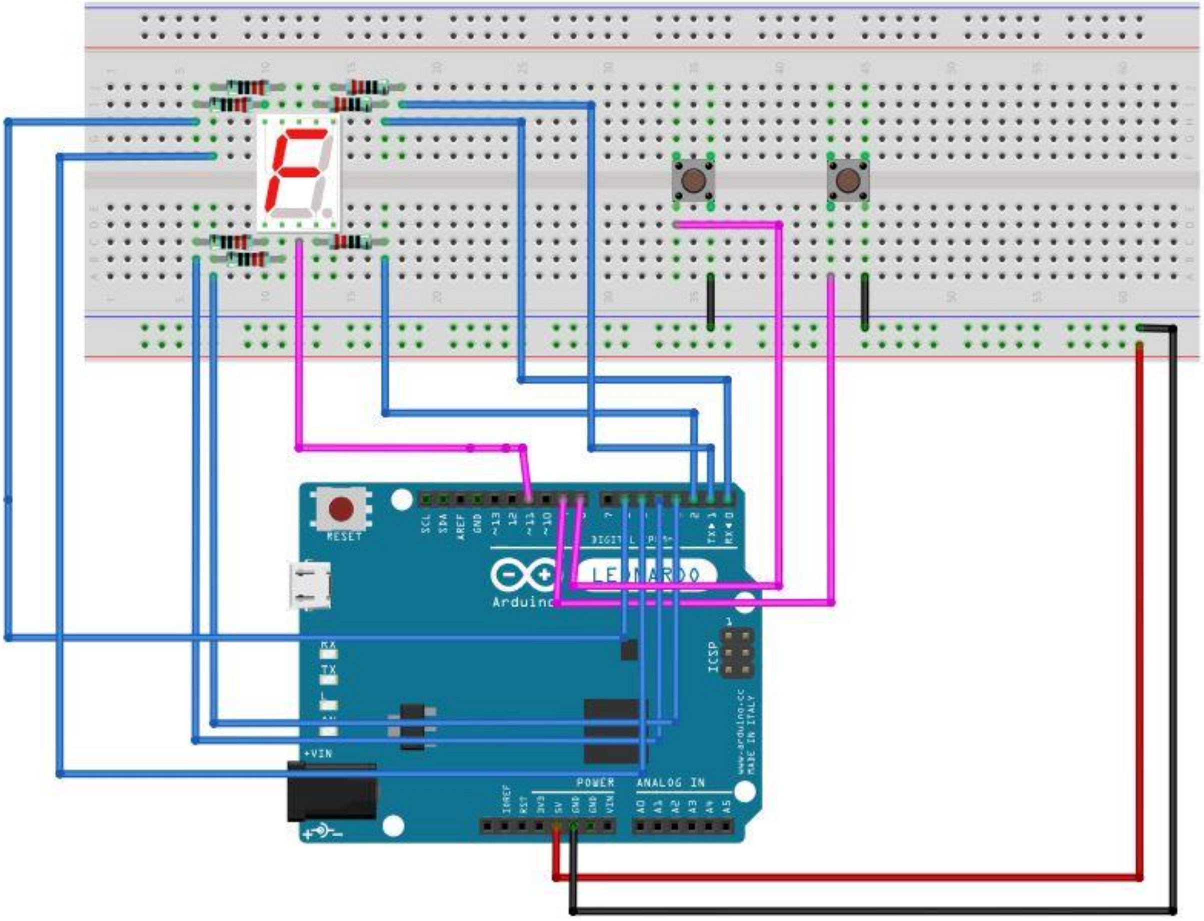
Record the status of:

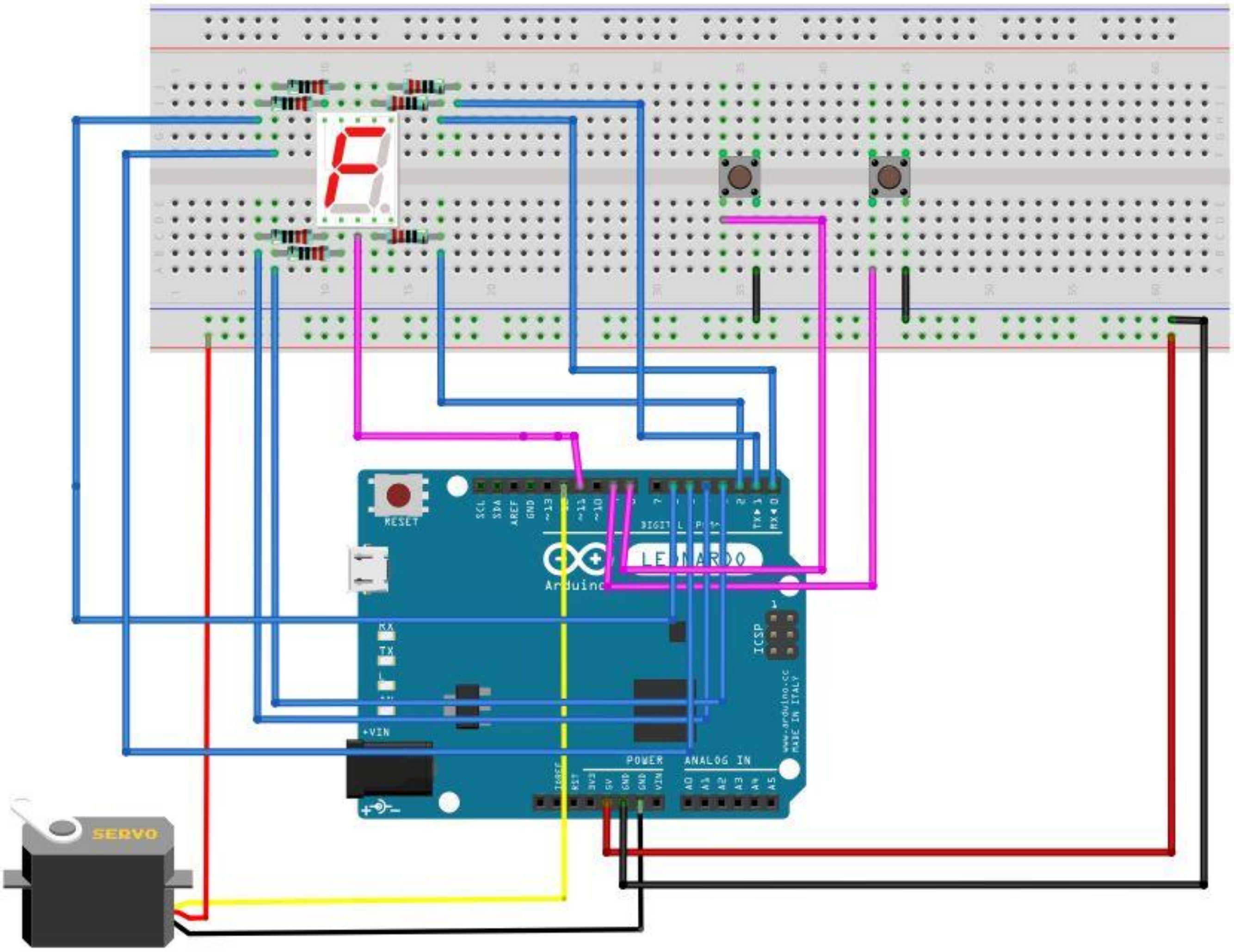
7-segment display: displays the number 0
Did parking allow the car to enter? □ yes 🟥 no

Step by step guide to build the circuit


RESET
SCL
SDA
AREF
GND
~13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM~)
LEONARDO
Arduino
ICSP
RX
TX
L
ON
+VIN
www.arduino.cc
MADE IN ITALY
POWER
ANALOG IN
IOREF
RST
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5


RESET
SCL
SDA
AREF
GND
~13
12
~11
~10
~9
8
7
2
TX▶1
RX◀0
DIGITAL (PWM~)
LEONARDO
Arduino
RX
TX
L
ICSP
+VIN
www.arduino.cc
MADE IN ITALY
POWER
ANALOG IN
IOREF
RST
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5


RESET
SCL
SDA
AREF
GND
~13
12
~11
~10
~9
DIGITAL (PWM~)
TX▶1
RX◀0
LEONARDO
Arduino
RX
TX
L
ON
ICSP
+VIN
POWER
IOREF
RST
3V3
5V
GND
GND
VIN
ANALOG IN
A0
A1
A2
A3
A4
A5
www.arduino.cc
MADE IN ITALY


RESET
SCL
SDA
AREF
GND
~13
12
~11
~10
TX▶1
RX◀0
DIGITAL (PWM~)
LEONARDO
Arduino
RX
TX
L
ON
ICSP
www.arduino.cc
MADE IN ITALY
+VIN
POWER
ANALOG IN
IOREF
RST
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
SERVO

Project 7: energy efficient street lighting system

Activity 4.22 (4.23)

Following the design process steps, ‘define’ the project requirements.

Objective:
Build an energy efficient street lighting system using an LDR and IR sensor modules.

Constrain(s):
You should use the components in the resources list only.

Writing the code

1- Define the variables

- Define the pin number for the LDR.
- Define the pin number for each LED.
- Define the pin number for each IR sensor.

```
// define the pins number for the LDR
int LDR= 0;

// define the pins number for the LEDs
int LED1 =6;
int LED2 = 7;
int LED3 = 8;
int LED4 = 9;
int LED5 = 10;
int LED6 = 11;

// define the pins number for the IR sensors
int ir1=5;
int ir2 = 4;
int ir3 = 3;
int ir4 = 2;

//set the starting Excel row
int row = 0;
//set an initial the value for the IR sensors
int proxy1 =0;
int proxy2 =0;
int proxy3 =0;
int proxy4 =0;
```

2- Void setup

- Start a serial communication to be able to send the data to the monitor. Set the baud rate to 128000.
- Define the IR pins as INTPUT, using pinMode function.

- Define the LED pins as output, using pinMode function.

```
void setup() {
//serial communication- baud rate 128000
  Serial.begin(128000);
  Serial.println("CLEARDATA"); //clears any residual data
  Serial.println("LABEL,Time,Pin,Light Level");

// define the IR pins as output
  pinMode(ir1, INPUT);
  pinMode(ir2, INPUT);
  pinMode(ir3, INPUT);
  pinMode(ir4, INPUT);

// define the LED pins as output
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
  pinMode(LED6, OUTPUT);
}
```

3- Void loop

- Take the readings from the LDR sensor.
- Take the readings from the IR sensors.
- Turn the LEDs when motion is detected by the IR sensors.

```
void loop() {
// readings from the LDR sensor
int ldrStatus = analogRead( LDR );

//readings from the IR sensors
proxy1 = digitalRead(ir1);
proxy2 = digitalRead(ir2);
proxy3 = digitalRead(ir3);
proxy4 = digitalRead(ir4);

//if it's night time
if (ldrStatus <= 1000) {

//if a car passing by the 1st IR sensor, turn LED1 and LED2 on
if (proxy1 == LOW ) {
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, LOW);
    digitalWrite(led4, LOW);
    digitalWrite(led5, LOW);
    digitalWrite(led6, LOW);
    }

//if a car passing by the 2nd IR sensor, turn LED2 and LED3 on
if (proxy2 == LOW ) {
    digitalWrite(LED1, LOW );
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, LOW);
    digitalWrite(LED6, LOW);
    }
//if a car passing by the 3rd IR sensor, turn LED3 and LED4 on
if (proxy3 == LOW) {
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
    digitalWrite(LED5, LOW);
    digitalWrite(LED6, LOW);
    }

//if a car passing by the 4th IR sensor, turn LED4, LED5 and LED6 on
if (proxy4 == LOW){
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, HIGH);
    digitalWrite(LED5, HIGH);
    digitalWrite(LED6, HIGH);
    }

//send data to the monitor
Serial.print('DATA,TIME,'); Serial.print(ldrPin); Serial.print(','); Serial.println(ldrStatus);
row++;
delay(100);
}
//else all lights are off
else {
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, LOW);
    digitalWrite(LED6, LOW);
  }
//send data to the
Serial.print('DATA,TIME,'); Serial.print(ldrPin); Serial.print(','); Serial.println(ldrStatus);
row++;
delay(100);
}
```

Testing – energy efficient street lighting system

Testing – Day time

Record the status of the LEDs

All LEDs are off

Testing – Night time

Cover the LDR sensor with your finger. Record the status of the LEDs when:

- Car passes by IR_1:
  LED_1: ■ on □ off
  LED_2: ■ on □ off
  LED_3: □ on ■ off
  LED_4: □ on ■ off
  LED_5: □ on ■ off
  LED_6: □ on ■ off
- Car passes by IR_2:
  LED_1: □ on ■ off
  LED_2: ■ on □ off
  LED_3: ■ on □ off
  LED_4: □ on ■ off
  LED_5: □ on ■ off
  LED_6: □ on ■ off
- Car passes by IR_3:
  LED_1: □ on ■ off
  LED_2: □ on ■ off
  LED_3: ■ on □ off
  LED_4: ■ on □ off
  LED_5: □ on ■ off
  LED_6: □ on ■ off
- Car passes by IR_4:
  LED_1: □ on ■ off
  LED_2: □ on ■ off
  LED_3: □ on ■ off
  LED_4: ■ on □ off
  LED_5: ■ on □ off
  LED_6: ■ on □ off

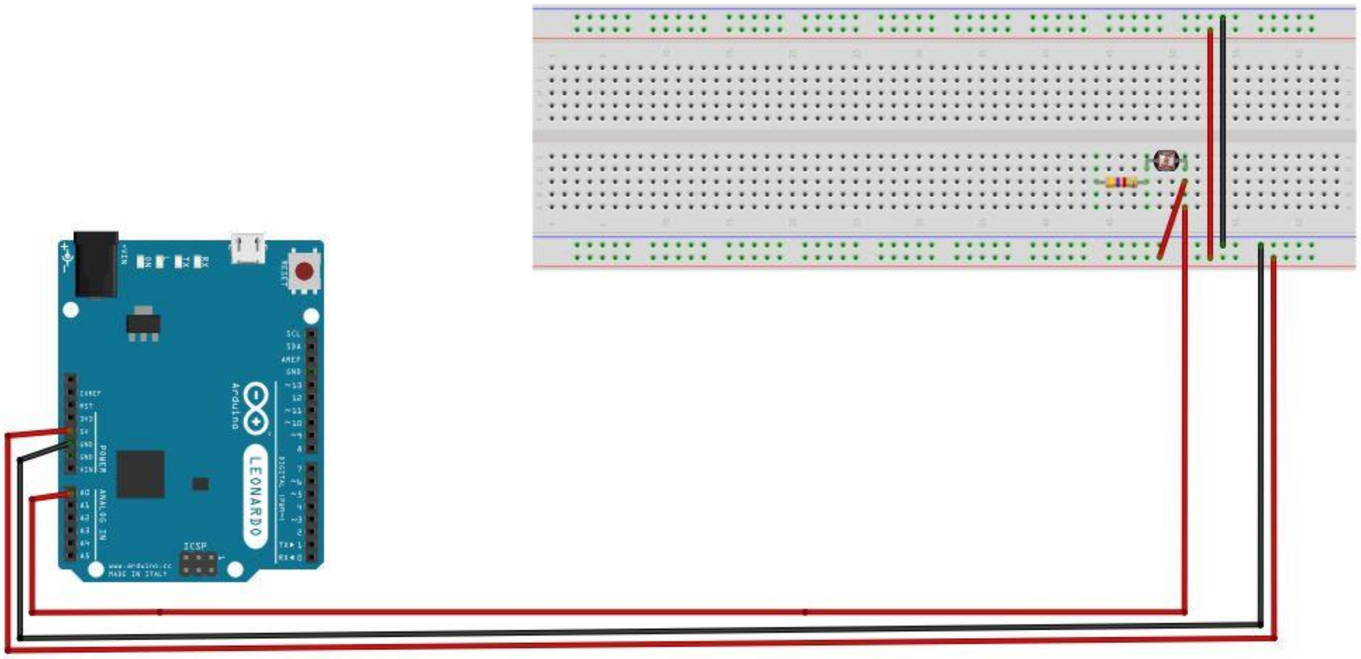
Critical thinking

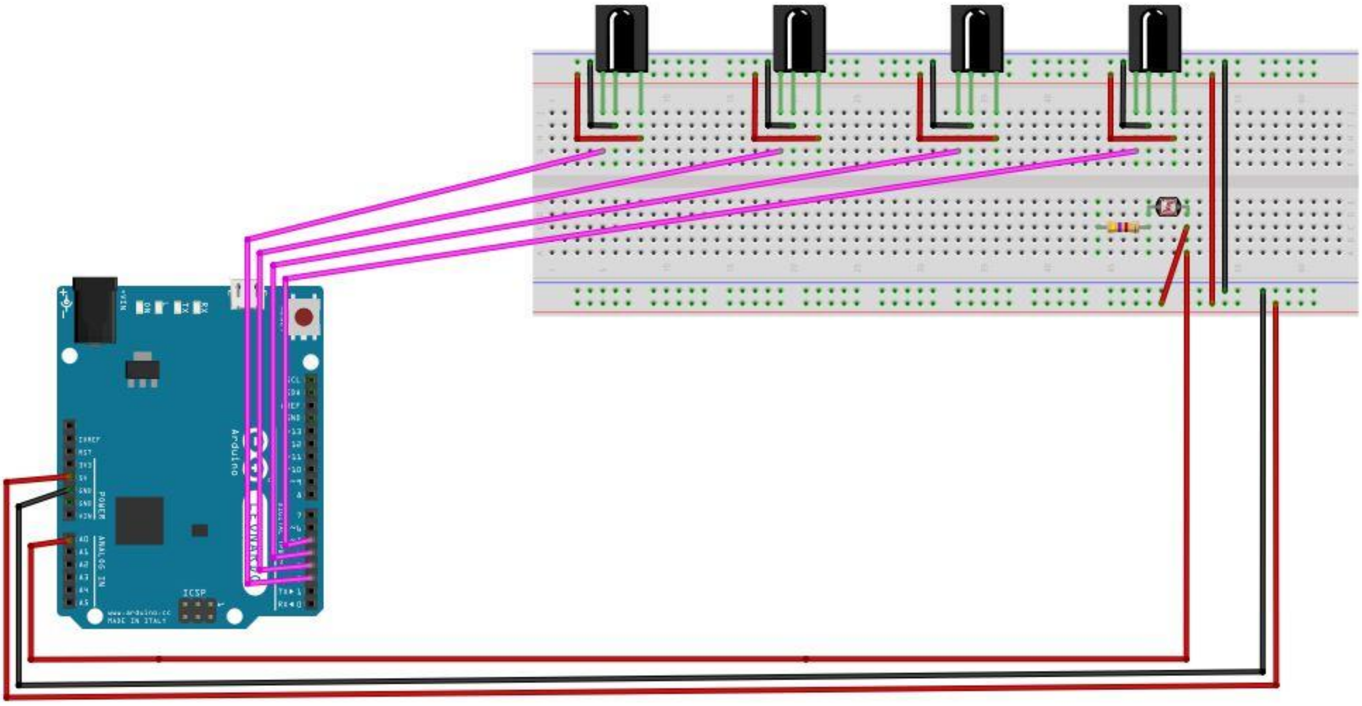
Did you notice that in the project only one IR sensors works at a time? When two cars pass by the IR sensors at two different speeds the system will not work properly. Think how you can modify the Arduino to solve this problem.

You will need to add more "if" statements studying the cases where more than one IR sensor module senses motion. For example, if the street was congested with cars, then probably all IR sensor modules will detect motion at the same time. Hence, the Arduino code for that scenario is going to be as follows::

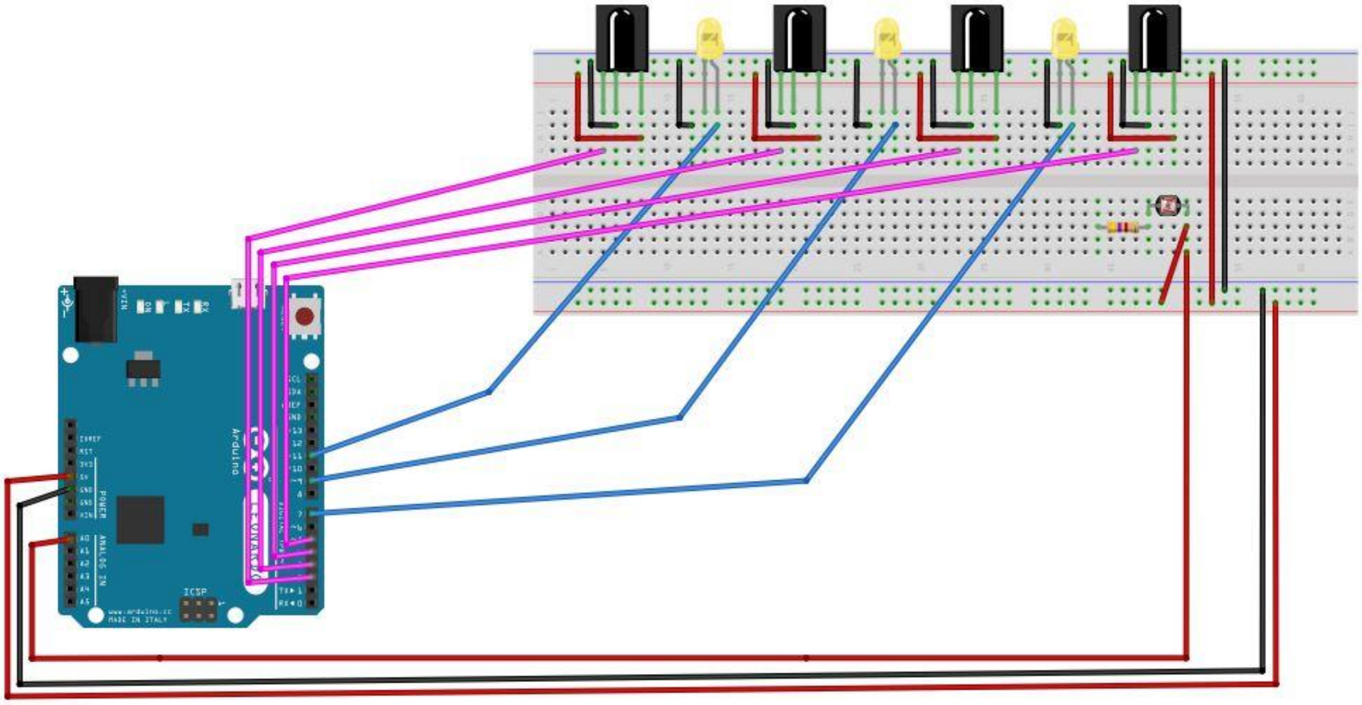
```
if (proxy1 == LOW && proxy2 == LOW && proxy3 == LOW && proxy4 == LOW){
   digitalWrite(led1, HIGH);
   digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);
digitalWrite(led5, HIGH);
digitalWrite(led6, HIGH);
}
```

Step by step guide to build the circuit

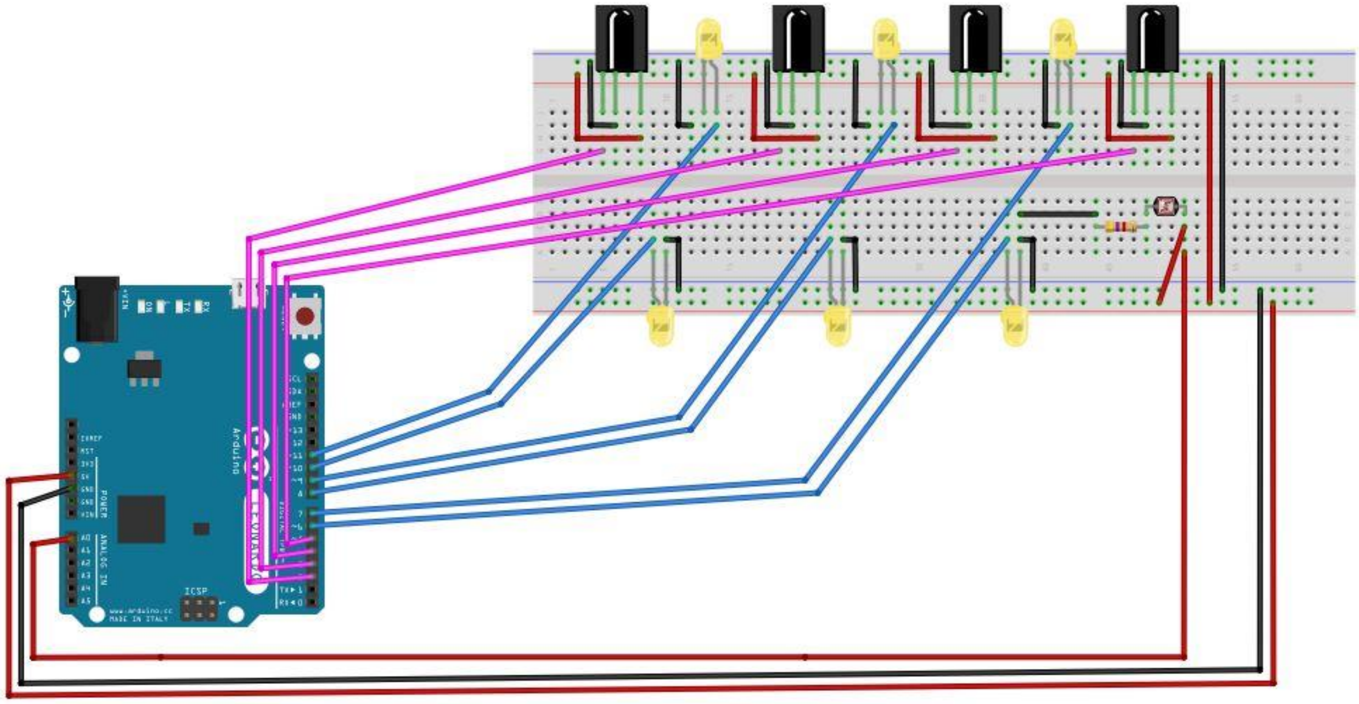




VIN
ON
L
TX
RX
IOREF
RST
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
Arduino
SCL
SDA
AREF
GND
13
12
11
10
9
8
7
6
TX▶1
RX◀0
ICSP
www.arduino.cc
MADE IN ITALY



VIN
ON
L
TX
RX
IOREF
RST
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
Arduino
SCL
SDA
AREF
GND
13
12
11
10
9
8
7
6
TX▶1
RX◀0
ICSP
www.arduino.cc
MADE IN ITALY



Activity 4.22 (4.24)

Following the design process steps, ‘define’ the project requirements.

Objective:
Build an adaptive building envelope using DC motor.

Constrain(s):
You should use the components in the resources list only.

Testing the DC motor

```
void setup() {
// define pin 5 and pin 6 of the motor as OUTPUT
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
}

void loop() {
//Running the DC motor in the first direction:
digitalWrite(5,HIGH);
digitalWrite(6,LOW);

//add 2 seconds delay
delay(2000);

//Running the DC motor in the opposite direction:
digitalWrite(5,LOW);
digitalWrite(6,HIGH);

//add 2 seconds delay
delay(2000);
}
```

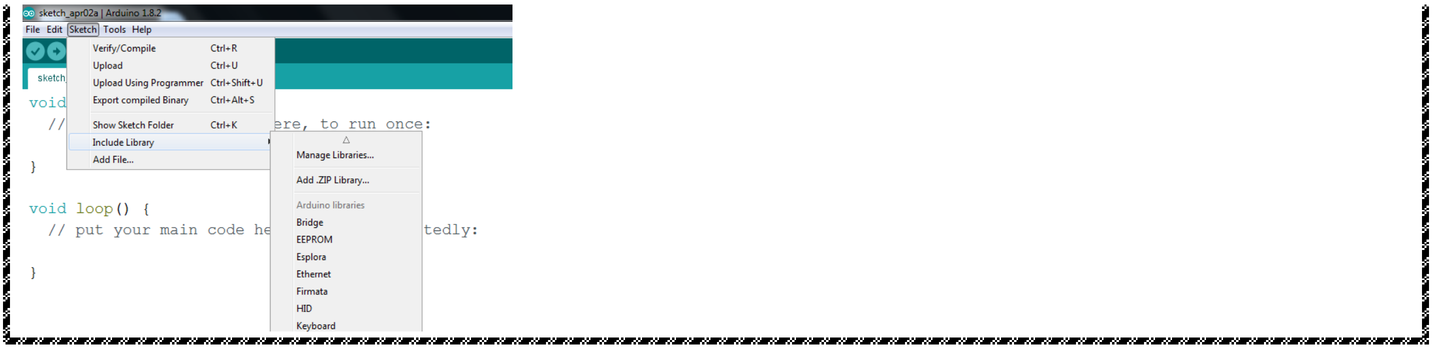
Note:

Before running the system’s code you must include “VirtualWire” and “dht” libraries.

Steps to include a library:

1- download the library’s zip file from sharepoint (you can find it on google as well)

2- go to sketch → include library → add .ZIP library



```
void
  //                                  ere, to run once:

}

void loop() {
  // put your main code he                tedly:

}
```

Testing – adaptive building envelope

Testing – Normal temperature

Record the LCD readings:
Temperature: 28.2
Humidity:45.32

Status of the DC motor: rotates for 3 seconds or 6 seconds based on the temperature value, in this case the motor moves for 6 seconds.

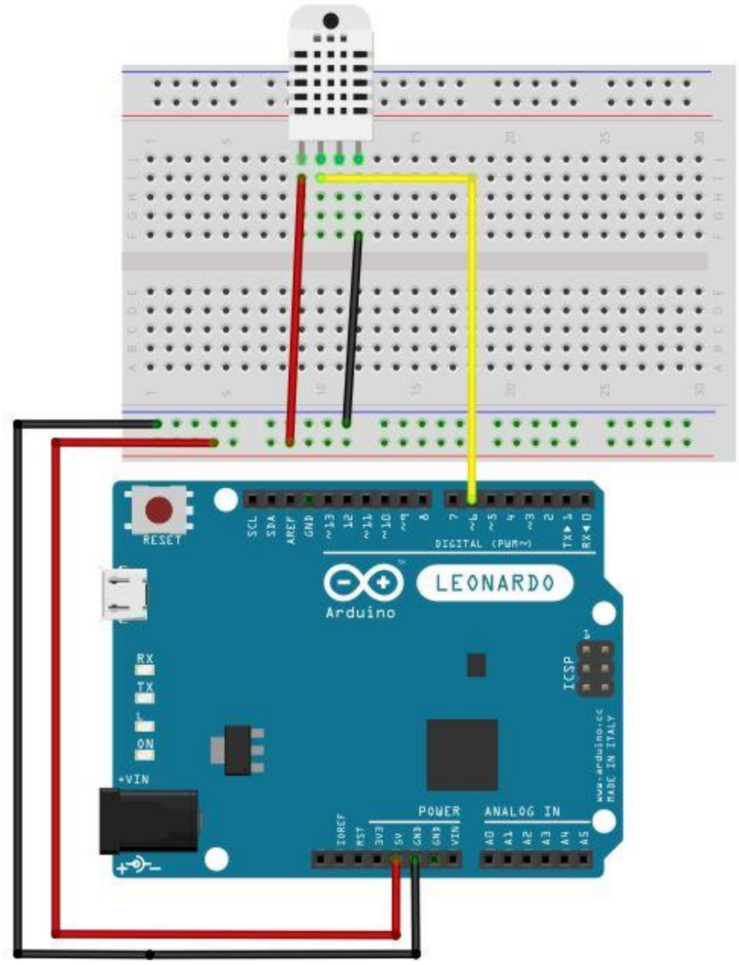
If temperature < 24 → motor runs for 3s
If temperature < 30 → motor runs for 6s

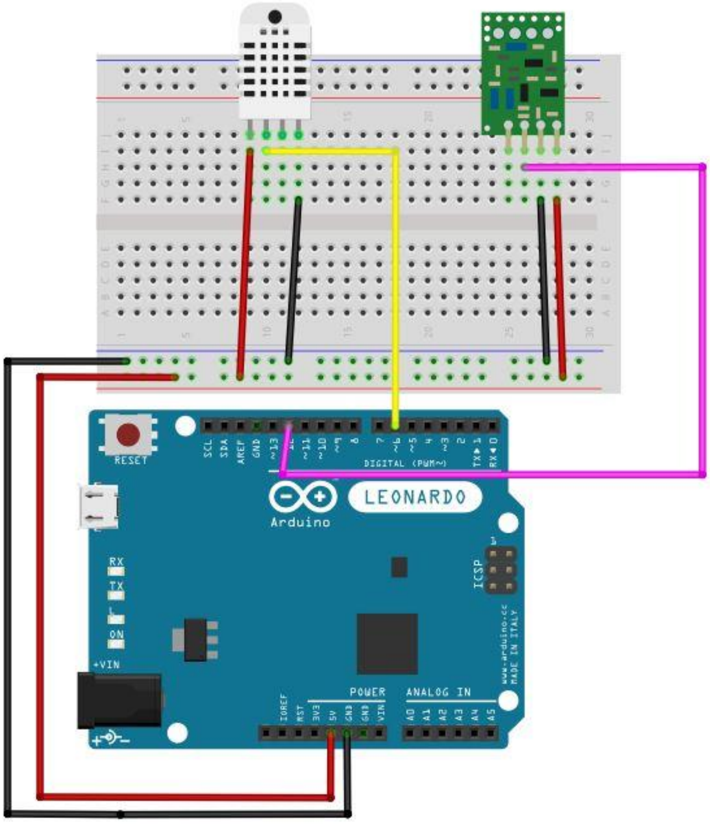
Testing – High temperature

Record the LCD readings:
Temperature: 39.21
Humidity:87.33

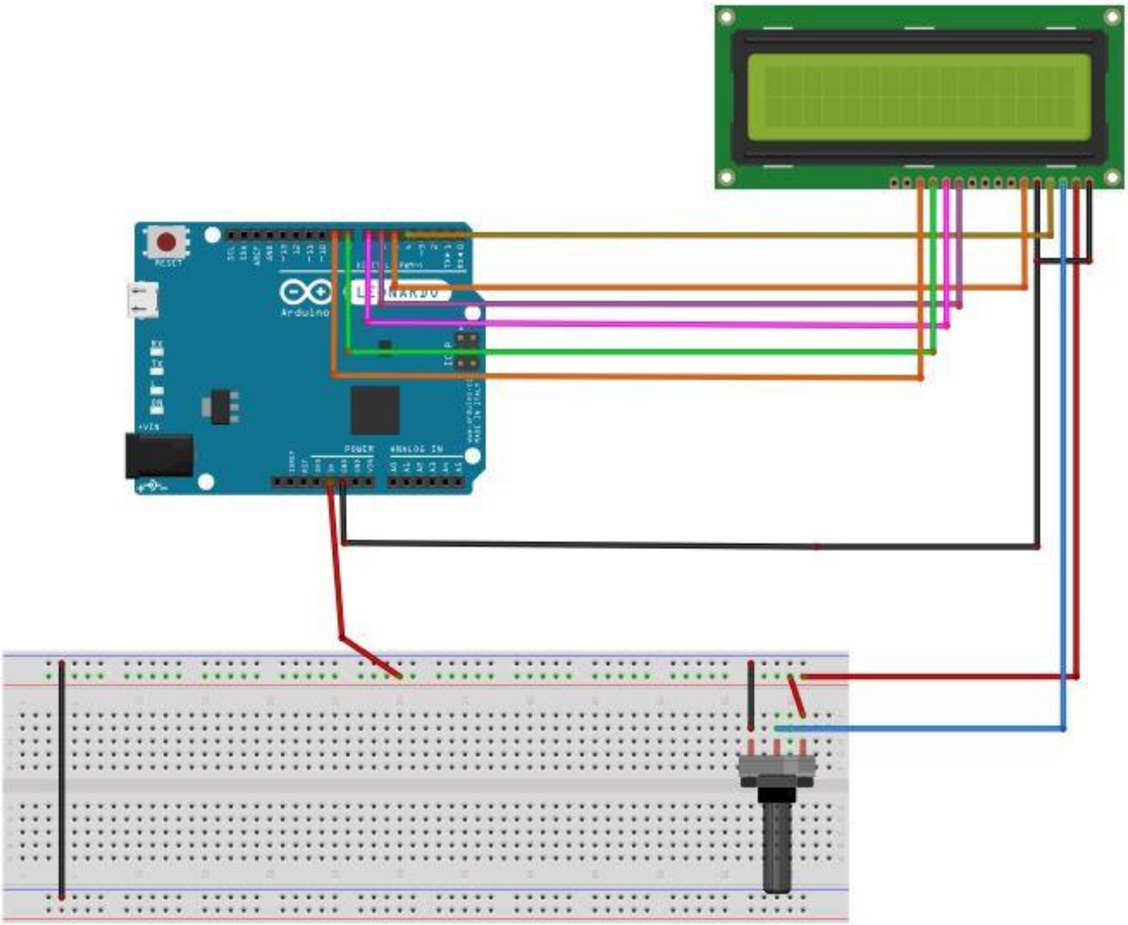
Status of the DC motor: off

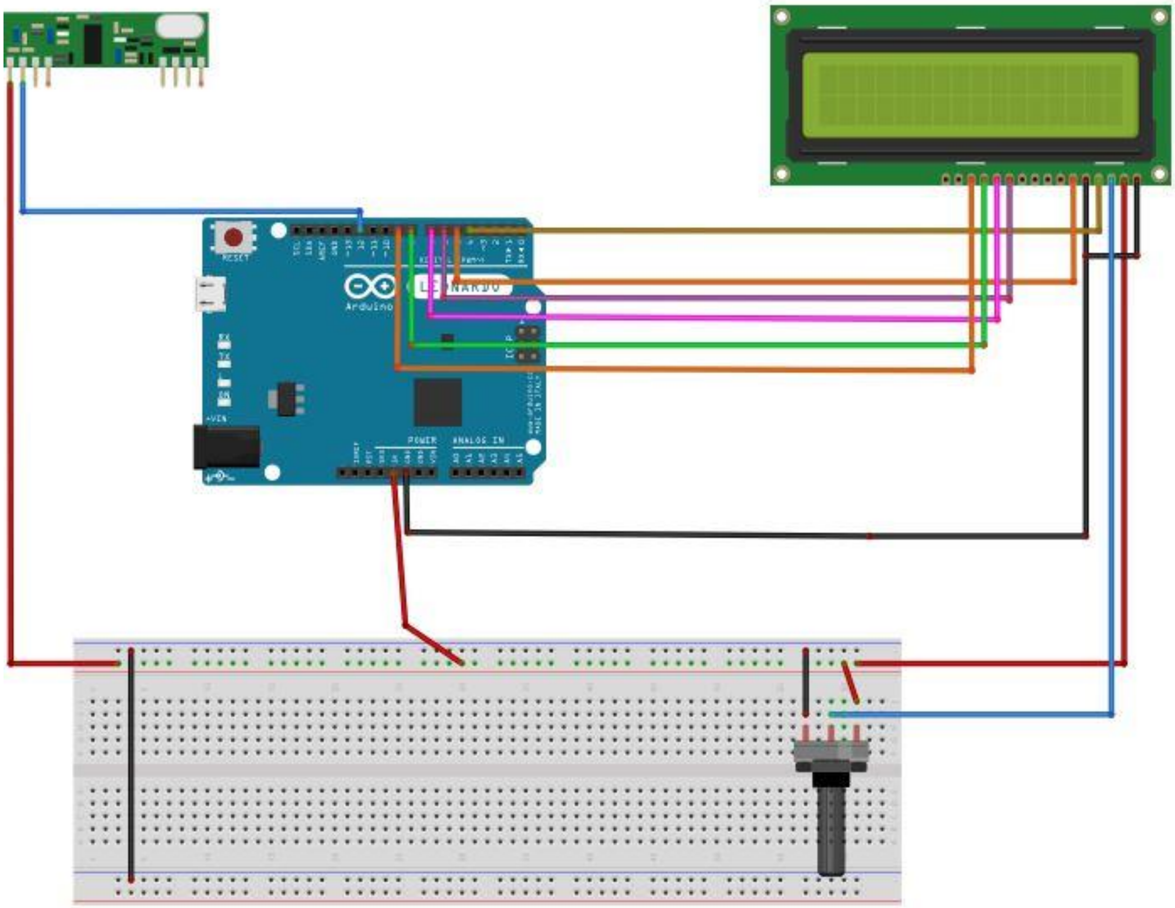
Step by step guide to build the circuit – transmitter

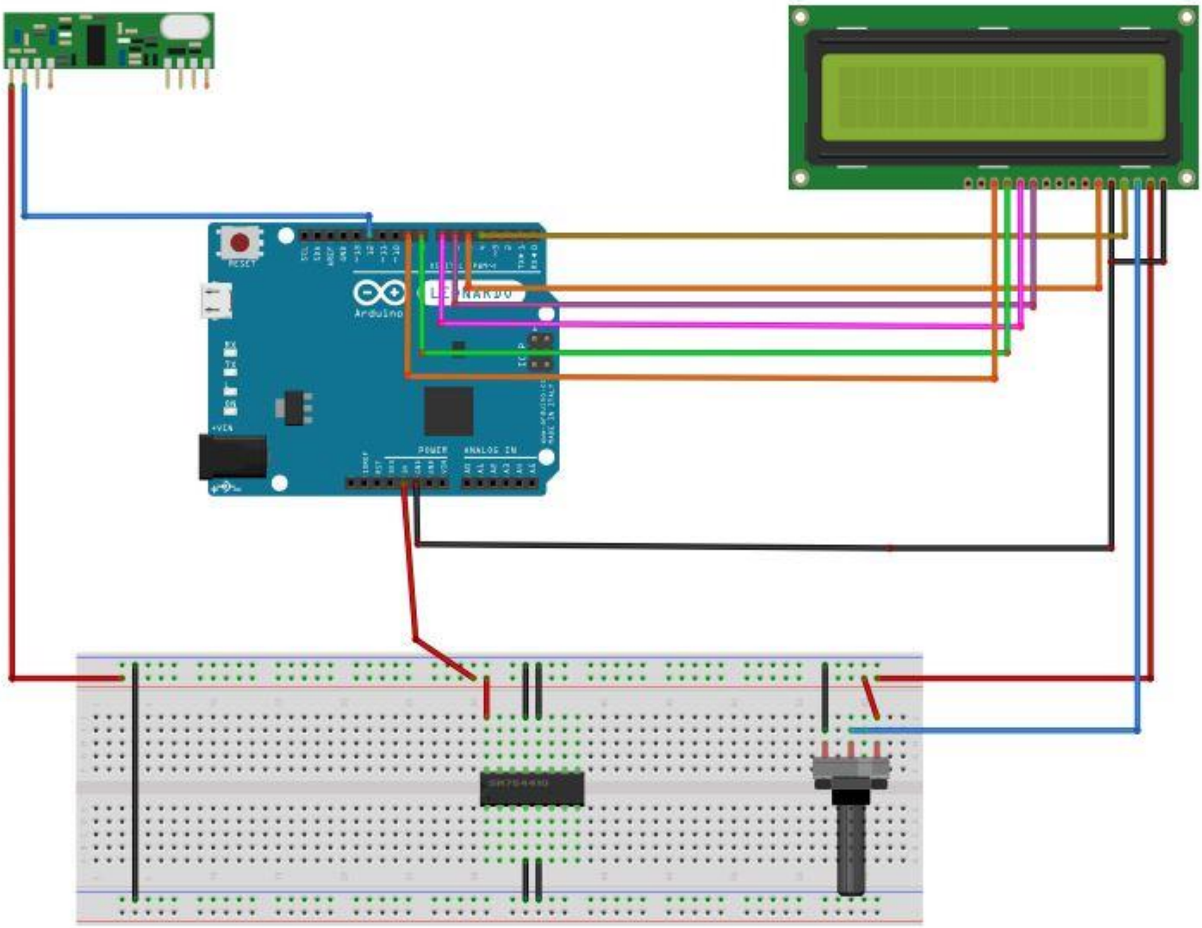


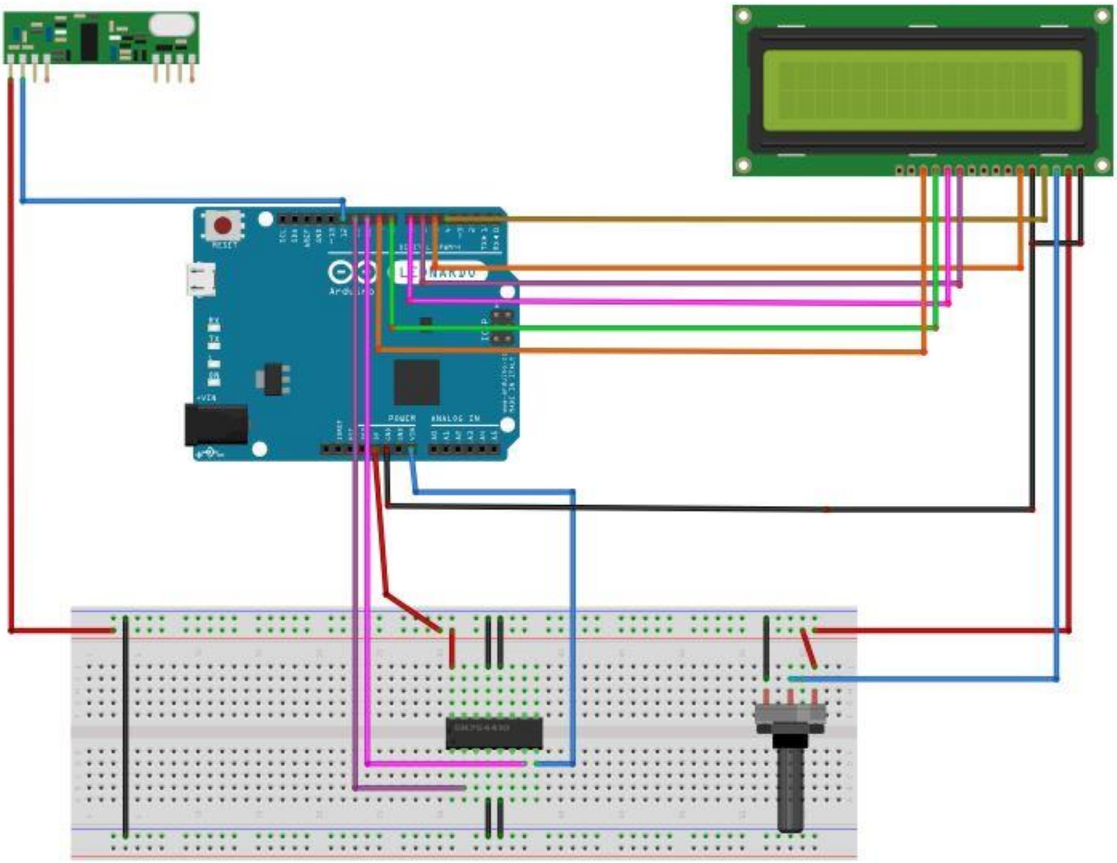


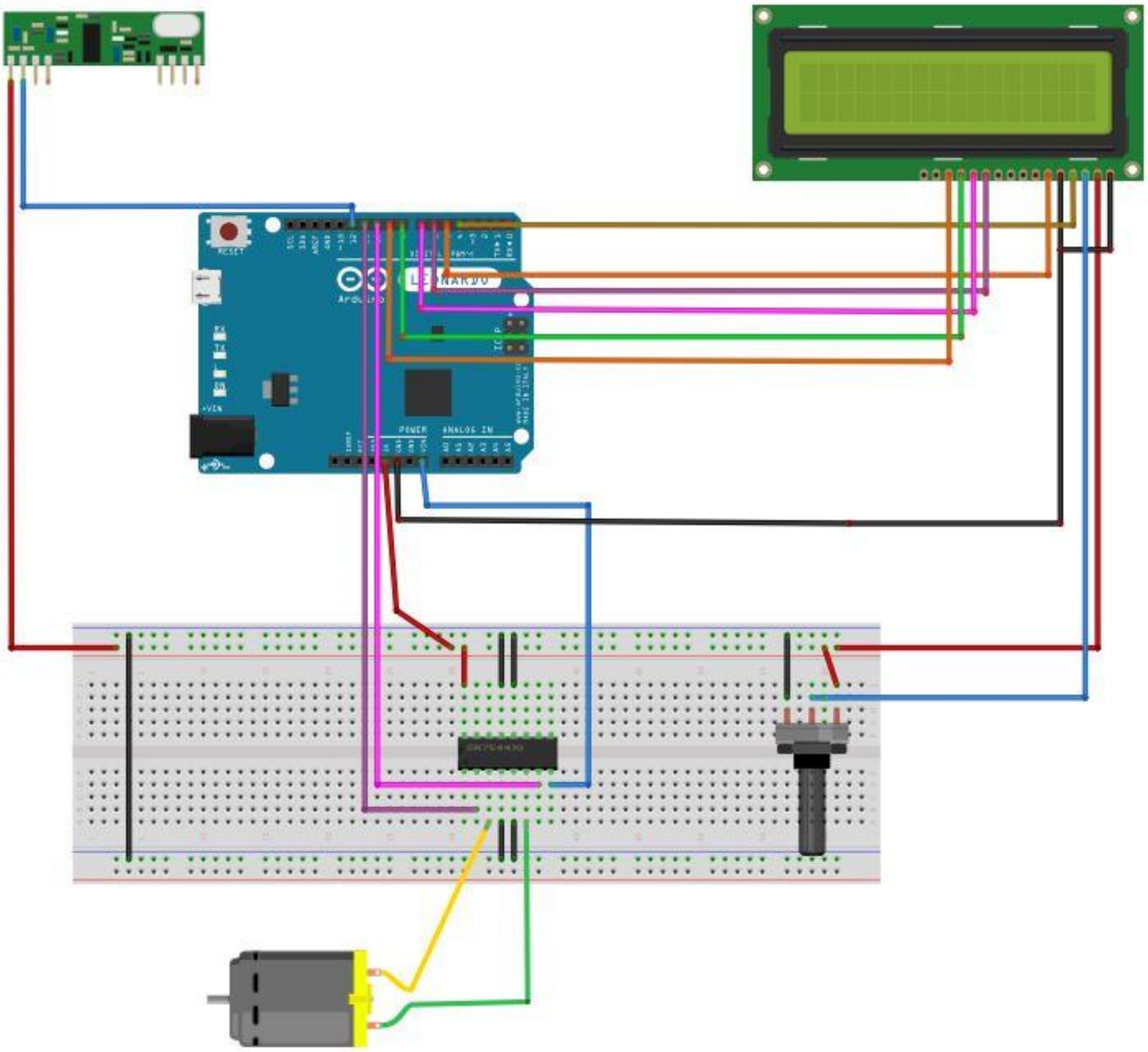
Step by step guide to build the circuit – receiver




RESET
Arduino
POWER
ANALOG IN


RESET
Arduino
LEONARDO
POWER
ANALOG IN
SN754410


RESET
Arduino
LEONARDO
POWER
ANALOG IN
SN754410


RESET
LEONARDO
Arduino
POWER
ANALOG IN