

# All-in-one Starter Kit for Micro:bit

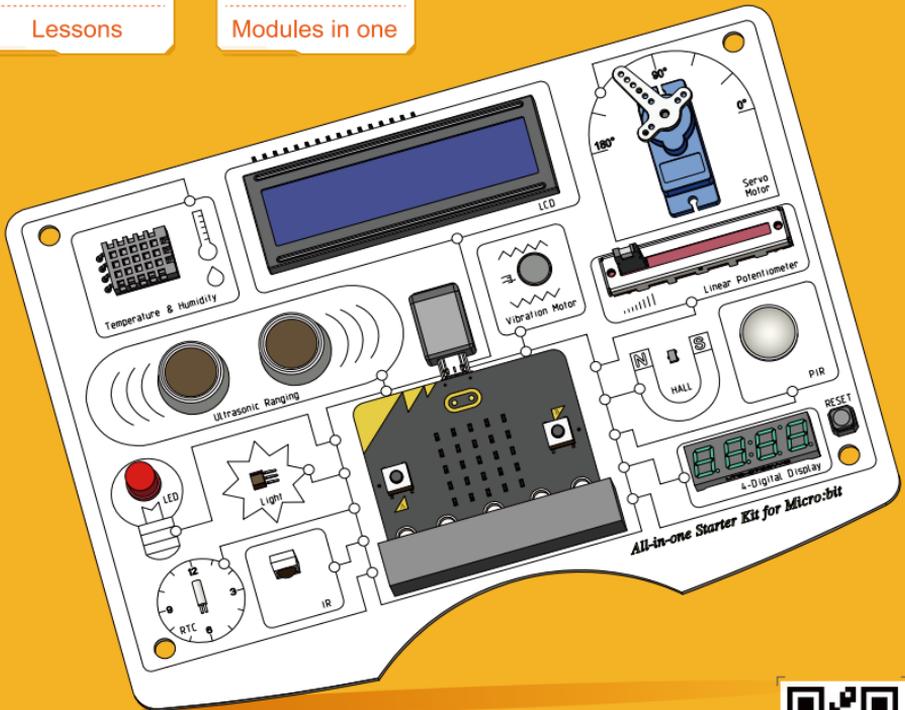
- STEAM Education
- Open-source Hardware

**20+**

Lessons

**13**

Modules in one



# Introduction

Welcome to the User Manual for the All-in-one Starter Kit for Micro:bit. Let's begin our journey into the world of the Micro:bit development board and its sensors.

This development board comes with 22 well-designed courses that gradually increase in challenge, engagement, and creativity. These courses will guide you step-by-step to master essential knowledge. Throughout this learning journey, you will become familiar with various electronic modules, develop logical thinking skills, enhance your creative design abilities, and implement module functions through programming.

The learning process starts with an introduction to the Micro:bit development board and its different sensors. You will then explore how to program these sensors and apply them in practical projects. Each step is clearly explained, making it easy for beginners to quickly get started with programming in a user-friendly environment. The All-in-one Starter Kit for Micro:bit includes 13 electronic modules, each with its own unique features and functions. These modules are specially designed for beginners and are an excellent choice for learning and experimentation. For example, the light sensor enables beginners to control real-world lighting devices through programming.

In summary, by using this development board, you will learn the basic knowledge and working principles of sensors, understand important concepts such as digital and analog signals, analog-to-digital conversion, and programming logic, and master the usage of some complex electronic modules. Most importantly, through Micro:bit programming, you will further strengthen your logical thinking skills.

For the programming software, we will use the official Micro:bit online programming platform provided on the Micro:bit website. No software installation is required, making it simple and convenient for beginners to start programming directly online.

**\*Note: Turn off the power before inserting the Micro:bit**

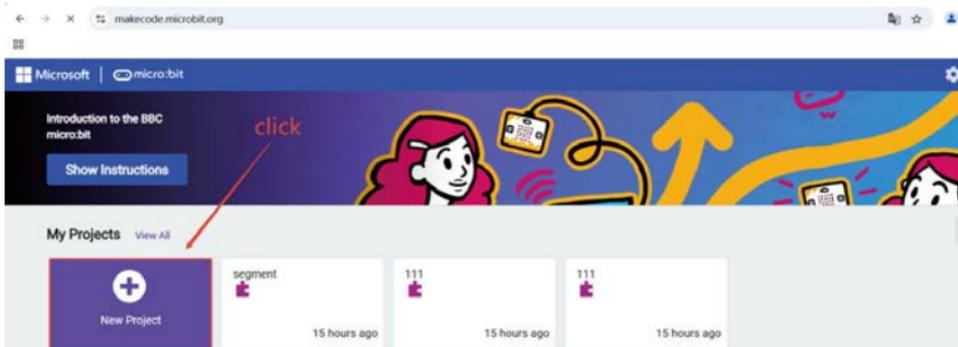
## List of Sensors

• Ultrasonic Ranging Sensor	x1
• Light Sensor	x1
• Linear Potentiometer	x1
• LED	x1
• Temperature& Humidity Sensor	x1
• LCD	x1
• Infrared Remote	x1
• Servo Motor	x1
• RTC	x1
• Vibration Motor	x1
• Hall Sensor	x1
• PIR	x1
• 4-Digit Display	x1

# Getting Started

## STEP 1:

Go to the Micro:bit website: <https://makecode.microbit.org/>



To create a project, enter a name and click "Create".

A screenshot of the 'Create a Project' dialog box. The title is 'Create a Project' with a smiley face emoji and a close button (X) in the top right. The main text says 'Give your project a name.' Below this is a text input field containing the word 'test'. There is a link '> Code options' below the input field. At the bottom right, there is a green button labeled 'Create' with a checkmark icon.

Enter the online editor.



# Lessons

## Lesson 1 - LED Control

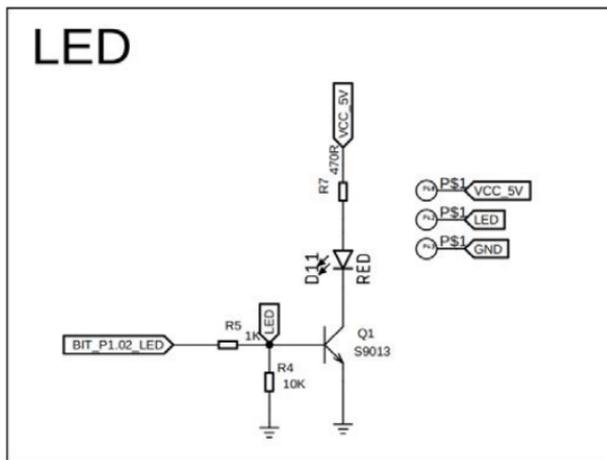
### Introduction

In this lesson, we will learn how to control LED lights. By manipulating the code, we can achieve the effect of the LED lights turning on and off at intervals.

### Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

### Hardware Schematic



### Code Explanation

The code consists of a "forever" loop containing four blocks:

- digital write pin P16 to 1**: This block is connected to the text "The LED lights up when the P16 pin is high".
- pause (ms) 1000**: This block is connected to the text "Wait 1000 milliseconds".
- digital write pin P16 to 0**: This block is connected to the text "The LED light is off when pin 16 is low".
- pause (ms) 1000**: This block is connected to the text "Wait 1000 milliseconds".

## What Will You See

---

After starting the course, you will see the LED light up for one second, then turn off for one second, and repeat this cycle. If this process is not observed, please ensure that the program is running correctly.

## Lesson 2 - Button Control LED

### Introduction

---

In this lesson, we will use the Micro:bit's built-in buttons to control output modules. The Micro:bit has two buttons labeled A and B.

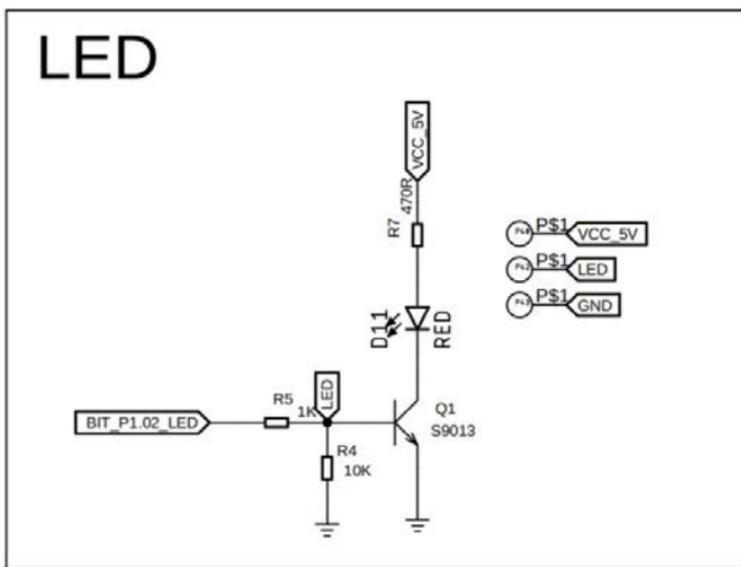
### Hardware Required

---

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

### Hardware Schematic

---



### Code Explanation

---

The image shows a Scratch code block for a Micro:bit. It consists of a 'forever' loop containing an 'if' block. The 'if' block has a condition 'button A is pressed'. If true, it executes 'digital write pin P16 to 1'. If false, it executes 'digital write pin P16 to 0'.

Annotations on the right side of the code block:

- Forever Loop
- If the onboard button A on the Micro:bit is pressed
- The LED lights up when the P16 pin is high
- "else" here means button A is not pressed
- The LED light is off when pin 16 is low

## What Will You See

After uploading the code, pressing button A on the Micro:bit will turn on the LED. Releasing the button will turn the LED off.

## Lesson 3 - Breathing LED

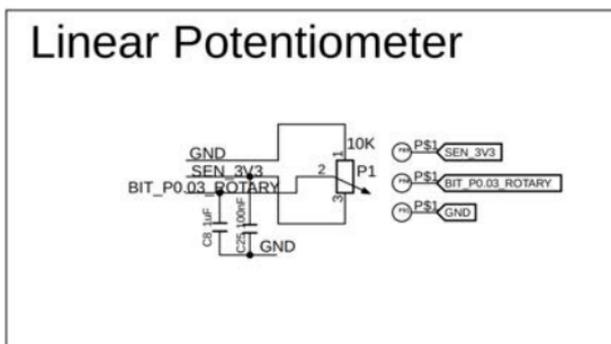
### Introduction

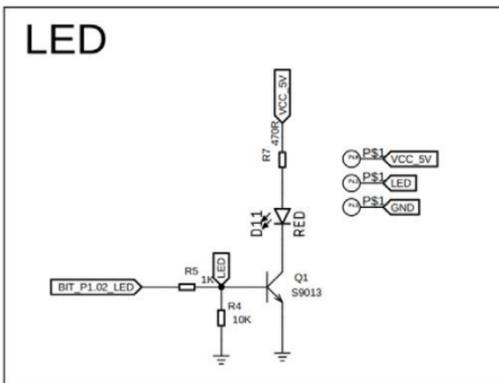
In this lesson, we will use a potentiometer with a maximum resistance of 10k $\Omega$  to achieve the breathing effect of the LED. As you slide it from left to right, its output voltage will range from 0V to 5V (VCC). In this session, we will adjust the LED using the potentiometer to create the breathing light effect!

### Hardware Required

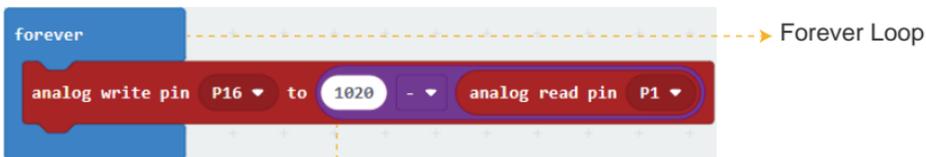
- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

### Hardware Schematic





## Code Explanation



The sliding rheostat outputs an analog value between 0 and 1020. A higher resistance means the LED will be dimmer. So we need to write:1020 - the value from the sliding rheostat

## What Will You See

After uploading the code, the LED brightness will change based on the position of the potentiometer. Turning it all the way to the left (minimum position) will make the LED the brightest. Turning it all the way to the right (maximum position) will turn the LED off.If the LED doesn't respond as expected, please check whether the program is running correctly.

# Lesson 4 - LCD Display

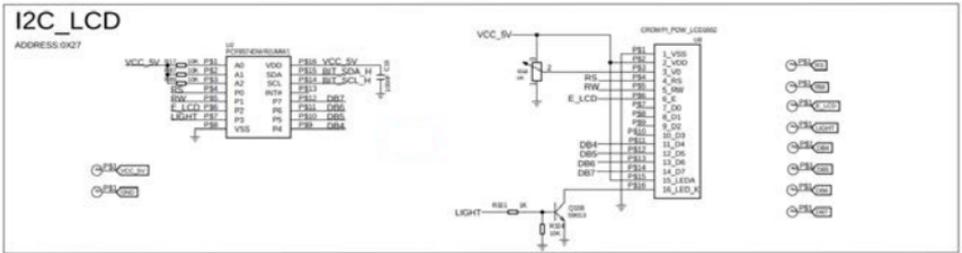
## Introduction

In this lesson, we will learn how to display text on an LCD screen. By the end of this lesson, you will understand the basic principles of using an LCD, know how to control the display content with code, and be well-prepared for more advanced projects.

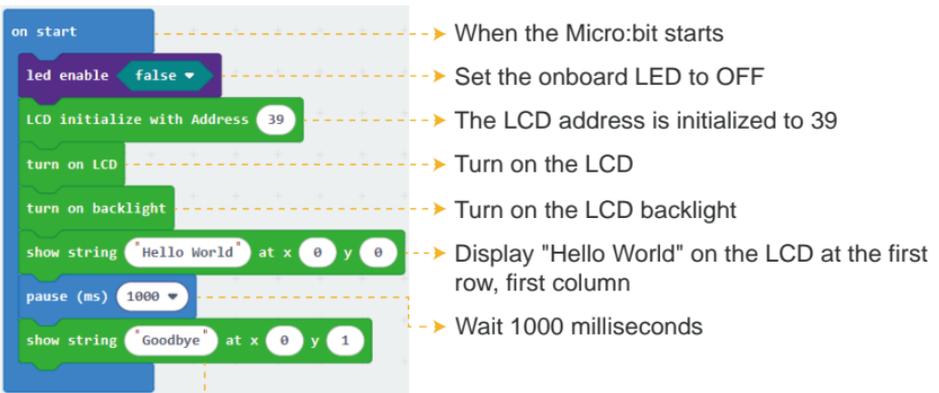
## Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

## Hardware Schematic



## Code Explanation



Display "Goodbye" on the LCD at the second row, first column

## What Will You See

After uploading the program, the LCD will display "Hello World" on the first line. After one second, "Goodbye" will appear on the second line. In the end, the LCD will turn off and the screen will go blank.

# Lesson 5 - Intelligent Street Light

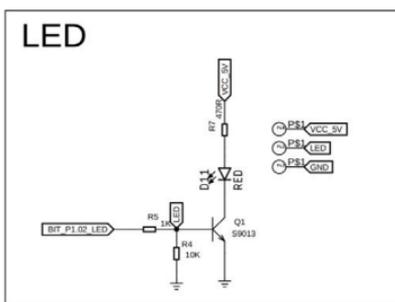
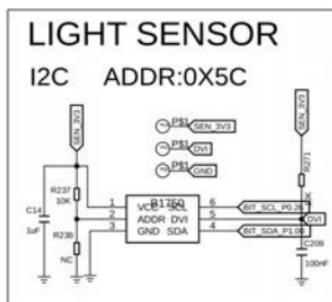
## Introduction

In this lesson, you will learn how to obtain light intensity information from the light sensor module and how to control the LED light based on this data. By determining different levels of brightness, you can achieve intelligent control of the LED's on/off state. This smart control of the LED allows you to avoid unnecessary energy consumption, thereby achieving energy-saving goals.

## Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

## Hardware Schematic



## Code Explanation

	→ When the Micro:bit starts
	→ Set the onboard LED to OFF
	→ The light sensor address is initialized to 92
	→ Example: Initializing the BH1750
	→ Forever Loop
	→ Get the ambient light intensity and save it to "light2"
	→ If the value of the variable "light2" is less than 100, the ambient light intensity is too low
	→ The LED lights up when the P16 pin is high
	→ If the conditions fail
	→ The LED light is off when pin 16 is low

## What Will You See

When you cover the top of the light sensor with your hand to simulate a dark environment, the red LED will light up. When you remove your hand to simulate a light environment, the red LED will turn off. If this does not happen, please check whether the program has been executed correctly.

## Lesson 6 - Ultrasonic Ranging Display

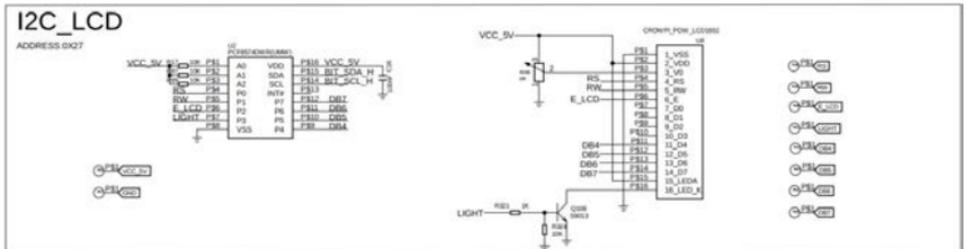
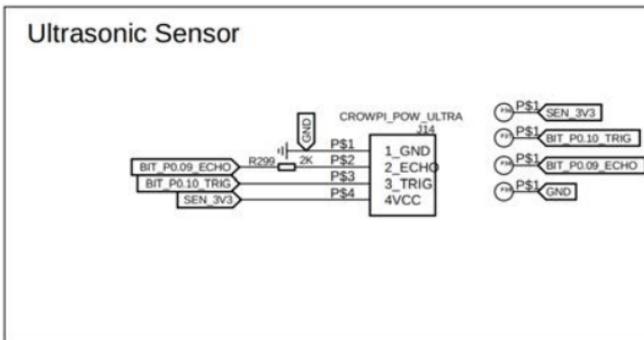
### Introduction

In this lesson, we will learn how to use the ultrasonic sensor module. With this module, we can measure the distance between the module and a flat surface in front of it. We can create an ultrasonic distance meter, and the measured distance value will be displayed on the LCD module.

### Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

### Hardware Schematic

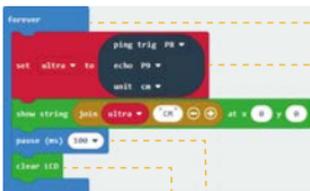


## Code Explanation

---



- When the Micro:bit starts
- Set the onboard LED to OFF
- The LCD address is initialized to 39
- Turn on the LCD



- Forever Loop
- Initialize the pins for the ultrasonic sensor and store its measured values in the variable "ultra"
- Display the value of the variable "ultra" on the LCD at the first row and first column
- Wait 100 milliseconds
- Clear the LCD display

## What Will You See

---

After starting the course, you will see the distance data measured by the ultrasonic sensor continuously are refreshed on the LCD screen. As the flat surface in front of the ultrasonic module moves, the measured distance values will also change accordingly. If this does not happen, please check whether the program is running correctly.

## Lesson 7 - Obstacle Close Range Alarm

### Introduction

---

In this lesson, we will delve deeper into the ultrasonic sensor module and learn how to integrate it with other modules. We will use the distance data obtained from the ultrasonic module to control the on/off state of the vibration motor module. In this way, we can achieve the effect of ultrasonic obstacle avoidance.

### Hardware Required

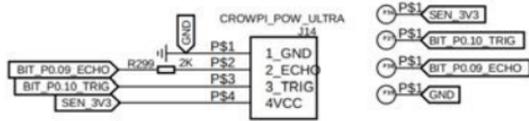
---

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

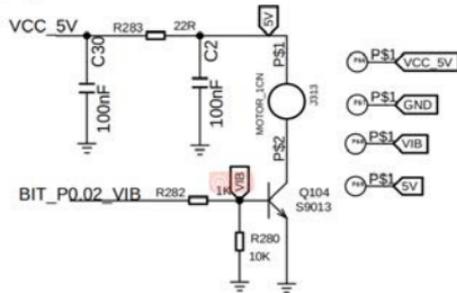
### Hardware Schematic

---

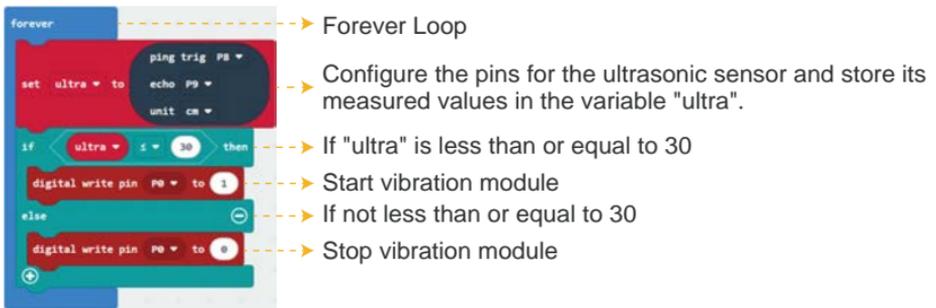
## Ultrasonic Sensor



## Vibration sensor



## Code Explanation



## What Will You See

After starting the course, you will observe that as the distance measured by the ultrasonic module changes, the vibration motor will activate when the distance is less than or equal to 30 centimeters, indicating that an obstacle has been detected. If the distance exceeds 30 centimeters, the vibration motor will turn off, indicating that the path is clear and there are no obstacles. If this process does not occur as described, please check whether the program is running correctly.

# Lesson 8 - Brightness Display

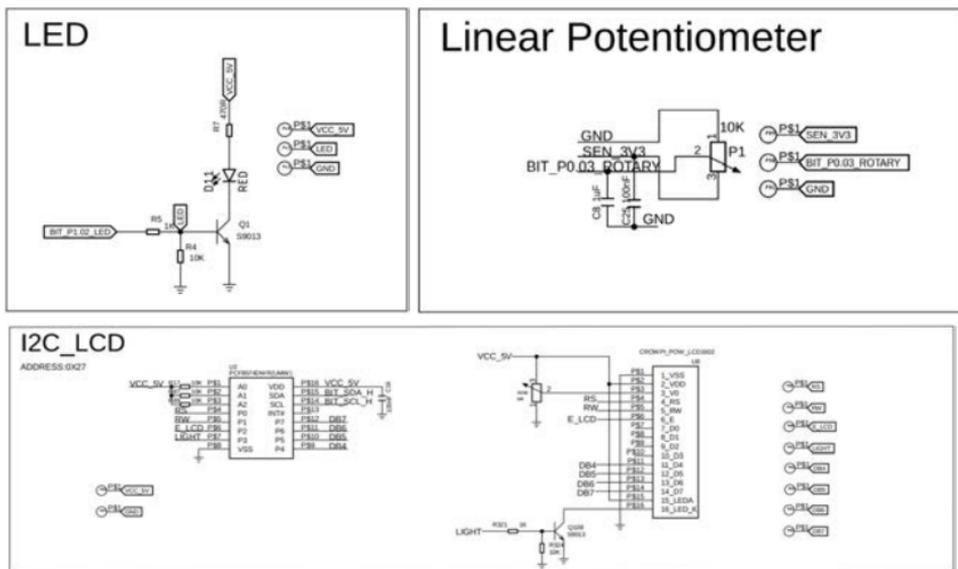
## Introduction

In this lesson, we will learn how to control the brightness of a light using a potentiometer and divide the brightness levels into 10 grades.

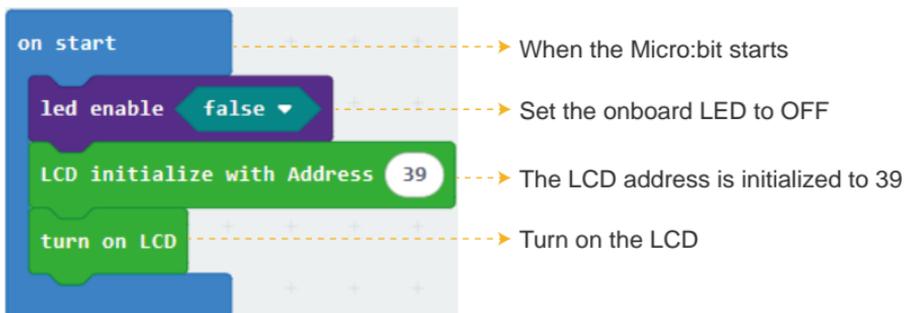
## Hardware Required

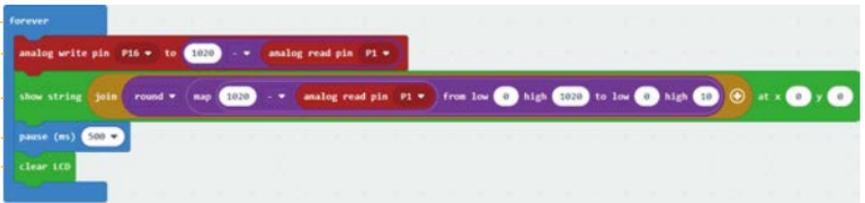
- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

## Hardware Schematic



## Code Explanation





➤ Forever Loop

➤ The sliding rheostat outputs an analog value between 0 and 1020. A higher resistance means the LED will be dimmer. So we need to write:1020 - the value from the sliding rheostat.

➤ Since the values we get are analog and range from 0 to 1020, we need to map them to a range of 0 to 10 and display the result as an integer on the LCD screen.

➤ Wait 500 milliseconds

➤ Clear LCD

## What Will You See

After starting the course, you will see that by sliding the potentiometer left or right, you can adjust the brightness level of the light from 0 to 10.

# Lesson 9 - Temperature&Humidity Detecting System

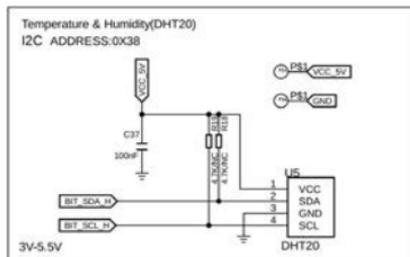
## Introduction

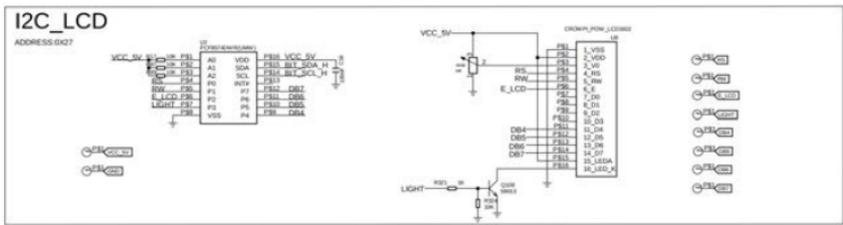
In this lesson, we will learn how to use the temperature and humidity sensor module.

## Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

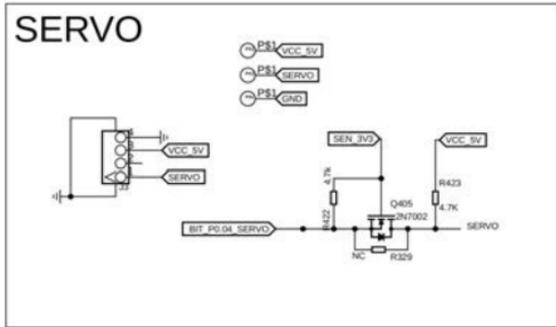
## Hardware Schematic



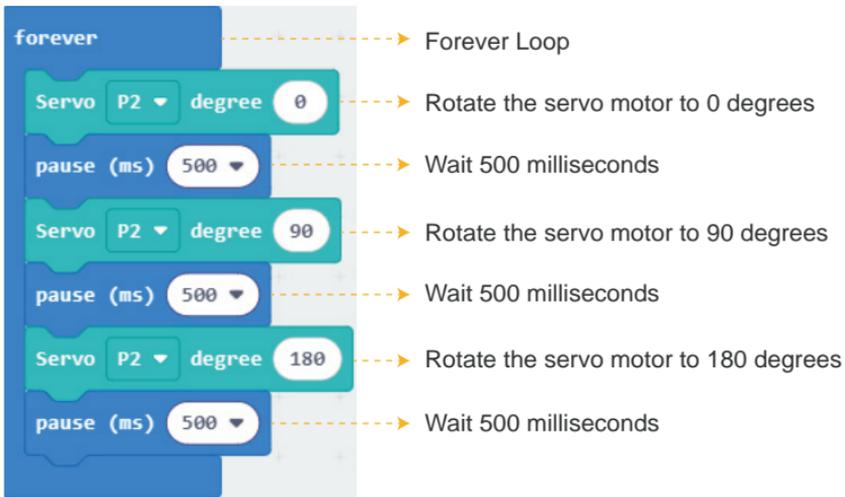


- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

## Hardware Schematic



## Code Explanation



## What Will You See

After starting the course, you will see the servo rotate from 0 degrees to 180 degrees, and then from 180 degrees back to 0 degrees. If this process does not occur, please check whether the firmware has been correctly flashed and whether the hardware connections are correct.

# Lesson 11 - IR Control LED

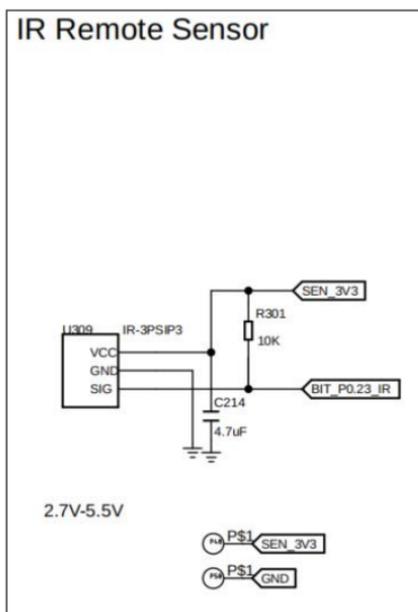
## Introduction

In this lesson, we will learn the basic operation of using an infrared remote control to control an LED.

## Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1
- Infrared Remote Control

## Hardware Schematic



## Code Explanation

The screenshot shows two code blocks in the Micro:bit IDE. The first block is a blue "on start" block. The second block is an orange "connect IR receiver at" block with a dropdown menu set to "P11".

on start → When the Micro:bit starts

connect IR receiver at P11 → Set pin P11 as the input for the IR receiver

forever

- set ir\_value to IR button → Store the value received from the IR sensor in the variable "ir\_value"
- if ir\_value = 12 then → If the returned value equals 12
  - set num to 1 → Set the value of the variable "num" to 1
- else if ir\_value = 24 then → If the returned value equals 24
  - set num to 2 → Set the value of the variable "num" to 2
- else if ir\_value = 94 then → If the returned value equals 94
  - set num to 3 → Set the value of the variable "num" to 3
- if num = 1 then → If the variable "num" equals 1
  - digital write pin P16 to 1 → The LED lights up when the P16 pin is high
- else if num = 2 then → If the variable "num" equals 2
  - digital write pin P16 to 1 → The LED lights up when the P16 pin is high
  - pause (ms) 100 → Wait 100 milliseconds
  - digital write pin P16 to 0 → The LED light is off when pin 16 is low
  - pause (ms) 100 → Wait 100 milliseconds
- else if num = 3 then → If the variable "num" equals 3
  - digital write pin P16 to 0 → The LED light is off when pin 16 is low

## What Will You See

After the course starts, you will observe the followings:

When you press the "1" button on the remote control, the red LED lights up.

When you press the "2" key on the remote control, the red LED flashes.

When you press the "3" button on the remote control, the red LED light goes off.

# Lesson 12 - PIR Control Light

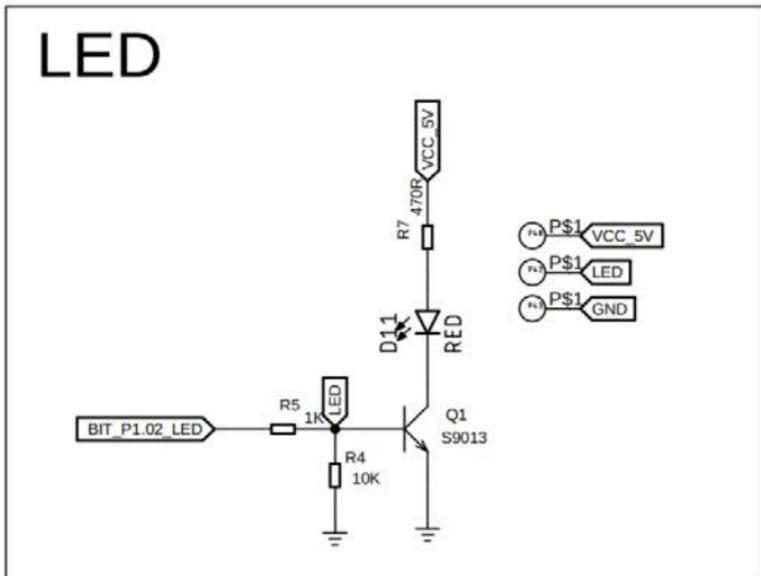
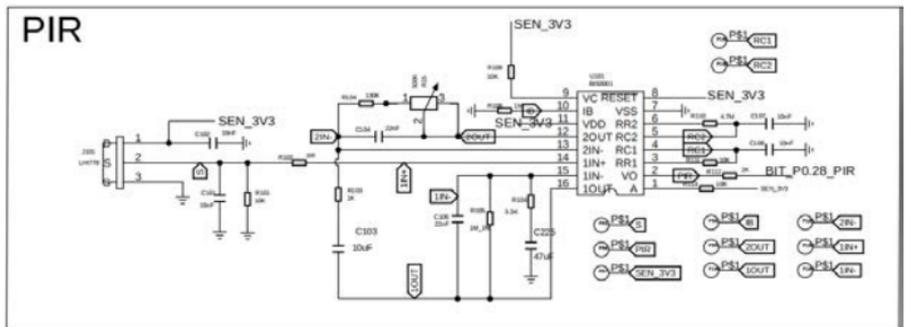
## Introduction

In this lesson, we will use the PIR sensor, which can detect infrared radiation from humans or animals. When a person or animal enters the detection range, the PIR sensor will send a signal to detect motion. PIR sensors are widely used in applications such as automatic lighting control, security alarms, and more.

## Hardware Required

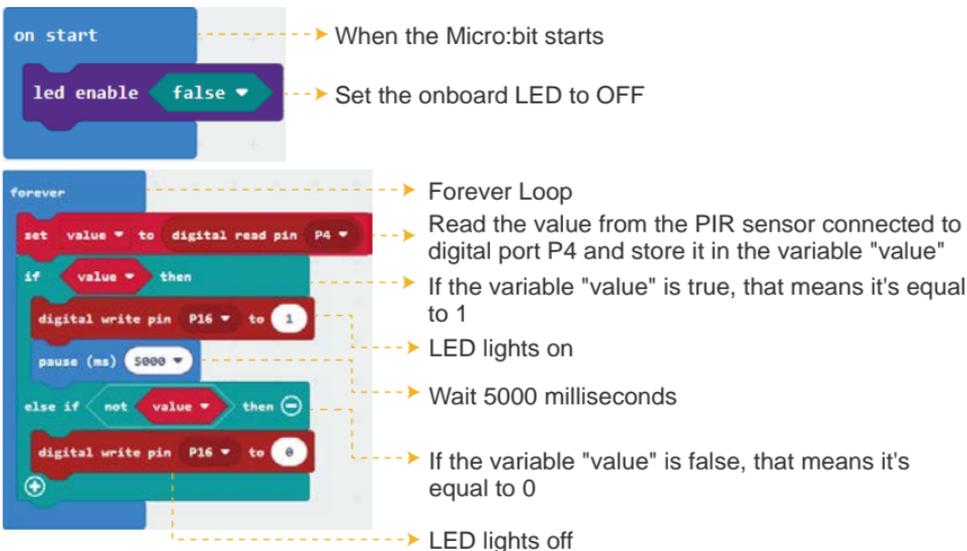
- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

## Hardware Schematic



## Code Explanation

---



The image shows a Scratch code editor with several blocks and annotations. The 'on start' block contains a 'set led enable to false' block. A 'forever' loop contains a 'set value to digital read pin P4' block, an 'if value then' block with a 'digital write pin P16 to 1' block and a 'pause (ms) 5000' block, and an 'else if not value then' block with a 'digital write pin P16 to 0' block. Annotations with arrows point to these blocks and explain their function.

on start

- When the Micro:bit starts

led enable false

- Set the onboard LED to OFF

forever

- Forever Loop

set value to digital read pin P4

- Read the value from the PIR sensor connected to digital port P4 and store it in the variable "value"

if value then

- If the variable "value" is true, that means it's equal to 1

digital write pin P16 to 1

- LED lights on

pause (ms) 5000

- Wait 5000 milliseconds

else if not value then

- If the variable "value" is false, that means it's equal to 0

digital write pin P16 to 0

- LED lights off

## What Will You See

---

After the course starts, you will observe the followings:

The LED turns on for 5 seconds when a motion, such as waving or walking, is detected within the range of the PIR sensor. If no motion is detected, the LED remains off.

## Lesson 13 - Servo Angle Control

### Introduction

---

In this lesson, we will use an infrared remote control to operate a servo motor. By pressing different buttons on the remote, we can adjust the angle of the servo, and the value of this angle will also be displayed on the LCD screen.

### Hardware Required

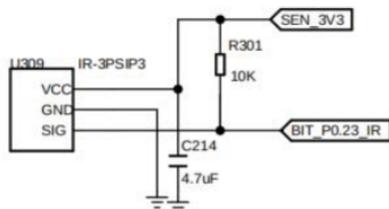
---

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1
- Infrared Remote Control

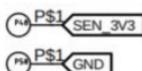
### Hardware Schematic

---

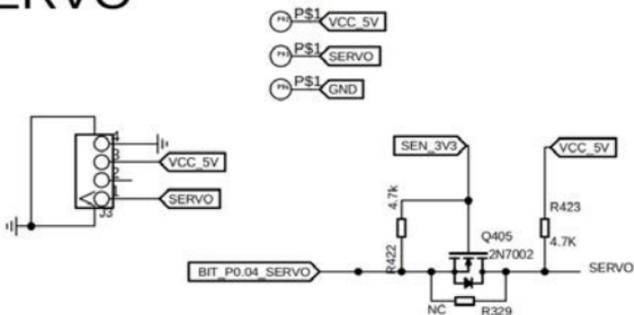
## IR Remote Sensor



2.7V-5.5V

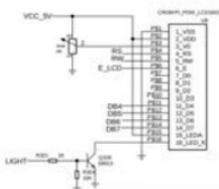


## SERVO



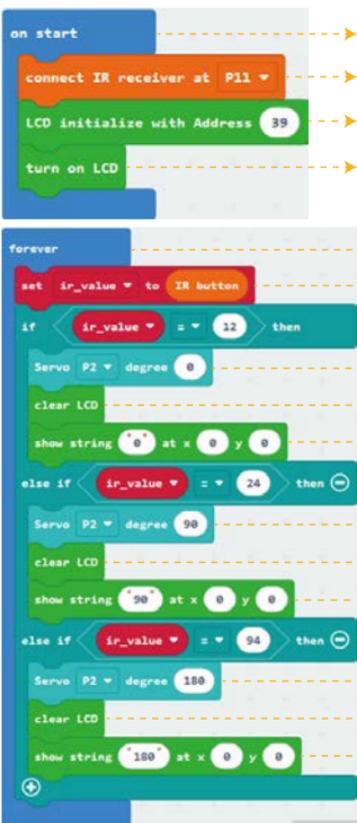
## I2C\_LCD

ADDRESS:0x27



## Code Explanation

---



The image shows a Scratch script for a Micro:bit. The script is divided into two main sections: an initialization section and a forever loop. The initialization section includes: 'on start' (When the Micro:bit starts), 'connect IR receiver at P11' (Set pin P11 as the input for the IR receiver), 'LCD initialize with Address 39' (The LCD address is initialized to 39), and 'turn on LCD' (Turn on the LCD). The forever loop section includes: 'set ir\_value to IR button' (Assign the value returned by the IR sensor to the variable 'ir\_value'), an 'if ir\_value = 12 then' block (If the returned value is 12) containing 'Servo P2 degree 0' (Set the steering angle to 0), 'clear LCD' (Clear LCD display), and 'show string "0" at x 0 y 0' (Display "0" at the first row, first column of the LCD); an 'else if ir\_value = 24 then' block (If the returned value is 24) containing 'Servo P2 degree 90' (Set the steering angle to 90), 'clear LCD' (Clear LCD display), and 'show string "90" at x 0 y 0' (Display "90" at the first row, first column of the LCD); and an 'else if ir\_value = 94 then' block (If the returned value is 94) containing 'Servo P2 degree 180' (Set the steering angle to 180), 'clear LCD' (Clear LCD display), and 'show string "180" at x 0 y 0' (Display "180" at the first row, first column of the LCD).

on start

- When the Micro:bit starts
- connect IR receiver at P11
- Set pin P11 as the input for the IR receiver
- LCD initialize with Address 39
- The LCD address is initialized to 39
- turn on LCD
- Turn on the LCD

forever

- Forever Loop
- set ir\_value to IR button
- Assign the value returned by the IR sensor to the variable "ir\_value"
- if ir\_value = 12 then
- If the returned value is 12
- Servo P2 degree 0
- Set the steering angle to 0
- clear LCD
- Clear LCD display
- show string "0" at x 0 y 0
- Display "0" at the first row, first column of the LCD
- else if ir\_value = 24 then
- If the returned value is 24
- Servo P2 degree 90
- Set the steering angle to 90
- clear LCD
- Clear LCD display
- show string "90" at x 0 y 0
- Display "90" at the first row, first column of the LCD
- else if ir\_value = 94 then
- If the returned value is 94
- Servo P2 degree 180
- Set the steering angle to 180
- clear LCD
- Clear LCD display
- show string "180" at x 0 y 0
- Display "180" at the first row, first column of the LCD

## What Will You See

---

The servo is controlled by a remote. When button 1 is pressed, it rotates to 0 degrees; button 2 sets it to 90 degrees, and button 3 to 180 degrees. After each press, the current angle is displayed on the LCD screen.

## Lesson 14 - Polite Automatic Door

### Introduction

---

In this lesson, you will learn how to open the door by pressing a button and show a "Welcome" message on the LCD.



## What Will You See

After uploading the program, press button A on the Micro:bit to rotate the servo to 180 degrees (open the door) and display a "Welcome" message on the LCD. Release the button to rotate the servo back to 0 degrees (close the door) and clear the message from the LCD.

## Lesson 15 - Sound Reminder

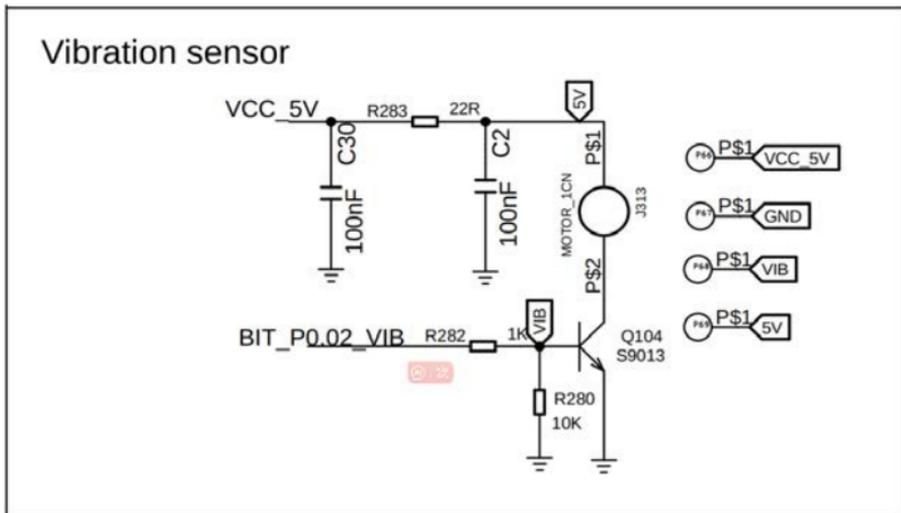
### Introduction

In this lesson, we will learn advanced operations of the Micro:bit microphone. We will use the sound sensor to detect the ambient noise in the current environment.

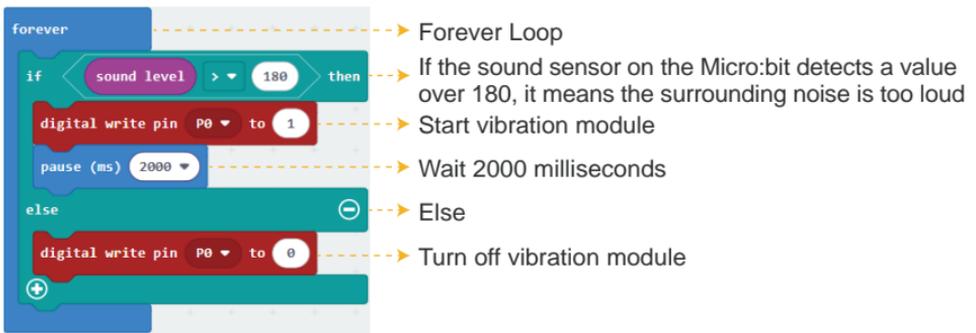
### Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

### Hardware Schematic



### Code Explanation



## What Will You See

If the surrounding noise is too loud, or if you are shouting into the microphone, the device will vibrate for 2 seconds to remind you to lower your voice. If the sound level remains high, the vibration motor will continue to vibrate. Once the noise level drops, the vibration will stop.

**Note:** When the Micro:bit is just powered on, the microphone may produce unusually high sound readings. To filter these out, we use a threshold of 180.

## Lesson 16 - Calculation Of Acceleration

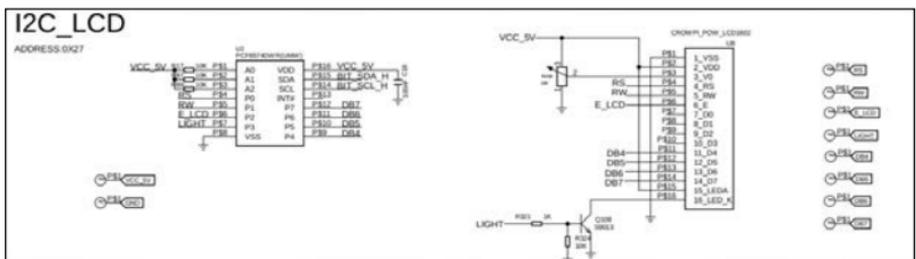
### Introduction

In this lesson, we will learn how to use an accelerometer to read x, y, and z axis values.

### Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

### Hardware Schematic



## Code Explanation

---



The code block for initialization consists of four blocks:

- on start** (blue block): The starting point of the program.
- led enable** (purple block): Set to **false**.
- LCD initialize with Address** (green block): Set to **39**.
- turn on LCD** (green block): Turns the LCD display on.

Annotations with arrows point from the text to the corresponding blocks:

- When the Micro:bit starts
- Set the onboard LED to OFF
- The LCD address is initialized to 39
- Turn on the LCD



The code block for the forever loop consists of four blocks:

- forever** (blue block): The loop structure.
- show string** (green block): Displays 'X:' followed by 'acceleration (mg) x' and 'Y:' followed by 'acceleration (mg) y' at x=0, y=0.
- show string** (green block): Displays 'Z:' followed by 'acceleration (mg) z' at x=0, y=1.
- pause (ms)** (blue block): Set to **500**.
- clear LCD** (green block): Clears the LCD display.

Annotations with arrows point from the text to the corresponding blocks:

- Forever Loop
- The x- and y-axis acceleration values are displayed in the first row and first column of the LCD display
- The z-axis acceleration values are displayed in the second row and first column of the LCD display
- Wait 500 milliseconds
- Clear LCD display

## What Will You See

---

Run the program, and you'll see the LCD display the accelerometer values for the x, y, and z axes. When you move the accelerometer quickly along an axis, you'll notice that the value for that axis changes accordingly.

# Lesson 17 - Smart Corridor Light

## Introduction

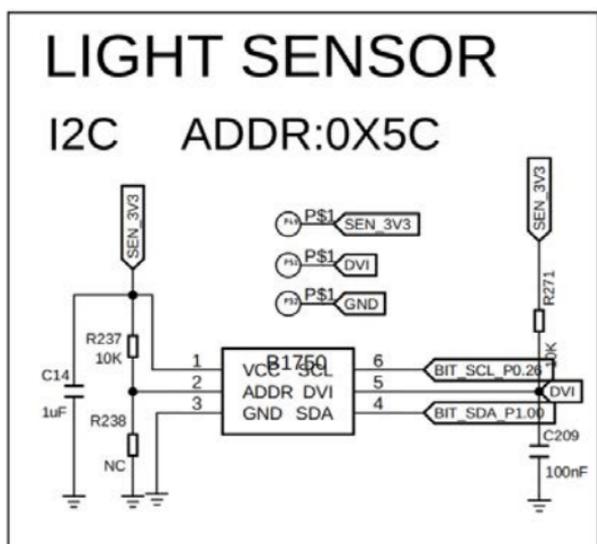
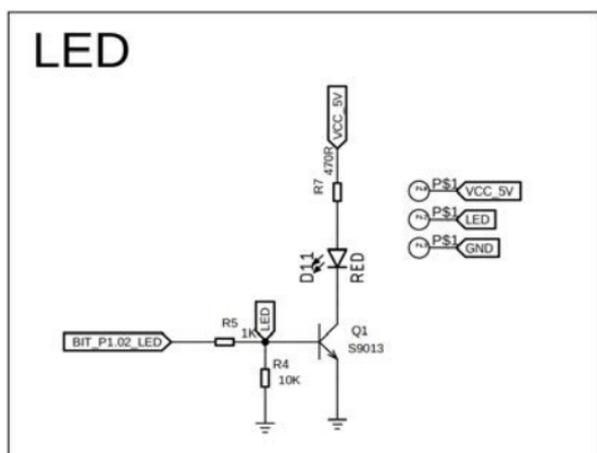
---

In this lesson, we will learn advanced operations involving the integration of the sound sensor, light sensor, and LED. By coordinating these sensors, we will achieve the effect of an intelligent corridor light.

## Hardware Required

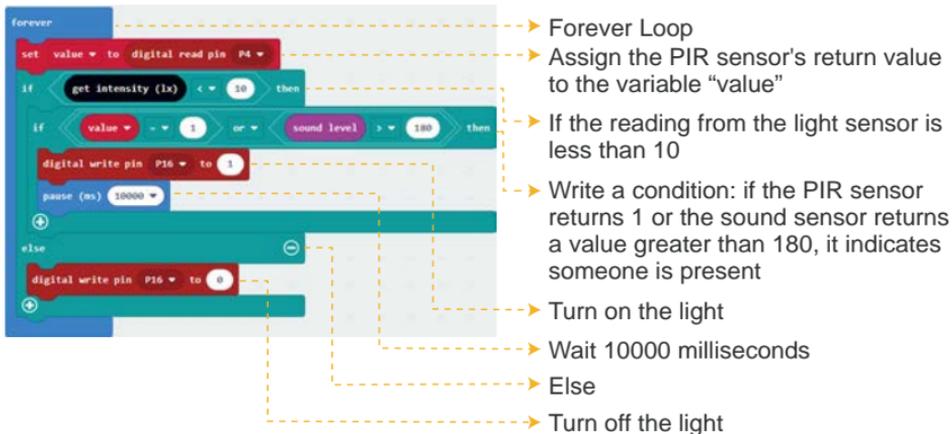
---

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1



## Code Explanation

on start	→	When the Micro:bit starts
led enable <span style="background-color: #008000; color: white; padding: 2px;">false</span>	→	Set the onboard LED to OFF
set Address <span style="background-color: #000000; color: white; padding: 2px;">92</span>	→	The light sensor address is initialized to 92
turn on	→	Turn on the light sensor



## What Will You See

When the ambient light is very strong, the LED will remain off regardless of movement or sound.

When the ambient light is very dim, or if the light sensor is covered by a hand, the LED will stay off by default. In this case, if someone moves or makes a sound, the LED will turn on for 10 seconds and then turn off. If there is continuous movement or continuous sound, the LED will stay on. Once there is no further movement or sound, the LED will remain on for 10 seconds before turning off.

## Lesson 18 - Hall Counter

### Introduction

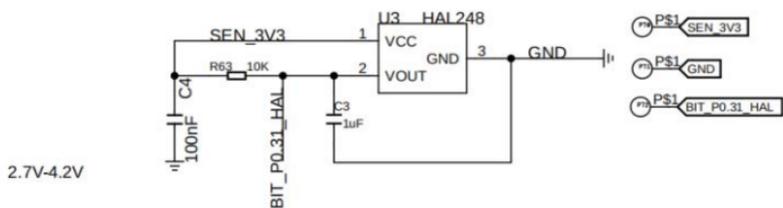
In this lesson, we will learn the basic usage of a Hall effect counter.

### Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

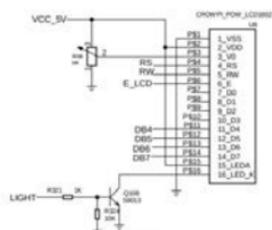
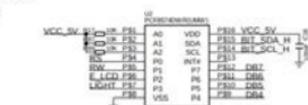
### Hardware Schematic

# HALL SENSOR

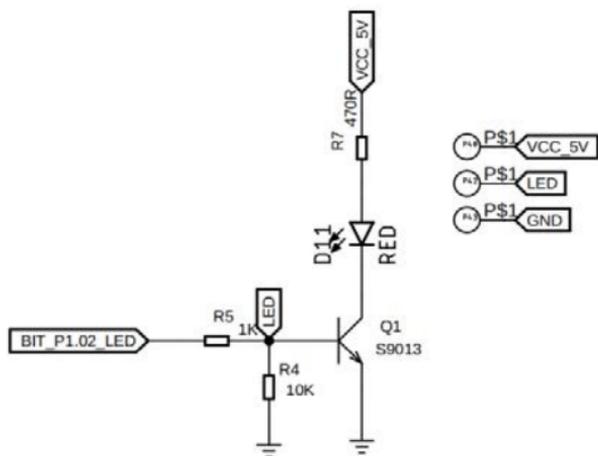


# I2C\_LCD

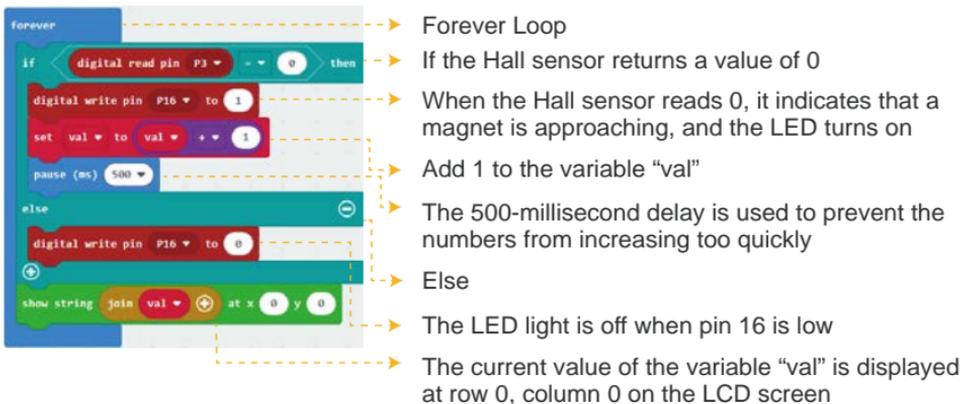
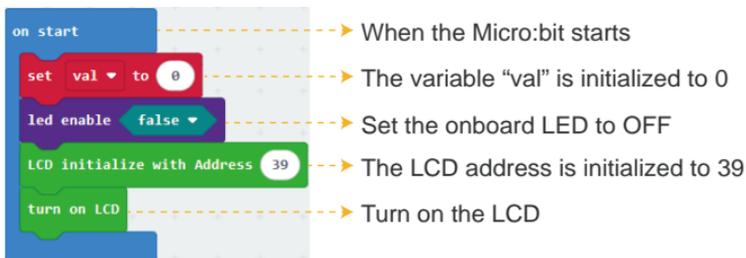
ADDRESS 0x27



# LED



## Code Explanation



## What Will You See

Run the program, and the LCD will display "Counter: 0". Each time a magnet approaches the Hall sensor, the red LED will turn on, and the count on the LCD will increase by 1. Repeat this process to continue counting.

## Lesson 19 - Show Number on 4-Digit Display

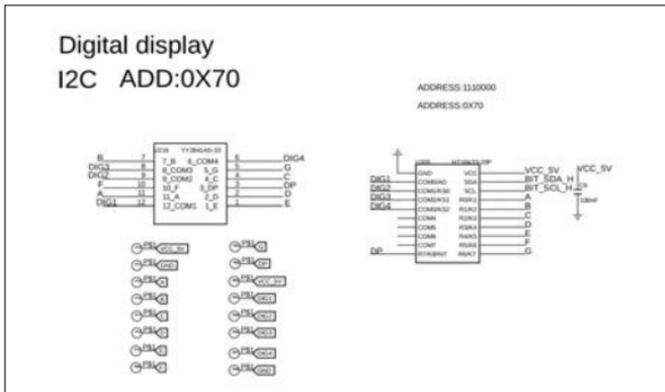
### Introduction

In this lesson, we will learn how to use a 4-digit LED display. We will learn how to display numbers on it, and control each digit through programming.

### Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

## Hardware Schematic



## Code Explanation



## What will you see

Run the program, and the 4-digit display will first show the numbers 1 to 4. After that, the numbers 5 to 8 will be displayed, replacing the previous numbers.

# Lesson 20 - Make an Accurate Clock

## Introduction

In this lesson, we'll learn how to use the RTC clock to keep track of time. We'll also display the RTC time on a 4-digit display.

## Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1



## What will you see

The time is calibrated using the RTC and then displayed on the 4-digit display.

# Lesson 21 - Alarm Clock

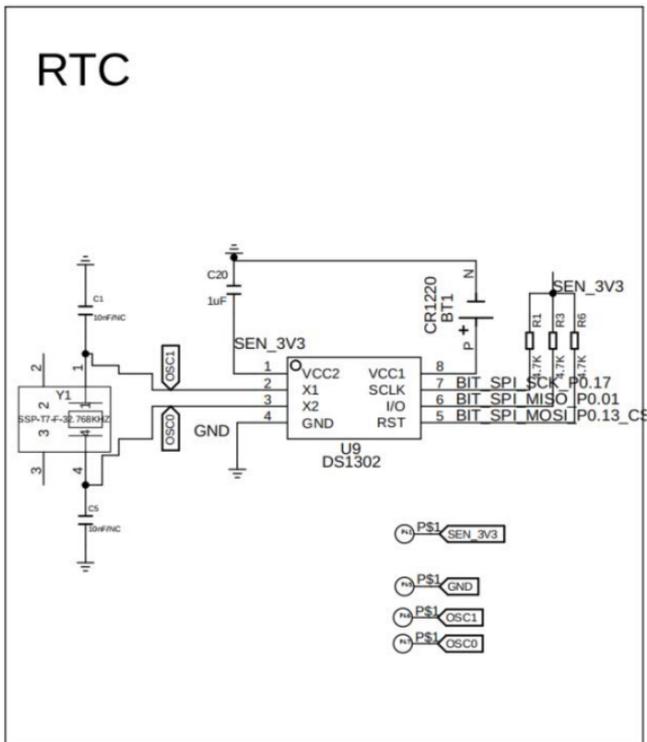
## Introduction

In this lesson, we will combine the RTC clock, a 4-digit Nixie tube, and the Micro:bit's the onboard speaker and the touch sensor to create an alarm clock. The alarm can be turned off using the touch sensor when it sounds.

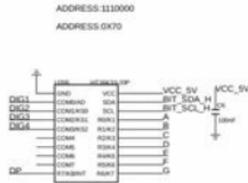
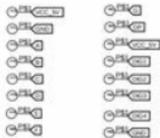
## Hardware Required

- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

## Hardware Schematic



## Digital display I2C ADD:0X70



## Code Explanation

on start

- When the Micro:bit starts

initialize 7-segment display

- Initialize the 7-segment display

set num to 0

- Initialize the value of the variable "num" to 0

set a to CLK P13 D10 P14 CS P15

- Initialize the pin connected to the RTC

set Date and Time: Year 2025

Month 4

Day 15

WeekDay 2

Hour 10

Minute 37

Second 0

- Initialize the real-time clock, which can be freely set here

forever

- Forever Loop

write number: parse to number join get hour get minute on 7-segment display with color on

- Combine the hour and minute into a string and display it on the Nixie tube

if get hour >= 24 or get minute >= 60 then

- If the time reaches the preset value

write number: parse to number join get hour get minute on 7-segment display with color on

- Combine the hour and minute into a string and display it on the Nixie tube

while touch sensor (D1) readable or num < 1200

- Repeat the loop as long as the Micro:bit's touch sensor is not pressed and the variable "num" is not equal to 1200

play tone (D1) 440Hz 0.5

- The Micro:bit plays a sound

wait 100 ms

- Wait 100 milliseconds

set num = 1200 + 1

- The variable "num" increases by 1 every 100 milliseconds. After 1200 increments, 2 minutes will have passed

stop all sounds

wait 100 ms

Stop all sounds

The variable "num" is set to 1200 to prevent from re-entering the loop when the Micro:bit's touch sensor is triggered again.

## What will you see

The alarm clock is set through the program. When the time is up, the speaker will sound for 2 minutes. If you press the touch sensor during this time, the speaker will turn off.

# Lesson 22 - Compass

## Introduction

In this lesson, we will use the Micro:bit's onboard compass to create a project.

## Hardware Required

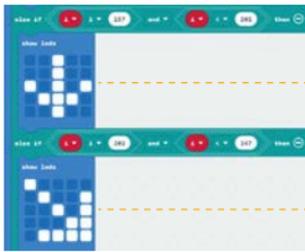
- All-in-one-Starter-Kit-for-Micro:bit x1
- USB Cable x1

## Code Explanation

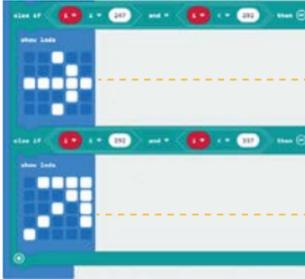


The code consists of the following blocks:

- on start** block: When the Micro:bit starts.
- calibrate compass** block: Calibrate the compass on the Micro:bit.
- Forever Loop** block: A loop containing several conditional blocks:
  - if < i < 23 or i >= 338 >** block: Store the compass heading from the Micro:bit in the variable 'i'. If the value of variable 'i' is less than 23 or greater than or equal to 338, then the direction is due north.
  - if < i < 23 >** block: If the direction is due north, pointing forward means pointing north.
  - if < i <= 67 >** block: If the value of variable 'i' is between 23 and 67, the direction is northwest.
  - if < i <= 67 >** block: If the direction is northwest, pointing northwest means pointing north.
  - if < i <= 112 >** block: If the value of variable 'i' is between 67 and 112, the direction is due west.
  - if < i <= 112 >** block: If the direction is due west, pointing due west means pointing north.
  - if < i <= 157 >** block: If the value of variable 'i' is between 112 and 157, the direction is southwest.
  - if < i <= 157 >** block: If the direction is southwest, pointing southwest means pointing north.



- If the value of variable 'i' is between 157 and 202, the direction is due south
- If the direction is due south, pointing due south means pointing north.
- If the value of variable 'i' is between 202 and 247, the direction is southeast
- If the direction is southeast, pointing southeast means pointing north.



- If the value of variable 'i' is between 247 and 292, the direction is due east
- If the direction is due east, pointing due east means pointing north.
- If the value of variable 'i' is between 292 and 337, the direction is northeast
- If the direction is northeast, pointing northeast means pointing north.

## What will you see

The LED matrix indicates the direction by displaying an arrow, which always points to the north. The pointing resolution is 45 degrees, which corresponds to the following directions:  $\uparrow$ ,  $\nearrow$ ,  $\rightarrow$ ,  $\searrow$ ,  $\downarrow$ ,  $\swarrow$ ,  $\leftarrow$ ,  $\nwarrow$ .

**Note:** After uploading the code, make sure to calibrate the compass by moving the red dot around the entire Micro:bit screen as prompted.

