

# SparkHive

**Arduino Workshop/Hackathon**

18 May 2024 @ Lecture Hall C5  
Universiti Teknologi PETRONAS



# ***Welcoming* *Speech***

# What we'll be doing today

Today, you will assume the role of engineer.

You are going to learn to pick things up quickly. To ask questions and to find the answers to them.

You will form 4 groups.

You will be introduced to the Arduino platform and face 3 challenges.

# ***Workshop at light speed***

## **What is Arduino?**

Arduino is a prototyping microcontroller platform mostly used in hobbyist applications.

In essence, microcontrollers are tiny computers that run your washing machine, TVs, air conditioners, cars, and pretty much everything.

## **How does it work?**

There are generally three types of pins:

Analog, digital, and digital PWM.

These pins connect to other devices such as buttons, relays, LEDs, etc.

Relays then connect to other high power devices such as motors and heaters.

By writing code, compiling them to 1s and 0s and uploading them to the Arduino board, you can receive input from and output to different devices to create a system.

# Section 1

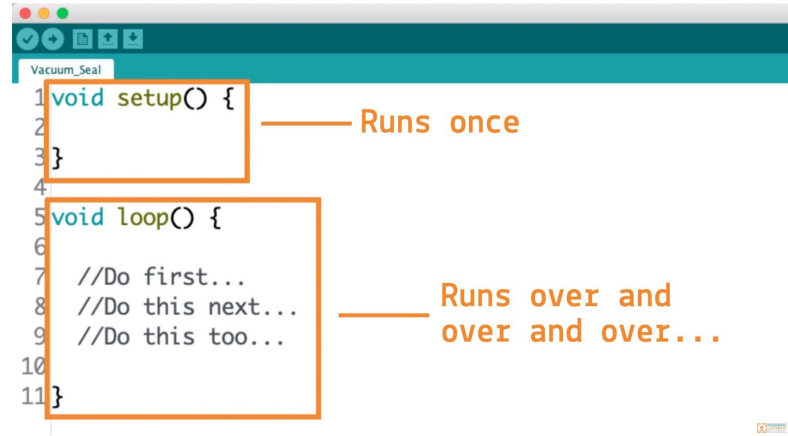
## setup() and loop()

Any code placed within setup() will run first, and will run only once.

Any code placed within loop() will run over and over unless the Arduino board is disconnected from a power supply.

## pinMode()

To use a pin, you must declare it as an INPUT or OUTPUT.



## Example

To use the digital pin 0 as INPUT, type:

```
pinMode(0, INPUT);
```

To use the analog pin 3 as OUTPUT, type:

```
pinMode(A3, OUTPUT);
```

# Section 1

## **digitalRead() and digitalWrite()**

`digitalRead(pin_num)` will measure boolean states. HIGH will be anything above the Arduino board's ground reference voltage.

`digitalWrite(pin_num, HIGH/LOW)` will send boolean states to the specified pin.

## **analogRead() and delay()**

`analogRead(pin_num)` will read a continuous analog value from the specified pin.

`delay(milliseconds)` will delay the execution of the next line of code by the specified amount of time.

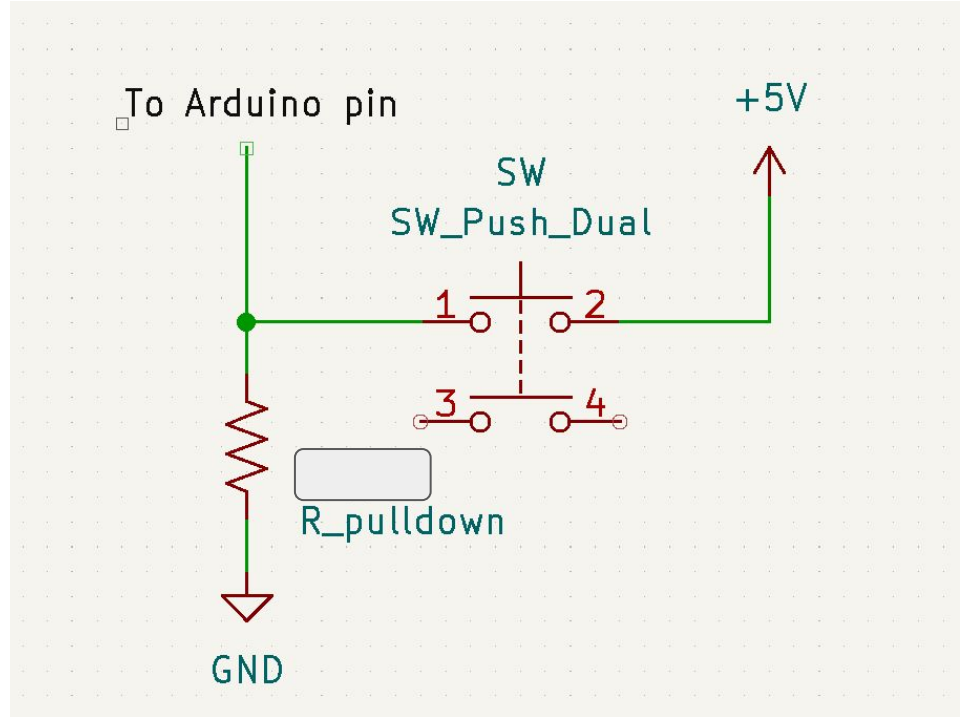
# Section 1 - Exercise

## Connect the switch button

Wire the button according to the schematic. You may use a 4-pin switch.

Connect two 1k $\Omega$  resistors in series for the pulldown resistance.

Ask your guides for help.



# Section 1 - Exercise

## Read the state of a button

Pick a digital pin. Digital pins are those labeled with only numbers.

Analog pins are labeled starting with the letter A.

Use `pinMode()` to declare the chosen pin as INPUT.

Use `digitalRead()` to fetch the state of a button. Check the result by adding the line `Serial.print("Button pressed")`.

## Within `setup()` loop:

Add the following line: `Serial.begin(9600);`

## Conditional statements

Initialize a state variable by adding the line:

```
bool [variable_name] = digitalRead(pin_num);
```

Use an if statement to print something to the Serial Monitor if the button is pressed:

```
if (variable_name == true) {  
    Serial.println("Button pressed");  
}
```



## Section 2

### Other conditionals: else, else if

`else { // Your code }` after an if statement will execute code contained in `else` whenever the if statement condition is not satisfied.

```
else if (condition) {  
    // Your code  
}
```

is the same as `else` except that it will only execute if the extra condition is true.

### **tone() and noTone()**

These functions control buzzers.

```
tone(pin_num, freq);
```

will make a buzzer connected at `pin_num` output a sound at the specified frequency.

```
noTone(pin_num);
```

will cancel the earlier command to the buzzer connected at `pin_num`.

## Section 2

### **millis()**

This function returns the time elapsed that is tracked by the Arduino microcontroller.

It does not measure time starting from 0 when called. It simply returns the time elapsed since your Arduino was turned on.

### **lcd.setCursor()**

```
lcd.setCursor(column, line);
```

This will set the cursor at the particular LCD column and line.

### **lcd.print()**

```
lcd.print("Your message here");
```

This will print a string starting from the cursor position defined by `lcd.setCursor()`

### **lcd.clear()**

```
lcd.clear();
```

This will reset the LCD.

## Section 2 - Exercise

### Write something on the LCD

First wire your LCD on the breadboard following the info on the right.

Then import the LiquidCrystal.h library. Put it at the top line of your source code.

```
#include <LiquidCrystal.h>;
```

### LCD wiring instructions

**K** to GND.

**A** to 5V rail through a 220Ω resistor.

**E** to digital pin 11.

**RW** to GND.

**RS** to digital pin 12.

**VO** to GND through two 1kΩ resistors.

**VDD** to 5V

**VSS** to GND.

**D7** to digital pin 2.

**D6** to digital pin 3.

**D5** to digital pin 4.

**D4** to digital pin 5.

## Section 2

### Write something on the LCD

Include these before your setup() loop:

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

Put this one within your setup() loop:

```
lcd.begin(16, 2);
```

Using the functions below, write a message to your LCD.

#### **lcd.setCursor()**

```
lcd.setCursor(column, line);
```

This will set the cursor at the particular LCD column and line.

#### **lcd.print()**

```
lcd.print("Hello world");
```

This will print a string starting from the cursor position defined by lcd.setCursor()

#### **lcd.clear()**

```
lcd.clear();
```

This will reset the LCD.

# Hackathon Challenges

Attempt as many challenges to the best of your ability.

Marks will be given based on **code readability**, your **understanding of the problem** and attempts at coming up with a solution.

Your ability to **cooperate**, **communicate**, and **brainstorm** with your team members will also be taken into account.

# Challenge 1

Boréno, a Mexican restaurant chain wants you to design a smart signboard that will be used in all of its branches throughout Malaysia.

The signboard will automatically illuminate when the sky is dark. Using the output of a light-dependent resistor, write a program that turns an LED on/off.



## Challenge 1 - Coding Hint

Pick an analog pin to receive input from the LDR. Use the `pinMode()` function. Don't forget that this function needs to be called within the `setup()` loop.

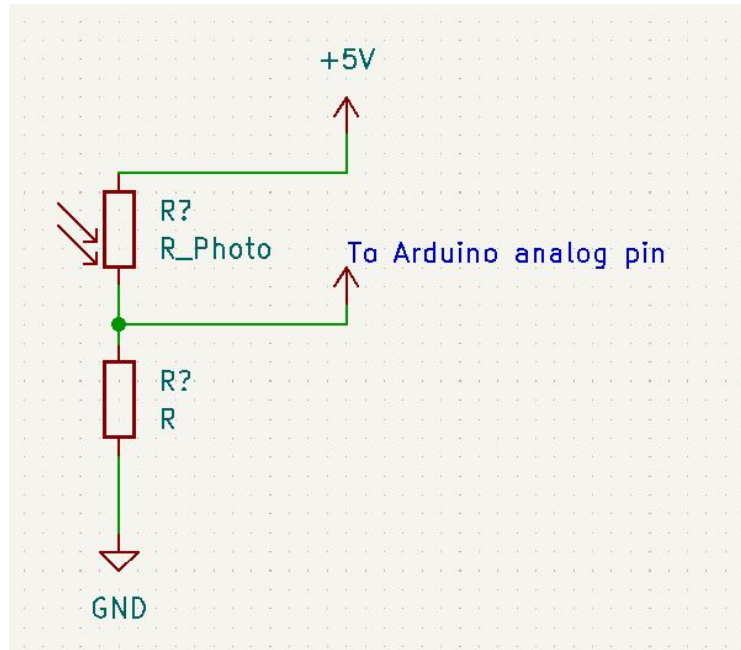
Then within `loop()`, you may, for example:

```
int variable = analogRead(pin_num);  
if (???? < ???) {  
    ?????;  
}
```



# Challenge 1

Tip for wiring the LDR:





## Challenge 2

Nabalu Metro has chosen you to implement a notification system to report sexual harassment on the upcoming Kota Kinabalu BRT's women's coach.

To prevent false triggers, the emergency button must be pressed down for 3 seconds to send a message.

Turn on only the red LED if a message  $> 3s$  is received

Turn on only the green LED if a signal  $< 3s$  is received.

Whenever the emergency button registers even a brief signal, the corresponding green LED lights up to indicate a potential emergency.



# Challenge 3

Cummings, a manufacturer of diesel combustion engines has employed you to program a mode selector for a diesel generator. Using a single button as an input, show to the user that the proper mode has been selected using a buzzer, two LEDs, and an LCD.

When the button is pressed once, the engine enters low emissions mode. When pressed again, the engine enters high performance mode. When pressed the third time, the engine reverts to its normal mode.

For each mode, show the mode name on the LCD.

Turn off both LEDs in NORMAL mode.

Turn on only the green LED in LOW EMISSIONS mode.

Turn on only the red LED in PERFORMANCE mode.



# ***Prize Giving Ceremony***

# ***Photography Session***