

Student name: Yasmin Pokia

Student ID: s222245206

SIT225: Data Capture Technologies

Activity 1.1: Arduino Blink

Welcome to Arduino!

Arduino is an electronic prototyping platform. Different types of sensors & actuators can be attached to Arduino boards to create our own sensing-thinking-acting systems.

Throughout this unit, we will use Arduino to create different sensing devices, and to retrieve the collected sensor data.

In this task, we will try out an introductory exercise, to learn the basic concepts of Arduino.

Hardware Required

- Arduino Board with in-built LED

- USB cable

Software Required

- Arduino programming environment

Pre-requisites: You must do the following before this task

Unit site weekly materials

Active learning sessions require reading material and/or videos available in the unit site, which you must read/view **BEFORE** you start the lab. If you are on-campus, this means that we expect you to have gone through these materials when you join active learning sessions.

Why should you read/watch pre-lab materials?

These materials will help you understand the background which the lab tasks require. Students come to university from diverse backgrounds. Some of you may be familiar with the background information, some of you may not. When you come to the lab prepared, you're already equipped

with confidence and will be able to participate in activities better. Ultimately, class time will be much more productive, dynamic, and fun for everyone.

Here are the pre-lab materials for our first task:

1. Watch TED Talk: <https://www.youtube.com/watch?v=UoBUXOOdLXY> (~15 minutes)
2. Read Arduino tutorial: <https://www.dummies.com/article/technology/computers/hardware/arduino/how-to-complete-your-first-arduino-sketch-164747>)
3. Read this task sheet from beginning to end.

Task Objective

“We have an Arduino board with an in-built LED light. We need the LED light to be turned on and off continuously, every one second.”

Activity Submission Details

Answer the questions below in this word document and other activities in this activity sheet to create a PDF and submit to OnTrack as described in this week’s OnTrack task. PDFs of this activity sheet and OnTrack task need to be merged for submission in the OnTrack portal.

Q1: The TED talk given under the Pre-Lab materials, shows how Arduino is being used for interesting projects to capture data from the environment, process it, and use it to carry out useful actions.

Fill the given table below to answer the following:

What are **three** projects that use captured data as given in the TED talk? What data do they capture? What sensors do you think they could use to capture this data?

Project name	Data captured	Sensors to capture the data
Pet/Cat feeder	Feeding times, amount of food dispensed, which food to dispense for the type of cat	Weight sensors to measure the amount of food, infrared sensors to detect the presence of the cat, motor controllers to operate the food dispenser mechanism
TxtBomber	Text input from the user	Buttons or a small keyboard for text input, position sensors to ensure accurate text printing, motor controllers to move the pens and print the text.
Sign language glove	Hand and finger movements	Flex sensors to measure the bending of fingers, accelerometers

		and gyroscopes to track hand orientation and movement, pressure sensors to detect touch and grip strength
--	--	---

Q2: Consider the given Task Objective. Think about how this simple system can be decomposed to ‘Sense-Think-Act’?

- a) What is the ‘sensing’ requirement in this system, if any?

There is no sensing requirement in this system as this system does not require any external data collection through sensor data as it is built to function on a predetermined timing mechanism.

- b) What is the ‘thinking’ requirement in this system, if any?

The thinking requirement in this system is the timing logic on the Arduino microcontroller. To decide when to turn the LED on and off, the microcontroller must monitor the amount of time that has passed.

- c) What is the ‘acting’ requirement in this system, if any?

The acting requirement in this system is to control the state of the LED light, where the Arduino must control the LED by turning it on and off based on the timing logic.

Q3: Please refer to the provided ‘Arduino Blink Activity Sheet’ and follow the steps.

- a) In Arduino-speak, what is a “sketch”?

In Arduino-speak, a “sketch” is a program created with the Arduino Software (IDE).

- b) `setup()` and `loop()` are key Arduino constructs. These are required in every Arduino sketch.

- i) Which of the above two, runs once at the very beginning of your program and never again (unless you reset or upload new code)?

The `setup()` function runs once at the very beginning of your program and never again, unless you reset or upload new code.

- ii) Which of the above two, is used to continuously run code over and over again?

The `loop()` function is used to continuously run code over and over again. The `loop()` function is called repeatedly once the `setup()` function is finished, enabling the Arduino to carry out its primary functions continually.

- c) What does **`pinMode()`** do?

Hint: <http://arduino.cc/en/Reference/HomePage>

The `pinMode()` function in Arduino is used to configure a specific pin to behave either as an input or an output. It sets the mode of the specified pin, defining whether it will be used to receive data (input) or send data (output).

d) What is a comment?

A comment in programming is a line or block of text in the code that is not executed by the program. Comments can be used to mark functionality for future reference, clarify the code, or offer explanations. They aid in the comprehension of the logic and purpose of the code by programmers. Comments are indicated with // for single-line comments and /*... */ for multi-line comments in Arduino sketches and many other programming languages.

e) What does the following line of code do:

```
delay(x);
```

Hint: <http://arduino.cc/en/Reference/HomePage>

The line 'delay(x)', pauses the execution of the Arduino program for 'x' milliseconds. The program doesn't go on to the following instruction during this period.

f) There is something you need to check before uploading your sketch. What is this?

Before uploading your sketch to the Arduino board, something you need to check is to make sure the correct Arduino board is selected in the Arduino IDE under 'Tools > Board', and the correct port is selected under 'Tools > Port' to communicate with your Arduino board.

Q4: How can you test the Blink program to make sure it is working as given in the Task Objective?

Activity 1.2: Write Arduino data to serial communication port

Now you can blink Arduino's built-in LED, it is time to talk to outside Arduino-world, your computer which connects the Arduino board using a USB cable. Arduino IDE shows what you write to the serial port.

Hardware Required

Arduino Board with in-built LED

USB cable

Software Required

Arduino programming environment

Steps:

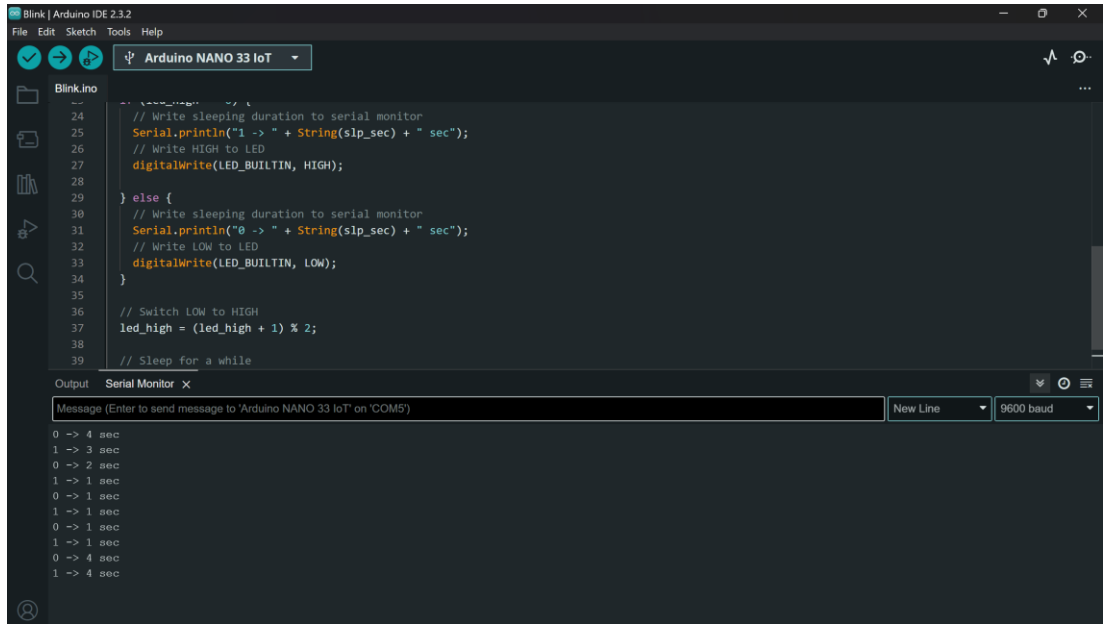
Steps	Actions
1	Identify the port through which Arduino is connected to your computer. You can find it in Arduino IDE Tools menu. Write Arduino sketch (or download from https://github.com/deakin-deep-dreamer/sit225/blob/main/week_1/sketch_blink.ino), which look like below, in Arduino IDE, deploy in your board and observe the output in IDE's serial monitor.

```

1  int led_high = 0; // variable to flag ON (1) or OFF (0)
2  int slp_sec = 0; // variable holds number of seconds to sleep
3
4
5  void setup() {
6      pinMode(LED_BUILTIN, OUTPUT); // set LED pin as output
7      digitalWrite(LED_BUILTIN, LOW); // switch off LED pin
8
9      // Baud rate (speed of communication, symbol transfer rate)
10     Serial.begin(6900);
11 }
12
13 void loop() {
14     // sleep randomly selected 1 to 5 seconds
15     slp_sec = random(1, 5);
16
17     if (led_high == 0) {
18         // Write sleeping duration to serial monitor
19         Serial.println("1 -> " + String(slp_sec) + " sec");
20         // Write HIGH to LED
21         digitalWrite(LED_BUILTIN, HIGH);
22     } else {
23         // Write sleeping duration to serial monitor
24         Serial.println("0 -> " + String(slp_sec) + " sec");
25         // Write LOW to LED
26         digitalWrite(LED_BUILTIN, LOW);
27     }
28
29     // Switch LOW to HIGH
30     led_high = (led_high + 1) % 2;
31
32     // Sleep for a while
33     delay(1000*slp_sec);
34 }
35
36

```

Question: Screenshot the serial monitor output and paste the image here.

	<p>Answer:</p> 
2	<p>Question: Observe the use of “Serial” such as functions Serial.begin() in setup and Serial.println() in loop. Describe what these functions are doing with respect to the serial monitor output you have attached above.</p> <p>Answer: The Serial.begin(6900) function initializes the serial communication at a baud rate of 6900 bits per second, which sets up the communication speed between the Arduino and the computer.</p> <p>The Serial.println(“message”) function sends a string message to the serial monitor, followed by a new line, which is used to print the current state of the LED (on or off) and the duration for which it will sleep.</p> <p>When the LED is turned on, the serial monitor prints ‘1 -> X sec’, where ‘X’ is the random sleep duration, whereas when the LED is turned off, the serial monitor prints ‘0 -> X sec’.</p>
3	<p>Question: If Arduino transfers data at 4800 bits per second (baud rate) and you're sending 12 bytes of data, how long does it take to send over this information?</p> <p>Answer: 8 bits in a byte, so 12 bytes is 96 bits. Time = number of bits / baud rate so 96 bits / 4800 bits per second = 0.02 seconds. Thus, it takes 0.02 seconds to send 12 bytes of data at a baud rate of 4800 bits per second.</p>

Activity 1.3: Arduino talks to Python

To listen to what Arduino sends, there will be a Python program running and keep listening to the same port where Arduino is writing data to receive it.

Hardware Required

Arduino Board with in-built LED

USB cable

Software Required

Arduino programming environment

Python 3.0 (Follow Python installation manual in unit site)

Steps:

Steps	Actions
1	Write Arduino sketch (or download from https://github.com/deakin-deep-dreamer/sit225/blob/main/week_1/sketch_serial_comm.ino) which looks like below. Open in Arduino IDE. Study the code. Upload the code to Arduino board and observe output in Arduino IDE serial monitor.

```

1
2 int x;
3
4 void setup() {
5     Serial.begin(9600); // set baud rate
6 }
7
8 void loop() {
9     while (!Serial.available()) {} // wait for data to arrive
10
11     // read string data from Serial, we know Python
12     // script is sending just an integer.
13     x = Serial.readString().toInt();
14
15     // write a string (no newline)
16     Serial.print("Arduino sends: ");
17
18     // Add 1 to what received.
19     // Write an integer with a newline (println vs print).
20     Serial.println(x + 1);
21
22     // Push the data through serial channel.
23     Serial.flush();
24 }
25

```

Question: Do you see any output in serial monitor? If not, then why?

Answer: No, there is no output in the serial monitor as the Arduino is waiting for data to arrive via the serial interface. The code has a 'while (!Serial.available()) {}' loop, which stops the execution until it receives data from the serial input.

To see any input in the serial monitor, type an integer in the input field and send it, and you will see a +1 increase in the integer you typed.

- 2 Write Python code as below (or download from https://github.com/deaki`n-deep-dreamer/sit225/blob/main/week_1/serial_comm_script.py) and save it to a file serial_comm_script.py.

```

1  import serial
2  import random
3
4  # set baud rate, same speed as set in your Arduino sketch.
5  boud_rate = 9600
6
7  # set serial port as suits your operating system
8  s = serial.Serial('/dev/ttyACM1', boud_rate, timeout=5)
9
10 while True: # infinite loop, keep running
11
12     # a random number between 5 and 50.
13     data_send = random.randint(5, 50)
14
15     # write to serial port, set data encoding.
16     # Raw bytes are sent through serial ports, Python bytes() needs
17     # to know the encoding to generate bytes from string.
18     #
19     # We send a single integer which is read from Arduino sketch.
20     #
21     d = s.write(bytes(str(data_send), 'utf-8'))
22     print(f"Send >>> {data_send} ({d} bytes)")
23
24     # Read from serial port.
25     #
26     # readline keeps reading until a newline found in the data stream.
27     # Unlike write above, we just send an integer with no newline.
28     # You should receive data the same way as it is sent.
29     #
30     d = s.readline().decode("utf-8")
31     print(f"Recv <<< {d}")
32

```

Run the Python file from command line using command:

\$ *python serial_comm_script.py*

Question: Run Python script in command line, does the script run or do you receive any error? If there is an error, analyse the error message and identify what went wrong.

Answer: When I run the Python script in command line, I receive an error.

	<p>Answer: The communication protocol used between the Arduino sketch and Python script is based on serial communication. The python script sends a random integer to the Arduino board, Arduino reads this integer, increments it by 1, and sends the result back to the Python script. Python script reads and prints the response.</p>
--	--