Student name: Hoang Long Tran

Student ID: s223128143

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
Botanicalls	The moisture data will be the language of the plants. There are thresholds set for the moisture level and each threshold there will be an output (e.g. I need water).	Sensor probe measures the moisture of the soil. A microcontroller captures the data and send them online via an embedded ethernet connection.
RDTN Geiger Shield	Radiation data gather from many different sources to inform people of places that are radioactive in Japan since the Fukushima Daiichi nuclear disaster.	Geiger counter and Arduino integrated to capture radiation data.
Tweets farts chair	Natural gas data, more	A office chair and a natural gas

	specifically, human fart that is naturally produced in the digestive system	sensor to capture every time someone farts on the chair, and tweets the fart on Twitter.

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?
Sensing comes from sensors, and since there is no sensor involves in this case, so there is no 'sensing' requirement.

b) What is the 'thinking' requirement in this system, if any?
The thinking or logic of our system is the ability to turn on and off the LED light continuously at 1 second interval.

c) What is the 'acting' requirement in this system, if any?
The acting is the action of the system, which is turning the LED light on and off at appropriate intervals.

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

a) In Arduino-speak, what is a "sketch"?
The term 'sketch' in this case comes from the functionality of Arduino that allows people to quickly prototype and test ideas using little code, similar to what we sketch on a paper.

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 every time the program starts. This is where we set up our initialization tasks.
   ii) Which of the above two, is used to continuously run code over and over again?
   The `loop()` is used to run the code continuously in a loop. This is where the logic of our program lies.

c) What does **pinMode()** do?
Hint: http://arduino.cc/en/Reference/HomePage
This is a function where we specified a pin to behave as an input or an output.

d) What is a comment?
Comment is code documenting. Denote with `//` to tell us what this function or variable or any logic in the code do.

e) What does the following line of code do:
```
delay(x);
```
Hint: http://arduino.cc/en/Reference/HomePage
Function to pause the program execution for a specified period of time.

f) There is something you need to check before uploading your sketch. What is this?
When we try to upload the sketch to the Arduino. First, the Arduino IDE will verify your code. This does not guarantee the code will work as expected, but it will

show that there are no syntax errors, and the program can be compiled. After successfully compiled, the code will then be uploaded to Arduino.

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

We need to first connect to Arduino Nano 33 to our computer. Then we download the correct board manager, choose the right port. Next, we download the sample code called `Blink` and try to upload that to our Arduino. If the code has been successfully uploaded and the built-in LED light is switching at correct interval, then we can conclude that the program is working correctly according to 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

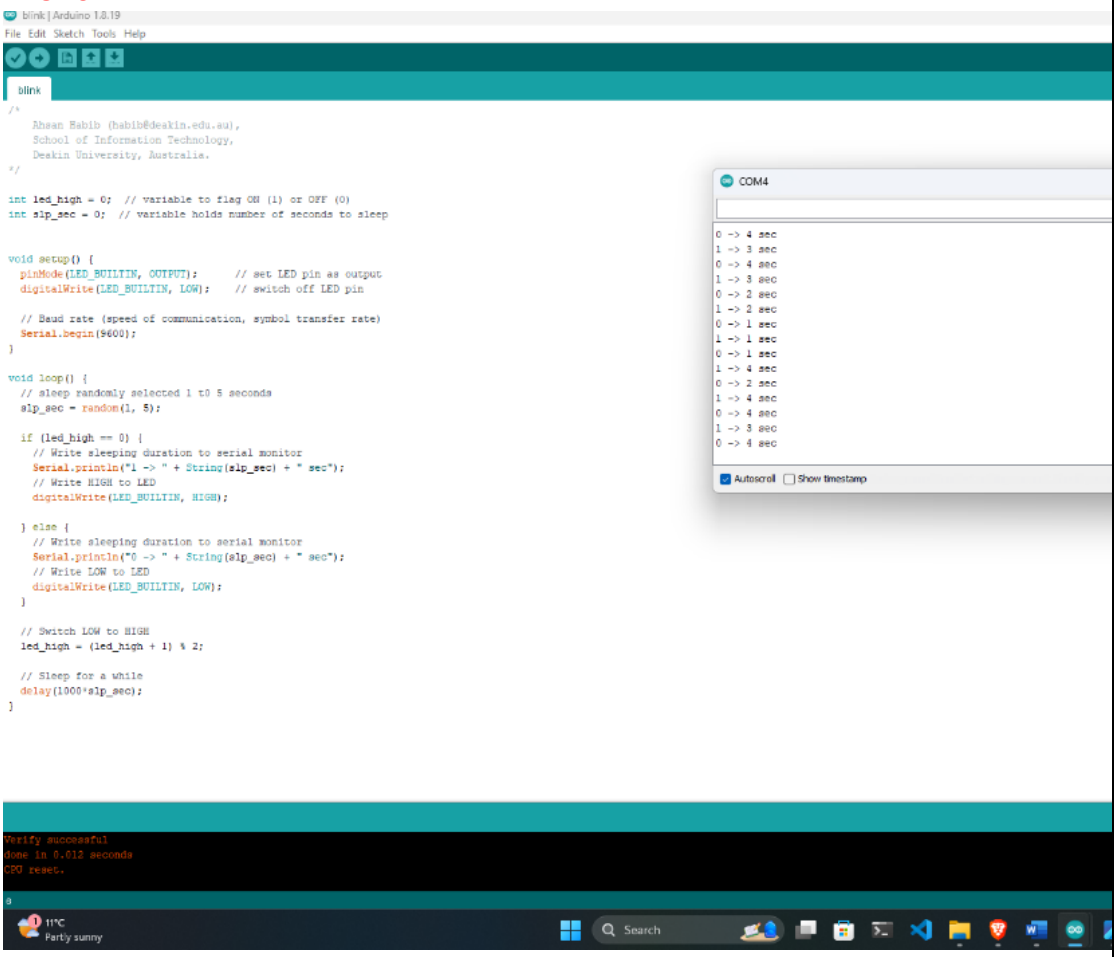
Arduino programming environment

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.

```
int led_high = 0;  // variable to flag ON (1) or OFF (0)
int slp_sec = 0;  // variable holds number of seconds to sleep


void setup() {
  pinMode(LED_BUILTIN, OUTPUT);      // set LED pin as output
  digitalWrite(LED_BUILTIN, LOW);    // switch off LED pin

  // Baud rate (speed of communication, symbol transfer rate)
  Serial.begin(6900);
}

void loop() {
  // sleep randomly selected 1 t0 5 seconds
  slp_sec = random(1, 5);

  if (led_high == 0) {
    // Write sleeping duration to serial monitor
    Serial.println("1 -> " + String(slp_sec) + " sec");
    // Write HIGH to LED
    digitalWrite(LED_BUILTIN, HIGH);

  } else {
    // Write sleeping duration to serial monitor
    Serial.println("0 -> " + String(slp_sec) + " sec");
    // Write LOW to LED
    digitalWrite(LED_BUILTIN, LOW);
  }

  // Switch LOW to HIGH
  led_high = (led_high + 1) % 2;

  // Sleep for a while
  delay(1000*slp_sec);
}

```

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

	Answer:
	 
	blink | Arduino 1.8.19
	File Edit Sketch Tools Help
	blink
	/*
	Ahsan Habib (habib@deakin.edu.au),
	School of Information Technology,
	Deakin University, Australia.
	*/
	int led_high = 0; // variable to flag ON (1) or OFF (0)
	int slp_sec = 0; // variable holds number of seconds to sleep
	void setup() {
	pinMode(LED_BUILTIN, OUTPUT); // set LED pin as output
	digitalWrite(LED_BUILTIN, LOW); // switch off LED pin
	// Baud rate (speed of communication, symbol transfer rate)
	Serial.begin(9600);
	}
	void loop() {
	// sleep randomly selected 1 to 5 seconds
	slp_sec = random(1, 5);
	if (led_high == 0) {
	// Write sleeping duration to serial monitor
	Serial.println("1 -> " + String(slp_sec) + " sec");
	// Write HIGH to LED
	digitalWrite(LED_BUILTIN, HIGH);
	} else {
	// Write sleeping duration to serial monitor
	Serial.println("0 -> " + String(slp_sec) + " sec");
	// Write LOW to LED
	digitalWrite(LED_BUILTIN, LOW);
	}
	// Switch LOW to HIGH
	led_high = (led_high + 1) % 2;
	// Sleep for a while
	delay(1000*slp_sec);
	}
	COM4
	0 -> 4 sec
	1 -> 3 sec
	0 -> 4 sec
	1 -> 3 sec
	0 -> 2 sec
	1 -> 2 sec
	0 -> 1 sec
	1 -> 1 sec
	0 -> 1 sec
	1 -> 4 sec
	0 -> 2 sec
	1 -> 4 sec
	0 -> 4 sec
	1 -> 3 sec
	0 -> 4 sec
	Autoscroll Show timestamp
	Verify successful
	Done in 0.012 seconds
	CPU reset.
	11°C
	Partly sunny
	Search

2	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. Answer: The `Serial.begin(9600);` sets up the serial data communication at 9600 rate in bits per second (baud). The baud rate of serial monitor needs to match `Serial.begin(9600);` for proper communication. There is a random sleep duration between 1 to 4 seconds to control how long the built in LED will stay on and off. The duration is then printed with `Serial.println`.
3	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? Answer: 12 bytes = 8*12 = 96 bits. The speed will be 96/4800 = 0.02 seconds.

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

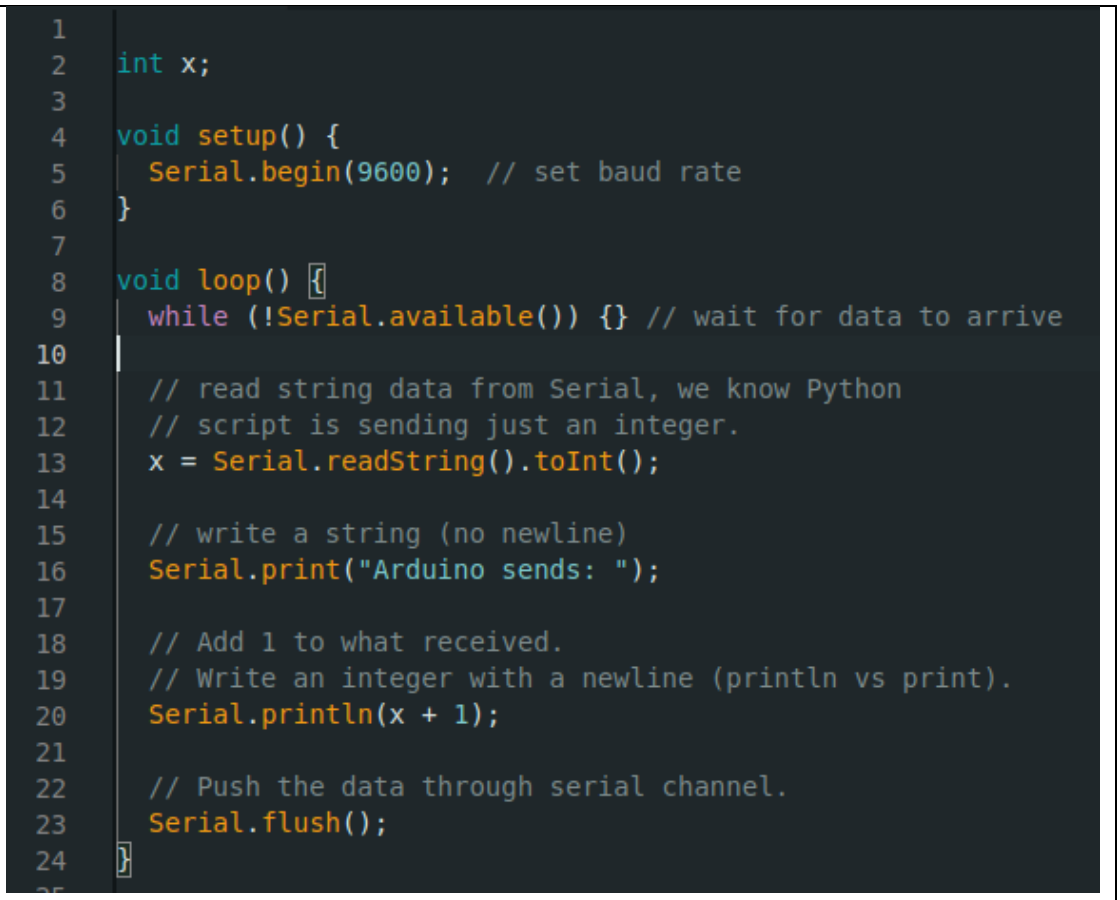
USB cable

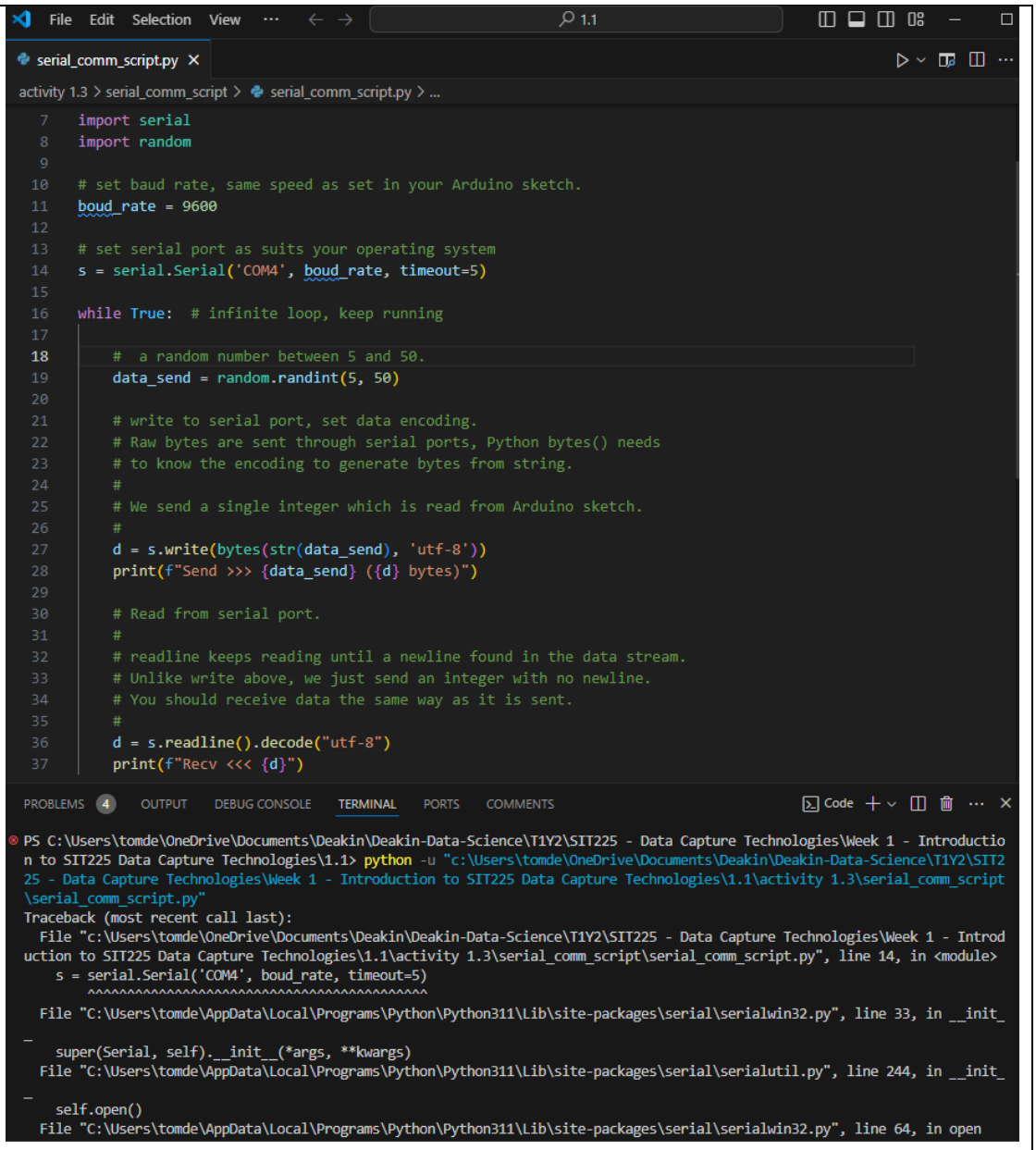
Software Required

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.

	 Question: Do you see any output in serial monitor? If not, then why? Answer: There is no output in the serial monitor because there is no python code that sent data to read.
2	Write Python code as below (or download from https://github.com/deakin-deep-dreamer/sit225/blob/main/week_1/serial_comm_script.py) and save it to a file serial_comm_script.py.



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: The python script when runs receive errors because the Arduino IDE is still running and using the port.

3	Question: Following the above step 2, do you think Arduino IDE is keeping the serial communication port busy talking to Arduino board? Now try to close the Arduino IDE and repeat step 2 above to run the Python script. Do previous errors show up again? If not, Python script should have print messages in the command line. Take the screenshot of the Python script output and paste here. Answer: `PS C:\Users\tomde\OneDrive\Documents\Deakin\Deakin-Data-Science\T1Y2\SIT225 - Data Capture Technologies\Week 1 - Introduction to SIT225 Data Capture Technologies\1.1> python -u "c:\Users\tomde\OneDrive\Documents\Deakin\Deakin-Data-Science\T1Y2\SIT225 - Data Capture Technologies\Week 1 - Introduction to SIT225 Data Capture Technologies\1.1\activity 1.3\serial_comm_script\serial_comm_script.py"` `Send >>> 9 (1 bytes)` `Recv <<< Arduino sends: 10`  `Send >>> 28 (2 bytes)` `Recv <<< Arduino sends: 29`  `Send >>> 27 (2 bytes)` `Recv <<< Arduino sends: 28`  `Send >>> 14 (2 bytes)` `Recv <<< Arduino sends: 15`  `Send >>> 13 (2 bytes)` `Recv <<< Arduino sends: 14`  `Send >>> 8 (1 bytes)` `Recv <<< Arduino sends: 9`  `Send >>> 35 (2 bytes)` `Recv <<< Arduino sends: 36`  `Send >>> 43 (2 bytes)` `Recv <<< Arduino sends: 44`  `Send >>> 34 (2 bytes)`
4	Question: Observe the Python script output and describe the communication protocol used between Arduino sketch and Python script. Answer: Python sends data integer type randomly between 5 to 50. Arduino takes in the data and outputs that random number + 1.

Video demonstration:

https://www.youtube.com/watch?v=SesJEGHjG5k