Student name: Disha Sharma

Student ID: 2410994787

SIT225: Data Capture Technologies

# Activity 8.1: Using smartphone to capture sensor data

The **Arduino IoT Remote** phone application lets you control and monitor all of your dashboards in the Arduino Cloud. With the app, you can also access your phone's internal sensors such as GPS data, light sensor, IMU and more (depending on what phone you have).

The phone's sensor data is automatically stored in Cloud variables, which you can also synchronize with other Things such as custom thing in Python board. This means your phone can become a part of your IoT system, acting as another node in your network.

In this activity, you will enable your smartphone to work as a custom device (like an Arduino board) and connect to your smartphone sensors such as accelerometers and GPS and streaming data to Arduino IoT Cloud dashboard.

## Hardware Required

* Your smartphone – compatible Android or iPhone
* NOTE: *The IoT Remote app requires iOS 12.4 or later for iOS the version. If you are using Android, version 8.0 or later is required. Make sure the iOS or Android version on your device is up to date before downloading the app.*

## Software Required

Android / iOS smart phone.

Arduino account

Arduino IoT Remote App (App Store or Google Play)

Python 3 (for custom Python Thing)

Steps

|  |  |
| --- | --- |
| **Step** | **Action** |
| 1 | **Install App**:  To use the Arduino IoT Remote app, visit Google Play / App Store and search for "Arduino IoT Remote".  After installing the app, you will need to log in to your Arduino account.  After you login, you will discover all your dashboards (if you have any), in the main menu. Based on the app version, home screen may vary. There will be 3 tabs at the bottom – Dashboards, Devices and Activity. You can follow the tutorial (<https://docs.arduino.cc/arduino-cloud/iot-remote-app/getting-started> ). |
| 2 | **Add device**:  Tap into the Devices tab. You will be able to create a new device. Alternatively, you can your profile (top right corner), in the settings section, you will see “Phone as device” which you can turn ON if it is OFF. There, you can select sensors in your smartphone such as accelerometer linear, accelerometer x/y/z and GPS among others.  Note: A free account is enough for this experiment. If you are asked to upgrade your account, you can remove all other Things from your Arduino IoT Cloud account since the Free account allows at most 2 Things to configure, see below image.    The add device wizard will allow you to setup your sensor and also create a dashboard which you can see in your smartphone app. If you login to Arduino IoT Cloud in web browser, you can see the dashboard for your smartphone is already created. |
| 3 | **Keep your smartphone screen ON for a while**:  Keep data coming through your smartphone for 10-15 minutes. During this time, keep moving your smartphone in a pattern so the accelerometer data can be analysed to discover the pattern.  You can download data from the Ardiuno Cloud dashboard page by clicking on a download icon at top right corner which shows – Download historic data. A data download link will be sent to your account email from there you can download data.  Question: Take a screenshot of your Ardiuno Cloud Dashboard where smartphone data is streaming and paste it here.  Answer: |
| 4 | **Plot accelerometer data**:  The zipped data file you downloaded from the cloud contains separate files per variable including accelerometer\_linear, accelerometer\_x, accelerometer\_y, accelerometer\_z and Gps. Each file has 2 columns – time and value.  Question: Open Jupyter Notebook by using command line, go to the data folder and write command ($ jupyter lab). Using Pandas, read CSV file and fetch the data column for accelerometer\_x and plot it using Python plotting library (matplotlib or any other convenient for you). Repeat the plotting process for accelerometer y and z to have 3 separate graphs. Now create a fourth graph with all 3 variables x, y and z. Screenshot the 4 graphs and paste here.  Answer: |
| 5 | Question: Analyse accelerometer variables to find any repeating pattern. Remember that you were repeatedly moving your phone in a single pattern which should be manifested in the graphs. Justify your answer.  Answer: The accelerometer graphs show this repeating pattern, which indicates that the phone has been moved consistently and rhythmically. The individual X, Y, Z graphs fluctuate at regular intervals rather than with random acceleration values, suggesting some structure of motion. The pattern is confirmed more by the combined graph in which all three axes show synchronized changes over time. Since we were moving the phone up and down in a repeating pattern, the readings correspond well with this kind of observation as periodic rises and falls show up in the data. It also implies that the changes are fairly regular and followed a well-predicted cycle, thus reflecting quite accurately that repeated motion has been captured by the accelerometer. |

# Activity 8.2: Receive smartphone sensor data from Python script

You can connect anything to Arduino Cloud including a wide range of compatible Arduino boards such as Arduino Nano 33 IoT or a third-party device that speaks Python. In activity 3.2, you have configured custom Python board and created a cloud variable that was synced to your Arduino Thing such as DHT22 sensor variables.

In this activity, you will need to synchronise smartphone’s accelerometer x, y and z variables to Python script. If you can recall, you have already done a similar function in Activity 3.2.

## Steps:

|  |  |
| --- | --- |
| **Step** | **Action** |
| 1 | Configure Python board in Arduino Cloud and create a Thing where define 3 variables at a time and sync to corresponding accelerometer variable of smartphone Thing. |
| 2 | Write Python script to keep listening to data from the 3 variables to come through. You may need to create 3 call-back functions – a single function per variable (x, y and z). |
| 3 | Question: Keep storing each variable data in a separate file. Append each value with a timestamp so each data reading forms a comma separated line - <timestamp>, <data-value>. New data is written in a separate line. Keep storing them in a CSV file, where there will be 3 separate files. Screenshot your Python script here and screenshot the files opened side-by-side you have created and paste it here.  Answer:    Python script:  import sys  import traceback  import time  from arduino\_iot\_cloud import ArduinoCloudClient  DEVICE\_ID = "8804f551-ab6a-4878-9b3d-9beae644d0e7"  SECRET\_KEY = "nl49dbaA4#Y#hQe8N9JArkMyx"  USERNAME = DEVICE\_ID  FILE\_X = "accelerometer\_x.csv"  FILE\_Y = "accelerometer\_y.csv"  FILE\_Z = "accelerometer\_z.csv"  for file\_path, header in [  (FILE\_X, "Timestamp,AccelerometerX\n"),  (FILE\_Y, "Timestamp,AccelerometerY\n"),  (FILE\_Z, "Timestamp,AccelerometerZ\n")  ]:  with open(file\_path, mode='a', newline='') as file:  file.write(header)  def on\_accelerometer\_x\_changed(client, value):  timestamp = time.strftime("%Y-%m-%d %H:%M:%S")  csv\_string = f"{timestamp},{value}\n"  with open(FILE\_X, mode='a', newline='') as file:  file.write(csv\_string)  file.flush()  print(f"Logged AccelerometerX: {csv\_string.strip()}")  def on\_accelerometer\_y\_changed(client, value):  timestamp = time.strftime("%Y-%m-%d %H:%M:%S")  csv\_string = f"{timestamp},{value}\n"  with open(FILE\_Y, mode='a', newline='') as file:  file.write(csv\_string)  file.flush()  print(f"Logged AccelerometerY: {csv\_string.strip()}")  def on\_accelerometer\_z\_changed(client, value):  timestamp = time.strftime("%Y-%m-%d %H:%M:%S")  csv\_string = f"{timestamp},{value}\n"  with open(FILE\_Z, mode='a', newline='') as file:  file.write(csv\_string)  file.flush()  print(f"Logged AccelerometerZ: {csv\_string.strip()}")  def main():  print("Initializing Arduino Cloud client for accelerometer data...")    client = ArduinoCloudClient(  device\_id=DEVICE\_ID, username=USERNAME, password=SECRET\_KEY  )    client.register(  "accelerometerX", value=None,  on\_write=on\_accelerometer\_x\_changed  )  client.register(  "accelerometerY", value=None,  on\_write=on\_accelerometer\_y\_changed  )  client.register(  "accelerometerZ", value=None,  on\_write=on\_accelerometer\_z\_changed  )    client.start()  if \_\_name\_\_ == "\_\_main\_\_":  try:  main()  except Exception as e:  print("Error occurred:", e)  traceback.print\_exc() |
| 4 | Question: Now manage 3 variable data so they can be stored in a single CSV file where each line consists of comma separated sensor values with a timestamp - <timestamp>, <x>, <y>, <z>. Store data once you gather 3 variables and repeat the process. Screenshot your Python script here and screenshot the file you have created opened and paste it here.  Answer:    import sys  import traceback  import time  from arduino\_iot\_cloud import ArduinoCloudClient  DEVICE\_ID = "8804f551-ab6a-4878-9b3d-9beae644d0e7"  SECRET\_KEY = "nl49dbaA4#Y#hQe8N9JArkMyx"  USERNAME = DEVICE\_ID  FILE\_DATA = "accelerometer\_data.csv"  with open(FILE\_DATA, mode='a', newline='') as file:  file.write("Timestamp,AccelerometerX,AccelerometerY,AccelerometerZ\n")  sensor\_data = {"x": None, "y": None, "z": None}  updated = {"x": False, "y": False, "z": False}  def log\_if\_ready():  """Log data to CSV once all three sensor values have been updated."""  if updated["x"] and updated["y"] and updated["z"]:  timestamp = time.strftime("%Y-%m-%d %H:%M:%S")  line = f"{timestamp},{sensor\_data['x']},{sensor\_data['y']},{sensor\_data['z']}\n"  with open(FILE\_DATA, mode='a', newline='') as f:  f.write(line)  f.flush()  print(f"Logged Data: {line.strip()}")  updated["x"] = updated["y"] = updated["z"] = False  def on\_accelerometer\_x\_changed(client, value):  sensor\_data["x"] = value  updated["x"] = True  log\_if\_ready()  def on\_accelerometer\_y\_changed(client, value):  sensor\_data["y"] = value  updated["y"] = True  log\_if\_ready()  def on\_accelerometer\_z\_changed(client, value):  sensor\_data["z"] = value  updated["z"] = True  log\_if\_ready()  def main():  print("Initializing Arduino Cloud client for accelerometer data...")  client = ArduinoCloudClient(  device\_id=DEVICE\_ID, username=USERNAME, password=SECRET\_KEY  )    client.register(  "accelerometerX", value=None,  on\_write=on\_accelerometer\_x\_changed  )  client.register(  "accelerometerY", value=None,  on\_write=on\_accelerometer\_y\_changed  )  client.register(  "accelerometerZ", value=None,  on\_write=on\_accelerometer\_z\_changed  )    client.start()  if \_\_name\_\_ == "\_\_main\_\_":  try:  main()  except Exception as e:  print("Error occurred:", e)  traceback.print\_exc() |

All the codes and csv files have been uploaded to GitHub. You can access them here: <https://github.com/disha-sharma11/SIT225_2024T2>