

Chancheep Mahacharoensuk 6288092

Kantapong Matangkarat 6288160

Activity 13 embedded system

Part A: NodeRED and MongoDB

Exercise 1: What is the version of your MongoDB?

Your result:

```
cc@cc-VirtualBox:~$ mongo --version
MongoDB shell version v3.6.8
git version: 8e540c0b6db93ce994cc548f000900bdc740f80a
OpenSSL version: OpenSSL 1.1.1f 31 Mar 2020
allocator: tcmalloc
modules: none
build environment:
  distarch: x86_64
  target_arch: x86_64
cc@cc-VirtualBox:~$
```

Exercise 2: Can you check the output of the following command? `mongo --eval 'db.runCommand({ connectionStatus: 1 })'`

Your result:

```
cc@cc-VirtualBox:~$ mongo --eval 'db.runCommand({ connectionStatus: 1 })'
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("052d962a-5b7d-4002-a64a-dca16fd910b5") }
MongoDB server version: 6.0.3
WARNING: shell and server versions do not match
{
  "authInfo" : {
    "authenticatedUsers" : [ ],
    "authenticatedUserRoles" : [ ]
  },
  "ok" : 1
}
cc@cc-VirtualBox:~$
```

Continue the flow from Activity11 with ESP32 and AHT20. Do the following instructions: 1. Install node-red-contrib-mongodb and node-red-node-mongodb at Manage Palette 2. import Activity13-ESP32-AHT20-NodeRED-MongoDB.txt 3. Add "Mongodb out" from Temperature and Humidity with the properties of 4. Find out the following results on your mongoDB. Go to Ubuntu terminal `$ mongo > use Activity13db > db.Temperature.find() > db.Humidity.find()`

Your result:

Part B: ESP32 & Firebase

Chancheep Mahacharoensuk 6288092
Kantapong Matangkarat 6288160

```

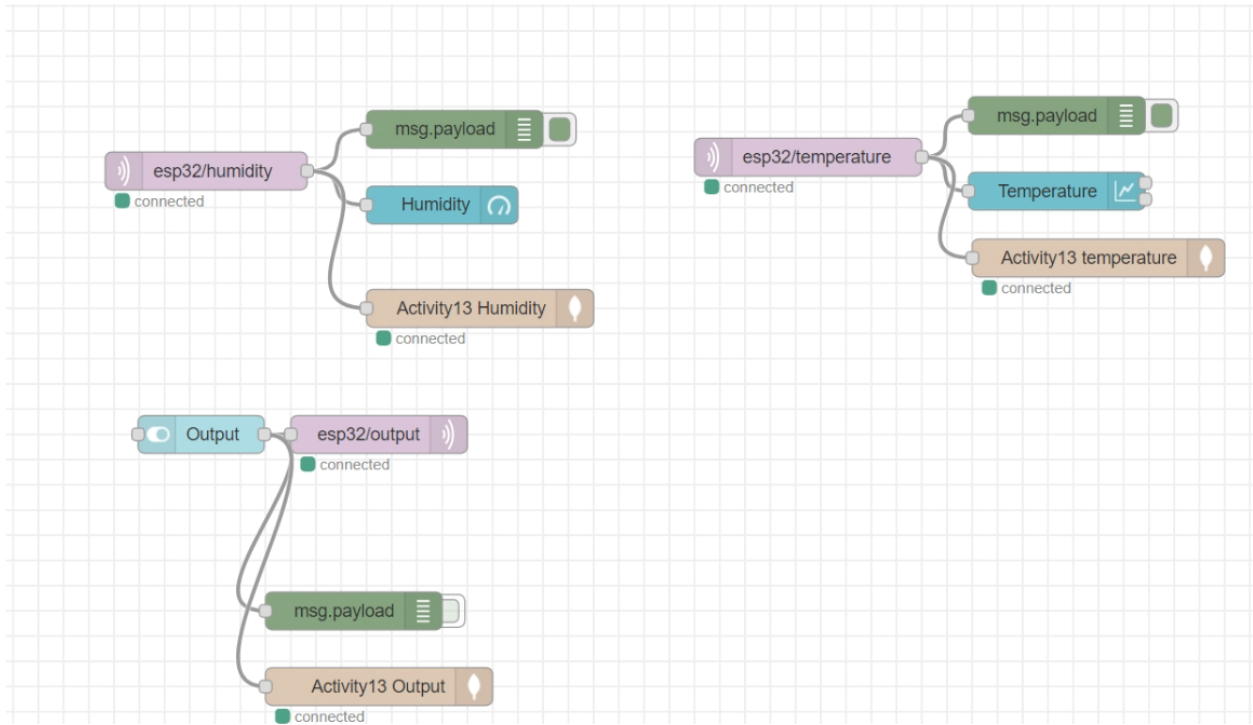
true
test> show collections

test> show dbs
Activity13db  40.00 KiB
admin         40.00 KiB
config        72.00 KiB
local         72.00 KiB
test> use Activity13db
switched to db Activity13db
Activity13db> db.temperature.find()
[
  {
    _id: ObjectId("6375d2de1730e510d909c667"),
    topic: 'esp32/temperature',
    payload: '27.97',
    qos: 0,
    retain: false,
    _msgid: '86e7b888c5f3c0fd'
  },
  {
    _id: ObjectId("6375d2df1730e510d909c668"),
    topic: 'esp32/temperature',
    payload: '27.97',
    qos: 0,
    retain: false,
    _msgid: '6f9c52a563bb7fa1'
  },
  {
    _id: ObjectId("6375d2e01730e510d909c669"),
    topic: 'esp32/temperature',
    payload: '27.96',
    qos: 0,
    retain: false,
    _msgid: '6e4a447bbb0b0304'
  },
  {
    _id: ObjectId("6375d2e11730e510d909c66a"),
    topic: 'esp32/temperature',
    payload: '27.98',
    qos: 0,
    retain: false,
    _msgid: '69f557ac5ebf3c8c'
  },
  {
    _id: ObjectId("6375d2e21730e510d909c66b"),
    _

```

```
Type "it" for more
Activity13db> db.Humidity.find()
[
  {
    _id: ObjectId('6375d3cc1730e510d909c74c'),
    topic: 'esp32/humidity',
    payload: '73.62',
    qos: 0,
    retain: false,
    _msgid: '5d970de42a435970'
  },
  {
    _id: ObjectId('6375d3cd1730e510d909c74e'),
    topic: 'esp32/humidity',
    payload: '73.63',
    qos: 0,
    retain: false,
    _msgid: '714e6e0125f8325e'
  },
  {
    _id: ObjectId('6375d3ce1730e510d909c750'),
    topic: 'esp32/humidity',
    payload: '73.58',
    qos: 0,
    retain: false,
    _msgid: '1df1b11eacebf08e'
  },
  {
    _id: ObjectId('6375d3cf1730e510d909c752'),
    topic: 'esp32/humidity',
    payload: '73.64',
    qos: 0,
    retain: false,
    _msgid: 'cfce0f8a9fb8fdbc'
  },
  {
    _id: ObjectId('6375d3d11730e510d909c754'),
    topic: 'esp32/humidity',
    payload: '73.63',
    qos: 0,
    retain: false,
    _msgid: '84224Fcd59f3a448'
  },
  {
    _id: ObjectId('6375d3d11730e510d909c756'),
    topic: 'esp32/humidity',
    payload: '73.64'
  }
]
```

Chancheep Mahacharoensuk 6288092
Kantapong Matangkarat 6288160



Chancheep Mahacharoensuk 6288092

Kantapong Matangkarat 6288160

Part B: ESP32 & Firebase Process Flow

Your result:

The screenshot displays the Realtime Database interface for the project `https://embeddedfirebase-1263a-default-rtdb.asia-southeast1.firebaseio.com`. The database structure is as follows:

```
UsersData
├── edwHDPkgK5OYVQFCiqR1eRp1Vyx1
│   └── readings
│       └── 0
│           ├── humidity: "72.24"
│           ├── temperature: "28.54"
│           └── timestamp: "0"
```

Below the database view, the Arduino IDE is shown with the code for `firebase_AHT20` in `config.h`. The code includes headers for the AHT20 sensor, WiFi, and Firebase. It defines the API key, email, password, and RTDB URL. The `firebaseData` object is initialized with the RTDB URL.

The serial monitor shows the output of the code, including the upload progress and the sensor readings:

```
13:59:25.873 -> 72.35
14:03:08.445 -> Connecting to WiFi ....192.168.0.102
14:03:11.448 ->
14:03:11.448 -> Token info: type = id token (GITKit token), status = on request
14:03:13.573 -> Token info: type = id token (GITKit token), status = ready
14:03:13.573 -> Getting User UID
14:03:13.573 -> User UID: edwHDPkgK5OYVQFCiqR1eRp1Vyx1
14:03:13.607 -> AHT10 or AHT20 found
14:03:13.640 -> temp 28.54
14:03:13.640 -> humid 72.24
14:03:18.632 -> time: 0
14:03:19.645 -> Set json... ok
14:03:20.691 -> temp 28.53
14:03:20.691 -> humid 72.25
```

Chancheep Mahacharoensuk 6288092

Kantapong Matangkarat 6288160

Part C: ESP32 & ThingSpeak

Your result:

```
File Edit Sketch Tools Help
ESP32_AHT20_ThingSpeak config.h
WiFi.begin(ssid, pass); // connect to an access point
delay(5000);
/* loop until ESP32 can successfully connect to the WiFi */
Serial.printf("Connecting to %s ", ssid);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

/* connection is successful */
Serial.println(" CONNECTED");
Serial.print("IP Address: ");
Serial.println(WiFi.localIP());

/* Initialize ThingSpeak */
ThingSpeak.begin(espClient);
}

void loop()
{
  if ((millis() - lastTime) > timerDelay)
  {
    // Get a new temperature and humidity reading

    COM4
    14:28:04.852 -> temp 28.58
    14:28:04.852 -> humid 72.86
    14:28:05.902 -> temp 28.58
    14:28:05.902 -> humid 72.85
    14:29:03.264 -> AHT10 or AHT20 found
    14:29:08.377 -> Connecting to MyWifi2G CONNECTED
    14:29:08.377 -> IP Address: 192.168.0.102
    14:29:33.261 -> temp: 28.56
    14:29:33.261 -> humid: 72.98
    14:29:34.173 -> Temp and Humid update successful.

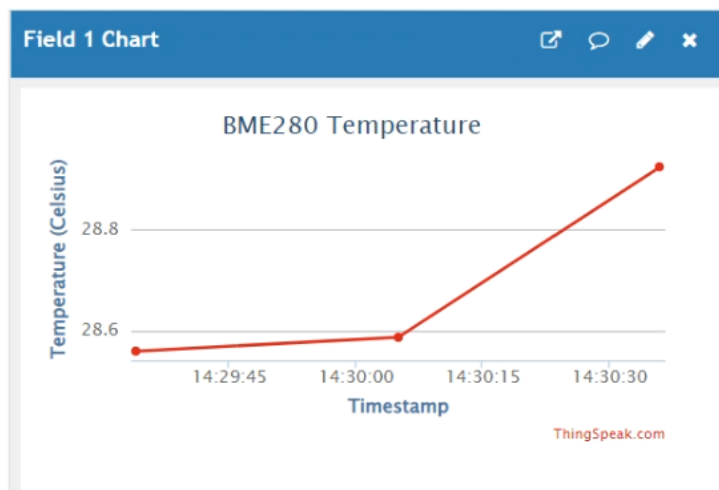
    [x] Autoscroll [x] Show timestamp Newline 115200 baud
    Done uploading.
    Writing at 0x00050000... (68 %)
    Writing at 0x00054000... (72 %)
    Writing at 0x00058000... (76 %)
    Writing at 0x0005c000... (80 %)
    Writing at 0x00060000... (84 %)
    Writing at 0x00064000... (88 %)
    Writing at 0x00068000... (92 %)
    Writing at 0x0006c000... (96 %)
    Writing at 0x00070000... (100 %)
    Wrote 664528 bytes (407209 compressed) at 0x00010000 in 6.9 seconds (effective 768.5 Kbit/s)...
    Hash of data verified.
    Compressed 3072 bytes to 128...
    Writing at 0x00080000... (100 %)
    Wrote 3072 bytes (128 compressed) at 0x00080000 in 0.0 seconds (effective 4096.0 Kbit/s)...
    Hash of data verified.
    Leaving...
    Hard resetting via RTS pin...
```

Channel Stats

Created: [6 minutes ago](#)

Last entry: [less than a minute ago](#)

Entries: 3



Chancheep Mahacharoensuk 6288092

Kantapong Matangkarat 6288160

Sending Multiple Fields (Temperature, and Humidity)

Your result:

```
ESP32_AHT20 ThingSpeak ...
Serial.print("Temp: "); Serial.println(temp.temperature);

Serial.print("Humid: "); Serial.println(humidity.relative_humidity);

Serial.print("Pressure (hPa): "); Serial.println(pressure.pressure);
// Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to st
// pieces of information in a channel. Here, we write to field 1.
ThingSpeak.setField(1, temp.temperature);
ThingSpeak.setField(2, humidity.relative_humidity);
ThingSpeak.setField(3, pressure.pressure);
int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

//int x = ThingSpeak.writeField(myChannelNumber, 1, temp.temperature, myWriteAPIKey);
//int y = ThingSpeak.writeField(myChannelNumber, 2, humidity.relative_humidity, myWriteAPIKey);

//uncomment if you want to get temperature in Fahrenheit
//int x = ThingSpeak.writeField(myChannelNumber, 1, temperatureF, myWriteAPIKey);

if(x == 200)
    Serial.println("Temp Humid and pressure update successful.");
else
    Serial.println("Problem updating Temp and Humid. HTTP error code " + String(X));

lastTime = millis();
}

Done uploading.
Writing at 0x00000000... (68 %)
Writing at 0x00004000... (72 %)
Writing at 0x00008000... (76 %)
Writing at 0x0000c000... (80 %)
Writing at 0x00010000... (84 %)
Writing at 0x00014000... (88 %)
Writing at 0x00018000... (92 %)
Writing at 0x0001c000... (96 %)
```

