MQTT IN ANDROID APPLICATION

***Why MIT app inventor?***

MQTT is very simple to use, and you can build any android application in minutes by just dragging and dropping components. Also, the MIT app inventor has been very popular among the young kids who start with STEM education, as block programming helps them understand the programming concept.

## *Introduction to MIT app inventor*

## MIT app inventor window

## 

#### You can do the back-end development in the block menu, like how a specific button will behave when clicking on it.

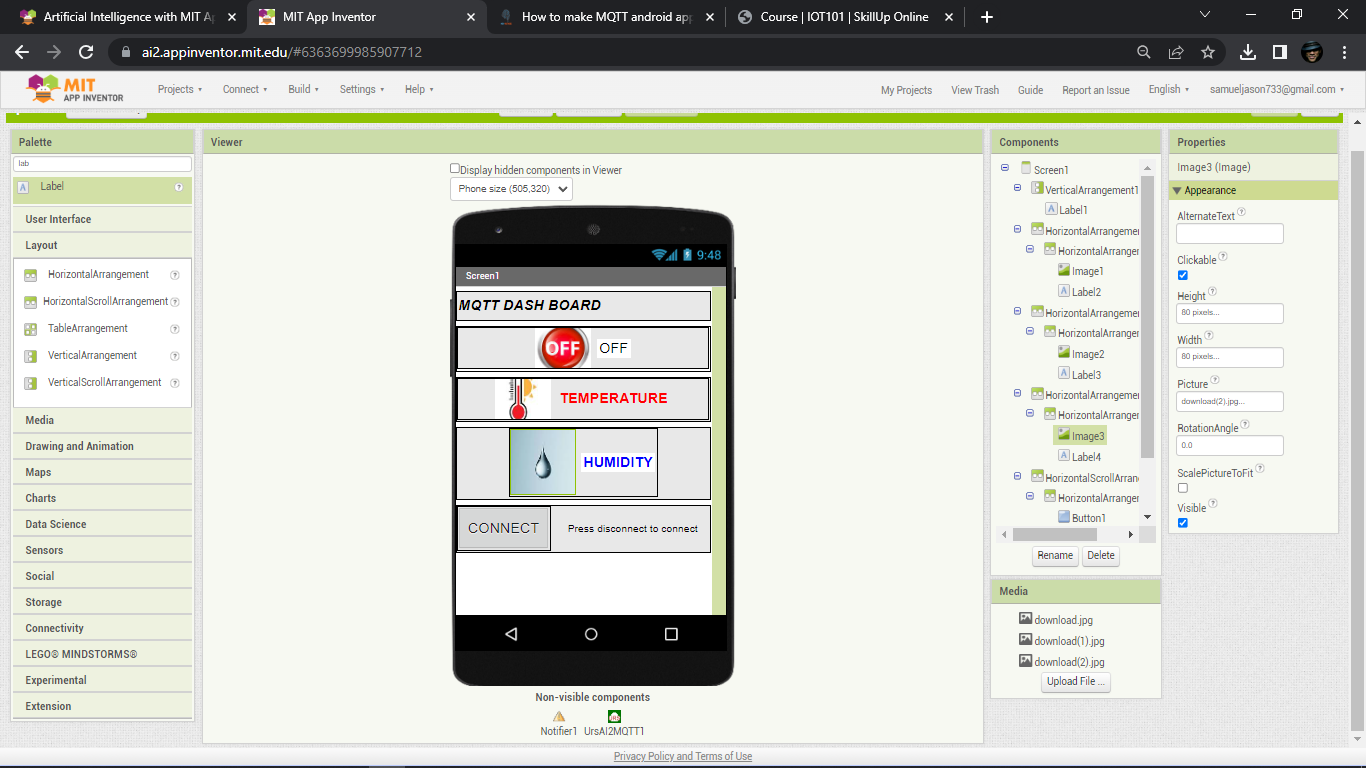
#### 

## *Creating Layout*

## By dragging a vertical layout in our screen, so all the items are in vertical arrangement and set the width of layout to fill-parent.

## we will add a horizontal layout, and inside it, we will add a label to show the title MQTT dashboard, and in the right menu, you can play with parameters to adjust the size, width, height alignment, etc

We need to add a button for connecting to the MQTT broker. So, I will copy the same horizontal layout, and instead of the image, i will put a button. Then rename it to “connect” and change the label text as well.



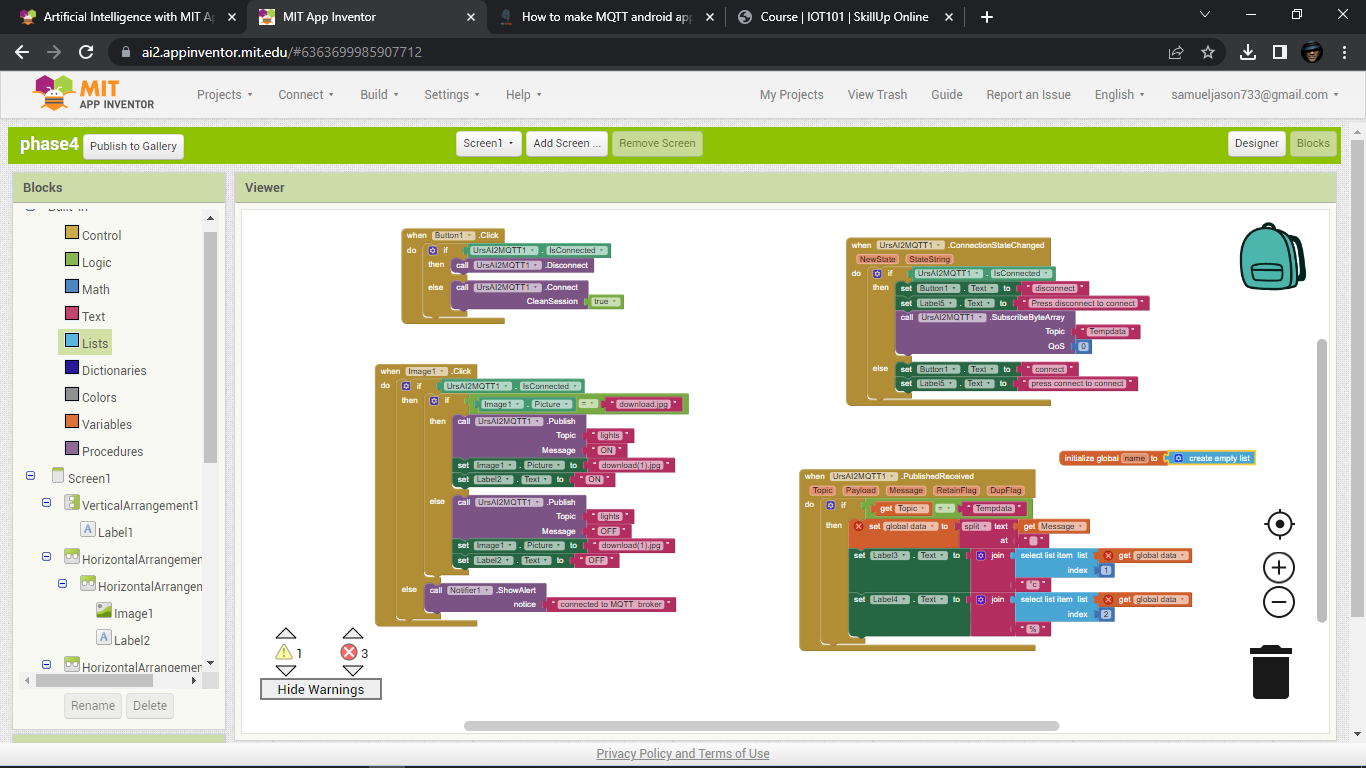
## *Programming MQTT android app*

we have to download the MQTT expansion after the download the file wil it l shown in the zip file extract the file after we upload the file to the mit app inventor and then we have to move to programming in Mqtt android app and it will shows like..

To perform this program first opens your browser and type A12 MQTT expansion

Go to the website and download the package after the download upload the extracted zip file to your screen 1

Here the Link for the file :https://ullisroboterseite.de/android-AI2-PahoMQTT.html

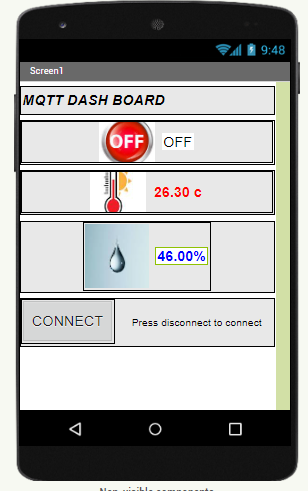


## *Building APK and running it*

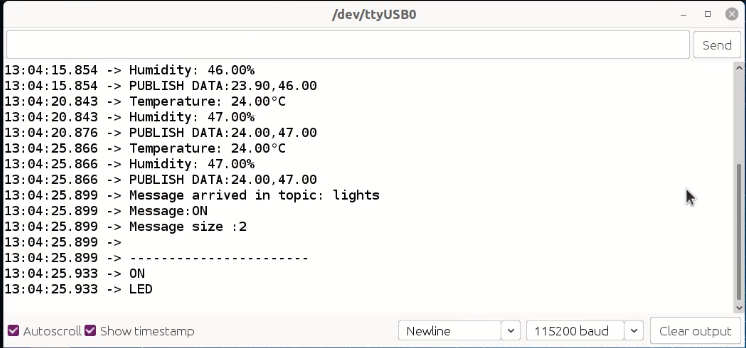
## 

##### **output**

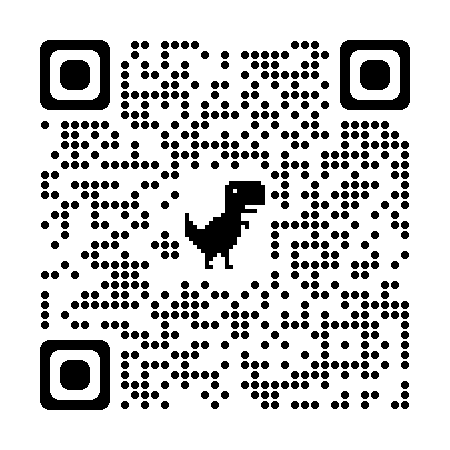
Click on build and then android app (apk) , you will get two options either to dowload it or you can directly download from your android device by scanning the qr ciode



After the installation process connect the probe from ESP32 of humidity and temperature monitoring project to the app then the results will shown like this



***To see my project using MIT app inventor I attached my project Qr code below***

******

## *ESP8266 code*

**#include <Adafruit\_Sensor.h> //Library for Adafruit sensors , we are using for DHT**

**#include <DHT\_U.h> //DHT library which uses some func from Adafruit Sensor library**

**#include <ESP8266WiFi.h> //library for using ESP8266 WiFi**

**#include <PubSubClient.h> //library for MQTT**

**#include <ArduinoJson.h> //library for Parsing JSON**

//defining Pins

**#define DHTPIN 5**

**#define LED D2**

//DHT parameters

**#define DHTTYPE DHT11 // DHT 11**

DHT\_Unified dht(DHTPIN, DHTTYPE);

**uint32\_t** delayMS;

//MQTT Credentials

const **char**\* ssid = "ssid";//setting your ap ssid

const **char**\* password = "password";//setting your ap psk

const **char**\* mqttServer = "iot.reyax.com"; //MQTT URL

const **char**\* mqttUserName = "mqtt username"; // MQTT username

const **char**\* mqttPwd = "mqtt password"; // MQTT password

const **char**\* clientID = "username0001"; // client id username+0001

const **char**\* topic = "Tempdata"; //publish topic

//parameters for using non-blocking delay

unsigned **long** previousMillis = 0;

const **long** interval = 5000;

String msgStr = ""; // MQTT message buffer

**float** temp, hum;

//setting up wifi and mqtt client

WiFiClient espClient;

PubSubClient client(espClient);

**void** setup\_wifi() {

delay(10);

WiFi.begin(ssid, password);

**while** (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

**void** reconnect() {

**while** (!client.connected()) {

**if** (client.connect(clientID, mqttUserName, mqttPwd)) {

Serial.println("MQTT connected");

client.subscribe("lights");

Serial.println("Topic Subscribed");

}

**else** {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

delay(5000); // wait 5sec and retry

}

}

}

//subscribe call back

**void** callback(**char**\*topic, byte\* payload, unsigned **int** length) {

Serial.print("Message arrived in topic: ");

Serial.println(topic);

Serial.print("Message:");

String data = "";

**for** (**int** i = 0; i < length; i++) {

Serial.print((**char**)payload[i]);

data += (**char**)payload[i];

}

Serial.println();

Serial.print("Message size :");

Serial.println(length);

Serial.println();

Serial.println("-----------------------");

Serial.println(data);

**if**(data=="ON"){

Serial.println("LED");

digitalWrite(LED, HIGH);

}

**else**{

digitalWrite(LED, LOW);

}

}

**void** setup() {

Serial.begin(115200);

// Initialize device.

dht.begin();

// get temperature sensor details.

sensor\_t sensor;

dht.temperature().getSensor(&sensor);

dht.humidity().getSensor(&sensor);

pinMode(LED, OUTPUT);

digitalWrite(LED, LOW);

setup\_wifi();

client.setServer(mqttServer, 1883); //setting MQTT server

client.setCallback(callback); //defining function which will be called when message is received.

}

**void** loop() {

**if** (!client.connected()) { //if client is not connected

reconnect(); //try to reconnect

}

client.loop();

unsigned **long** currentMillis = millis(); //read current time

**if** (currentMillis - previousMillis >= interval) { //if current time - last time > 5 sec

previousMillis = currentMillis;

//read temp and humidity

sensors\_event\_t event;

dht.temperature().getEvent(&event);

**if** (isnan(event.temperature)) {

Serial.println(F("Error reading temperature!"));

}

**else** {

Serial.print(F("Temperature: "));

temp = event.temperature;

Serial.print(temp);

Serial.println(F("°C"));

}

// Get humidity event and print its value.

dht.humidity().getEvent(&event);

**if** (isnan(event.relative\_humidity)) {

Serial.println(F("Error reading humidity!"));

}

**else** {

Serial.print(F("Humidity: "));

hum = event.relative\_humidity;

Serial.print(hum);

Serial.println(F("%"));

}

msgStr = String(temp) +","+String(hum);

byte arrSize = msgStr.length() + 1;

**char** msg[arrSize];

Serial.print("PUBLISH DATA:");

Serial.println(msgStr);

msgStr.toCharArray(msg, arrSize);

client.publish(topic, msg);

msgStr = "";

delay(50);

}

}