

# Weekly Progress Report

**Project: TinyML-Based Project on FPGA Board with RISC-V Core**

**Name:** Aman Chauhan

**Branch:** Btech. Computer Science and Engineering (CSE Core)

**Roll Number:** 22BCE0476

**College:** Vellore Institute of Technology, Vellore

**Mentor:** Dr. Sudip Roy

**Week:** 6th report

**Date:** 26th June 2025





# Summary of Week 6 Activities

- Successfully completed the development and training of an OCR (Optical Character Recognition) model using Edge Impulse, specifically for recognizing the digits 0 and 1 and also completed the temperarute ,humidity based anomaly detection based model deployment.
- Uploaded and deployed the trained model onto the ESP32 chip, ensuring compatibility and functionality for edge-based inference.
- Verified the model performance on the ESP32 by running test inputs and confirming that the OCR model correctly recognized 0 and 1, completing the integration from cloud to embedded hardware.



# Data Acquisition(Data Collection)

Screenshot of the Edge Impulse web interface showing a dataset for digit recognition.

The interface includes:

- Toolbar:** Download thank you, CP210x USB to UART, how to run a esp32, (1) WhatsApp, PawanCoder786-pro, Thank you for downl, cam module to use, GitHub - lutzroeder/, NSL-KDD.
- Project Information:** PawanCoder786 / PawanCoder786-project-1 v1 (PERSONAL), Target: Raspberry Pi 5.
- Left Sidebar:** EDGE IMPULSE, Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (Create impulse, Image, Transfer learning, Retrain model, Live classification, Model testing, Deployment), Versioning, Upgrade Plan (Get access to higher job limits and more collaborators, View plans).
- Dataset Overview:** DATA COLLECTED: 52 items, TRAIN / TEST SPLIT: 79% / 21%.
- Dataset View:** Dataset, Training (41), Test (11). Shows six samples of handwritten digits labeled '1'.
- Collect Data:** A section with a 'Connect a device' button and a message: "Click on a sample to load...".
- Raw Data:** A dark blue area where users can click on a sample to load it.
- System Status:** Hot days ahead, 3:45 PM, 6/24/2025.





# Creating Impulse(Transfer Learning)

Screenshot of the Edge Impulse Studio interface showing the creation of an impulse using transfer learning.

The interface includes a sidebar with navigation links such as Dashboard, Devices, Data acquisition, Experiments (selected), EON Tuner, Impulse design, Create impulse, Image, Transfer learning, Retrain model, Live classification, Model testing, Deployment, Versioning, and an Upgrade Plan section.

The main content area displays the "Experiments" tab, showing a single experiment named "Impulse #1". The experiment details are as follows:

NAME	INPUT	DSP BLOCKS	LEARN BLOCKS	F32_V_ACC	F32_T_ACC	IB_V_ACC	IB_T_ACC	F32_LATENCY	F32_FLASH	IB_LATENCY	IB_FLASH
Impulse #1	96x96	Image	Transfer learning	100.0%	100.0%	55.6%	45.5%	2 ms.	1.6M	5 ms.	547.1K

The interface also shows tabs for "Experiments" and "EON Tuner". A message at the bottom states "© 2025 Edgimpulse Inc. All rights reserved".

The taskbar at the bottom of the screen includes icons for File, Search, Task View, Start, Taskbar settings, and a system status bar indicating "Hot days ahead" and the time "3:45 PM".



# Setting the parameters

Download thank you | CP210x USB to UART | how to run a esp32 | (1) WhatsApp | PawanCoder786-project | Thank you for downl | cam module to use | GitHub - lutzroeder/ | NSL-KDD

studio.edgeimpulse.com/studio/725817/impulse/1/create-impulse

Target: Raspberry Pi 5

## Impulse #1

An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

**Image data**

**Input axes**: image

**Image width**: 96    **Image height**: 96

**Resize mode**: Fit shortest axis

**Image**

**Name**: Image

**Input axes (1)**: image

**Transfer Learning (Images)**

**Name**: Transfer learning

**Input features**:  Image

**Output features**: 2 (0, 1)

**Output features**: 2 (0, 1)

**Save Impulse**

**Add a processing block**

**Add a learning block**

**Upgrade Plan**

Get access to higher job limits and more collaborators.

[View plans](#)

© 2025 EdgeImpulse Inc. All rights reserved

Type here to search

Hot days ahead 3:45 PM 6/24/2025



# Features Generation

EDGE IMPULSE

PawanCoder786 / PawanCoder786-project-1 v1 PERSONAL

Target: Raspberry Pi 5

Raw data

Show: All labels 1.5ss0fc64 (1)

Raw features

Parameters

Image

Color depth: RGB

Save parameters

DSP result

Image

Processed features

On-device performance

PROCESSING TIME: 1 ms.

PEAK RAM USAGE: 4 KB

Upgrade Plan

Get access to higher job limits and more collaborators.

Type here to search

Hot days ahead

3:45 PM 6/24/2025

IIT ROORKEE



# Data Training and Retraining Process

Screenshot of the Edge Impulse Studio interface showing the Data Training and Retraining Process.

The interface includes:

- Left Sidebar:** EDGE IMPULSE navigation menu with options like Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design, Create impulse, Transfer learning, Retrain model, Live classification, Model testing, Deployment, Versioning, and Upgrade Plan.
- Central Area:**
  - Neural Network settings:** Training processor set to CPU.
  - Advanced training settings:** Neural network architecture code (Python script) shown below:
- Right Area:**
  - Training output:** Model version: Quantized (Int8). Last training performance (validation set): Accuracy 55.6%, Loss 0.55. Confusion matrix (validation set):

	0	1
0	50%	50%
1	40%	60%
F1 SCORE	0.50	0.60

Metrics (validation set): Area under ROC Curve: 0.55, Weighted average Precision: 0.56, Weighted average Recall: 0.56, Weighted average F1 score: 0.56.
  - Data explorer:** (full training set) showing a scatter plot of data points categorized by correctness.

# Classification Result(Data Exploration)



The screenshot shows the Edge Impulse classification result page. The left sidebar contains navigation links: Dashboard, Devices, Data acquisition, Experiments, EON Tuner, Impulse design (with sub-options Create impulse, Image, Transfer learning), Retrain model, Live classification, Model testing, Deployment, Versioning, and an Upgrade Plan section. The main content area has a title 'Classification result' and a summary card with fields for Name (1.5ss0cs8q) and Label (1). Below this is a table showing category counts: Category 0 (Count 0), Category 1 (Count 1), and uncertain (Count 0). A 'Detailed result' chart shows a distribution between categories 0 and 1, with values 0.14 and 0.86 respectively. To the right, there are two large visualizations: 'Raw data' showing a blurred image of a red digit '1' labeled '1.5ss0cs8q', and 'Image' showing a scatter plot of processed features with points colored by category (0, 1, classified, classification 0).



# Model Testing

studio.edgeimpulse.com/studio/725817/impulse/1/validation

PawanCoder786 / PawanCoder786-project-1 V1 PERSONAL Target: Raspberry Pi 5 P

**EDGE IMPULSE**

Dashboard Devices Data acquisition Experiments EON Tuner Impulse design Create impulse Image Transfer learning Retrain model Live classification Model testing Deployment Versioning

**Upgrade Plan**  
Get access to higher job limits and more collaborators.  
[View plans](#)

This lists all test data. You can manage this data through [Data acquisition](#).

**Test data**

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCOME	ACCURACY	RESULT	⋮
1.5ss0cs8q	1	100%	1.1	⋮
1.5ss0c9f5	1	100%	1.1	⋮
1.5ss0c12d	1	100%	1.1	⋮
1.5ss0a7td	1	100%	1.1	⋮
1.5ss09m6r	1	100%	1.1	⋮
1.5ss09ja6	1	100%	1.1	⋮
0.5ss094s5	0	100%	1.0	⋮
0.5ss08ig7	0	100%	1.0	⋮
0.5ss07udi	0	100%	1.0	⋮
0.5ss06cdm	0	100%	1.0	⋮
0.5srvr7r2	0	100%	1.0	⋮

**Model testing output**

Model version: ② Unoptimized (float32)

**Results**

ACCURACY 100.00%

**Metrics for Transfer learning**

METRIC	VALUE
Area under ROC Curve ②	1.00
Weighted average Precision ②	1.00
Weighted average Recall ②	1.00
Weighted average F1 score ②	1.00

**Confusion matrix**

	0	1	UNCERTAIN
0	100%	0%	0%
1	0%	100%	0%
F1 SCORE	1.00	1.00	

**Feature explorer ②**

0 - correct  
1 - correct



# Final Deployment Process(Arduino Lib.)

Screenshot of the Edge Impulse deployment interface for a project titled "PawanCoder786-project-1 v1".

The interface shows the "Configure your deployment" section where the "Arduino library" is selected for deployment to a Raspberry Pi 5. It also displays "Run this model" options, including a QR code for testing.

On the left sidebar, the "Deployment" section is expanded, showing options like "Create impulse", "Image", "Transfer learning", and "Retrain model".

At the bottom, the Windows taskbar shows the Edge Impulse studio icon, along with other pinned icons for File Explorer, Microsoft Edge, Google Chrome, and the Start button.

Quantized (int8)	IMAGE	TRANSFER LEARNING	TOTAL
Selected	1 ms.	4 ms.	5 ms.
LATENCY	1 ms.	4 ms.	5 ms.
RAM	4.0K	334.7K	334.7K
FLASH	-	536.8K	-
ACCURACY			45.45%

Unoptimized (float32)	IMAGE	TRANSFER LEARNING	TOTAL
Select	1 ms.	1 ms.	2 ms.
LATENCY	1 ms.	1 ms.	2 ms.
RAM	4.0K	893.7K	893.7K
FLASH	-	1.6M	-
ACCURACY			100.00%



# Zip File Creation(TFlite Model)

The screenshot shows a Windows File Explorer window with the following details:

**File Explorer Title Bar:** Extract PawanCoder786-project-1\_inferencing

**Left Navigation Bar:** Quick access, Desktop, Downloads, Documents, Pictures, geometry\_plots, plots, Screenshots, Simulation\_2x3, OneDrive, This PC, 3D Objects, Desktop, Documents, Downloads, arduino-ide\_ni, COMSOL MATI, CP210x\_Windo, Geometry2x3, Python, Simulation\_2x2, Simulation\_2x4, Simulation\_2x1, Simulation\_3x1, Simulation\_3x2, Simulation\_3x4, Simulation\_4x1, Simulation\_4x2, Simulation\_4x4, Velocity2x3, VS CODE, arduino-ide\_ni, CP210x\_Windo, ei-pawancoder.

**Current Path:** This PC > Downloads > ei-pawancoder786-project-1-arduino-1.0.6.zip > PawanCoder786-project-1\_inferencing > src > tflite-model

**File List:**

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
tflite_learn_2_compiled.cpp	CPP File	707 KB	No	1,948 KB	64%	6/20/2025 9:51 AM
tflite_learn_2_compiled.h	H File	2 KB	No	3 KB	57%	6/20/2025 9:51 AM
trained_model_ops_define.h	H File	2 KB	No	5 KB	79%	6/20/2025 9:51 AM



# Adruino Code for ESP32 Tiny ML Model

esp32\_camera | Arduino IDE 2.3.7-nightly-20250624

File Edit Sketch Tools Help

ESP32 Dev Module

esp32\_camera.ino

```
27 #include <PawanCoder786-project-1_inferencing.h>
28 #include "edge-impulse-sdk/dsp/image/image.hpp"
29
30 #include "esp_camera.h"
31
32 // Select camera model - find more camera models in camera_pins.h file here
33 // https://github.com/espressif/arduino-esp32/blob/master/libraries/ESP32/examples/Camera/CameraWebServer/camera\_pins.h
34
35 #define CAMERA_MODEL_ESP_EYE // Has PSRAM
36 // #define CAMERA_MODEL_AI_THINKER // Has PSRAM
37
38 #if defined(CAMERA_MODEL_ESP_EYE)
39 #define PWDN_GPIO_NUM     -1
40 #define RESET_GPIO_NUM   -1
41 #define XCLK_GPIO_NUM     4
42 #define SIOD_GPIO_NUM    18
43 #define SIOC_GPIO_NUM    23
44
45 #define Y9_GPIO_NUM       36
46 #define Y8_GPIO_NUM       37
47 #define Y7_GPIO_NUM       38
48 #define Y6_GPIO_NUM       39
49 #define Y5_GPIO_NUM       35
50 #define Y4_GPIO_NUM       14
51 #define Y3_GPIO_NUM       13
52 #define Y2_GPIO_NUM       34
53 #define VSYNC_GPIO_NUM    5
54 #define HREF_GPIO_NUM     27
55 #define PCLK_GPIO_NUM     25
56
57 #elif defined(CAMERA_MODEL_AI_THINKER)
58 #define PWDN_GPIO_NUM     32
59 #define RESET_GPIO_NUM   -1
60 #define XCLK_GPIO_NUM     0
61 #define SIOD_GPIO_NUM    26
62 #define SIOC_GPIO_NUM    27
63
64 #define Y9_GPIO_NUM       35
65 #define Y8_GPIO_NUM       34
66 #define Y7_GPIO_NUM       39
67 #define Y6_GPIO_NUM       36
68 #define Y5_GPIO_NUM       21
69 #define Y4_GPIO_NUM       19
70 #define Y3_GPIO_NUM       18
```

Output Serial Monitor X

Message (Enter to send message to 'ESP32 Dev Module' on COM4)

New Line 115200 baud Ln 112, Col 7 ESP32 Dev Module on COM4 C 2

Type here to search 34°C Haze 3:49 PM 6/24/2025

Adding the Model in the form of a zip add library



# Compiling the Sketch

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch\_jun25a | Arduino IDE 2.3.7-nightly-20250624
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Includes icons for Open, Save, Print, and others.
- Library Manager:** Shows installed libraries:
  - ESP32Servo** by Kevin Harrington, John K. Bennett (Version 3.0.8, Installed)
  - ESP32Servo360** by Sébastien Matos (Initial development release Control Parallax Feedback 360° High Speed Servos with a ESP32 dev-board, Version 0.2.2, Not Installed)
  - ESP32ServoController** by MDO (A PWM and servo library for ESP32 platform, Version 0.6.0, Not Installed)
- Sketch Editor:** Displays the code for `sketch_jun25a.ino`. The code includes headers for `<ESP32Servo.h>`, `<DHT.h>`, and `<soumadipm-project-1_inferencing.h>`. It defines pins for DHT and servo, initializes serial communication at 115200 bps, starts the DHT sensor, attaches the servo, and prints a welcome message. The `loop()` function handles serial input and reads sensors.

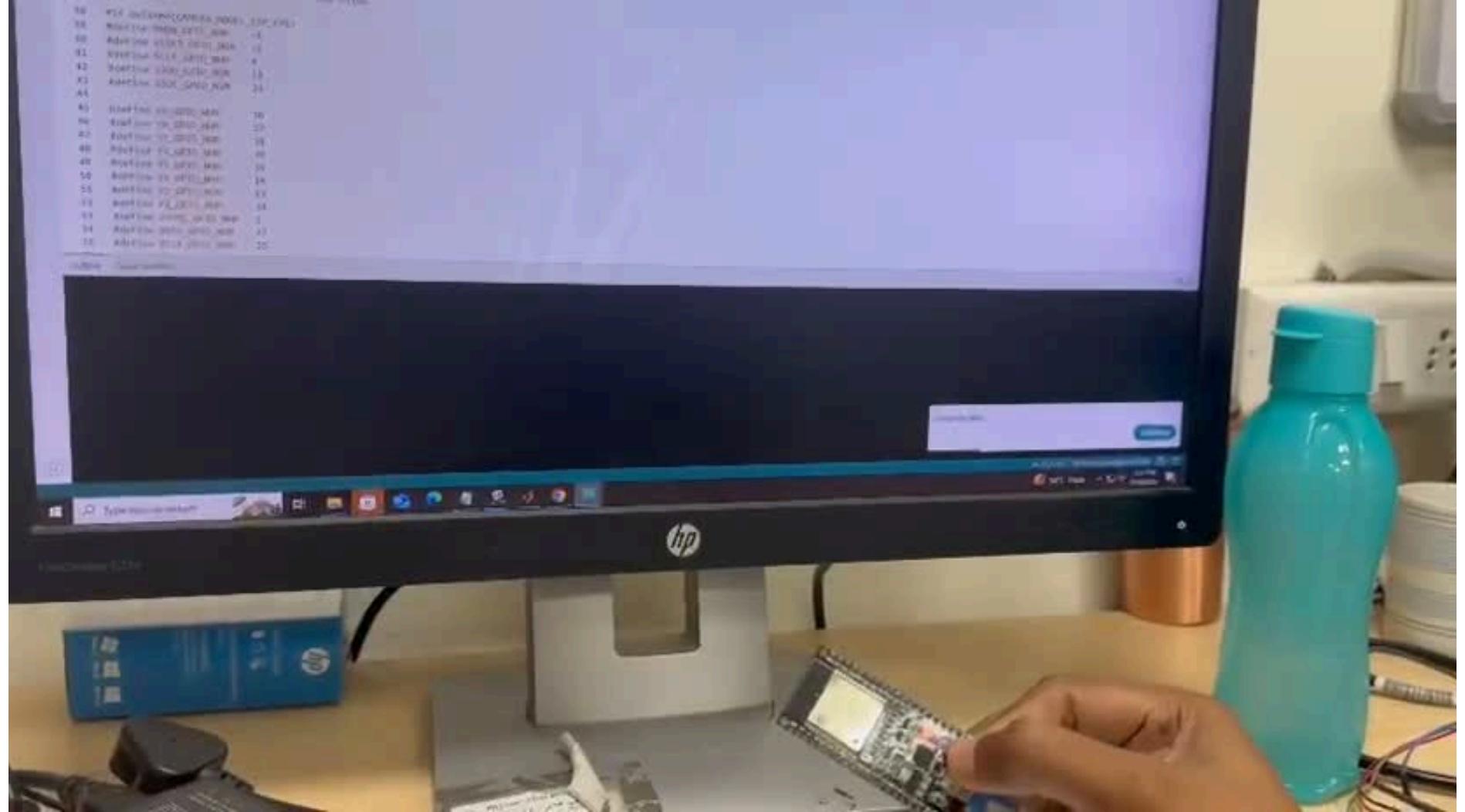
```
1 #include <ESP32Servo.h>
2 #include <DHT.h>
3 #include <soumadipm-project-1_inferencing.h> // Keep only your project header
4
5 #define DHTPIN 14
6 #define DHTTYPE DHT11
7 #define SERVOPIN 18
8
9 DHT dht(DHTPIN, DHTTYPE);
10 Servo myservo;
11
12 // Edge Impulse variables
13 static float ei_features[2]; // [temperature, humidity]
14
15 void setup() {
16     Serial.begin(115200);
17     delay(1000);
18     dht.begin();
19     myservo.attach(SERVOPIN);
20     myservo.write(0);
21     Serial.println("System Ready. Send 'q' to sweep servo.");
22 }
23
24 void loop() {
25     handleSerial();
26     readSensors();
27 }
28
29 void handleSerial() {
```
- Output Panel:** Shows "Serial Monitor". A progress bar indicates the sketch is being compiled.
- Taskbar:** Shows the Windows Start button, a search bar, and various pinned application icons. The system tray shows the date and time as 6/25/2025, 3:37 PM, with a temperature of 33°C and a cloudy weather icon.

Not able to Get Output through the Because of NO cam Module

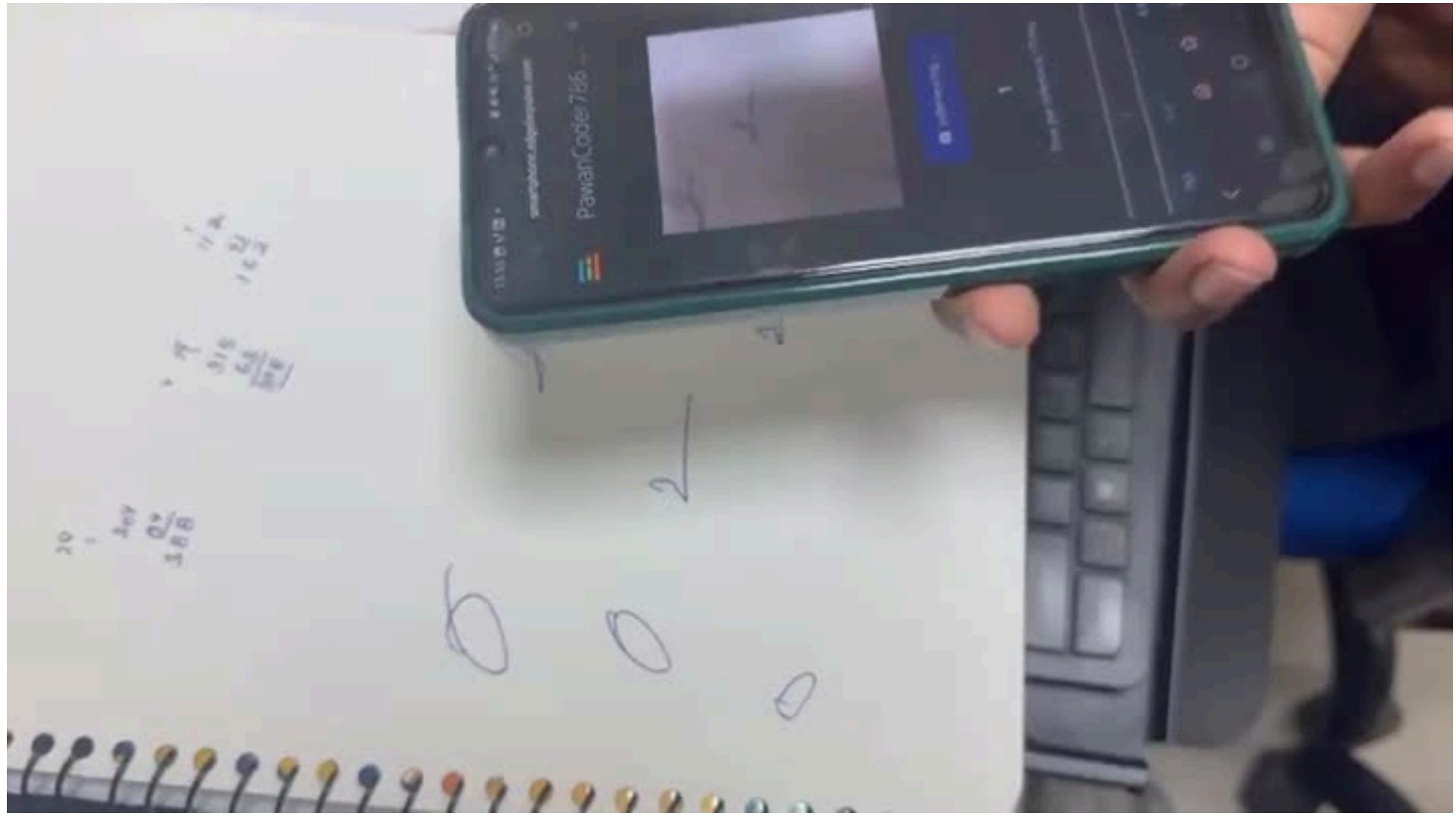
# Uploading the Code in ESP32 DEV Module



10	#12_jeanphilippeGARRIGUE_20091112_0001.XML	
11	Reactive_0001_0001.XML	-1
12	Reactive_0002_0001.XML	1
13	Reactive_0003_0001.XML	1
14	Reactive_0004_0001.XML	24
15	Reactive_0005_0001.XML	10
16	Reactive_0006_0001.XML	29
17	Reactive_0007_0001.XML	18
18	Reactive_0008_0001.XML	10
19	Reactive_0009_0001.XML	10
20	Reactive_0010_0001.XML	10
21	Reactive_0011_0001.XML	14
22	Reactive_0012_0001.XML	22



# Demonstration on Android Phone(Secondary Cam)





# Anomaly Detection Project

## Summary of Anomaly Detection Project Using DHT11 Sensor

- Data Collection and Categorization: Real-time temperature and humidity data were collected from the DHT11 sensor and categorized using AI tools to build a dataset of approximately 100 samples for analysis.
- Model Training with Edge Impulse: The categorized data was used to train a K-means clustering model in Edge Impulse, enabling the identification of normal and anomalous patterns in the sensor readings.
- Deployment and Anomaly Detection: The trained model was deployed on an Arduino, where it processes live sensor data and flags anomalies in temperature and humidity in real time, enhancing the reliability of environmental monitoring.



# Data Collection Through the Sensors

The screenshot shows the Edge Impulse web interface for dataset management. On the left, a sidebar lists various project management and development tools. The main area displays a summary of data collection (1m 38s total, 82% train / 18% test split), a dataset table with 80 training and 18 test samples, and a 'Collect data' section prompting to connect a device.

SAMPLE NAME	LABEL	ADDED
data.s99	anomaly	Yesterday, 15:...
data.s98	anomaly	Yesterday, 15:...
data.s97	anomaly	Yesterday, 15:...
data.s96	anomaly	Yesterday, 15:...
data.s94	anomaly	Yesterday, 15:...
data.s92	anomaly	Yesterday, 15:...
data.s93	anomaly	Yesterday, 15:...
data.s91	anomaly	Yesterday, 15:...

**Data Acquisition and Servo Control for Anomaly Detection Using DHT11**



# CSV FILE CREATED

nestamp,temperature,humidity,label  
29.50,33.00,normal  
29.40,33.00,normal  
29.40,34.00,normal  
29.40,34.00,normal  
29.40,33.00,normal  
29.30,34.00,normal  
29.30,34.00,normal  
29.30,34.00,normal  
29.30,34.00,normal  
,29.30,34.00,normal  
,29.30,34.00,normal  
,29.30,34.00,normal  
,29.30,34.00,normal  
,29.30,34.00,normal  
,29.30,34.00,normal  
.29.30.34.00.normal

**Dataset Creation  
Using DHT11 Sensor  
and Refinement with  
Perplexity Pro**



# Impulse Creation

soumadipm-project-1 - Create X (280) WhatsApp X + studio.edgeimpulse.com/studio/728890/impulse/1/create-impulse ☆ New Chrome available :

soumadipm / soumadipm-project-1 PERSONAL Target: Espressif ESP-EYE... S

## EDGE IMPULSE

Impulse #1

An impulse takes raw data, uses signal processing to extract features, and then uses a learning block to classify new data.

**Features** Input features (2) temperature, humidity

**Raw Data** Name: Raw data  
Input axes (2):  temperature,  humidity

**Anomaly Detection (K-means)** Name: Anomaly detection  
Input features:  Raw data  
Output features: 1 (Anomaly score)

**Output features** 1 (Anomaly score)

**Save Impulse**

**Upgrade Plan**  
Get access to higher job limits and more collaborators.



# Raw Features Of the data

soumadipm-project-1 - Raw c x (280) WhatsApp x | + studio.edgeimpulse.com/studio/728890/impulse/1/dsp/raw/2 New Chrome available :

EDGE IMPULSE

soumadipm / soumadipm-project-1 PERSONAL Target: Espressif ESP-EYE...

Parameters Generate features

Raw data

KEY	VALUE
temperature	29.1
humidity	87

Show: All label: data.s99 (anom:)

Raw features 29.1000, 87.0000

DSP result

Processed features 29.1000, 87.0000

Parameters

Scaling

Scale axes 1

Save parameters

On-device performance

PROCESSING TIME 1 ms.

PEAK RAM USAGE 8 Bytes

Upgrade Plan

Get access to higher job limits and more collaborators.



# Anomaly Detection

studio.edgeimpulse.com/studio/728890/impulse/1/learning/anomaly/3

soumadipm / soumadipm-project-1 PERSONAL Target: Espressif ESP-EYE... S

**EDGE IMPULSE**

- Dashboard
- Devices
- Data acquisition
- Experiments
- EON Tuner
- Impulse design
- Create impulse
- Raw data
- Anomaly detection
- Retrain model
- Live classification
- Model testing

**Anomaly detection settings**

Cluster count: 32

Axes: Select feature axes to include in model training.  
Select suggested axes

temperature  
humidity

**Training output**

**Anomaly explorer**

X Axis: temperature  
Y Axis: humidity  
Test data: -- No test data --

● training data

Get access to higher job limits  
and more collaborators.

## Upgrade Plan



# Retrain the Model

studio.edgeimpulse.com/studio/728890/impulse/1/retrain

soumadipm / soumadipm-project-1 PERSONAL Target: Espressif ESP-EYE... 5

**EDGE IMPULSE**

- Dashboard
- Devices
- Data acquisition
- Experiments
- EON Tuner
- Impulse design
  - Create impulse
  - Raw data
  - Anomaly detection
  - Retrain model
  - Live classification
  - Model testing

**Retrain model with known parameters**

- ✓ Raw data
- ✓ Anomaly detection
- ✓ Model testing

**Train model**

**Build output**

Classifying data for Anomaly detection...  
Classifying data...  
✓ Job scheduled at 26 Jun 2025 08:36:39  
[spinner-done] Job started at 26 Jun 2025 08:36:39  
Classifying data for Anomaly detection OK

Generating model testing summary...  
Finished generating model testing summary  
Running model testing OK

Job completed (success)

© 2025 EdgeImpulse Inc. All rights reserved

**Upgrade Plan**  
Get access to higher job limits  
and more collaborators.



# Live Classification (Data Exploration)

studio.edgeimpulse.com/studio/728890/impulse/1/classification?sampleId=1996901934

New Chrome available :

**EDGE IMPULSE**

Dashboard

Devices

Data acquisition

Experiments

EON Tuner

Impulse design

- Create impulse
- Raw data
- Anomaly detection
- Retrain model
- Live classification
- Model testing

Upgrade Plan

Get access to higher job limits and more collaborators.

**Summary**

Model version: Unoptimized (float32)

Name: data.s100

Label: anomaly

CATEGORY	COUNT
anomaly	0
no anomaly	1

**Detailed result**

Show only unknowns

TIMESTAMP	ANOMALY
0	0.06

**RAW DATA**

**data.s100**

KEY	VALUE
temperature	28.6
humidity	89

**Raw features**

28.6000, 89.0000

**Raw data**

Legend:

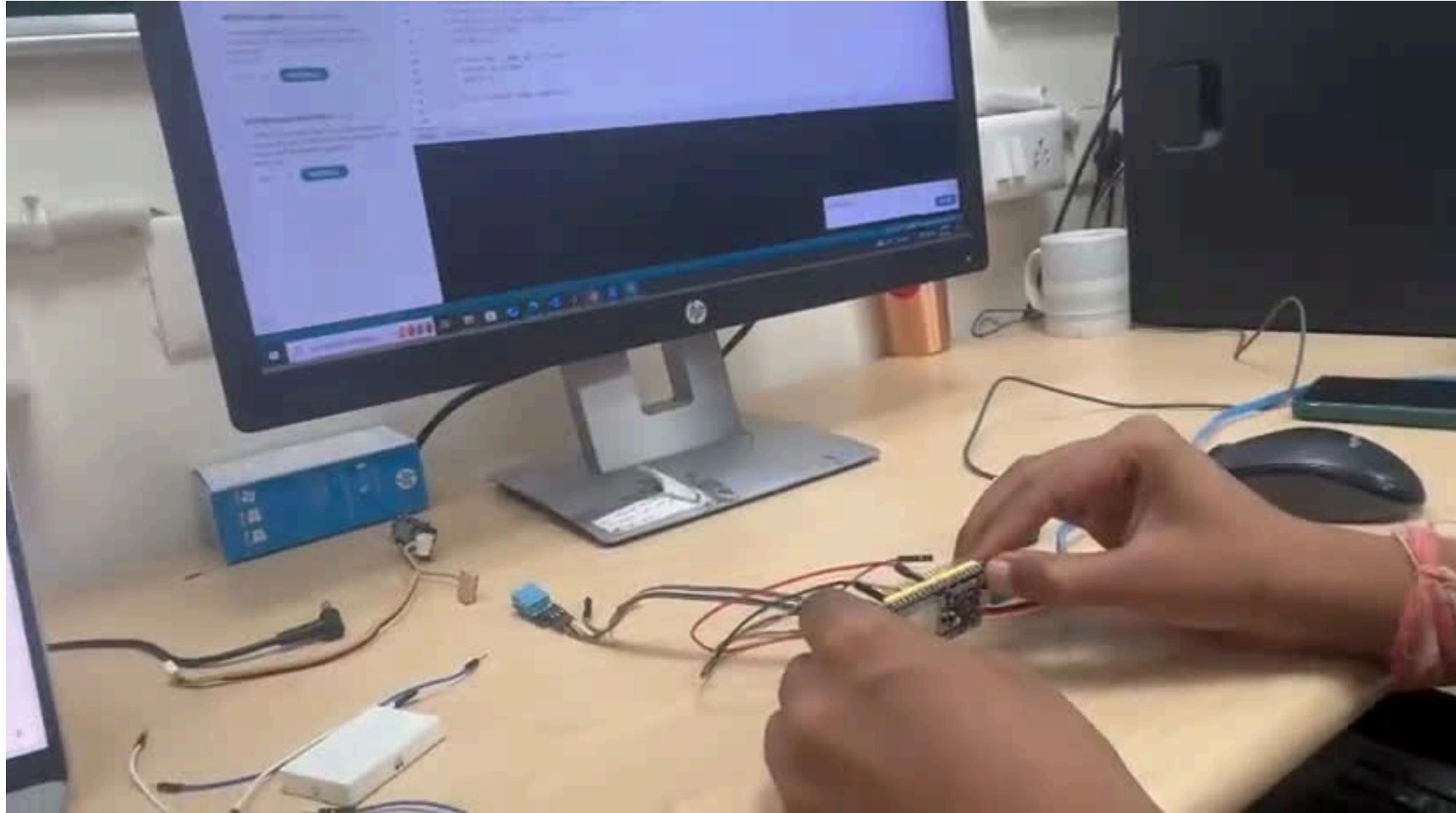
- anomaly (blue)
- classified (orange)
- normal (green)
- classification 0 (cyan)

**Processed features**

28.6000, 89.0000

**Anomaly explorer**

# Deployment and Testing(Demonstration)





# Output through the Model

SKECHES | PROJECTS | LIBRARIES | TUTORIALS | EDUCATION

File Edit Sketch Tools Help

ESP32 Dev Module

LIBRARY MANAGER

ESP32Servo

Type: All Topic: All

**ESP32Servo** by Kevin Harrington, John K. Bennett  
3.0.8 installed  
Allows ESP32 boards to control servo, tone and analogWrite motors using Arduino semantics. This library can control many types of servos. <br />...  
More info

3.0.8 REMOVE

**ESP32Servo360** by Sébastien Matos  
Initial development release Control Parallax Feedback 360° High Speed Servos with a ESP32 dev-board.  
More info

0.2.2 INSTALL

**ESP32ServoController** by MDO  
A [PWM and servo] library for ESP32 platform. Follows the ledc architecture from Espressif. Auto scales with hardware capabilities.  
More info

0.6.0 INSTALL

sketch\_jun25a.ino

```
16 void setup() {
17   Serial.begin(115200);
18   delay(1000); // Give time for Serial Monitor to connect
19   dht.begin();
20   myservo.attach(SERVO_PIN);
21   myservo.write(0);
22   lastTemp = dht.readTemperature();
23   lastHumidity = dht.readHumidity();
24   Serial.println("Ready. Send 'q' to sweep servo!");
25 }
26
27 void loop() {
28   // Check for 'q' command from Serial Monitor
29   if (Serial.available()) {
30     char c = Serial.read();
31     if (c == 'q' || c == 'Q') {
32       Serial.println("Received 'q'. Sweeping servo from 0 to 180 and back.");
33       // Sweep from 0 to 180 and back to 0 (standard servo)
34       for (int pos = 0; pos <= 180; pos += 5) {
35         myservo.write(pos);
36         delay(15);
37       }
38       for (int pos = 180; pos >= 0; pos -= 5) {
39         myservo.write(pos);
40         delay(15);
41       }
42       Serial.println("Sweep complete.");
43     }
44   }
}
```

Output Serial Monitor X

Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')

Temperature: 29.00 °C Humidity: 33.00 %  
Temperature: 28.90 °C Humidity: 32.00 %  
Temperature: 29.00 °C Humidity: 33.00 %  
Temperature: 29.00 °C Humidity: 33.00 %  
Temperature: 28.90 °C Humidity: 32.00 %  
Temperature: 29.00 °C Humidity: 35.00 %  
Temperature: 29.00 °C Humidity: 44.00 %  
Temperature: 29.00 °C Humidity: 46.00 %  
Temperature: 30.70 °C Humidity: 95.00 %  
Temperature increased! Servo at 180 degrees.  
Temperature: 30.70 °C Humidity: 95.00 %

Ln 81, Col 2 ESP32 Dev Module on COM4 6 Oklahoma City Thur... 3:43 PM



# TFLITE MODEL AND MODEL LINK

Screenshot of the Arduino IDE showing the ESP32 library manager and serial monitor.

**ESP32Servo** by Kevin Harrington, John K. Bennett  
3.0.8 installed  
Allows ESP32 boards to control servo, tone and analogWrite motors using Arduino semantics. This library can control a many types of servos.  
More info  
3.0.8 **REMOVE**

**ESP32Servo360** by Sébastien Matos  
Initial development release Control Parallax Feedback 360° High Speed Servos with a ESP32 dev-board.  
More info  
0.2.2 **INSTALL**

**ESP32ServoController** by MDO  
A [PWM and servo] library for ESP32 platform. Follows the ledc architecture from Espressif. Auto scales with hardware capabilities.  
More info  
0.6.0 **INSTALL**

```
20 myservo.attach(SERVOPIN);
21 myservo.write(0);
22 lastTemp = dht.readTemperature();
23 lastHumidity = dht.readHumidity();
24 Serial.println("Ready. Send 'q' to sweep servo!");
25 }

26 void loop() {
27 // Check for 'q' command from Serial Monitor
28 if (Serial.available()) {
29     char c = Serial.read();
30     if (c == 'q' || c == 'Q') {
31         Serial.println("Received 'q'. Sweeping servo from 0 to 180 and back.");
32         // Sweep from 0 to 180 and back to 0 (standard servo)
33         for (int pos = 0; pos <= 180; pos += 5) {
34             myservo.write(pos);
35             delay(15);
36         }
37         for (int pos = 180; pos >= 0; pos -= 5) {
38             myservo.write(pos);
39             delay(15);
40         }
41         Serial.println("Sweep complete.");
42     }
43 }
44 }
```

Output Serial Monitor X

Message (Enter to send message to 'ESP32 Dev Module' on 'COM4')  
Temperature: 29.00 °C Humidity: 33.00 %  
Temperature: 28.00 °C Humidity: 32.00 %  
Temperature: 29.00 °C Humidity: 33.00 %  
Temperature: 29.00 °C Humidity: 33.00 %

This PC

- 3D Objects
- Desktop
- Documents
- Downloads
- arduino-ide\_ni
- COMSOL MATR
- CP210x\_Windo
- GammaRay\_2



# References

- **RVfpga HarvardX edX** and **RVfpga source code**
- TinyML foundational courses and **TensorFlow Lite Micro documentation.**
- **WorldQuant University Applied AI Lab content**
- RVfpga simulator **Vibodo,Rvfpga Nexys Board,Piplines,Whisper**(Debugging and simulation) and Tracer (**GTK wave** usage), then **whisper for c program files compilation.**
- **Report and internal study materials.**
- **To complete for hardware Devices.**

**THANK**  

---

**S**