PROJECT IDEAS · M.TECH · IOT AND EMBEDDED SYSTEMS

M.Tech projects on *embedded systems and IoT*

These 14 M.Tech projects on embedded systems and IoT each start from an open-access IEEE paper published from 2023 to 2026, and run on boards you can buy in India, such as the ESP32, STM32 and Raspberry Pi. Each card names **one change to measure on the board** and the comparison table your thesis would contain.

Last updated 2026-11-12

Every paper on this page was checked against its DOI record on Crossref on 12 November 2026, so the title, first author, journal and year are as IEEE registered them, and all 15 are open access. For our M.Tech service for ECE students, see M.Tech projects for ECE. For VLSI, antennas and wireless papers, see the IEEE base papers for ECE; for simpler builds for B.Tech and diploma, see our IoT project ideas.

What makes a good M.Tech project in IoT or embedded systems?

Three things: a published method you rebuild, hardware you can buy, and one change you measure on that hardware.

1. **A base paper you can reproduce.** Pick a recent, open-access IEEE paper whose data is public or which you can collect yourself. Our guide on how to select a base paper covers the checks.
2. **Hardware you can buy.** ESP32, STM32 Nucleo, Arduino and Raspberry Pi boards, common sensors and LoRa modules are sold by Indian electronics

shops and online. Avoid papers that need a custom chip, an instrument your college does not have, or a radio band India does not allow.

3. **An extension measured on the board.** Not only accuracy on a laptop. Measure latency, memory, current, battery life, range or packet delivery on the device, for the paper's method and for yours.

What to measure on the board in an M.Tech IoT or embedded project, and how

MEASURE	HOW	TYPICAL TOOL
Time per inference or per message	Time the code on the device over many runs; report the mean and the worst case	esp_timer on the ESP32, the DWT cycle counter on Cortex-M4 and M7 STM32 boards, or a GPIO pin watched on a logic analyser
Memory	Flash and RAM from the build, plus peak RAM while the model or cipher runs	The size report in ESP-IDF or the Arduino IDE, the Build Analyzer in STM32CubeIDE
Current and energy	Current over a full cycle of sensing, computing, sending and sleeping, times voltage and time	A USB power meter for a first look; an INA219 or INA226 sensor logged by a second board for detail
Battery life	Run the node on its battery to cut-off, or estimate it from the measured average current, and state the cell and the reporting interval	A data logger and the battery's datasheet
Radio link	Packet delivery ratio, RSSI and SNR at set distances, with and without line of sight	Receiver or gateway logs with packet counters
Accuracy	On a held-out test set, split by person, field or day where the data allows	Python and scikit-learn, with the board's own outputs logged over serial

Measure the base paper's method yourself on your board. Do not compare your board's numbers with figures the paper got on different hardware.

Which M.Tech IoT and embedded project ideas can you build?

Fourteen ideas in four groups: TinyML on microcontrollers, LoRa and wireless sensor networks, IoT security, and energy, water and agriculture. Each card gives the problem and what you build and measure. Open "The base paper and the M.Tech extension" for the paper, the boards and tools, and one change with the comparison table your thesis would contain.

Topic ideas, not delivered projects

Each card is built on a published, open-access IEEE paper, checked against its DOI record. The project around it is an idea to discuss with your guide. Our 8 delivered case studies are M.Tech and M.E. projects, each shown with its real paper pages, screens and results.

TinyML: machine learning on the microcontroller

Train in Python on a laptop, then run the model on the board. The thesis numbers come from the board: accuracy after quantisation, flash and RAM, time per inference and current.

TOPIC IDEA · M.TECH · TINYML

Atrial fibrillation screening on an STM32 wearable

Problem: Paroxysmal atrial fibrillation shows up only in episodes, so it is caught by wearing a monitor for days, and streaming every heartbeat to a phone or the cloud drains the battery.

Output: Port the paper's Local Binary Pattern features and SVM to an STM32 Nucleo board, feed it MIT-BIH records over serial, and log accuracy, flash use, time per inference and current beside the paper's own figures.

− The base paper and the M.Tech extension

Base paper: Atrial Fibrillation Detection From Electrocardiogram Signal on Low Power Microcontroller, M. Yazid et al., IEEE Access, 2024
Boards, tools and M.Tech extension: Data, boards and tools: the MIT-BIH Atrial Fibrillation Database on PhysioNet, an STM32 Nucleo board with a Cortex-M4 (the paper used an STM32F413 at 100 MHz), STM32CubeIDE and Python with scikit-learn. M.Tech extension: replace the hand-made features and SVM with a small INT8 1D CNN on the raw ECG, and tabulate sensitivity, specificity, flash, RAM, milliseconds per inference and average current for the paper's method and yours on the same board.

TOPIC IDEA · M.TECH · TINYML

Rice leaf disease detection on a Cortex-M board with a camera

Problem: A farmer who spots discoloured rice leaves needs an answer in the field, and many paddy fields in India have weak mobile data, so a detector that works without the internet is more useful than a cloud app.

Output: Train and quantise the paper's small CNNs on its four disease classes, deploy the best one on a Cortex-M board with a camera module, and measure accuracy, RAM, flash and time per image on the board.

− The base paper and the M.Tech extension

Base paper: Effective Edge Solution for Early Detection of Rice Disease on ARM- M Microcontroller, Chew Jing Xan et al., IEEE Access, 2024
Boards, tools and M.Tech extension: Data, boards and tools: Rice Leaf Disease Image Samples on Mendeley Data, a free collection that matches the paper's 5,932 images and four classes; an Arduino Nano 33 BLE Sense with a camera or an STM32 board with a camera module; TensorFlow Lite for Microcontrollers. M.Tech extension: fine-tune the quantised model on a small set of leaf photos you take in local fields, and tabulate accuracy on the paper's images and on a held-out set of your field photos, before and after fine-tuning, with RAM and time per image.

TOPIC IDEA · M.TECH · TINYML

Camera-free activity recognition from Wi-Fi signals on an ESP32

Problem: Families want to know whether an elderly parent living alone has fallen or stopped moving, but cameras inside a home feel intrusive and wearables are often left on the table.

Output: Run a small INT8 CNN on Wi-Fi channel state information directly on an ESP32-WROOM-32, and measure accuracy, working memory, milliseconds per window and energy against the paper's results on UT-HAR.

− The base paper and the M.Tech extension

Base paper: Feasibility and Deployment Analysis of Wi-Fi-Based Human Activity Recognition on a Memory-Constrained Microcontroller, A. W. Malik et al., IEEE Access, 2026
Boards, tools and M.Tech extension: Data, boards and tools: the public UT-HAR and CSI-HAR sets, two ESP32 boards (one sending, one receiving) running Espressif's open esp-csi code, and TensorFlow Lite for Microcontrollers. M.Tech extension: the paper reports that accuracy falls by about 25–33 percentage points for a person left out of training, so record a one-minute calibration from each new person and fine-tune only the last layer, and tabulate leave-one-person-out accuracy, memory and inference time with and without it.

TOPIC IDEA · M.TECH · TINYML

Solar yield forecasting that runs on a microcontroller

Problem: Homes and small factories with rooftop solar could plan battery charging and heavy loads around a forecast, and running it on a cheap controller avoids a cloud subscription and keeps working offline.

Output: Train the paper's four recurrent models on solar yield records, then convert one for an ESP32 or STM32 board and compare forecast error, model size and time per forecast on the board with the laptop version.

− The base paper and the M.Tech extension

Base paper: Intelligent Solar Forecasts: Modern Machine Learning Models and TinyML Role for Improved Solar Energy Yield Predictions, A. M. Hayajneh et al., IEEE Access, 2024
Boards, tools and M.Tech extension: Data, boards and tools: solar farm yield records as in the paper (the abstract does not name the source, so read its data section first), an ESP32 or STM32 board, TensorFlow and TensorFlow Lite for Microcontrollers; check early that your recurrent model converts, because the ESP32 Wi-Fi sensing paper above reports that its recurrent models could not be converted for that runtime. M.Tech extension: replace the recurrent layers with a 1D temporal CNN that converts cleanly to INT8, and

converts cleanly to INT8, and tabulate RMSE, MAE, flash, RAM and milliseconds per forecast for the best recurrent model on the laptop and your CNN on the board.

LoRa and wireless sensor networks

In India, LoRa uses the 865–867 MHz band with the IN865 plan, so buy modules and antennas made for it. A network result needs several nodes, so budget for more than two boards.

TOPIC IDEA · M.TECH · LORA

A LoRa mesh network with time-on-air routing, simulated and then built

Problem: Villages, farms and hilly sites often have no single spot from which one LoRaWAN gateway reaches every node, so nodes have to pass each other's packets along, and the routing rule decides how many arrive.

Output: Reproduce the paper's distance-vector routing with its multi-spreading-factor time-on-air metric in FLoRa and OMNeT++, and plot packet delivery ratio, goodput and latency against traffic load in several topologies, including random ones.

− The base paper and the M.Tech extension

Base paper: A Minimalistic Distance-Vector Routing Protocol for LoRa Mesh Networks, R. Pueyo Centelles et al., IEEE Access, 2024
Boards, tools and M.Tech extension: Data, boards and tools: FLoRa on OMNeT++ for the simulation, as in the paper, then five or more ESP32 boards with SX1276 or SX1262 radios set to 865–867 MHz. M.Tech extension: run the same protocol on real boards spread across your campus, and give a table of simulated and measured packet delivery ratio and latency for the time-on-air metric and for plain hop count.

TOPIC IDEA · M.TECH · LORA

Locating LoRa nodes without GPS using machine learning

Problem: GPS drains a tracker's battery and fails indoors, so finding a LoRa tag on a cattle collar, a bicycle or a parcel from the signals the receivers already pick up would save power.

Output: Reproduce the paper's comparison of k-NN, CNN, SVR, ANN, XGBoost and LightGBM, and its hybrid of convolutional features with gradient-boosted regression, on its urban LoRaWAN dataset, with the location error and computing time of each.

− The base paper and the M.Tech extension

Base paper: Enhancing LoRa-Based Outdoor Localization Accuracy Using Machine Learning, N. Keleşoğlu et al., IEEE Access, 2025
Boards, tools and M.Tech extension: Data, boards and tools: the open-source urban LoRaWAN dataset the paper uses (its data section names it), Python with scikit-learn, XGBoost and LightGBM, and for the extension three or more ESP32 boards with LoRa radios as fixed receivers plus one moving tag. M.Tech extension: collect RSSI and SNR fingerprints on your own campus, and tabulate the mean and 90th-percentile location error and the prediction time of the best model on the public data and on yours.

TOPIC IDEA · M.TECH · LORA

LoRa wearable for vital signs and fall alerts that lasts longer on one charge

Problem: When an elderly person living alone falls, the minutes before someone is told matter, and in many Indian towns the home has neither Wi-Fi nor steady mobile data for a phone-based alarm.

Output: Build the paper's wearable with a MAX30102 sensor, an accelerometer and an SX1262 link that sends routine DATA packets and urgent ALERT packets, and measure vital-sign error, fall sensitivity, packet reception at distance and alert latency.

The base paper and the M.Tech extension

Base paper: LoRa-Based Wearable Monitoring System for Elderly Vital Signs and Emergency Alerts, N. Pukrongta et al., IEEE Access, 2026
Boards, tools and M.Tech extension: Data, boards and tools: an ESP32 or similar microcontroller, an SX1262 module set to 865–867 MHz, a MAX30102 sensor, an accelerometer, a LoRa gateway or receiver board and a Li-ion cell. M.Tech extension: the paper reports battery durations of 1 h 59 min and 3 h 21 min at 12-second and 60-second DATA intervals, so add deep sleep and send DATA only when a reading changes, and tabulate battery life, packet reception ratio and alert latency for the paper's schedule and yours.

TOPIC IDEA · M.TECH · LORA

LoRa-linked orchard irrigation with fuzzy PID control

Problem: Farm plots and orchards can sit hundreds of metres from the pump house over uneven ground, and a timer-based valve gives the same water whether the soil is dry or wet.

Output: Build the paper's soil-moisture collection nodes and LoRa link to a controller running fuzzy PID, and measure packet loss at distance over open and cluttered ground and the error between planned and actual irrigation time.

The base paper and the M.Tech extension

Base paper: Remote Irrigation Control Based on LoRa Technology and Fuzzy PID Algorithm, Y. Li, IEEE Access, 2025
Boards, tools and M.Tech extension: Data, boards and tools: capacitive soil-moisture sensors, ESP32 or STM32 nodes with SX1276 modules at 865–867 MHz, a relay driving a 12 V solenoid valve, and MATLAB or Python to tune the controller before field tests. M.Tech extension: run fuzzy PID, plain PID and a simple on-off threshold on the same plot, and give a table of water used, hours outside the target moisture band, overshoot and packet loss for each.

IoT security on small devices

Measure the cost on the device as well as the detection score. Time, memory and energy decide whether a method can run where the attack happens.

TOPIC IDEA · M.TECH · IOT SECURITY

ASCON lightweight encryption on ESP32 and Raspberry Pi nodes

Problem: Sensor nodes often send readings unencrypted because standard security stacks seem heavy for a small microcontroller, which leaves the data open to reading and tampering on the way.

Output: Reproduce the paper's ASCON authenticated encryption in the CupCarbon simulator and on a Raspberry Pi link, and measure latency, response time and resource use for messages of several sizes.

The base paper and the M.Tech extension

Base paper: Implementation and Performance of Lightweight Authentication Encryption ASCON on IoT Devices, G. Cagua et al., IEEE Access, 2025
Boards, tools and M.Tech extension: Data, boards and tools: CupCarbon as in the paper, a Raspberry Pi, two ESP32 boards and the ASCON reference code in C. M.Tech extension: port ASCON-128 to the ESP32 and compare it with AES-128-GCM from the ESP32's mbedTLS library, which can use the chip's AES hardware, in a table of time per message, throughput, code size, RAM and energy per message at several payload sizes.

TOPIC IDEA · M.TECH · IOT SECURITY

Federated attack detection across Raspberry Pi gateways, with privacy noise

Problem: Hostels, shops and factory floors each see only a few attacks on their own IoT devices, and none of them wants to send its raw traffic to a central server.

Output: Reproduce the paper's federated deep neural network on CIC IoT 2023, first in simulation and then with three or more Raspberry Pi clients, and compare accuracy and macro F1 with the same model trained centrally.

The base paper and the M.Tech extension

Base paper: A Novel Federated Edge Learning Approach for Detecting Cyberattacks in IoT Infrastructures, S. Abbas et al., IEEE Access, 2023
Boards, tools and M.Tech extension: Data, boards and tools: the CIC IoT 2023 dataset, Python with PyTorch or TensorFlow and the Flower framework, and three to five Raspberry Pi boards on one Wi-Fi network. M.Tech extension: add differential-privacy noise to the model updates each Pi sends, and tabulate macro F1, rounds to converge and training time per round on the Pi at three noise levels and with no noise.

TOPIC IDEA · M.TECH · IOT SECURITY

Explainable intrusion detection that runs on an industrial IoT gateway

Problem: In a plant, an operator will not stop a line on an unexplained alarm, so an intrusion detector has to show which traffic features made it fire, and do so quickly.

Output: Reproduce the paper's ensemble deep learning detector and its extreme learning machine baseline on ToN_IoT, add SHAP and LIME explanations, and time detection and explanation on a Raspberry Pi.

− The base paper and the M.Tech extension

Base paper: An Explainable Ensemble Deep Learning Approach for Intrusion Detection in Industrial Internet of Things, M. M. Shtayat et al., IEEE Access, 2023

Boards, tools and M.Tech extension: Data, boards and tools: the ToN_IoT dataset, Python with TensorFlow or PyTorch, the SHAP and LIME libraries, and a Raspberry Pi as the gateway. M.Tech extension: kernel SHAP calls the model many times per explanation, so train a gradient-boosted tree detector explained with TreeSHAP, and tabulate F1, false-positive rate and seconds per explanation on the Pi for the ensemble with kernel SHAP and the tree model with TreeSHAP.

Energy, water and smart agriculture

These builds run outdoors or on live supplies for weeks, so start logging early in Stage-II. Keep mains wiring out of the student build: use clamp-on current sensors, and low-voltage pumps and valves.

TOPIC IDEA · M.TECH · ENERGY AND AGRICULTURE

ESP32 irrigation controller that predicts the water a crop needs

Problem: Farmers with a borewell pump often water on a fixed routine, which wastes groundwater on cool days and under-waters on hot ones.

Output: Build the ESP32 node from the paper, with sensors for temperature, humidity and water level, log your own readings for several weeks, fit the linear regression it uses, and switch a small pump from the prediction.

The base paper and the M.Tech extension

Base paper: An Innovative Smart Irrigation Using Embedded and Regression-Based Machine Learning Technologies for Improving Water Security and Sustainability, A. Morchid et al., IEEE Access, 2025
Boards, tools and M.Tech extension: Data, boards and tools: readings from your own node, an ESP32, a temperature and humidity sensor such as the DHT22, a water-level sensor, and a relay with a 12 V pump. M.Tech extension: the abstract reports an R^2 of 0.4044 for its regression, so keep the same sensors, replace linear regression with a small quantised neural network running on the ESP32, and tabulate MAE, RMSE, R^2, prediction time and flash use for both models on the same logged data.

TOPIC IDEA · M.TECH · ENERGY AND AGRICULTURE

Reading old water meters with a camera board on solar power

Problem: Many Indian housing societies and municipal connections still use analogue water meters read by hand once a month, so a leak or a running tap can go unnoticed for weeks.

Output: Retrofit an analogue meter with a camera board that reads the dial on the device, sends the reading to the cloud and runs from a solar-charged battery, as in the paper, and measure reading accuracy, current draw and battery charge over several weeks.

The base paper and the M.Tech extension

Base paper: IoT-Enabled Water Monitoring in Smart Cities With Retrofit and Solar-Based Energy Harvesting, N. Bawankar et al., IEEE Access, 2024
Boards, tools and M.Tech extension: Data, boards and tools: an ESP32-CAM or ESP32-S3 camera board, a small solar panel with a charge controller and a Li-ion cell, an INA219 current sensor for logging, and a cloud dashboard such as ThingSpeak; three of the paper's four authors are at IIIT Hyderabad. M.Tech extension: make the reading interval adaptive, reading more often while water is flowing and rarely when it is not, and tabulate reading accuracy, energy per day, battery charge and missed flow events for a fixed interval and the adaptive one

over the same weeks.

TOPIC IDEA · M.TECH · ENERGY AND AGRICULTURE

Appliance-level energy monitoring from one current sensor on a microcontroller

Problem: A household electricity bill shows one total, so a family cannot see which appliance drives it, and a smart plug on every socket costs too much.

Output: Build the paper's low-cost terminal, in which a microcontroller samples the current and takes an FFT, then reproduce its change-point detection and its XGBoost and LightGBM layer with logistic regression on top to name the appliance that switched on or off.

− The base paper and the M.Tech extension

Base paper: A Non-Intrusive Load Identification Method Based on Novel Data Acquisition Terminals and Model Fusion, J. Zhuge et al., IEEE Access, 2024

Boards, tools and M.Tech extension: Data, boards and tools: an STM32 or ESP32 board, a clamp-on current transformer such as the SCT-013 fitted on one conductor by an electrician, a burden resistor and bias circuit, and Python with XGBoost and LightGBM; the abstract reports more than 95% accuracy with high-power and low-power appliances in use together. M.Tech extension: run the whole two-layer classifier on the microcontroller as pruned, compact tree models, and tabulate accuracy for each appliance, flash, RAM and time per

KEY POINTS

- **14 M.Tech IoT and embedded project ideas,** each built on an open-access IEEE paper from 2023 to 2026 and checked against its DOI record.
- **Four groups:** TinyML on microcontrollers (4), LoRa and wireless sensor networks (4), IoT security (4), and energy, water and agriculture (3).
- **All 14 papers are in IEEE Access,** so anyone, your examiner included, can read the full text free.
- **Boards you can buy in India:** ESP32, STM32, Arduino and Raspberry Pi, with LoRa radios for the 865–867 MHz band.
- **Every card names one measured change** and the table that compares it with the base paper on the same board.

Choosing an M.Tech IoT project?

Send your specialisation, the boards your lab has and your Stage-I date. The consultation is free.

Message in the next 7 h 19 min for a reply today.

Free consultation →

WhatsApp

How does an M.Tech embedded project run, stage by stage?

Most universities split the M.Tech project into two stages, each with a report and a review. For an IoT or embedded project, the work usually falls like this.

Stage-I: literature, simulation and reproduction

1. **Literature and the base paper.** Read the recent papers on one problem, choose the base paper, and write down the gap your change will address.
2. **Simulation and reproduction.** Rebuild the paper's method where it is cheapest: train the model in Python, run the protocol in a simulator such as

FLoRa on OMNeT++ or CupCarbon, or test the firmware on one board on your desk. Put your numbers beside the paper's.

3. **Design of the extension.** Fix the one change, the boards, the test data and the table you will fill in Stage-II. Order the hardware now.
4. **Stage-I report and review.** Literature, gap, the reproduced baseline and the plan. Our base paper presentation guide covers the slides.

Stage-II: hardware, measurements, thesis and paper

1. **Hardware build.** Port the method to the board, wire the sensors and the radio, and run it from a supply whose current you can measure.
2. **Measurements.** Run the base method and your change on the same board and the same test data, and fill the comparison table.
3. **Thesis.** Write it in your university's format. Our M.Tech thesis format guide sets out the chapters.
4. **Paper.** If your course asks for one, write the result as a conference or journal paper. See our IEEE paper format guide.

For the reproduction step in detail, including what to do when your numbers do not match the paper's, read how to implement a base paper.

Which boards and tools do you need, and what should you check on cost?

Every idea here runs on common boards. We do not list prices, because they change with stock and seller. Check these points before you buy.

Boards for M.Tech IoT and embedded projects, what each suits and what to check before buying

BOARD OR PART	SUITS	CHECK BEFORE YOU BUY
ESP32 (ESP32-WROOM-32, ESP32-S3)	Wi-Fi nodes, Wi-Fi sensing, small TinyML models, camera boards, LoRa nodes with an add-on radio	Whether the module has PSRAM if your model or camera needs it, and the exact module name your paper or library uses

BOARD OR PART	SUITS	CHECK BEFORE YOU BUY
STM32 Nucleo (Cortex-M4 or M7)	Low-power signal processing, ECG and sensor classification, precise timing and current measurement	Flash and RAM of the exact part number, and that your toolchain and TensorFlow Lite for Microcontrollers support it
Arduino Nano 33 BLE Sense	TinyML with on-board motion and audio sensors	Which revision you are buying, since the on-board sensors differ, and that your sensors work at 3.3 V
Raspberry Pi 4 or 5	Gateways, security models, federated learning clients, camera pipelines	An official power supply, cooling for long runs, and enough RAM for your model
LoRa modules (SX1276, SX1262)	Long-range links, mesh networks, localisation, field sensors	That the module and antenna are made for 865–867 MHz, and whether your test needs a full LoRaWAN gateway or receiver boards are enough
Measurement parts	Current, energy and timing evidence for the thesis	An INA219 or INA226 current sensor, a USB power meter, and whether your lab lends a logic analyser or an oscilloscope

Beyond the boards, plan for these costs:

- **Spares.** Buy two of each board you depend on, so that one failure in Stage-II does not stop your work.
- **Enough nodes.** A LoRa mesh or localisation test needs five or more boards, and a federated learning test needs three or more Raspberry Pi boards.
- **Software.** The tools named on this page have free or academic versions: the Arduino IDE, ESP-IDF, STM32CubeIDE, TensorFlow Lite for Microcontrollers, OMNeT++ with FLoRa, CupCarbon and Python.
- **Time.** Field tests for irrigation, water and energy need weeks of logging, and imported modules can take weeks to arrive, so order in Stage-I.

How can we help with an M.Tech IoT or embedded project?

We check the base paper first: whether its data is public, whether the boards are easy to buy, and whether the result can be reproduced in your time. Then we reproduce it, plan one measured extension with you, build and test it on the board, and support the Stage-I and Stage-II reports, the thesis and, if you need one, a paper and viva preparation.

See how an ECE project runs, track by track, on our M.Tech projects for ECE page, which covers embedded systems, VLSI and signal processing. For the M.Tech service across branches, with the thesis and the paper, see M.Tech projects. Want more papers to choose from? See the IEEE base papers by domain.

We build, guide, help with the writing and explain every result. The thesis you submit must follow your university's rules and cite the base paper.

More M.Tech domains, each built on verified IEEE base papers: M.Tech projects in machine learning, M.Tech projects in image processing, M.Tech projects in cyber security. For the full service, see our M.Tech projects for CSE and AI.

Frequently asked questions

Which is the best M.Tech project in embedded systems?

There is no single best one. A good M.Tech embedded project rebuilds a recent IEEE paper on a board you can buy, then makes one change you measure on that board, such as lower latency, less current or better accuracy on new data. Each of the 14 cards on this page names such a change. Our guide to selecting a base paper covers the checks.

Can I do an M.Tech IoT project without buying hardware?

Partly. Stage-I can run on a laptop: train the model in Python, or simulate a LoRa network in FLoRa on OMNeT++. An embedded or IoT thesis is much stronger with measurements from real boards, so plan to buy them for Stage-II. The boards on this page are common hobby parts, not lab instruments.

Which board is best for an M.Tech embedded project: ESP32, STM32 or Raspberry Pi?

It depends on what you will measure. The ESP32 suits Wi-Fi nodes and small TinyML models. An STM32 Nucleo suits low-power signal processing and precise timing. A Raspberry Pi suits gateways, security models and federated learning clients, but it

is a Linux computer, not a microcontroller. Our IoT project ideas page compares the common boards.

Can I use LoRa for an M.Tech project in India?

Yes, in the 865–867 MHz band, which India keeps open for low-power wireless devices; the LoRaWAN channel plan for India is called IN865. Buy LoRa modules and antennas made for that band, and set your radio library to it before any field test.

What should the M.Tech extension be in an IoT or embedded project?

One change you can measure on the board against the base paper's method: a quantised or smaller model, a faster cipher, a better routing metric, a sleep schedule that extends battery life, or a test on new people or new fields. Put both methods in one table, on the same board and data. Our M.Tech projects for ECE page gives more examples.

Can The Ultimate Project World help with an M.Tech IoT or embedded project?

Yes. Send the DOI of your base paper, the boards your lab has and your stage review dates in the free consultation. We tell you which parts we can take on, such as the reproduction, the hardware build and measurements, the thesis, an IEEE-format paper and viva preparation. Our M.Tech projects service explains how it runs.

RELATED SERVICES

From a topic to a project you can present

M.Tech projects

A base paper, one measured extension, a paper and the thesis.

IEEE projects

Start from a published base paper and extend it.

Research papers

A paper in IEEE, Springer or journal format, and a journal shortlist.