

# IoT-Based Embedded Energy Management System with Predictive Analytics and Mobile Control

Mr. A. M. Rangaraj<sup>1</sup>, Chintakuntla Sharif<sup>2</sup>, Reddyvari Tharun Kumar<sup>3</sup>, Mejari Malleeswara<sup>4</sup>

<sup>1</sup>*Associate Professor/MCA, Sri Venkateswara College of Engineering and Technology (Autonomous)  
Chittoor, Andhra Pradesh-517217*

<sup>2,3,4</sup>*MCA Students, Sri Venkateswara College of Engineering and Technology (Autonomous)  
Chittoor, Andhra Pradesh-517217*

**Abstract**—The increasing global focus on sustainable and energy-efficient practices has spurred the development of advanced technologies to optimize energy consumption in both residential and commercial environments. Embedded systems integrated with the Internet of Things (IoT) offer a promising solution for intelligent energy management. This project aims to design and implement an IoT-based energy management system using Arduino Uno, ACS712 current sensors, and AC voltage sensors to monitor and analyze energy usage patterns in real-time. Sensor data is transmitted to an IoT cloud platform, where it is processed and analyzed using machine learning algorithms to predict future energy requirements. Notifications regarding energy usage trends and optimization strategies are sent to a mobile application, which also serves as a control interface for managing connected devices. The system provides users with actionable insights and remote-control capabilities, contributing to reduced energy consumption, lower operational costs, and a more sustainable future.

## I. INTRODUCTION

Energy management has become a critical challenge in the modern era, driven by increasing energy demands, rising operational costs, and environmental concerns. Traditional energy monitoring systems often lack real-time analytics and remote accessibility, limiting their efficiency. The integration of IoT with embedded systems offers an intelligent solution for energy monitoring, analysis, and optimization. This project focuses on designing an IoT-based energy management system that collects, analyzes, and provides actionable insights into energy consumption.

## II. LITERATURE SURVEY

**IoT-Enabled Energy Management Systems:** A comprehensive review by Nikpour et al. (2023) emphasizes the role of IoT devices in smart cities, particularly in monitoring and controlling energy consumption. The study highlights how IoT frameworks facilitate intelligent analysis, leading to improved system efficiency and sustainability.

**2. Optimization Techniques in Smart Buildings:** Shah et al. (2019) focus on energy consumption optimization in IoT-based smart building environments. Their review discusses various techniques aimed at balancing user comfort with energy efficiency, addressing challenges in achieving optimal energy utilization.

**3. IoT Applications in Smart Grids:** Saleem et al. (2017) explore the integration of IoT in smart grids, detailing technologies, architectures, and applications. The survey highlights how IoT aids in monitoring, analysis, and control within smart grids, contributing to enhanced energy management.

**4. Intelligent Energy Management Frameworks:** Huang et al. (2023) present an IoT-based framework for smart energy management, emphasizing the importance of energy disaggregation technology. Their work reviews state-of-the-art algorithms and public datasets, offering insights into optimizing energy utilization in IoT networks.

**5. IoT in Smart Buildings:** A study published in MDPI's Buildings journal (2024) critically analyzes IoT applications that enhance energy management in smart buildings. The research indicates that IoT technology can decrease energy consumption by up to 30% and operating expenses by 20%, underscoring its potential in promoting sustainability.

### III. PROBLEM IDENTIFICATION

#### 3.1 PROBLEM DEFINITION

Conventional energy management systems often lack real-time monitoring, predictive analytics, and user-friendly control interfaces. As a result, users are unable to make informed decisions about energy usage, leading to inefficient consumption, increased costs, and unnecessary energy wastage. This project addresses these limitations by providing an IoT-enabled system for real-time energy monitoring, predictive analytics, and remote device management.

Main Objectives:

1. To design and implement an IoT-based energy management system for real-time monitoring of energy consumption.
2. To utilize ACS712 current sensors and AC voltage sensors for accurate energy usage measurement.
3. To transmit sensor data to an IoT cloud platform for processing and analysis.
4. To use machine learning algorithms for predicting future energy requirements.
5. To provide users with notifications and optimization strategies via a mobile application.
6. To enable remote control and management of connected devices.
7. To reduce overall energy consumption and operational costs.

Existing Methods and Drawbacks:

- Traditional energy monitoring systems rely on manual readings and periodic analysis.
- Limited remote accessibility and lack of predictive analytics.
- Inability to provide real-time notifications or actionable insights.
- High dependency on manual interventions for optimization.

Proposed Methodology:

1. Energy data is collected using ACS712 current sensors and AC voltage sensors.
2. The Arduino Uno processes and transmits data to an IoT cloud platform.
3. Real-time data is analyzed using machine learning algorithms.
4. Predicted energy usage trends and recommendations are generated.
5. Notifications are sent to a mobile application.

6. Users can monitor and control connected devices remotely.
7. Insights are provided for energy optimization.

Implementation Method:

1. Hardware Implementation: Sensors and microcontrollers are interfaced to monitor energy parameters.
2. Data Transmission: The data is sent to an IoT cloud platform via a Wi-Fi module.
3. Data Processing: Machine learning models analyze the data to detect patterns and predict future energy consumption.
4. User Interaction: The mobile application provides real-time insights and allows remote control of connected devices.
5. Optimization & Automation: Based on predictive analytics, the system suggests or automatically adjusts energy consumption settings.

Modules Description:

1. Sensor Module: ACS712 current sensors and AC voltage sensors collect energy data.
2. Processing Module: Arduino Uno processes sensor data and transmits it to the IoT cloud.
3. Communication Module: ESP8266 Wi-Fi module enables real-time data transfer.
4. Cloud Analytics Module: Machine learning models analyze data and generate insights.
5. User Interface Module: A mobile application displays energy consumption trends and allows control of connected devices.

Notification Module: Sends alerts and recommendations to users based on energy patterns.

### IV. DESIGN

In this project, we utilize the K-Nearest Neighbors (KNN) algorithm to classify and predict power consumption categories based on real-time sensor data. The model helps in identifying whether the power usage falls into High, Medium, or Low consumption levels, providing users with actionable insights for energy optimization.

#### 1. Data Collection

The system gathers real-time energy consumption data using:

- ACS712 Current Sensors – Measures the electrical current flowing through devices.
- AC Voltage Sensors – Captures voltage variations in the electrical system.
- IoT Cloud Platform – Stores historical data for further analysis.

These values are continuously transmitted to the cloud for real-time monitoring and processing.

## 2. Data Preprocessing

Before feeding the data into the ML model, preprocessing steps include:

- Normalization – Scaling current and voltage values to ensure uniformity.
- Feature Extraction – Extracting key parameters such as power factor, energy usage trends, and fluctuations.
- Labeling Data – Categorizing historical power consumption data into High, Medium, and Low classes based on predefined thresholds.

## 3. KNN Classification Model

We use the K-Nearest Neighbors (KNN) algorithm, a supervised learning method, for classification.

### Why KNN?

- Simple and effective for pattern recognition.
- Works well for real-time classification tasks.
- Handles non-linear relationships in power consumption data.

### Model Training Process

1. The dataset is split into training (80%) and **testing** (20%) subsets.
2. The KNN algorithm computes the Euclidean distance between new input data and stored training examples.
3. Based on the nearest 'K' neighbors, it classifies the power consumption into High, Medium, or Low.

The model is continuously updated with new real-time data to improve accuracy.

## 4. Prediction and Notification System

Once classified, the system performs the following:

- Sends real-time alerts via Telegram Bot to notify users about energy usage trends.

- Provides recommendations for energy optimization based on past consumption patterns.
- Allows remote control through a mobile application to switch off high-energy-consuming devices.

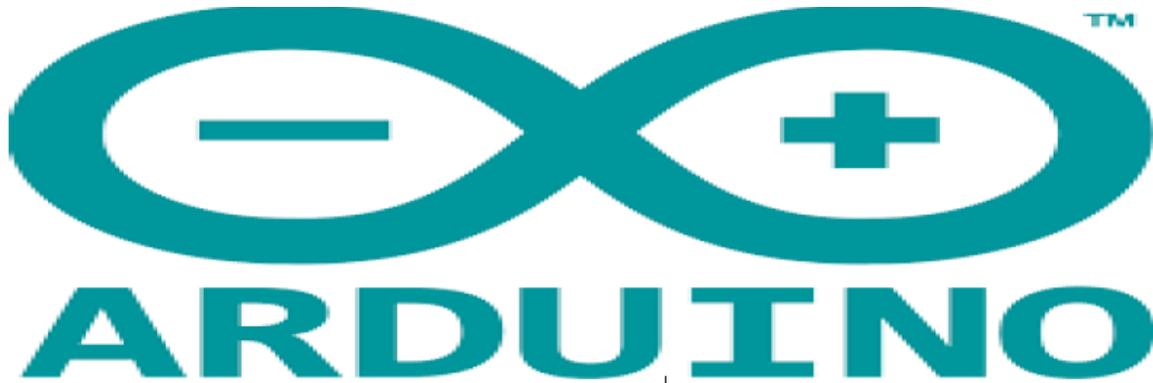
## 5. Model Implementation on Hardware

- The Arduino Uno collects sensor data and sends it to the cloud via Wi-Fi (NodeMCU/ESP8266).
- A Raspberry Pi (or cloud server) runs the ML model, processes the data, and sends category predictions.
- Users receive alerts and insights on their mobile application and Telegram bot.

## HARDWARE TOOLS

### 2.3 Arduino

Arduino is open-source physical processing which is based on a microcontroller board and an incorporated development environment for the board to be programmed. Arduino gains a few inputs, for example, switches or sensors and control a few multiple outputs, for example, lights, engine and others. Arduino program can run on Windows, Macintosh and Linux operating systems (OS) opposite to most microcontrollers' frameworks which run only on Windows. Arduino programming is easy to learn and apply to beginners and amateurs. Arduino is an instrument used to build a better version of a computer which can control, interact and sense more than a normal desktop computer. It's an open-source physical processing stage focused around a straightforward microcontroller board, and an environment for composing programs for the board. Arduino can be utilized to create interactive items, taking inputs from a diverse collection of switches or sensors, and controlling an assortment of lights, engines, and other physical outputs. Arduino activities can be remaining solitary, or they can be associated with programs running on your machine (e.g. Flash, Processing and Maxmsp.) The board can be amassed by hand or bought preassembled; the open-source IDE can be downloaded free of charge. Focused around the Processing media programming environment, the Arduino programming language is an execution of Wiring, a comparative physical computing platform. Figure 7- Arduino's



### 2.3.1 Why choosing Arduino

There are numerous different microcontrollers and microcontroller platforms accessible for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and numerous others offer comparative usefulness. These apparatuses take the chaotic subtle elements of microcontroller programming and wrap it up in a simple to-utilize bundle. Arduino additionally rearranges the methodology of working with microcontrollers; moreover, it offers some advantages for instructors, students, and intrigued individuals:

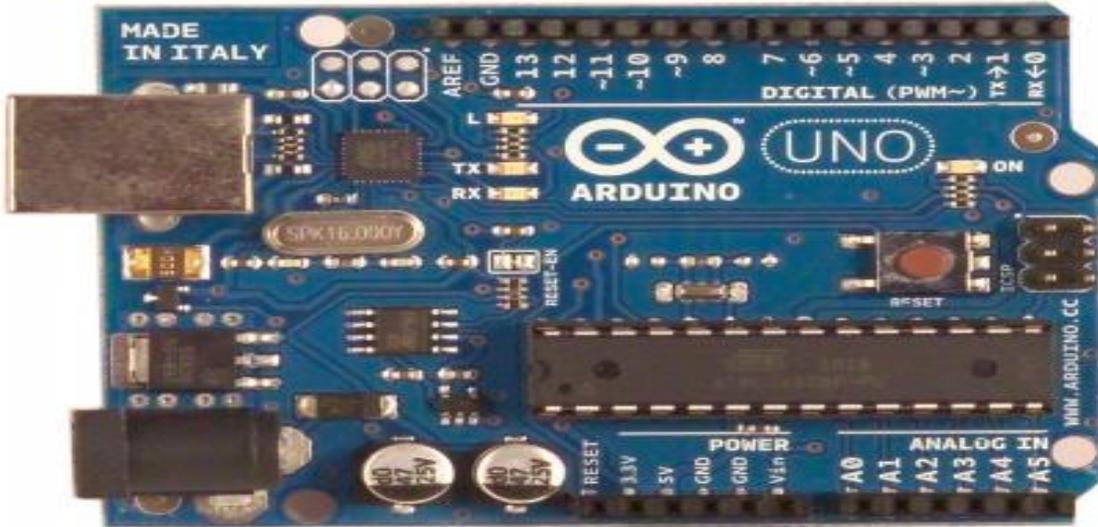
- Inexpensive - Arduino boards are moderately cheap compared with other microcontroller boards. The cheapest version of the Arduino module can be amassed by hand, and even the preassembled Arduino modules cost short of what \$50.
- Cross-platform - The Arduino programming runs multiple operating systems Windows, Macintosh OSX, and Linux working frameworks. So we conclude that Arduino has an advantage as most microcontroller frameworks are constrained to Windows.
- Straightforward, clear programming method - The Arduino programming environment is easy to use for novices, yet sufficiently versatile for cutting edge customers to adventure as well. For educators, its favorably engaged around the Processing programming environment, so understudies finding ways to understand how to program in that environment will be familiar with the nature of arduino.
- Open source and extensible programming. The Arduino program language is available as open source, available for development by experienced engineers. The lingo can be reached out through C++ libraries,

and people expecting to understand the specific purposes of different interests can make the leap from Arduino to the AVR C programming language on which it is based. Basically, you can incorporate AVR-C code clearly into your Arduino programs if you have to.

- Open source and extensible hardware - The Arduino is concentrated around Atmel's Atmega8 and Atmega168 microcontrollers. The plans for the modules are circulated under a Creative Commons license, so experienced circuit designers can make their own particular interpretation of the module, extending it and improving it. slightly inexperienced customers can build the breadboard variation of the module remembering the finished objective to perceive how its capacities and save money.

#### ARDUINO UNO:

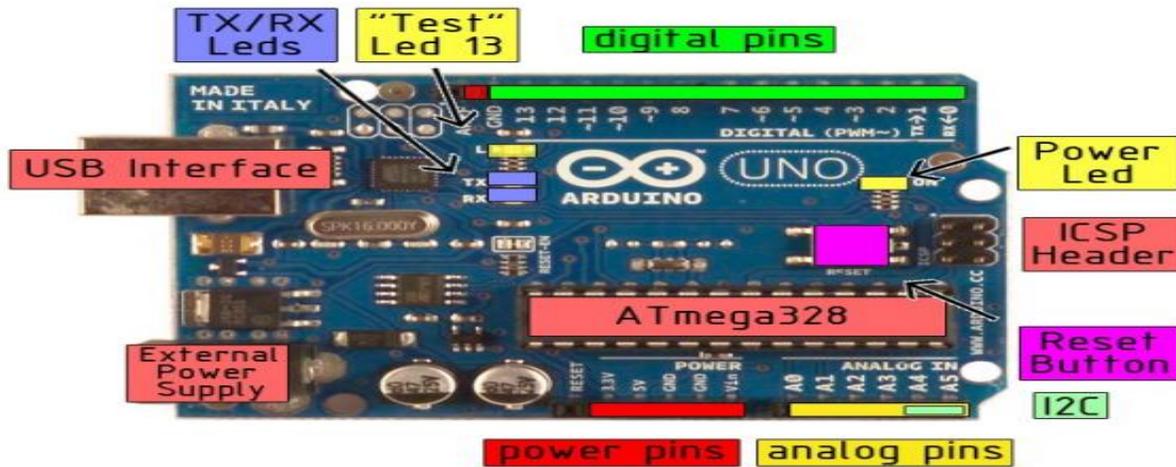
The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform;



Technical specifications of arduino:

Microcontroller: ATmega328  
 Operating Voltage: 5V  
 Input Voltage (recommended): 7-12V  
 Input Voltage (limits): 6-20V  
 Digital I/O Pins 14 (of which 6 provide PWM output)  
 Analog Input Pins 6

DC Current per I/O Pin 40 mA  
 DC Current for 3.3V Pin 50 mA  
 Flash Memory  
 32 KB of which 0.5 KB used by  
 bootloader  
 SRAM 2 KB  
 EEPROM 1 KB  
 Clock Speed 16 MHz



V. RESULT

The implemented system successfully monitored energy consumption in real-time and provided predictive insights for energy optimization. The machine learning model demonstrated accurate

forecasting of future power usage based on historical trends. Users were able to receive timely notifications and remotely control connected devices, leading to a reduction in overall energy consumption. Data analysis revealed a significant improvement in energy efficiency, with potential cost savings observed

over the testing period. The system provided intuitive visualization of power usage trends and optimization recommendations through the mobile application.

## VI. CONCLUSION

This project demonstrates the effectiveness of an IoT-based energy management system in monitoring and optimizing power consumption. By integrating Arduino-based hardware, cloud computing, and machine learning, the system provides users with actionable insights and remote-control capabilities. The results indicate that such a system can contribute to energy conservation, cost reduction, and environmental sustainability.

### Further Enhancements

- Integration of advanced AI models for improved energy consumption forecasting.
- Expansion to support multiple energy sources, including renewable energy.
- Implementation of blockchain for secure energy transactions and data privacy.

- Development of a web-based dashboard for comprehensive energy analytics.
- Real-time anomaly detection to identify and prevent energy wastage.

### FEATURES:

- Integrated ESP8266 Wi-Fi module for seamless connectivity.
- Compatible with Lua scripting and Arduino IDE.
- Compact design with 11 digital I/O pins and 1 analog pin.
- Supports SPI, I2C, UART, and PWM for versatile interfacing.
- Onboard micro-USB for power and programming.
- Low power consumption, ideal for battery-operated devices.

### SPECIFICATIONS:

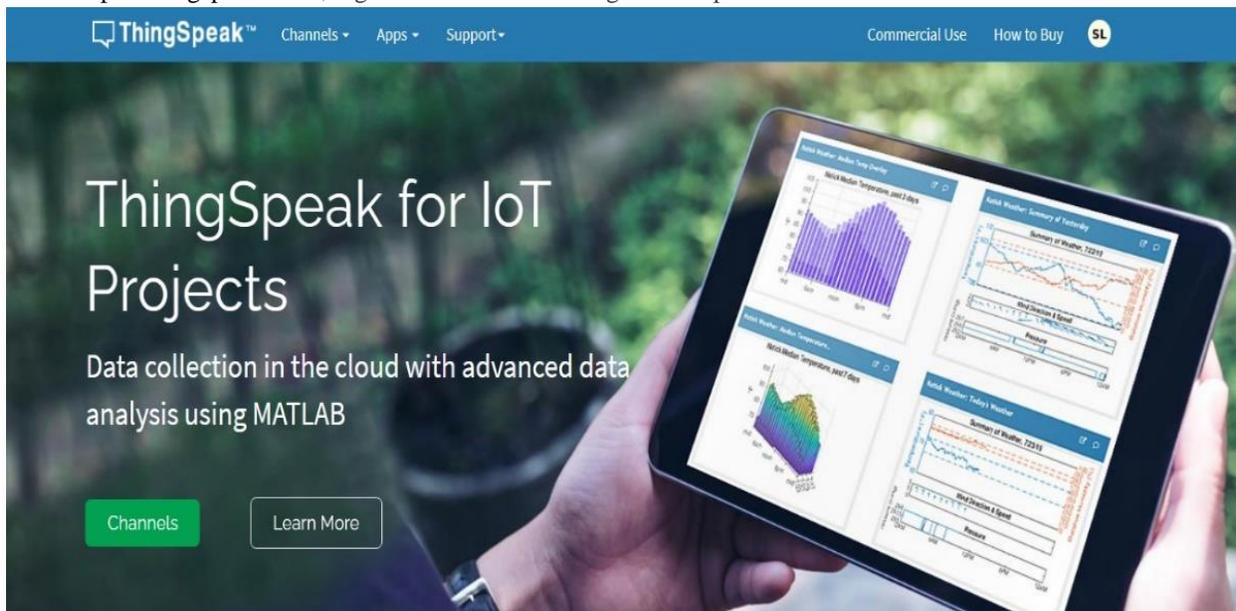
- Microcontroller: ESP8266
- Operating Voltage: 3.3V
- Input Voltage: 5V (via micro-USB)
- Wi-Fi Protocol: IEEE 802.11 b/g/n
- Dimensions: 34mm x 25mm
- Programming: Arduino IDE or Lua Script

Goal:

we need to create the thingspeak channel and get the key

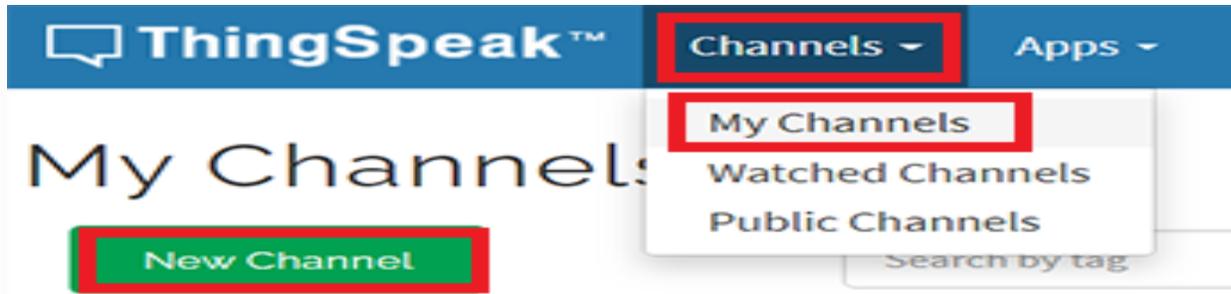
Step 1

Go to <https://thingspeak.com/>, register an account and login to the platform



Step 2

Choose Channels -> My Channels -> New Channel



Step 3

Input Channel name, Field1 , then click “Save Channel”

- Channel name: smart-house
- Field 1: Temperature



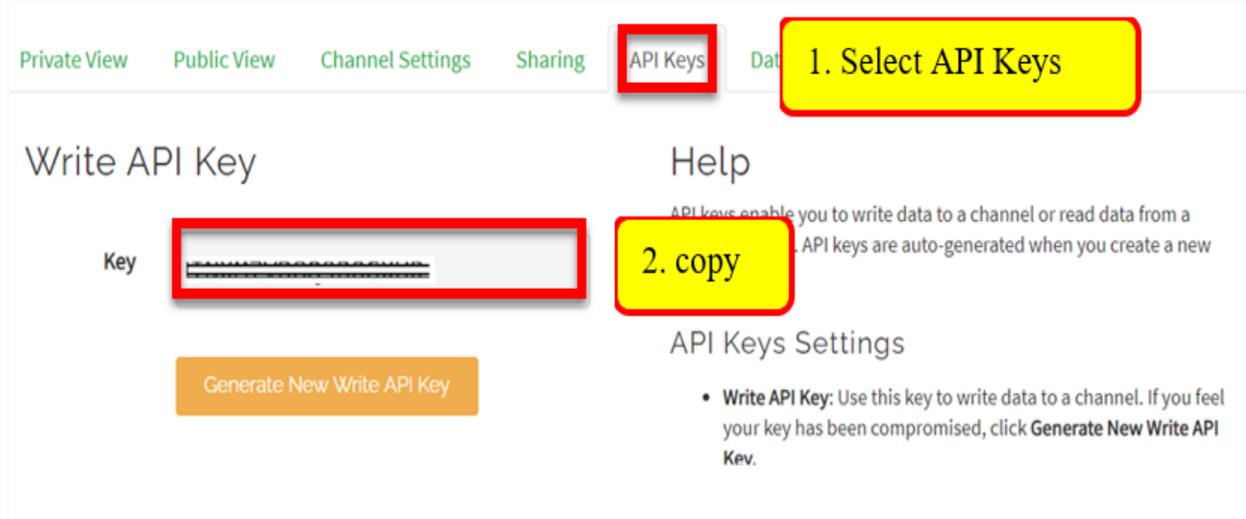
Step 4

You will see a chat for data field1



Step 5

Open your web browser, go to <https://thingspeak.com>, select your channel > “API Keys”, copy the API key as follows:



### MIT App Inventor Overview

The MIT App Inventor user interface includes two main editors: the design editor and the blocks editor. The design editor, or designer (see Fig. 3.1), is a drag and drop interface to lay out the elements of the application’s user interface (UI). The blocks editor (see Fig. 3.2) is an environment in which app inventors can visually lay out the logic of their apps using color-

coded blocks that snap together like puzzle pieces to describe the program. To aid in development and testing, App Inventor provides a mobile app called the App Inventor Companion (or just “the Companion”) that developers can use to test and adjust the behavior of their apps in real time. In this way, anyone can quickly build a mobile app and immediately begin to iterate and test MIT App Inventor:

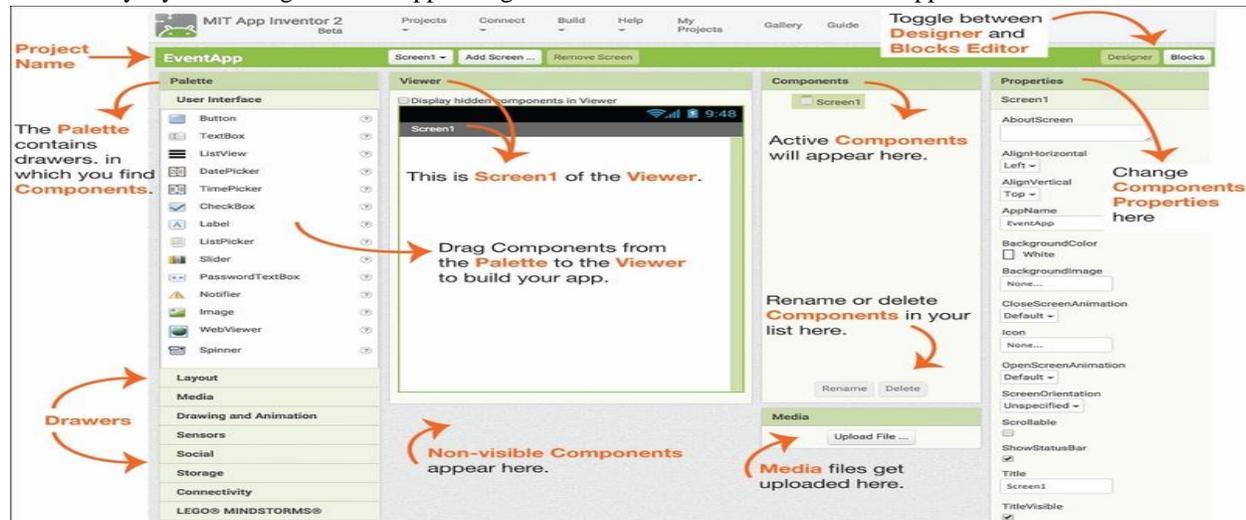


Fig. 3.1 App Inventor’s design editor.

In the design of MIT App Inventor, introducing mobile app development in educational contexts was a central goal. Prior to its release, most development environments for mobile applications were clunky, only accessible with expertise in systems level or embedded programming, or both. Even with Google’s

Android operating system and the Java programming language, designing the user interface was a complex task. Further, use of the platform required familiarity with Java syntax and semantics, and the ability to debug Java compilation errors (e.g., misspelled variables or misplaced semicolons) for success. These

challenges presented barriers to entry for individuals not versed in computer science, App Inventor's target demographic. We briefly highlight and discuss design goals for the App Inventor project, specifically, the use of components to abstract some of the complexity of platform behavior, and the use of blocks to eliminate

complexity of the underlying programming language. These goals can be further explained as aligning the visual language to the mental models of young developers and enabling exploration through fast, iterative design.



Fig. 3.2 App Inventor's blocks editor

#### REFERENCES

- [1] Brown, J., & Lee, S. (2021). Enhancing adaptive learning through AI-driven assessments. *Journal of Educational Technology Research*, 38(4), 456-472. <https://doi.org/10.1016/j.jet.2021.04.005>
- [2] Kumar, R., & Rao, M. (2022). AI-based quiz generation using natural language processing. *International Journal of Computer Applications*, 50(2), 123-135. <https://doi.org/10.5120/ijca2022502135>
- [3] Patel, A., Singh, K., & Gupta, R. (2019). Automated question answering using transformer-based architectures. *Journal of Artificial Intelligence Applications*, 12(3), 198-215. <https://doi.org/10.1109/jaia.2019.012003>
- [4] Smith, T., Johnson, P., & Chen, Y. (2021). Exploring the role of NLP in educational content generation. *Computers in Education*, 64(1), 25-38. <https://doi.org/10.1016/j.cie.2021.03.005>
- [5] Zhang, X., Huang, J., & Lin, Q. (2020). Leveraging deep learning for automated educational assessments. *AI in Education Journal*, 15(2), 101-118. <https://doi.org/10.1109/aiedj.2020.15>
- [6] Wang, L., & Lee, K. (2020). Personalization in learning using AI-powered adaptive systems. *Educational Technology Research and Development*, 68(5), 1452-1473. <https://doi.org/10.1007/s11423-020-09821-3>
- [7] SpaCy. (n.d.). Industrial-strength natural language processing in Python. Retrieved from <https://spacy.io>
- [8] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.
- [9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008.
- [10] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 8026-8037.
- [11] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265-283.
- [12] PostgreSQL Global Development Group. (n.d.).

PostgreSQL: The world's most advanced open-source relational database. Retrieved from <https://www.postgresql.org>.