

Implementation Of Internet Of Things (Iot) Based Lecture Schedule In Building G Floor 3 Padang State Polytechnic

Anggelina Wulandaria^a, Lifwardaa^{b*}, Firdaus^c

^{a,b,c}Department of Electrical Engineering, Telecommunication Engineering Study Program, Padang State Polytechnic
Corresponding author: lifwarda@pnp.ac.id

Abstract. The internet of things-based lecture schedule aims to improve the efficiency and effectiveness of time management. Thus, students, lecturers, and staff can know the time change so that they can prevent delays. By using Arduino Uno as the main component microcontroller, RTC as a clock module that calculates the time that will be displayed on the LCD. Bell and Speaker will sound when the clock on the RTC is the same as the inputted schedule clock, then sent to IoT via the ESP01 wifi module. This tool can also be used as a medium for delivering information because it is equipped with a mic. The RTC in this tool successfully calculates the time according to real time where the Bell and Speaker can work on time according to the schedule. When the bell sounds the measured voltage is 225 volts while the speaker is 2 volts. This tool has also been connected to ThingSpeak using the ESP01 wifi module.

Keywords— Arduino Uno, Bell, RTC, Speaker, ThingSpeak

*Manuscript received 15 Des. 2023; revised 29 Jan. 2024; accepted 31 Jan. 2024. Date of publication 17 Feb. 2024.
International Journal of Wireless And Multimedia Communications is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.*



I. Introduction

Internet of things (IoT) is a concept that aims to expand the utilization of continuous internet connectivity that allows us to connect equipment, and other physical objects with networked sensors and actuators to obtain data and manage their own performance. Internet of things or often referred to as IoT is currently experiencing many developments. The development of IoT can be seen from the level of convergence of wireless, microelectromechanical (MEMS), internet, and QR (Quick Responses) Code technologies. IoT is also often identified with RFID (Radio Frequency Identification) as a communication method [7]. Internet of Things (IoT) with automatic bell refers to the use of IoT technology to control or manage the bell or alarm automatically through the internet network. At this time there are still many campuses that still do not have an IoT-based lecture schedule such as the 3rd floor of building g of Padang State Polytechnic has not implemented an IoT-based lecture schedule system.

Arduino Uno is a microcontroller board based on ATmega328 (datasheet). It has 14 input pins of digital output where 6 of them can be used as PWM output and 6 analog input pins, 16 MHz crystal oscillator, USB connection, power jack, ICSP header, and reset button. To support the microcontroller to be used, it is enough just to connect the Arduino Uno Board to a computer using a USB cable or electricity with an AC-to-DC adapter or battery to run it. The main function of the Arduino Uno is as a microcontroller development board that can be programmed easily using the C/C++-based Arduino programming language. By using Arduino Uno, users can quickly develop electronics and programming projects involving various sensors and actuators. has digital and analog I/O pins that allow interaction with various devices. Pin on Arduino is a place where to connect cables between Arduino pins with input / output devices (usually connecting to the project circuit on the breadboard). Arduino pins are usually in the form of female headers so that to get a connection from the Arduino pin it is only

enough to plug the cable into the pin header hole. There are several pins on Arduino with different functions, each pin is labeled according to its name and function on the PCB.

The RTC (Real Time Clock) module is an electronic module that has a function to provide accurate time information in the form of digital data. The RTC module is equipped with an electronic circuit consisting of a crystal oscillator, a backup battery, and an electronic circuit that is responsible for storing the time and date. The RTC module used in this tool is the DS3231 RTC module. DS3231 is a device with six terminals, two of which are not mandatory for use, so basically we have 4 (four) main pins. Features of the DS3231 RTC Module are; RTC calculates seconds, minutes, hours and years, Accuracy: +2ppm to -2ppm for 0°C to +40°C, +3.5ppm to -3.5ppm for -40°C to +85°C, Digital Temperature sensor with ±3°C accuracy, can sound an alarm twice a day, square wave output can be programmed, aging Trim Register, 400Khz I2C interface, low power consumption, the circuit can handle the switch automatically if there is a battery failure, backup CR2032 battery with a life span of two to three years, and portable size. The specifications of the DS3231 RTC Module are; DS3231 Module operating voltage: 2.3V - 5.5V, can operate at low voltages, consumes about 500nA when using batteries, maximum voltage on SDA, SCL: VCC + 0.3V, and operating temperature: -45°C to +80°C.

Speakers are output hardware that functions to output the results of processing by the CPU in the form of audio / sound. Speakers can also be called a tool for outputting sound produced by music devices such as MP3 Players, DVD Players and so on. In this research, the speakers used are midrange types. This mid range speaker is designed to be able to produce sound from a frequency range of 500 Hz - 5,000Hz. This type of speaker is also capable of creating sounds that sound clearer and more focused. The size of these speakers is in the range of 4 to 6 inches only. These speakers are widely used to produce vocal sounds. Mid range speakers are usually part of a 3 way type speaker. However, mid range speakers cannot produce high-pitched sounds. The work of the midrange type speaker is only to perfect the other two types of speakers in the 3 way type. The function of the midrange itself is to capture high frequency sounds that cannot be captured by the tweeter and midbass. By using a midrange, the resulting vocal sound is thicker and clearer.

Types of speakers based on type include; 1) 3 Way speakers are speakers consisting of a midrange that functions to capture high-frequency sounds that cannot be captured by tweeters or midbass. 3 Way speakers are a type of speaker with three types of frequencies. This type of speaker uses low, mid and high frequencies. 3 Way speakers are capable of producing frequencies ranging from 20 Hz to 20 KHz. 3 Way speakers include midrange, tweeter and full range speakers. 2) 2 Way speakers are a type of speaker that uses two frequency systems, namely high and low. 2 way speakers are speakers consisting of

two tweeters which are audio components that play a role in spouting high-frequency sound. While two midbasses play a role in producing intermediate tones in an audio system, and consist of two x-overs whose job is to sort out the sound so that no 2 sounds overlap. The 2 Way Speaker includes a woofer and a tweeter.

Buzzer Electronics is an electronic component that can produce sound vibrations in the form of sound waves. In general, this electronic buzzer is often used as an alarm because its use is quite easy, namely by providing an input voltage, the electronic buzzer will produce sound vibrations in the form of sound waves that can be heard by humans. The working principle of the buzzer is very simple. When an electric current flows into the buzzer circuit, mechanical movement occurs in the buzzer. As a result, there is a change in energy from electrical energy to sound energy that can be heard by humans. Generally, the type of buzzer on the market is a piezoelectric buzzer that works at a voltage of 3 to 12 volts DC. The types of buzzers in the Arduino circuit based on the sound are divided into two, namely: active Buzzer, which is a buzzer that already has its own sound when given an electric voltage. This type of Arduino active *buzzer* is often also called a stand alone buzzer. Then passive buzzer, which is a buzzer that does not have its own sound. This type of *buzzer* is very suitable to be combined with Arduino because we can program the high and low tones. One example is a speaker. The specifications of the buzzer components are as follows: 1) Piezoelectric, which is in the form of a black tube which is the source of the sound. 2) Negative pin foot, which is a short buzzer foot to be connected to negative current or GND. 3) Positive pin foot, which is a long buzzer foot pin and its use is to be connected to positive current or VCC / 5V.

Furthermore, the power amplifier is an electronic circuit that serves to strengthen or enlarge the output signal on the sound. By using an amplifier, the sound coming from a microphone, speaker, megaphone will sound clearer and stronger.

LCD (Liquid Crystal Display) is a type of display media that uses liquid crystals as the main viewer. The features presented in this LCD are: consists of 16 characters and 2 lines, has 192 stored characters, there is a programmable character generator, can be addressed with 4-bit and 8-bit modes, equipped with a back light. The initialization process of arduino pins connected to LCD pins RS, Enable, D4, D5, D6, and D7, is carried out in the LiquidCrystal line (2, 3, 4, 5, 6, 127), where lcd is a variable that is called every time LCD-related instructions will be used.

The ESP01 Wifi Module is a wifi module that functions as an additional device for microcontrollers such as Arduino to connect directly to wifi and establish a TCP / IP connection. This module is also a complete chip which includes a processor, memory and also access to GPIO. ESP01 was developed by a developer from China called

"Espressif". ESP01 series products have many variants. One of the variants that we most often encounter is the ESP01 ESP-01 series. In 2014, AL-Thinker, a third-party manufacturer of this module, released the ESP-01 module which uses AT-Command as its configuration. The price is cheaper, has small module dimensions and low power usage. The ESP-01 module has a 2x4 DIL form factor with dimensions of 14.3 x 24.8 mm.

This module requires power around 3.3v and has three wifi modes, namely Station, Access Point and Both. This module is also equipped with a processor, memory and GPIO where the number of pins depends on the type 01 we use. So that this module can stand alone without using any microcontroller because it already has equipment like a microcontroller which we can see in Figure 1 below.



Figure 1. ESP01 Wifi Module

DFPlayer is an MP3 player module for Arduino that has a small size and its output can be directly paired to the speaker. This DFPlayer can function as a stand-alone module by adding batteries, speakers, and push buttons, or it can also use a combination of Arduino and other microcontrollers that have TX/RX capabilities. This DFPlayer supports several audio formats in general, such as MP3, WAV, and WMA and has supported micro SD with FAT16 and FAT32 file system types.

IDE is short for Integrated Development Environment, is a software used within the scope of Arduino because it is through this software that Arduino is programmed to perform the functions that are embedded through programming syntax to enter a command into the microcontroller so that Arduino can be used according to what is ordered.

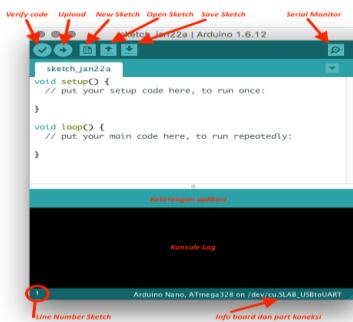


Figure 2. Arduino IDE Initial View.

In Figure 2 is a display of the Arduino IDE sketch,

where this sketch serves to write a microcontroller device program that supports the Arduino IDE. The following is an explanation of the parts on the initial display of the Arduino IDE: Verify in previous versions was known as Compile. Before the application is uploaded to the Arduino board, make it a habit to verify the sketch first. If there are errors in the sketch, an error will appear. The Verify / Compile process converts the sketch to binary code to be uploaded to the microcontroller. Upload this button serves to upload the sketch to the Arduino board. Even if we don't click the verify button, the sketch will be compiled, then uploaded directly to the board. Unlike the verify button which only serves to verify the source code. New Sketch Opens the window and creates a new sketch. Open Sketch Opens a sketch that has already been created. Sketches created with the Arduino IDE will be saved with the .ino file extension Save Sketch saves the sketch, but does not compile it. Serial Monitor Opens the interface for serial communication, we will discuss further in the next section. Application Description messages that the application does will appear here, for example *Compiling* and *Done Uploading* when we compile and upload the sketch to the Arduino board Console log Messages that the application does and messages about the sketch will appear in this section. For example, when the application compiles or when there is an error in the sketch we made, the error information and line will be informed in this section. This section will show the position of the active cursor line in the sketch. Board and Port Information This section informs the port used by the Arduino board.

MIT App Inventor is an open-source web application originally developed by Google, and currently maintained by the Massachusetts Institute of Technology (MIT). App Inventor allows new users to program computers to build software applications for the Android operating system. App Inventor uses a graphical interface, similar to the user interfaces in Scratch and StarLogo TNG, that allows users to drag-and-drop visual objects to build applications that can run on Android devices.

ThingSpeak is an IoT platform that allows us to collect, store, analyze, visualize, and act on data from sensors or actuators, such as Arduino, Raspberry Pi, Black BeagleBone, and other hardware. ThingSpeak functions as a data collector that gathers data from node devices and also allows the data to be captured into a software environment for historical analysis of the data. The distinctive features of the ThingSpeak workflow are: create Channels and collect data, analyze and Visualize data, Act on data using one of several Apps

IoT-based lecture schedule can be used as a reminder or marker of entry, return, change of hours and break time and can be used to convey information or announcements. Therefore, the implementation of an IoT-based lecture schedule system can help improve efficiency in lecture schedule management and reduce the possibility of delays [4]. By looking at the urgency of IoT, the author is

interested in conducting research by making a tool using Arduino Uno which can later be used as an IoT-based lecture schedule.

II. MATERIALS AND METHOD

The method used is to make a tool using Arduino Uno. The purpose of making this tool is 1) to provide automatic notifications and reminders to students and lecturers about scheduled lecture times, 2) the existence of an IoT-based lecture schedule system can make it easier for students, lecturers and staff to manage time, lecture schedules can be changed according to conditions such as regular schedules and Ramadan schedules using android applications and buttons on the *display*.

As a first step, it starts with the design stage. The design stage is a planning process before carrying out the tool manufacturing process. The process of making tools is carried out according to the design that has been done so that the tools made can function properly. Designing and making tools is the whole process of activities until this tool is complete. After the tool is finished, tool testing is carried out. Tool testing is one of the stages carried out to determine the success of the tool / system can work as expected. This tool testing was carried out in several trials.

III. RESULTS AND DISCUSSION

A. Design

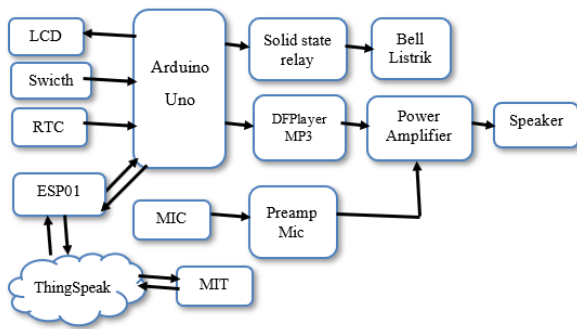


Figure 3. Block Diagram

The block diagram above shows the working system of the IoT-based lecture schedule system design tool using Arduino Uno as the controller of the RTC, Bell, Speaker and LCD and can change the schedule through a predetermined android application and ESP01 which functions to connect the device to android which in the block above explains the working system in outline which represents the main components that work according to their functions in the circuit. The design of the system block diagram can be seen that this tool uses internet media to access data on the ThingSpeak database. Arduino Uno in this tool is used as a microcontroller of an automatic lecture schedule tool that communicates with the ESP01

wifi module in order to access data on ThingSpeak. Real Time Clock is used to store real time displayed on the LCD and when the time on the RTC is right with the specified schedule, the Bell or Buzzer will sound and the Speaker will make a sound to notify the change of lecture hours. In this tool there is also a Mic and Speaker which is used for the delivery of information or announcements. MIT App Inventor on this tool is used as an application to set or change the lecture schedule where the data will be stored in ThingSpeak, the data on ThingSpeak will be sent to a device that has been connected to the internet. The design of this tool requires an electronic circuit that supports the work system. The following is a circuit used in hardware design.

B. Hardware Design.

LCD circuit with Arduino Uno

LCD is used for media displaying time from RTC. In this LCD the pins used are SCA, SCL, VCC and GND. The following LCD circuit image with Arduino Uno can be seen in Figure 4

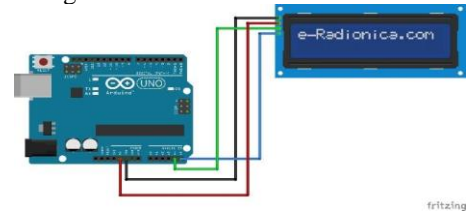


Figure 4. LCD circuit with Arduino Uno

Table 1. LCD Pin to Arduino Uno Pin Connection

No	LCD Pins	To Arduino Uno
1	VCC +5V	VCC +5V
2	GND (Ground)	GND (Ground)
3	SDA	A4
4	SCL	A5

RTC circuit with Arduino Uno

RTC is a module that can store time and date. Because it can store time, RTC is used as a clock for the lecture schedule system. The pins on the RTC are SCL, SDA, VCC and GND which are connected to the Pins on the Arduino Uno, namely A5, A4, 5V and GND. The following picture of the RTC circuit with Arduino Uno can be seen in Figure 5.

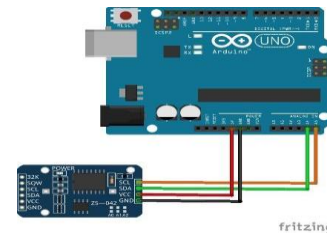


Figure 5. RTC circuit with Arduino Uno

Table 2 RTC Pin Connection to Arduino Uno.

No	RTC pins	To Arduino Uno
1	VCC +5V	VCC +5V
2	GND (Ground)	GND (Ground)
3	SDA	A4
4	SCL	A5

Buzzer circuit with Arduino Uno

The buzzer in this tool is used as a sound output for markers or reminders of time changes. The pins on the Buzzer, namely Positive (+) and Negative (-) are connected to the Arduino Uno Pin, namely Pin 12 and GND. Buzzer circuit image with Arduino can be seen in Figure 6

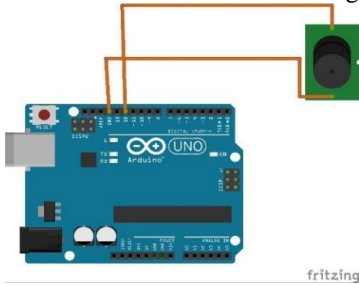


Figure 6. Buzzer circuit with Arduino Uno

Table 3 Pin Connection of Buzzer to Arduino Uno

No	Buzzer Pin	To Arduino Uno
1	Positive	Pin 13
2	Negative	GND

Mp3 DFPlayer and Speaker circuit with Arduino Uno

The circuit on this tool is used as a sound output where the music file or sound on the DFPlayer will be played by the Speaker according to the command that has been inputted on the Arduino Uno. A picture of the DFPlayer and Speaker circuit with Arduino can be seen in Figure 7.

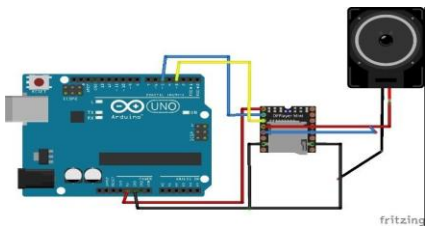


Figure 7. DFPlayer and Speaker circuit to Arduino Uno

Table 4. DFPlayer and Speaker Pin Connections to Arduino Uno

No	Speaker	DFPlayer	Arduino Uno
1	Positive	DAC_R DAC_L	
2		VCC	VCC +5V
3		RX	D5
4		TX	D3
5	Negative	GND	GND

ESP01 circuit with Arduino Uno

This circuit is used to connect the device to the internet using the ESP01 Wifi Module to Arduino Uno so that data can be sent to IoT and can be connected to a cellphone or laptop so that it can manage the device as desired. The ESP01 circuit image with Arduino can be seen in Figure 8.

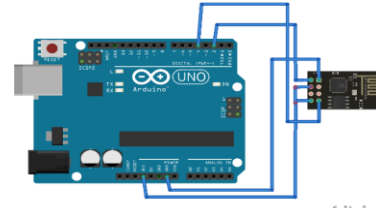


Figure 8. ESP01 circuit to Arduino Uno

Table 5 Pin Connection of ESP01 with Arduino Uno

NO	Pin ESP01	To Arduino Uno
1	RXD	D4
2	TXD	D2
3	VCC	VCC +3V
4	GND	GND

Mic circuit with speaker

This circuit is a mic circuit that is connected to the mic preamp and continued to the power amplifier. This circuit serves to amplify the mic voltage.

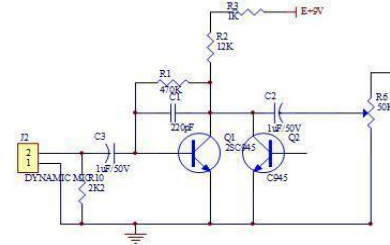


Figure 9. Mic circuit with preamp and power amplifier.

Button circuit with Arduino Uno

The button in this circuit functions to change the regular and Ramadhan schedule menus.

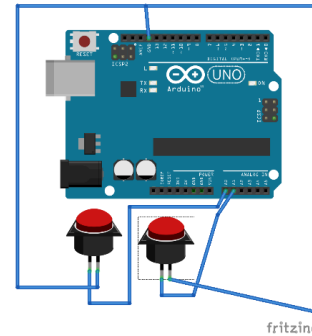


Figure 10 Button circuit with Arduino Uno.

Overall Tool Set

From the system design of each component that has been described, it can be seen how the overall system design is in the picture. Overall circuit image can be seen in Figure 10

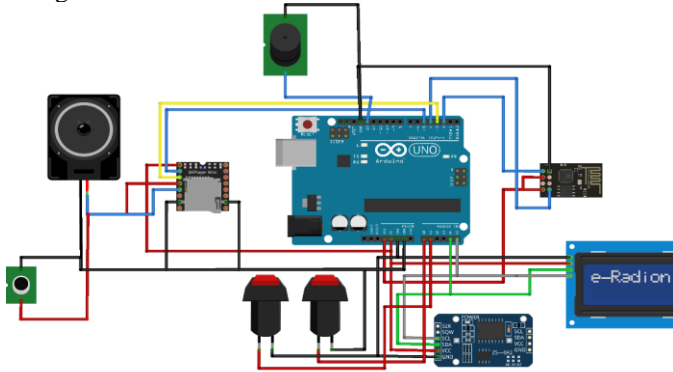


Figure 11. Overall circuitry of the tool.

Circuit Schematic and Layout

The following is a schematic view of the overall circuit made using proteus software as well as the layout of the device on the PCB. The schematic image can be seen in Figure 12 and the layout image can be seen in Figure 13.

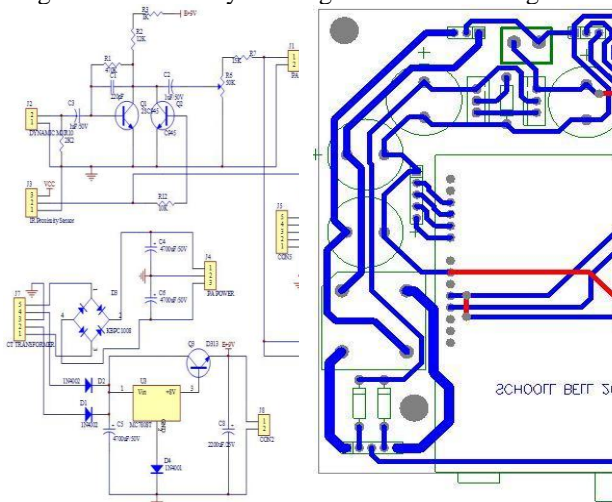


Figure 12. Schematic circuit

Figure 13. Layout

Software Design

After the manufacture of hardware devices, the next step is done right is software design. Software is a control program for the overall performance of the device which aims to make the designed hardware work as desired where the microcontroller will be coded so that all modules can work and be controlled by this Arduino Uno microcontroller.

Flowchart and Algorithm

Flowchart is a diagram used to explain an algorithm. Flowchart consists of symbols connected by an arrow (directed). The goal is to facilitate writing and facilitate reading

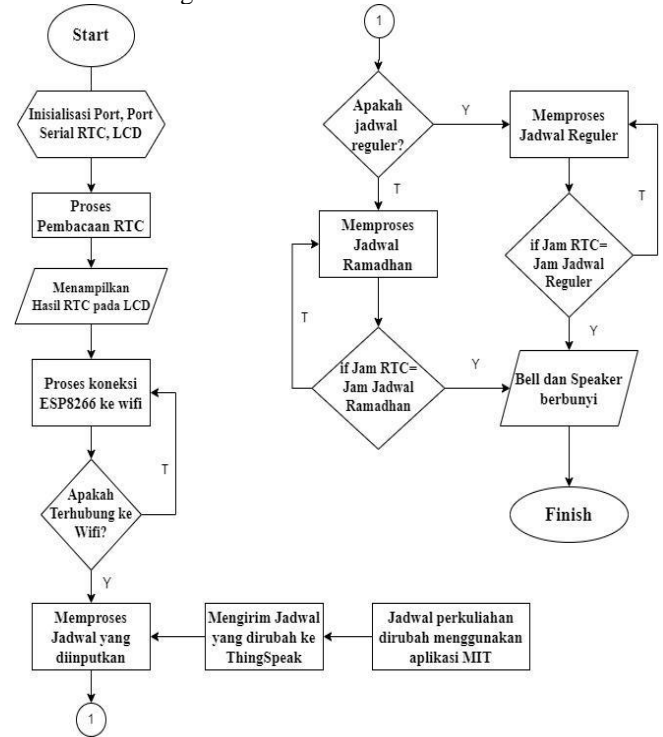


Figure 14. Flowchart of Tool Design

Thingspeak Creation

The Thingspeak creation step in this research starts by entering the *thingspeak* web and then pressing *new channel* as shown in Figure 15 below.

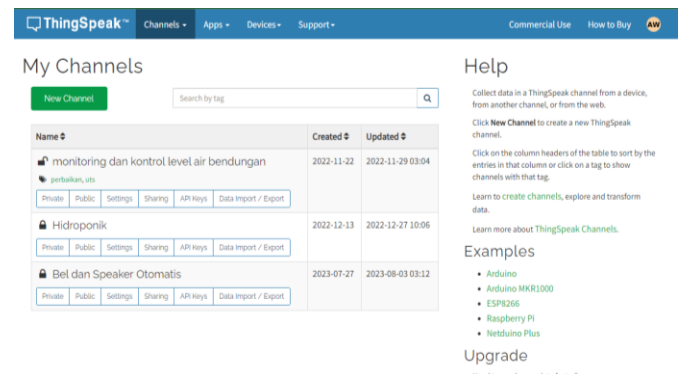


Figure 15. ThingSpeak channel creation view

Next, enter all the required parameters and press *save*. Then on the *sharing* menu change to *share channel view with everyone*, take the *API Keys* from the channel

created, so that it can read and send data, as shown in Figure 16 below.

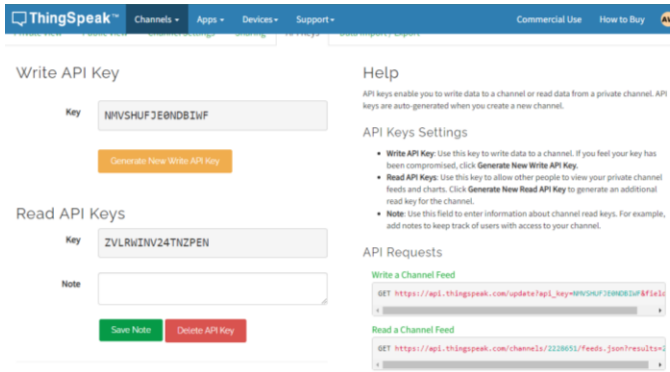


Figure 16 Channel Keys API View

In the next step, data transmission testing can be carried out. The results of the data transmission test can be seen in Figure 17 below.

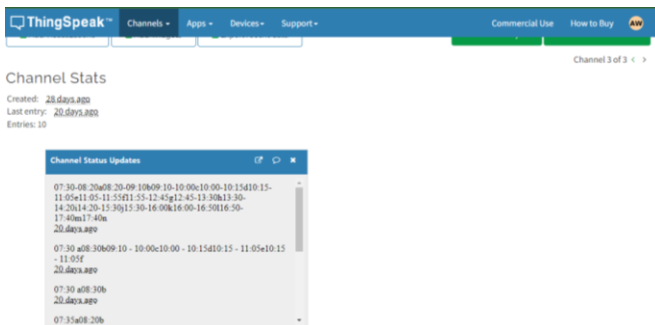


Figure 17 Display of Data Logged into ThingSpeak

Application Development

The application used to organize lecture schedules uses MIT App Inventor to create it. The steps for making this tool are; 1) make the application display as desired by entering the textbox, button, and others needed. Can be seen in Figure 18 and Figure 19



Figure 18 Initial view

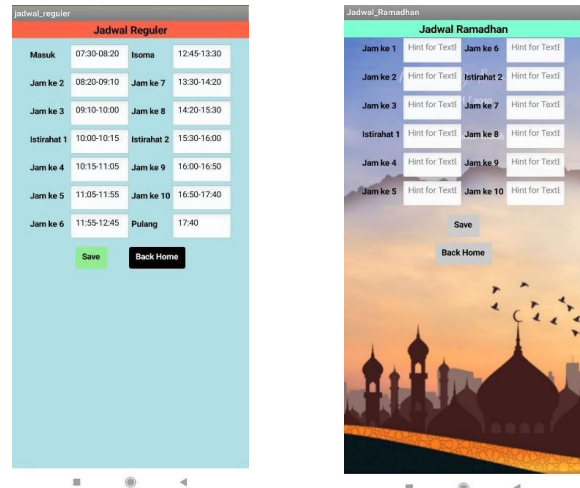


Figure 19 Regular and Ramadan Schedule Menu Display

Entering API Keys is useful for sending and receiving data which can be seen in Figures 20 and 21



Figure 20 Input Read API Keys

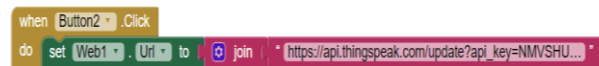


Figure 21 Input Write API Keys

Testing

Testing of this tool was carried out in several trials.

1. Bell/Buzzer Testing

Buzzer testing by measuring the voltage when the buzzer does not sound and when the buzzer sounds according to the schedule. When the buzzer does not sound the measured voltage is 0 volts, while when the buzzer sounds the measured voltage is 225 volts. This indicates that the buzzer can work according to the inputted program.

Table 6 Buzzer Voltage Measurement Results.

No	Hours	Voltage (V) Buzzer
1	07:30	225V
2	08:20	225V
3	09:10	225V
4	10:00	225V
5	10:15	225V
6	11:05	225V
7	11:55	225V
8	12:45	225V
9	13:30	225V
10	14:20	225V
11	15:30	225V
12	16:00	225V
13	16:50	225V
14	17:40	225V

2. Mic and Power Amplifier Testing

This test is carried out to find out how much the Mic voltage magnification is on the power amplifier.

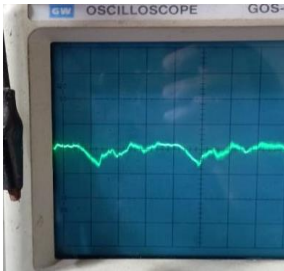


Figure 22 Oscilloscope Measurement Results on Mic



Figure 23 Oscilloscope Measurement Results on Power Amplifier

In the Oscilloscope, the Volts / Div is set at 20 mv / div, in Figure 22 it can be seen that the highest peak of the wave is as high as 0.4 div, so it can be calculated

$$\begin{aligned} \text{Voltage1} &= \text{Volts/div} \times \text{div} \\ &= 20 \text{ mv/div} \times 0.4 \text{ div} \\ &= 8 \text{ mv} \end{aligned} \quad (1)$$

In Figure 23, it can be seen that the highest peak of the wave is 2.8 div, so it can be calculated

$$\begin{aligned} \text{Voltage2} &= \text{Volts/div} \times \text{div} \\ &= 20 \text{ mv/div} \times 2.8 \text{ div} \\ &= 56 \text{ mv} \end{aligned} \quad (2)$$

Then it can be calculated that the voltage magnification on the power amplifier is as much as

$$\begin{aligned} \text{Magnification} &= \text{Voltage 2/Voltage 1} \\ &= 56 \text{ mv} / 8 \text{ mv} \\ &= 7 \end{aligned} \quad (3)$$

So, the magnification of the mic voltage on the power amplifier is 7 times.

3. Speaker Testing

Speaker testing is done using an oscilloscope. In Figure 24 we can see the highest peak of the wave as high as 1 div with Volts/Div of 2V/div.

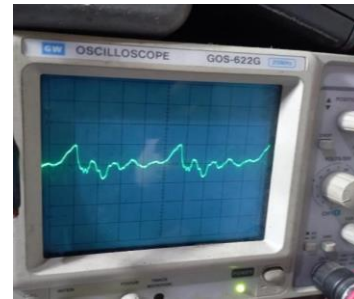


Figure 24 Oscilloscope Measurement Results on Speaker

Then the voltage on the speaker can be calculated as follows:

$$\begin{aligned} \text{Voltage} &= \text{Volts/Div} \times \text{div} \\ &= 2\text{V/div} \times 1 \text{ div} \\ &= 2\text{V} \end{aligned} \quad (4)$$

So, the voltage on the Speaker is 2 Volts when the Speaker is working.

4. Testing the RTC and LCD

This test is carried out to see if the RTC time is in accordance with real time and to see if the LCD and Buzzer work on time when changing hours.



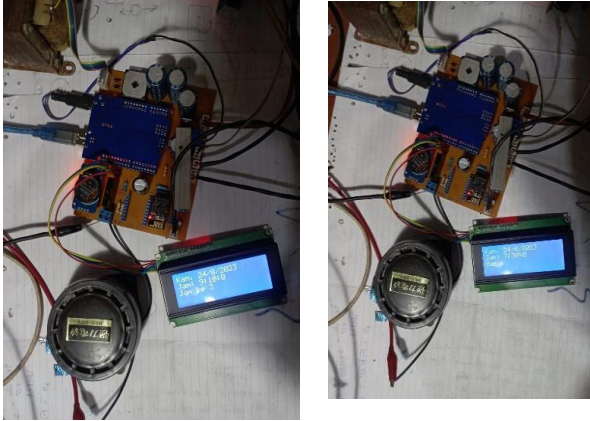


Figure 25 Results of RTC Display on LCD

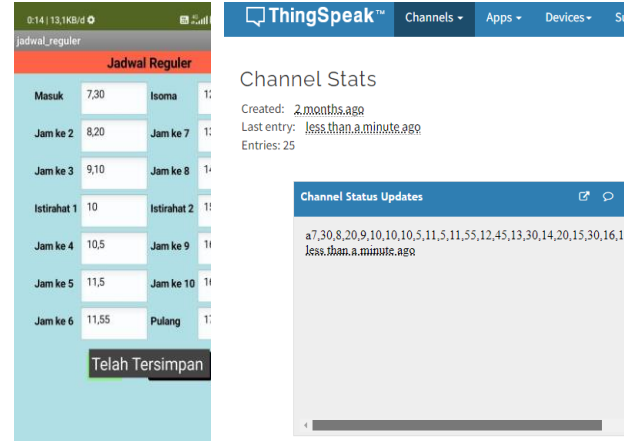


Figure 27 ThingSpeak and MIT App Inventor test results.

5. ESP01 Testing

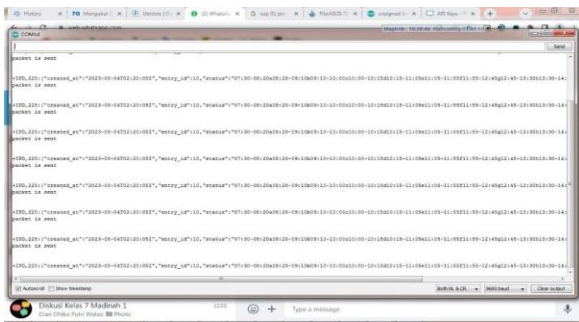


Figure 26 ESP01 Testing Results

Testing is done to find out whether the tool can connect to the internet using ESP01. Checking using the Serial Monitor on the Arduino IDE which can be seen in Figure 26. The Serial Monitor displays the data stored on ThingSpeak.

6. ThingSpeak and MIT App Inventor testing

The following are the results of ThingSpeak testing, when the lecture schedule is changed using the MIT application, the schedule data will be sent to the IoT platform, namely ThingSpeak. The picture of the test results of ThingSpeak and MIT App Inventor can be seen in Figure 27.

After testing and measuring this tool, some analysis is obtained, the Buzzer or Bell will sound according to the inputted schedule, it can be seen by measuring the Buzzer voltage when the Buzzer does not sound the measured voltage is 0 volts while when the Buzzer sounds the measured voltage is 225 volts. The second test is the voltage magnification of the Mic on the Power Amplifier. measurements were taken using an oscilloscope and then calculated using the formula $\text{Voltage} = \text{Volts} / \text{Div} \times \text{Div}$ which obtained a voltage on the mic of 8 mv and on the power amplifier the calculated voltage was 56 mv. To find out how many times the voltage magnification on the power amplifier, the result of the power amplifier voltage calculation is divided by the result of the mic voltage calculation, the result obtained is the magnification of the mic voltage by the power amplifier which is 7 times.

Speaker testing is done to find out if the speaker is functioning properly and see how much voltage is on the speaker when used. Measurements are made using an oscilloscope with Volts / Div of 2 volts / div where the peak wave height is 1 div. Then the calculation is done using the voltage formula $= \text{Volts} / \text{Div} \times \text{Div}$ which is the result of the calculation of 2 volts. RTC and LCD testing is done to see if the time on the RTC does not have an error or is not the same as real time. In this test the RTC works well. ESP01 testing can be seen in the Serial Monitor, the data taken is from ThingSpeak which imports data from the MIT App Inventor Application.

IV. CONCLUSION

After designing and making the "IoT-Based Lecture Schedule" tool, it can be concluded that;

By using Arduino Uno as the control of the RTC, ESP01, LCD, Buzzer and Speaker modules and Arduino IDE as a medium for making lecture schedule

programming and programs to read data from ThingSpeak so that the tool can work properly,

The lecture schedule can be changed using a cellphone through an application made using MIT APP Inventor. Schedule data will be sent to ThingSpeak to be stored and will be sent to a device connected to the ESP 01 Wifi Module so that the device can connect to the internet. The device will process the data received and run the device according to the schedule sent earlier.

Managing the lecture schedule can be done using a mobile phone and a button on the display box. On the mobile phone, managing the schedule uses an application made with MIT App Inventor which has 3 screens, namely the first screen as the Home menu, the second screen as the Regular Lecture Schedule and the third screen as the Lecture Schedule in the month of Ramadan. While in the display box used 2 buttons for each schedule. Information delivery can be done directly using a mic connected to the speaker.

REFERENCES

- [1] A.wRahim, "Design of E-Information Application for Funding Lecture Schedule Based on MobilewAndroid," *J. Process.*, vol. 12, no. 1, pp. 1000-1010, 2017
- [2] Moh. Dwi Chaidier Rizaldi, Ahmad R. Pratama "Development of a Lecture Schedule Information System for the Faculty of Medicine, Islamic University of Indonesia Based on Android" vol. 2 No. 2, 2021
- [3] Pauzan Muh, Yanti Indri, "Arduino-based Automatic School Bell Controlled Using *Mobile Applications*" vol. 10, No. 2, pp: 163-169, 2023
- [4] R. Malhotra, D. Kumar, and D. P. Gupta, "An android application for campus information system," *Procedia Comput. Sci.*, vol. 172, pp. 863-868, 2020, doi: 10.1016/j.procs.2020.05.124.
- [5] R. R. Putra, H. Hamdani, S. Aryza and N. A. Manik, "Automatic School Bell Scheduling System Based on RTC Using Microcontroller," *Media Informatika Budidarma*, vol. 4 No. 2, 2020.
- [6] Sarmidi and A. Nurtado, "Arduino Uno-based Automatic School Bell Simulation," *J. Manaj. and Tech. Inform.*, Vol. 3, No. 1, pp. 121-130, 2019.
- [7] Zein Afrizal, Farizy Salman, and Eriana Emi Sita, "Community Service to Socialize the Application of Internet Of Thing at Stmik Pranata Indonesia" vol. 1 No.1, 2023.