

---

# Design and build a DMX512 fixture based on Atmega2560.

Fikri M. Sabaya<sup>1\*</sup>, Iskandar Z. Nasibu<sup>2</sup>, Syahrir Abdussamad<sup>3</sup>, Amirudin Y. Dako<sup>4</sup>, Salmawaty Tansa<sup>5</sup>, Rahmad Hidayat Dongka<sup>6</sup>

Electrical Engineering, Universitas Negeri Gorontalo<sup>156</sup>  
Computer Engineering, Universitas Negeri Gorontalo<sup>234</sup>

[fikrisabaya97@gmail.com](mailto:fikrisabaya97@gmail.com)

---

**Abstract** -- Stage lighting systems require reliable communication protocols to support real-time control. DMX512 is a widely used industry standard because it is able to control multiple devices simultaneously with good synchronization. This research designed and implemented the ATmega2560-based DMX512 fixture that supports master and slave modes and integrates addressable WS2812B LEDs as visual outputs. The methods used include hardware design, firmware development, DMX512 signal measurement, and compatibility testing with commercial devices. The results showed that the system was able to produce signal parameters that conformed to standards, namely 90  $\mu$ s break, 10  $\mu$ s MAB, and 44  $\mu$ s time slot, and could operate stably in both master and slave mode. Compatibility testing also showed that the prototype could communicate well with commercial DMX512 devices. Thus, the developed system is feasible as an alternative solution that is more economical, flexible, and compatible with the DMX512 standard.

**Keywords:**

Arduino;  
DMX512;  
Usart;  
Input capture;  
Embedded.

**Article History:**

Received: March 15, 2026  
Revised: April 26, 2026  
Accepted: April 28, 2026  
Published: April 26, 2026

Copyright © 2026 IJEmCE. All rights reserved.

DOI: 10.xxxxx/ijemce.v1i1.xxxxx

---

## I. INTRODUCTION

### A. Background

Stage lighting systems have evolved from mere lighting devices to essential elements that support visual quality in the modern entertainment industry, architectural lighting, and event production. The development of digital control systems and LED technology has allowed for the creation of more flexible and dynamic lighting effects, requiring reliable communication protocols for real-time control. A wide range of communication protocols, such as KNX, DALI, ZigBee, and BACnet, the DMX512 remains a key standard in the entertainment lighting industry due to its ability to control multiple devices simultaneously with high levels of reliability and synchronization [1]. This protocol uses a data frame structure consisting of a break signal, Mark After Break (MAB), and data slots to efficiently transmit control information between connected devices [2]. Although it has been widely used, most commercial DMX512 fixture devices are relatively high in price and use a proprietary (closed-source) architecture. These limitations reduce flexibility in the customization process and become an obstacle for small to medium-scale event organizers who need more economical and adaptive lighting solutions. The development of microcontroller technology, especially the ATmega2560 (Arduino Mega2560), presents new opportunities in the development of embedded-based lighting control systems at lower costs and higher flexibility. The Arduino Mega2560 has several interfaces, including USART and Timer input capture and 54 input/output pins operating with a 16MHz oscillator crystal[3], [4]. However, existing research generally still focuses on the implementation of specific modes with more than 1 controller [5], [6], [7] and has not discussed much integration of addressable LED systems with dual-mode (master and slave) operating capabilities in a single platform [8], [9]. Therefore, this study proposes the design and implementation of a DMX512-based

lighting fixture using the Arduino Mega microcontroller (ATmega2560). The developed system is designed to support transmitter (master) and receiver (slave) modes, as well as integrate addressable LEDs WS2812B to produce dynamic lighting control. This system is expected to be a more economical, flexible, and open-source alternative solution, while remaining compatible with the existing DMX512 standard. The main contribution of this research lies in the development of an embedded-based lighting control system that is able to integrate DMX512 master and slave communication with addressable LED technology in real-time using an Arduino Mega microcontroller. The performance of the system is evaluated through signal measurement and functional testing to ensure compliance with the DMX512 standard and the operational stability of the system.

## II. JOURNAL PAPER PREPARATION

This study uses a Research and Development (R&D) approach to design the DMX512 fixture. Development begins with an analysis of the ANSI E1.11-2024 standard to determine the timing parameters and signal integrity. Next, the design of the hardware and software architecture is carried out. The final stage includes signal measurement and several tests. The research flow is systematically shown in Figure 1.

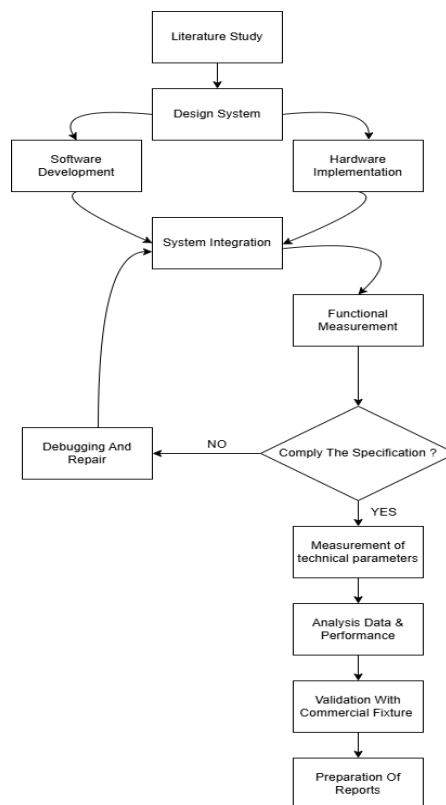


Figure 1. Research Flow

### A. Hardware implementation

Units *Transmitter* built using Arduino Mega 2560 which is based on the Atmega2560 chip and *transceiver* RS-485 (MAX485) features a 120 ohm terminator resistor [10], [11], the transmitter functions as a DMX Master with a transmission speed of 250 kbps and a capacity of 512 *Channel* via XLR-3 pin connector [1]. Receiver side built using peripherals *USART* and *Timer input capture* to detect *Break* and *MAB* in precision and processing data. The output side is composed of *LED* WS2812B (RGB) 800 kHz [12],[13], the formation of 56 LEDs in a 7-branch formation with each branch 8 pixel LEDs as validation of the DMX data received. All processing activities are carried out by the Arduino Mega2560 16MHz which

is able to manage various effects such as *Chase* and *strobe* through priority interrupt management. Power management system using 12V 5A main supply with regulation *Step-down* XL4005 with overcurrent protection [14], [15]. *Hardware* Equipped with a system *User:Stuttgar* in the form of navigation button control and a 0.96 inc OLED screen using 128x64 resolution [16]. System flow *Hardware* work can be seen in figure 2 System flow *Hardware*.

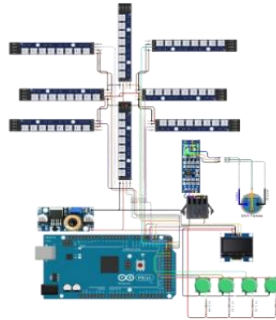


Figure 2. Hardware Design

### B. Software Implementation

Ensuring that the workflow runs regularly requires a key image to be worked on by the microcontroller. The *software workflow* is designed with a continuous loop that prioritizes data processing. The process begins with the system initialization phase which includes the configuration of the EEPROM. Next, the operation mode selection is performed to determine the mode, such as master mode (controller), slave mode (data receiver), or *settings* mode. Master mode focuses on data packet generation and transmission through the RS-485 interface, while slave mode works to process incoming signals through RS-485 to process DMX data or run a safety pattern in the event of a *timeout*. The entire process runs in a continuous main loop that integrates multiple LED visual effects with interrupt priority management. *The software flowchart* in this study can be seen in figure 3.

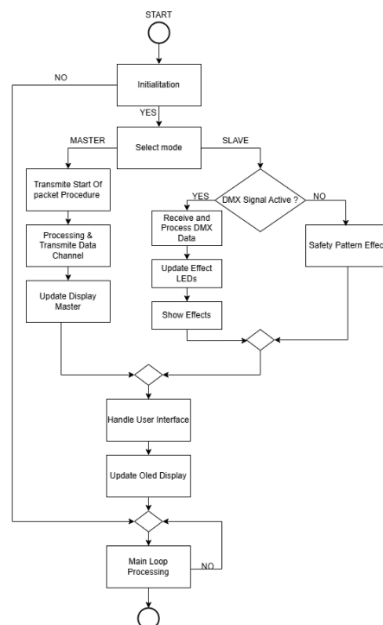


Figure 3. Software Flowchart

C. *Data Acquisition & testing method*

Data acquisition in this study was carried out to evaluate the characteristics of the DMX512 communication signal and the performance of the system that has been designed. Measurements were carried out to observe the main parameters of the DMX512 signal, including the duration of the break, Mark After Break (MAB), and the timing of the data in each channel slot. These parameters are used as a reference to assess the suitability of the system to the standard. The test is carried out in several scenarios to ensure the function and stability of the device. The test scenario includes operation in transmitter (master) mode, receiver (slave) mode, and combination testing of the two modes. In addition, compatibility testing is carried out by connecting the prototype with a commercial fixture. This test aims to ensure that the system is able to communicate with other devices that use the same standard. The parameters observed in this test include communication stability, the accuracy of the transmitted data, and the system's response to changes in channel values. To support the test process, a combination of tests is used that describes the interaction between devices in the system. This scheme is used as a reference in the implementation of the test. Figure 4 shows the combination scheme of testing used in this study. The data obtained from the measurement and testing process is then used as the basis for performance analysis in the results and discussion sections.

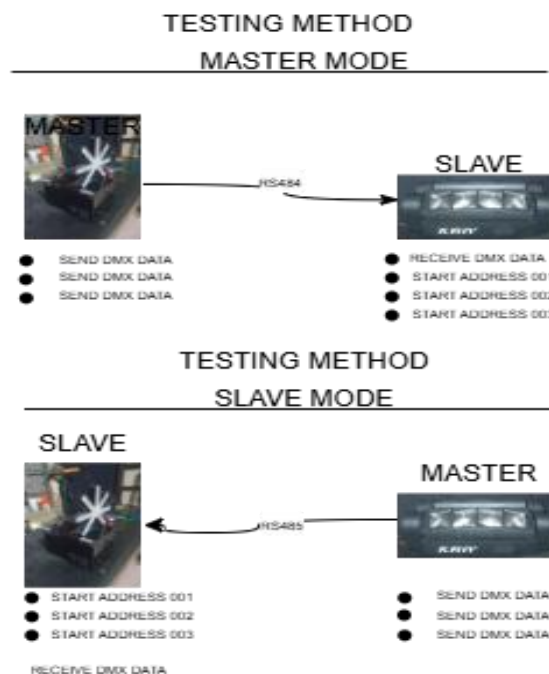


Figure 4. Combination of experiments

III. RESULTS AND DISCUSSION

Entering the third phase, the design was realized through the implementation stage which included hardware prototype assembly, firmware development, and integration of all supporting components. The implementation process was carried out in a modular manner to facilitate the debugging and validation process of the system on each unit. The final stage of the research was focused on functional testing and validation. In this phase, signal parameter measurements were carried out to ensure compliance with industry standards, system analysis in handling data, and direct comparison tests with commercial *DMX512* fixtures to prove the interoperability of the developed devices. The assembly results can be seen in Figure 5.



Figure 5. Assembly results

Technically, the DMX512 protocol has a fairly strict packet initial procedure. The initial procedure of the DMX512 packet consists of a *Low signal* of at least 88 microseconds, followed by a High signal of at least 8 microseconds and if the non-RDM device, data channel 0 (StartCode) must contain NULL. Furthermore, the channel is sent with an 8n2 asynchronous serial format and a transmission speed of 250 kbps which must be agreed between the sender and receiver. This protocol is equipped with a master and slave architecture that can be created separately, but the devices (fixtures) that are widely available on the market generally have master and slave modes, so the description below is in accordance with the master & slave architecture.

#### A. Master mode

The DMX512 protocol sets very strict timing specifications for the initial packet procedure, where the Break signal must have a duration of at least 88  $\mu\text{s}$ , followed by a Mark After Break (MAB) of at least 8  $\mu\text{s}$ . On devices with a *non-Remote Device Management* (non-RDM) classification, the Start Code or 0 th slot must contain a NULL value before proceeding with data transmission using the 8n2 format at a *baud* rate of 250 kbps. In the implementation of the master mode on the Arduino Mega2560 microcontroller, a technical problem was found in the built-in serial hardware library that did not support the direct method of generating break signals. To overcome this limitation, USART register manipulation was performed by dynamically changing the baud rate to force the TX pin to be at the logical level Low according to the standard duration. Based on the calculation of bit duration through the equation  $(1 \text{ bit})/(\text{Baud Rate}) = \text{duration of 1 bit}$ , the 100 kbps configuration was chosen because it resulted in a duration of 10  $\mu\text{s}$  per bit. By sending a NULL character (0x00) in the 8n1 format at that speed, the system succeeded in producing a total low duration of 90  $\mu\text{s}$  (accumulation of 1 start bit and 8 bits of data), which was automatically followed by a stop bit that formed a MAB signal of 10  $\mu\text{s}$  in duration. Once this initialization procedure is met, the system immediately switches the UART configuration back to the 8n2 format and a speed of 250 kbps to transmit 512 channel data slots. At this rate, each data frame has an exact duration of 44  $\mu\text{s}$  according to the 11 bit/250,000bps calculation results. The switching baud rate approach between 100 kbps for Break and 250 kbps for functional data is the main technical solution in realizing stable DMX512 communication without the need for additional hardware. Below is how to calculate it.

$$(1 \text{ bit})/(\text{Baud Rate})=?\mu\text{s}$$

$$\frac{1 \text{ bit}}{100.000 \text{ bps}} = 0.00001\mu\text{s} = 10\mu\text{s}$$

$$\text{Become}(8N1) = 1\text{startbit} + 8 \text{ bit data} + 1 \text{ stopbit} = 100\mu\text{s}$$

The total duration generated for 1 bit is 10 μs, so for the 8n1 configuration the total is 90 μs low 10 μs high for Mark After Break, (1startbit + 8 bits 0x00 = 90 μs + 10 μs High). In master mode after the break signal and MAB are sent, then the slot data format is sent in sequence 0-512 slots. The first slot 0 is the default NULL value for DMX512 non RDM (remote device management), then slot 1-512 is sent using the 8n2 data format with a duration of 44 μs for 1 channel slot. Creating a 44 μs channel slot frame is obtained from the following calculation.

$$\frac{1 \text{ bit}}{250.000 \text{ bps}} = 0.000004\mu\text{s} \text{ So as } = 4\mu\text{s} \times 11 \text{ bit} = 44\mu\text{s}$$

The generate break process can be seen in figure 6.

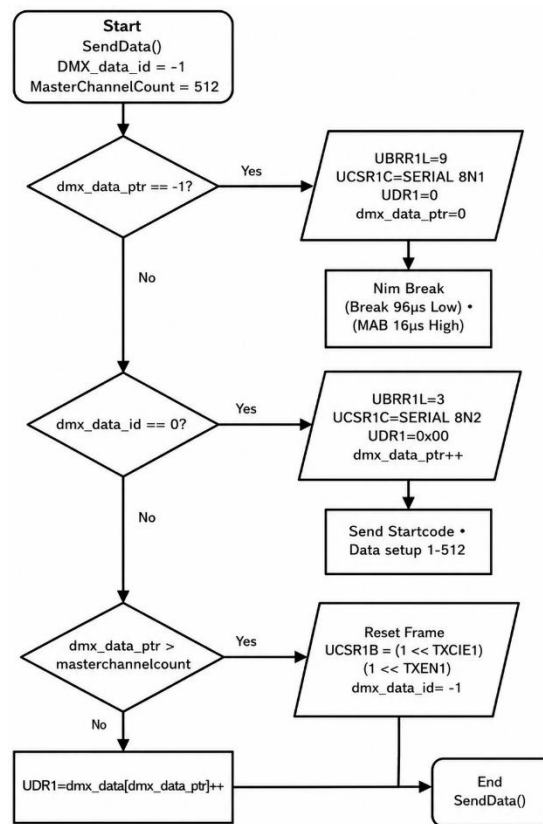


Figure 6. Flowchart Generate Break

The results obtained confirm the technical specifications of the DMX512, but also demonstrate the advantages of the method implementation in improving the reliability of the system compared to conventional approaches. The measurement data presented includes data frames, breaks, mark after breaks, channel data slots, start bits, stop bits. This validation includes testing the accuracy of the start bit and stop bit duration on each channel to ensure that no phase shift occurs during the transmission of 512 data slots.

*B. Slavic Mode*

The implementation of slave mode is focused on the accuracy of data packet detection and the mitigation of reading errors due to electromagnetic interference (EMI). To achieve high precision, the system utilizes the Input Capture unit on the Arduino Mega2560's Timer 5 (Pin 48). In contrast to conventional frame error (FE)-based break detection on USART, which is prone to software interruption latency, the use of input capture offers duration resolution at the microsecond level because it works directly at the hardware level.

The system also enables the Input Noise Canceler (ICNC5) feature that filters noise signals through repeated sampling over four clock cycles to ensure signal logic stability.

A. *Timer Configuration and Calculation*

To obtain accurate time resolution without the risk of overflow on a 16-bit timer, frequency division (prescaling) is applied. Using a prescaler 8 on a clock frequency of 16 MHz, the duration of each tick is determined by calculation:

$$Tick = \frac{Prescaler}{Fosc}$$

Fosc = 16MHz oscillator frequency

Become

$$\frac{8}{16.000.000} = 0,0005\mu s \times 1000 = 0,5\mu s$$

Calculating the value of the register:

$$Break\ Threshold = \frac{Target\ Time}{Tick\ Duration}$$

Become :

$$Break\ Threshold = \frac{88\ \mu s}{0,5\mu s} = 176\ tick$$

Based on this value, the threshold for detecting a break signal of at least 88  $\mu s$  is converted to a register value of 176 ticks (88  $\mu s$  / 0.5  $\mu s$ ). This value is used as a comparative constant in the program to validate the duration of the incoming low signal.

A. *Three-Signal Transition Detection Mechanism*

The system uses the flag polling technique on the TIFR5 register to detect logic changes. Transition analysis is performed by dynamically arranging the ICES5 (Input Capture Edge Select) bits through three main stages:

**First Transition (Falling Edge):** Early detection when the signal switches from *High* to *Low*. The *current timer* value is recorded as the starting point of the *break signal*.

**Second Transition (Rising Edge):** The system captures the reverse transition to *High logic*. The difference in the *value of the timer* is calculated to obtain the *duration of the low*. If the calculation results are within the standard range (85  $\mu s$  – 900  $\mu s$ ), then the pulse is confirmed as a *valid break*.

**Third Transition (Next Falling Edge):** This transition marks the end of the *Mark After Break (MAB)* phase and the beginning of the *Start Code*. Third transition detection guarantees that the synchronization between master and slave is perfectly aligned before serial data acquisition begins. The signal transition detection process can be seen in Figure 7. Duration calculations remain accurate despite overflow on the timer by applying logic:

$$\{T1 = Falling\ Edge\}\{T2 = Rising\ Edge\} \frac{T1}{T2} = Break\ Duration$$

*If timer overflow :*

$$T2 + T_{max} - T1 = \text{Break Duration}$$

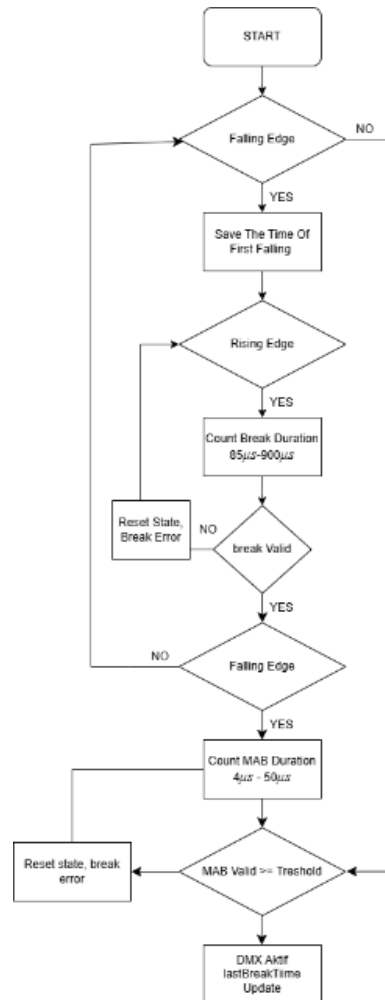


Figure 7. Break Detection Flowchart

### C. Serial Data Acquisition Management

Once the break signal is validated, the system switches to data reception mode via the USART1 unit. Each byte of data received on the UDR1 register triggers an Interrupt Service Routine (ISR) for the process of moving data into the buffer. The data sequence settings are managed by the finite state machine to verify the Start Code. If the Start Code is 0x00 value, the next channel data is stored in SRAM memory according to the predefined start address. This method ensures that memory usage remains efficient and allows flexible customization of device addresses through the settings menu. The comprehensive flow from transition detection to data storage is illustrated in Figure 8.

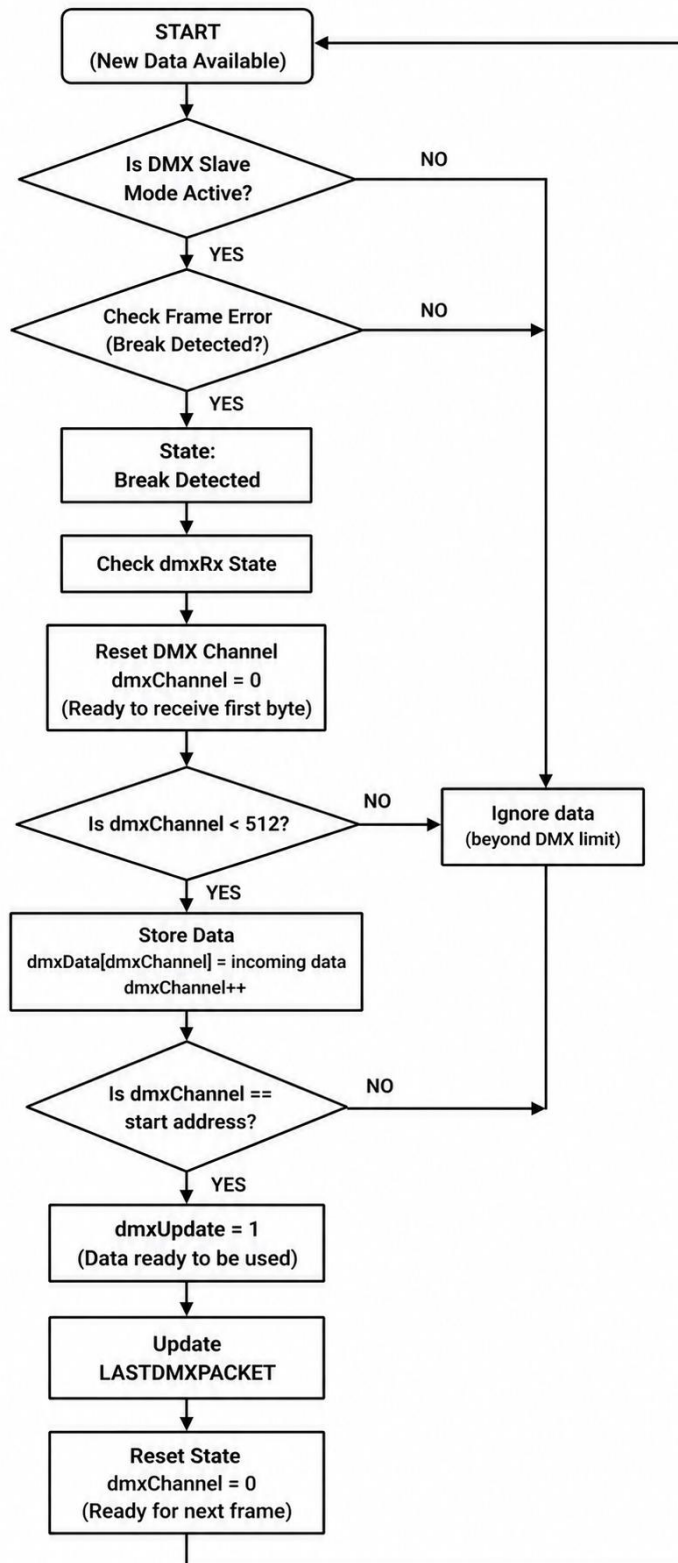


Figure 8. Data Processing Flowchart

D. Validation Testing

Validation of the protocol implementation was carried out through a series of quantitative measurements using a digital oscilloscope to ensure the system's conformity with the international standard USITT DMX512-A. The measurement was focused on critical parameters including timing accuracy, electrical characteristics, and signal integrity. The measurement point was taken on the TTL interface side between the Arduino Mega2560 microcontroller and the RS-485 transceiver in master mode. This aims to obtain pure data from the microcontroller without any distortions that may arise due to bandwidth limitations. The data obtained showed that the register manipulation method applied was able to produce precise signal parameters, including break duration, mark after break (MAB), startcode, and data channel slot structure. The test was carried out under active operational conditions using the rainbow effect to verify the stability of dynamic data transmission. To obtain in-depth signal analysis, a portable oscilloscope (Owon HDS200) is used with the following data capture procedures: (1) Visualization Optimization: Vertical scale settings (Volt/Div) and the Auto-Set feature are used to ensure that the entire wave frame is displayed at optimal resolution on the screen. (2) Probe Configuration: The probe attenuation is set at the 10x position to ensure the accuracy of voltage readings and minimize the effect of loading on the circuit. (3) Digital Data Acquisition: Signal recording data is downloaded in binary (.bin) format via a USB connection to a computer. This step allows for detailed identification of signal instruments through third-party analysis software. Visualization and in-depth analysis of the integrity of the captured signal can be seen in Figure 9 and Table I.

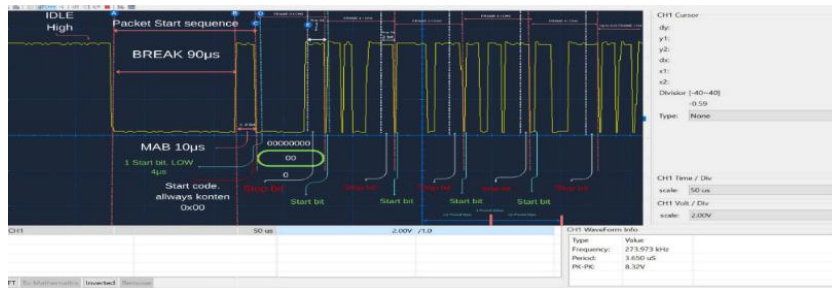


Figure 9. Measurement Results of All DMX Master Data Frames

Table I. DMX Master Signal Measurement Results

No.	Parameters	Standard Specifications	Measurement Results
1.	<i>Break Duration</i>	88 - 176 $\mu$ s	90 $\mu$ s
2.	<i>Mark After Break</i>	$\geq$ 8 $\mu$ s	10 $\mu$ s
3.	<i>Slot Time</i>	$\sim$ 44 $\mu$ s	44 $\mu$ s
4.	<i>Start Bit</i>	1 bit (4 $\mu$ s)	4 $\mu$ s
5.	Data Format	8N2	8 data bits, 2 stop bits

1) Analysis of Break Signal Characteristics

Identification and validation of the break signal was performed to ensure that the data packet headers generated by the prototype met the minimum duration standards of the DMX512 protocol. Measurements were made with an oscilloscope configuration on a 10µs div time base to obtain optimal measurement

resolution, as well as a 2V/div voltage scale to monitor the signal logic level. During the test, multiple channels were set at a maximum value (255) to create a clear transition contrast between the data phase and the break phase. Based on the results of waveform analysis, the duration of the low signal is calculated by multiplying the number of horizontal squares (grid) by the set time scale. The measurement results show that the prototype is able to produce a break signal with a constant duration of 90  $\mu\text{s}$ . This value is within the standard specification range of at least 88  $\mu\text{s}$  with a safety margin of 2  $\mu\text{s}$  from the operational minimum. The characteristics of the signal produced are in the form of sharp square waves with minimal rise and fall time, resulting in precision detection by various types of DMX receivers in the field. A detailed visualization of the measurement of *this break* signal is presented in figure 10.

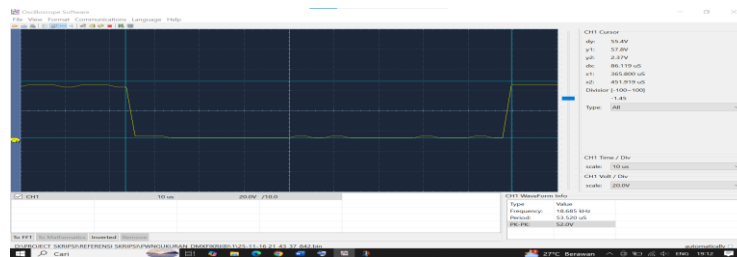


Figure 10. Break Signal Measurement Results

### 2) Analysis of Mark After Break (MAB) Signal Characteristics

The identification of the Mark After Break (MAB) signal is performed immediately after the validation of the break signal is met. In the DMX512 package structure, MAB is a transition phase in the form of high logic that functions as a pause and prepares to receive data. Based on international standards, the minimum duration for MAB is 8  $\mu\text{s}$  to ensure that the receiver unit has sufficient synchronization time before data processing begins. Testing on this prototype shows that the system consistently produces MAB signals with a duration of 10  $\mu\text{s}$ . The results of these measurements prove that the *baud rate switching* mechanism implemented in the USART register successfully creates a mark after break (MAB) phase. The resulting margin of 2  $\mu\text{s}$  aims to maintain the integrity of the communication despite the clock variation in the receiving device. The measured MAB signal characteristics show a high level of precision with clean *edge* transitions, as documented in the oscilloscope capture in figure 11.

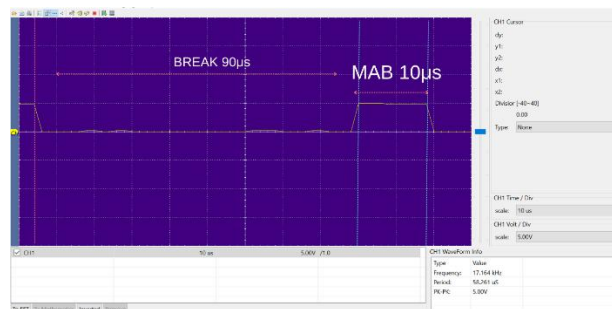


Figure 11. Mark After Break (MAB) Measurement Results

### 3) Channel slot frame analysis and start code

The data slot data frame transmission begins right after the *Mark After Break* (MAB) phase ends. Each *channel slot* in the DMX512 protocol uses an 8n2 asynchronous format consisting of one *start bit* (low logic), eight data bits, and two *stop bits* (high logic). The *start bit* serves as a *trigger* for the receiving unit to start reading the incoming bits and clock synchronization, while the two *stop bits* serve as a signal to detect the next *start bit*. According to the USITT standard, the first *byte* sent after MAB is the *Start Code*.

In this prototype, *the Start Code* is configured at a NULL value (0x00) to indicate that the distributed data packet is a standard light intensity control instruction for 512 channels and non-RDM. The oscilloscope capture confirms that the prototype is able to maintain the integrity of the 8n2 format consistently. The measurement results showed that the total duration of one *channel slot* was  $44 \mu\text{s}$ , according to theoretical calculations at a *baud rate* of 250 kbps ( $11 / 250,000 \text{ bps}$ ) The test was conducted in *master mode* by activating the *rainbow effect* on eight channels with a maximum intensity value (255) to verify the system's ability to handle the full data load without *jitter* or phase shift. The characteristics of the resulting data signal can be seen in Figure 12.

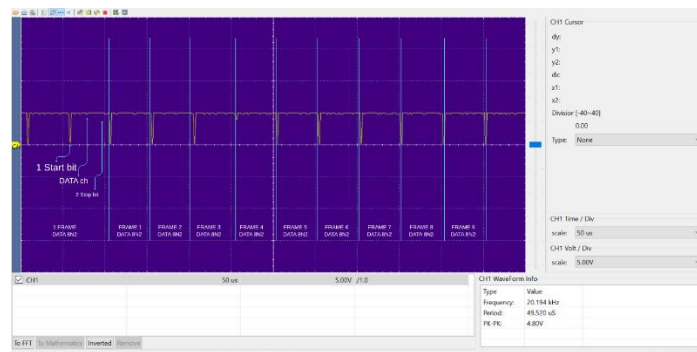


Figure 12. Channel Data Measurement Results

#### 4) Analysis of the Characteristics of the Start Bit

Observation of the *start bit* is made by adjusting the time base of the oscilloscope to  $1 \mu\text{s}$  div. This timescale decrease aims to increase the temporal resolution of the signal so that the duration of the logical transition can be measured with a higher level of precision. *The start bit* is the first component in each *channel slot frame* that serves as an initial synchronization marker for the receiving device to start the data acquisition cycle. Based on the measurement results, the *start bit* generated by the prototype has a constant duration of  $4 \mu\text{s}$  at the low logic level. This duration is equivalent to the duration of 1 bit at a *baud rate* of 250 kbps, which confirms that the system has met the technical specifications of the ANSI E1.11-2024 standard. The accuracy of *the start bit* duration is crucial to prevent *sampling errors* on the *receiver side*, especially when processing 512 channels continuously. A visualization of the start bit measurement results that show the sharpness of the signal transition can be seen in Figure 13.

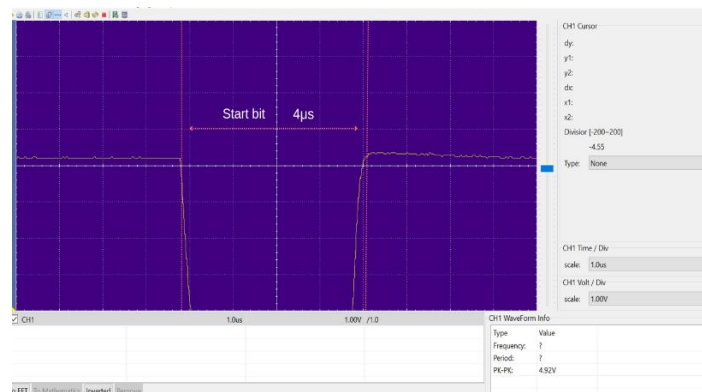


Figure 13. Start Bit Measurement Results

#### 5) Signal Quality Analysis

Based on visual observations and quantitative measurements using an oscilloscope, the signal produced by the prototype shows excellent quality characteristics with high data integrity. The details of the signal quality analysis are described as follows: Digital Signal Integrity: The resulting square waveform shows a sharp edge transition between *high* and *low* levels without any significant distortion. The *signal baseline* was observed to be stable without significant *drift* or *offset* phenomena, with amplitudes consistent at 5V TTL levels across the data packets. In addition, the system exhibited minimal *overshoot* values, thus minimizing the risk of logic misreadings on the receiver side. Timing Stability: Statistical analysis of ten consecutive data packets revealed a high level of precision with very low deviations. The *break duration* was recorded consistently at 90  $\mu$ s, MAB stability at 20  $\mu$ s and *slot time precision* at 44. With the periodicity of the package, the system was able to achieve a *refresh rate* of 44 Hz, which is the ideal standard for smooth stage light movement.

#### 6) Operational Validation and System Testing

Validation is not only carried out on the technical parameters of the protocol, but also includes functional testing through integration with commercial fixture devices (15-channel KAIY). The test device has an architecture of 8 double-sided RGBW LED units and two independent drive motor units. The test was carried out by transmitting control data on *specific channel mapping*, including motor controls (channels 1 and 2), *master dimmers* (channel 3), and *strobe master* (channel 4). The test results showed that the prototype was able to control all the functions of the commercial fixture smoothly without any latency or data errors (*glitches*). The success of this integration proves that the proposed timing manipulation method is fully compatible with the existing DMX512 infrastructure on the market. A summary of the results of the system's functional testing is presented in detail in Table II.

Table II. Operational Validation Test Results

NO.	Compatibility Validation	Test Results
1.	Start Address <i>test combinations</i>	<i>Start Address</i> in the execution accordingly.
2.	Dimmer Execution	<i>Dimmer Controll</i>
3.	Channel Execution	<i>Channels</i> are sent and received accordingly.
4.	Value <i>Channel</i>	Value <i>channels</i> are run accordingly, and can be controlled through <i>Dimmer</i> .
5.	Limit the number of <i>channels</i> sent and received.	<i>Channels</i> can be set to 1-512 to send or run.
6.	<i>Real-time effect</i>	The effect is executed according to the range and can be adjusted with <i>the Dimmer</i>
7.	Internal effects	The effect runs accordingly without the DMX512 protocol.

#### 7) Validation of Compatibility with Commercial Devices

Validation of compatibility with commercial devices is a fundamental parameter in this study to prove that the prototype system can operate within the standard stage lighting ecosystem. Given that the lighting

industry has established standardization, the level of interoperability with existing devices is a key indicator of the success of the system design. Integration testing was carried out using the Spyder model commercial DMX512 fixture. The results of the experiment showed that the commercial device responded to instructions from the prototype with a high level of reliability. Each *color mixing* command is executed with consistent color accuracy, while the dimming control displays linear and predictable response characteristics. In the execution of complex visual effects, the system has been proven to be able to maintain timing precision without any symptoms of lag or desynchronization. The successful integration is the result of the strict implementation of the protocol according to the USITT standard, combined with the provision of a timing margin of 4  $\mu$ s. The margin proved to be effective in accommodating different variations in clock tolerance and electrical characteristics in each receiving device in the field. Comprehensive data on the results of these integration tests is summarized in Table III, which confirms that the prototype is ready for use for small to medium-scale applications.

Table III. Compatibility Test Results

NO.	<i>Prototype</i>	<i>Commercial Fixture</i>	Remarks
1.	Master	Slave	No <i>packet loss</i>
2.	Slave	Master	No <i>packet loss</i>
3.	<i>Master Transmits 25 channel</i>	<i>Slave. Start Address (A001)</i>	Slave receives and processes data accordingly.
4.	<i>Slave. Start Address (A001)</i>	<i>Master</i> <i>Mode Rndom</i>	Data Ch1 <i>master</i> (Motor X) – Ch1( <i>Dimmer</i> ). Dimmer follows Motor X so it looks random.
5.	<i>Slave. Start Address (A002)</i>	<i>Master</i> <i>Mode Rndom</i>	Ch2 <i>master</i> (Y motor) – Ch1( <i>Dimmer</i> ). The dimmer follows the Y motor so it looks random.
6.	<i>Slave. Start Address (A003)</i>	<i>Mode Rndom</i>	Ch3 <i>master</i> ( <i>Dimmer</i> ) - Ch1 ( <i>Dimmer</i> ). Dimmer is executed with the same channel so it looks smooth.

Based on the data in Table 2, tests 1 and 2 prove that the system has high interoperability, where the prototype is able to act as a sender and receiver without packet *loss*. In tests numbers 3 to 6, the system was tested to respond to *changes in start addresses* and dynamic data mapping. The results in the random mode scenario showed that the prototype was able to extract data from various control channels (Motor X, Motor Y, and Dimmer) accurately according to the shift in the regulated address. When addresses are synchronized on the same channel (Test 6), the resulting light transition appears smooth, indicating that the data

processing latency on the *slave* side is minimal. Overall, these results confirm that the three-signal transition detection method and memory management applied have successfully created a DMX512 control system that is reliable and compatible with industry standards.

Based on the results of the design, implementation, and testing that have been carried out, the design and construction of *the DMX512 fixture based on the Arduino Mega2560 provides several main conclusions as follows:*

1. **Protocol Implementation:** The DMX512 control system in *master* and *slave* mode was successfully implemented with a high level of accuracy as per ANSI E1.11-2024 standard. The use of a *hybrid reception* method that combines a USART unit and an *Input Capture Timer* has been proven to be effective in generating and receiving data packets with precise *timing* parameters (*Break*: 90  $\mu$ s, *MAB*: 10  $\mu$ s, and *Slot time*: 44  $\mu$ s).
2. **Hardware Integration:** Control of *addressable* LEDs WS2812B as DMX512 channel outputs demonstrate highly responsive performance. The 56 LED configurations can be individually controlled with a *refresh rate* of up to 578 FPS, which guarantees visual stability without flicker symptoms.
3. **Performance and Optimization:** The prototype of the *developed fixture* demonstrates signal integrity and operational stability on par with industry-standard devices. Optimization of the Arduino Mega2560 microcontroller resources through a *real-time effect engine* enables simultaneous processing of complex lighting effects without disrupting the continuity of DMX data reception.
4. **Compatibility Validation:** Interoperability testing confirms that the system is fully compatible with the commercial stage lighting ecosystem, making it an economical yet professional alternative solution.

#### IV. ACKNOWLEDGMENT

I would like to thank Universitas Negeri Gorontalo, the Faculty of Engineering, and the Department of Electrical Engineering for their support. My sincere gratitude goes to my supervisors and examiners for their guidance and valuable feedback. I also thank the Electronics Laboratory staff for their technical assistance. Finally, I am grateful to my family and friends for their encouragement.

#### V. REFERENCES

- [1] A. S. B. Purba, "ELECTRICAL ENGINEERING STUDY PROGRAM, FACULTY OF ENGINEERING, UNIVERSITY OF MEDAN AREA," UNIVERSITY OF MEDAN AREA, MEDAN, 2024. [Online]. Available: [repository.uma.ac.id](https://repository.uma.ac.id)
- [2] A. E. 11 – 2024 ESTA, "ANSI E1.68 – 2024." Entertainment Services and Technology Association (ESTA), Apr. 25, 2024. Accessed: Oct. 27, 2025. [Online]. Available: <https://tsp.esta.org>
- [3] A. S. Ance, S. Tansa, and I. Z. Nasibu, "Design and Build a Household Scale Electrical Power Loss Prototype Based on Arduino ESP8266".
- [4] Atmel Corporation, "ATmega640/V-1280/V-1281/V-2560/V-2561/V Datasheet." Microchip Technology Inc, 2014. Accessed: Oct. 30, 2025. [Online]. Available: [https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561\\_datasheet.pdf](https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf)
- [5] C. Chousidis, I. Rigakis, M. Hadjinicolaou, and E. Antonidakis, "A MIDI to DMX512 Interfacing Protocol Implemented using Microcontroller".

- [6] S. Ilchev, "Design and Implementation of firmware for an embedded system that creates lighting and laser effects," in *Proceedings of the 24th International Conference on Computer Systems and Technologies*, Ruse Bulgaria: ACM, Jun. 2023, pp. 9–14. doi: 10.1145/3606305.3606310.
- [7] G. M. Vela, "Title: Degree in Industrial Electronics and Automotive Engineering," Google Sch., Jul. 2025, [Online]. Available: <https://hdl.handle.net/10459.1/468454>
- [8] T. Danielsson, "DMX-communication over Bluetooth low energy mesh network," 2019, doi: DiVA.org:uu-388635.
- [9] F. Yang, J. Chen, and Z. Cai, "A New Control System of Point Light Source," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 234, p. 012009, Mar. 2019, doi: 10.1088/1755-1315/234/1/012009.
- [10] N. M. Duy, "Development of an Arduino-controlled Smart Storage Rack for component reels using MQTT and RS485 communication protocols", doi: urn:nbn:de:hbz:1383 opus4-22237.
- [11] M. S. Tanvir, F. Ahmed, M. M. Islam, M. Islam, M. Hasan, and R. your Rashid, "Optimizing Master-Slave Broadcast Communication in Multi-Node Network Using RS485 Standard," in *2024 27th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh: IEEE*, Dec. 2024, pp. 1962–1967. doi: 10.1109/ICCIT64611.2024.11022430.
- [12] Y. Lozynskiy and T. Torubka, "Smart Information Board Based On The PSOC 6 MCU," vol. 7, Oct. 2022.
- [13] ws2812b WorldSemi Co., Limited, "ws2812b," *WORLDSEMI CO., LIMITED, WS2812B Datasheet WS2812B*. [Online]. Available: <https://www.alldatasheet.com/datasheet-pdf/view/1179113/WORLDSEMI/WS2812B.html>
- [14] I. Syafiq and S. Nurfidyah, "Prototype of Air Quality Monitoring Based on the Internet of Things," *PhD Thesis, Bangka Belitung State Manufacturing Polytechnic, Sungaliat*, 2025.
- [15] J. Tongkad, W. Ridwan, I. Z. Nasibu, S. Abdussamad, and S. Tansa, "Design of an IoT-Based Nursing Assistant Robot Using Nodemcu Esp8266 and Blynk," *Jambura J. Electr. Electron. Eng.*, vol. 6, no. 2, pp. 138–143, Jul. 2024, doi: 10.37905/jjee.v6i2.20740.
- [16] H. R. Dhani and S. Riskitasari, "The Effect of Load and Speed on Predicting Conversion Electric Bicycle Mileage," vol. 12, 2025, doi:<http://dx.doi.org/10.33795/elkolind.v12i1/7373>.