A Multi-UART Ultrasonic Sensor Controller for Real-Time Parallel Distance Measurement and Data Aggregation in Underwater Robotic Applications

Prabhat Thakur[1], Rishabh Pratap Singh[1], Kishor Kumar Das[1], Jagadeesh Kadiyam[1*] and Santhakumar Mohan[2]

[1] Indian Institute of Technology Mandi, Mandi, Himachal Pradesh, India
[2] Indian Institute of Technology Palakkad, Palakkad, Kerala, India

*Corresponding author. Email: kadiyamjagadeesh@gmail.com

Abstract. A multiple UART-based ultrasonic sensor controller that effectively integrates distance measurements from numerous ultrasonic sensors and combines their outputs into a single UART communication channel is demonstrated in this article. In order to reduce signal interference and enable concurrent data collection, each ultrasonic sensor functions via its own UART interface. All sensors real-time distance data is acquired by a central controller with several UART modules or serial ports. The controller then processes or filters the readings according to requirements and transmits the combined data frame to a host system, like a PC or microprocessor, via a single UART output. For multi-sensor applications, this method improves scalability, streamlines wiring complexity, and offers a single data stream for additional analysis or control. Applications demanding synchronized multi-point distance measuring, such as robotics, obstacle detection, and industrial automation, can benefit from this method of measurement.

Keywords: UART, ultrasonic sensor, microcontroller, multi-sensor application, obstacle detection, robotics.

1 Introduction

The Applications for ultrasonic sensors are multiple and include industrial automation, robotic navigation, obstacle detection, and distance measurement [1]. These sensors operate by sending out ultrasonic sound waves and measuring the time it takes for the echo to return after striking an object. The distance is then calculated based on the measured time of flight [2]. The low cost, non-contact nature, and reliability of ultrasonic sensors in detecting both metallic and non-metallic objects make them extremely useful. To provide a full view of the surrounding environment, distance data from multiple ultrasonic sensors positioned at various orientations must regularly be gathered in many modern embedded systems and robotic applications [3]. However, controlling the

V. Kumar et al. (eds.), *Proceedings of the International Conference on Advanced Design, Manufacturing, and Sustainable Energy Systems (ICADMSES 2026)*, Atlantis Highlights in Engineering 45,
https://doi.org/10.2991/978-94-6239-750-7_71

communication channels becomes highly challenging when multiple sensors are connected to a single controller. Traditional techniques that utilize a shared trigger/echo pin or a single UART interface frequently experience lower measurement precision, timing conflicts, and signal interference. Real-time, synchronized, and scalable distance measurement with several sensors is challenging to achieve due to these limitations. In order to address these issues, this project proposes a multi-UART-based Ultrasonic Sensor Controller that uses dedicated UART channels [4] that allow for the simultaneous capture of data from numerous ultrasonic sensors. To ensure reliable and interference-free communication, each sensor is interfaced via a separate hardware UART port on each microcontroller [5] of the ultrasonic sensor controller. For monitoring or decision-making, the controller gathers distance data from individual sensors, processes it, and then sends the entire data packet to an external device such as a master microcontroller, Raspberry Pi, or PC, via a single UART output [6]. This design has a number of advantages. By combining multi-sensor data into a single data stream, it streamlines overall data collection. It also allows the system expansion in a flexible way by adding more UART modules to add more sensors, if needed in the future. Also, the system uses a UART connection to make sure that serial data can be sent over long distances without any noise and with very little extra resources.

2 Existing technology

In embedded systems, multiple sensors often use the UART (Universal Asynchronous Receiver and Transmitter) protocol for serial communication [7]. The baud rate controls how many bits or signal changes are sent per second. Some of the most commonly used baud rates in serial transmission include 9600, 19200, 38400, 57600, and 115200 bps [8]. Most microcontrollers feature one or two UART connections [9], reducing the number of sensors that can be connected directly. This is solved by using hardware and software solutions for multi-UART sensor interfacing. No extra hardware, such as UARTs, can be added to the MCU hardware. Microcontrollers with fewer UARTs often use UART expander ICs like SC16IS752 (I2C/SPI to dual UART) or MAX3107 (SPI to UART). These chips allow parallel communication by converting I2C or SPI buses into multiple UART ports. Protocol conversion causes some latency and hardware costs. Using a multiplexer (MUX) circuit to link many UART sensors to one line is easy. The active channel can be selected using a GPIO pin. This method can't collect data fast or simultaneously because only one sensor can send and receive data. In systems that are sensitive to cost or don't have a lot of resources, software (bit-banged) UARTs can use general-purpose I/O ports to act as a UART port. But in this method, there is high CPU usage, timing errors, and performance that isn't always reliable at higher baud rates. These methods let a microcontroller talk to many UART-based sensors in a flexible and scalable way, but speed and reliability are always an issue.

3 Underwater ultrasonic sensor

This underwater ultrasonic obstacle-avoidance sensor, as shown in Fig. 1, is designed specifically for submerged use. It is small, accurate, well-protected, and needs very

Fig. 1. Underwater ultrasonic sensor [10]

little space for installation. It can handle a lot of different voltages, from 5 to 24 volts [10]. It is between 200 and 300 centimeters long while it is underwater. This sensor can work as a camera to avoid obstacles, as it has an IP68 protection grade and can handle normal underwater navigation speeds. The sensor emits a sound wave and waits for the

Fig. 2. Representation of ultrasonic sensor applications [10]

target item to return it. The echo pulse width shows how long it takes for the sound to get to an object and back [11]. This sensor helps underwater vehicles avoid obstacles so they can drive autonomously without any support, as shown in Fig. 2. At a depth of 6 meters, it can still easily avoid obstacles. Because this sensor is compact and has few blind spots, it is easy to use in a lot of different projects.

4 Methodology

A multiple UART-based ultrasonic sensor controller system is designed for collecting and merging data from multiple ultrasonic sensors, as shown in Fig. 3. Each controller is connected to one or more ultrasonic sensors via independent UART channels (UART 1, UART 2, UART 3). These dedicated UART lines en-able the concurrent communication of sensors and controllers, thereby reducing data interference and timing issues.

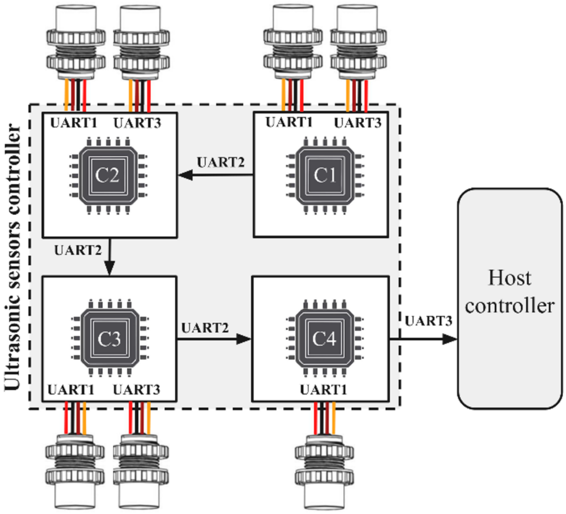

Fig. 3. Block diagram of the proposed system

1) Controller 1 receives data from Sensor 1 and Sensor 2. It then uses UART 2 to transmit the processed data to controller 2.

2) Controller 2 communicates with Sensor 3 and Sensor 4, merges the data it receives, and then transmits it via a UART 2 channel to Controller 3.

3) Controller 3 gathers data from Sensor 5 and Sensor 6 and merges it with data received from controller 2 and sends it to controller 4 via UART 2 channel.

4) Controller 4 processes data collected by Sensor 7 and integrates data from the previous controller, and transmits final data output via UART 3 channel.
Finally, the Controller 4 transmits the entire sensor dataset to a higher-level system, such as a PC, master microcontroller, or embedded processor, via a single UART line for display or further analysis. The schematic diagram of the ultrasonic sensor controller is shown in Fig. 4. This modular and hierarchical design provides:

- Scalability: An increase in the number of controllers enables the addition of additional sensors.

- Data efficiency: Each controller minimizes communication overhead by managing a limited number of sensors.
- Reliability: UART provides asynchronous, noise-resistant, and robust serial data transfer. This system provides the efficient, effective, and simultaneous measurement of distance for multi-sensor applications such as industrial automation, robotics, and obstacle detection.

5 UART-based data transmission and verification algorithm

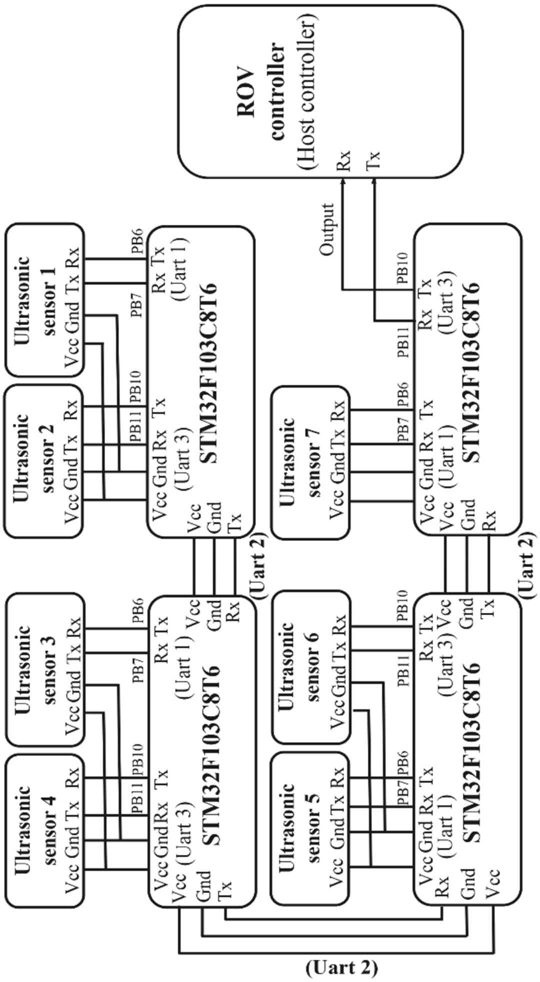

Fig. 4. Schematic diagram of the proposed system

The algorithm used for reliable data transmission and verification between the transmitter and receiver using the UART protocol. On the receiver side, the following steps are performed:

1) A structured data frame is put together from data from various ultrasonic sensors (such as Sensor 1 and Sensor 2).

2) The frame consists of a final Check Byte, Data Bytes, and a Signature Byte [12].

3) The Check Byte is generated by sending all data elements to an XOR procedure. This offers an easy and simple method of verifying for errors.

4) The entire data frame is subsequently transmitted serially to the receiver using the UART communication channel.

On the receiver side, the following steps are performed:

1) Each byte is read sequentially from the UART buffer.

2) The receiver compares the result with the received check byte after executing an XOR operation [13] on the received data bytes.

3) If the two values are identical, the data frame is considered to be error-free and valid.

4) The data is separated into different sensor channel values for display or further analysis after verification.

Thus, this methodology develops a communication framework that is both dependable and efficient for multi-sensor embedded systems. Despite the fact that a software serial can be used to carry out this process, it is ineffective for this purpose due to its limitations.

6 Fabrication

This two-layer PCB was designed with Easy EDA software to connect several ultrasonic sensors via a UART connection. The fabricated PCB of the ultrasonic sensor controller is shown in Fig. 5.

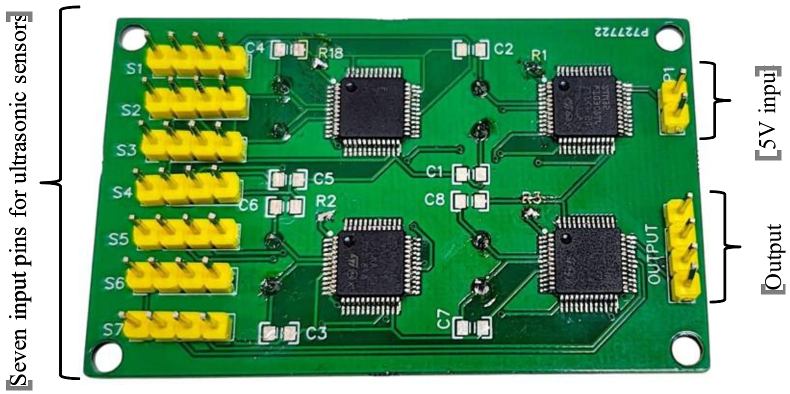

Fig. 5. Fabricated PCB

Data from seven ultrasonic sensors is received by seven UART input channels on the board, while processed data is transmitted to a host controller or microprocessor via a single UART output channel. A 5V input pin is used to supply power to the controller board and the seven ultrasonic sensors.

7 Results and discussion

Fig. 6 shows how the seven ultrasonic sensors are connected to the controller through their serial ports. The ultrasonic sensors are put in their glasses, which are filled with water. The glasses are around 70 mm tall. The STM32F103C8T6 controller connects to the ultrasonic sensor controller. This controller is built into the ship hull cleaning controller to get data and act as a test controller. The USB connection sends the estimated distance value to the host computer, where it is shown on the serial monitor in real time. This makes it easier to find and fix bugs and check that the sensor is working. The sensor comes with a standard Arduino program that is used at first. The Arduino sends a trigger command to the sensor periodically, and the sensor sends back a structured data frame with distance information.

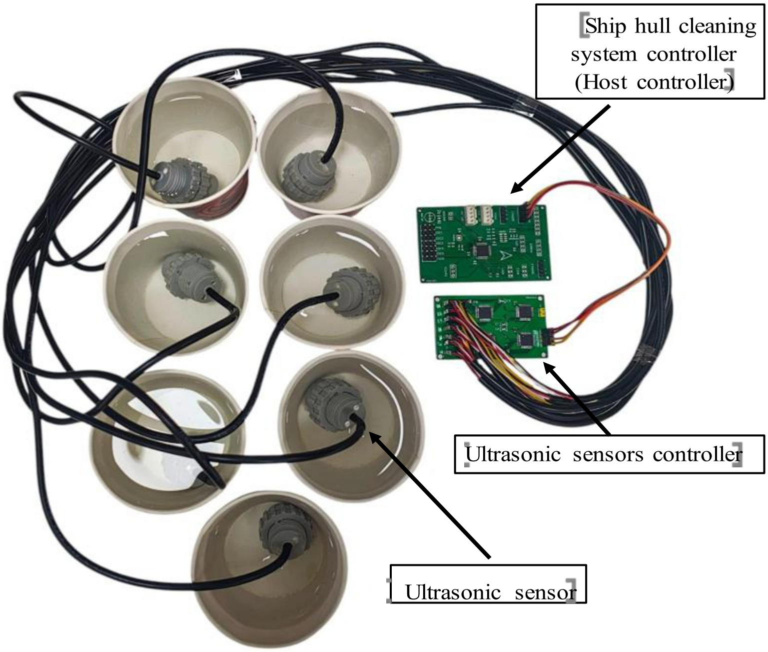

Fig. 6. Ultrasonic sensors interfacing with the controller

The distance sensor works in polling mode, which means it needs a command byte to start measuring. The Arduino transmits a single-byte command 0x55. When this

command is sent, the sensor sends back a 4-byte data frame with a set length. The response frame sent by the sensor consists of four bytes. To ensure correct data alignment, the program first checks for the presence of a frame header byte 0xFF.
Only frames that start with this header are processed further. This lowers the chance of decoding incorrect or improper data. A checksum method checks the integrity of the data

$$CS = B_0 + B_1 + B_2 \tag{1}$$

where B_0, B_1, and B_2 represent the header, high byte, and low byte, respectively. The received checksum byte is compared against the computed checksum to validate the frame. After successful checksum verification, the distance value is reconstructed from the high and low bytes:

$$Distance_{(mm)} = (B_1 << 8) + B_2 \tag{2}$$

The technique uses a 16-bit unsigned integer to measure distance in millimeters. It uses blocking delays (100 ms after sending a command and 4 ms before retrieving data) to make sure that communication is stable. These delays are effective, but they make the system less responsive and slow down the rate at which updates happen. Replacing the blocking delays with non-blocking timer-based routines makes the system run more efficiently. This lets the CPU do other things while it waits for the sensor, which makes the system more responsive and speeds up overall performance. Fig. 7 shows that the blocking code has a loop time that can be varied from 1000 to 5000 µs, depending on which processes are executed in each loop.

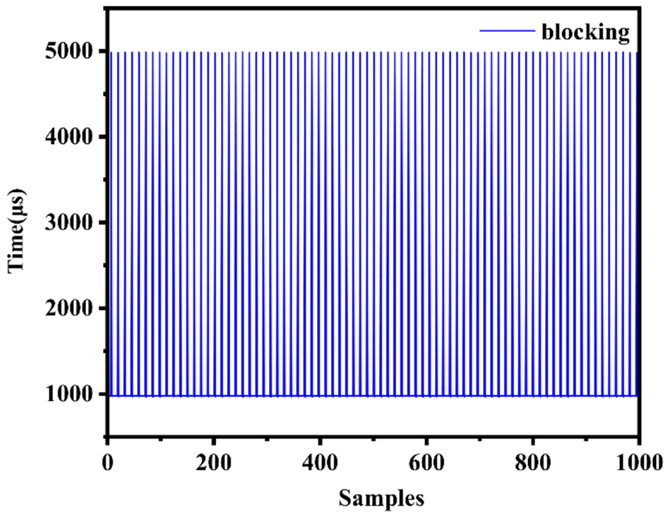

Fig. 7. Loop time with blocking code

After modifying the code to non-blocking using timers, the loop time decreases to 3–12 μs (Fig. 8), which makes the controller much more responsive and faster at updating.

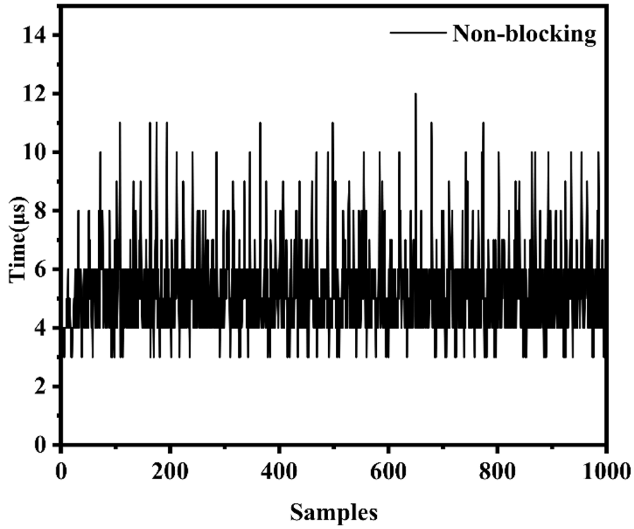

Fig. 8. Loop time with non-blocking code

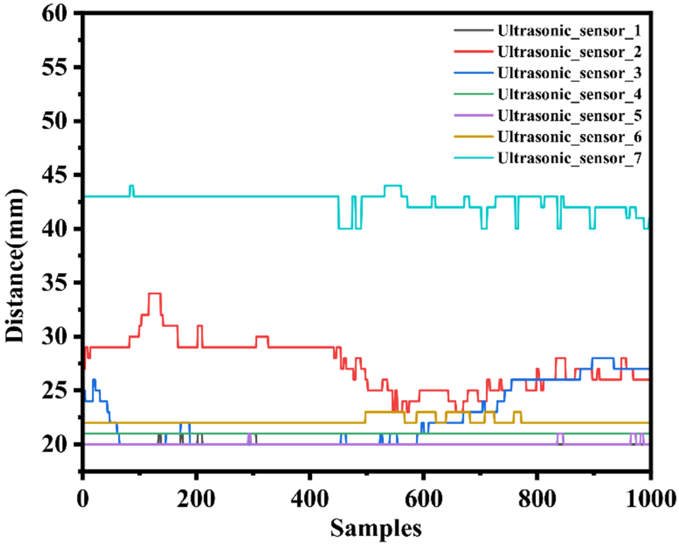

Fig. 9. Ultrasonic sensor controller output with all seven ultrasonic sensors data.

The STM32F103C8T6 receives data from ultrasonic sensors through UART for debugging and data extraction. The host computer gets encoded data packets that have to follow certain rules in order to decode and get the data from each sensor. After decoding the data on the host computer, the data from each ultrasonic sensor is taken out. Fig. 9 shows the output of the seven ultrasonic sensors' data that was decoded on the host computer.

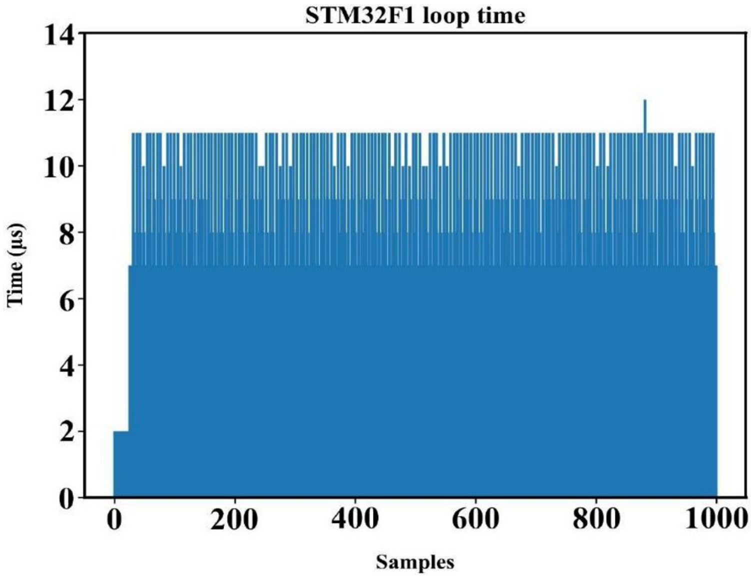

Fig. 10. Host controller loop time

The loop time of the host computer (STM32F103C8T6) is a maximum of 11 microseconds, as shown in Fig. 10.

The output waveform of an oscilloscope is shown in Fig 11. The output of the ultrasonic sensor controller (Tx pin of output pins) is connected to the oscilloscope for signal visualization. Each burst represents a complete frame of all 7 sensor values. The signal is seen in clusters (bursts) but not in continuous transmission, indicating periodic data delivery in the form of packets. The relatively clean waveform edges in the output of the oscilloscope indicate a robust UART signal, proper timing, minimal noise, or distortion.

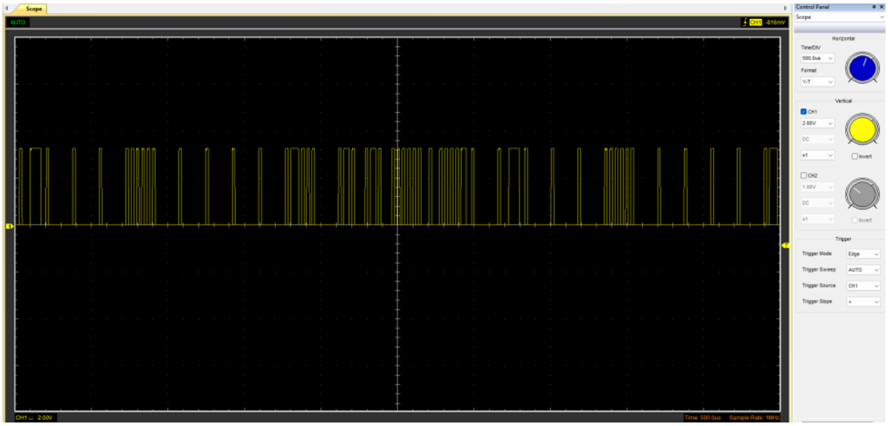

Fig. 11. Waveform of the serial output of the ultrasonic sensors controller in the oscilloscope

8 Conclusion

The implemented method shows that it is feasible to measure distance accurately and easily from multiple underwater ultrasonic sensors with a UART-based con-troller and a host microcontroller. The system is good for embedded sensing and educational purposes since it uses frame synchronization, checksum checking, and byte-wise decoding to make sure that the distance is measured correctly. This methodology is not limited to an underwater ultrasonic sensor only and can be applied to any UART-based sensor. This concept is applicable to robotic systems, autonomous vehicles, and distance measuring applications, where precise obstacle identification and environmental mapping require the use of many ultrasonic sensors.

Acknowledgments. Partially supported by the Indian Institute of Technology Mandi seed grant, India (# IITM/SG/JKD/110), IIT Guwahati Technology Innovation and Development Foundation (TIDF), India, (# TIH/TD/0213), Deep Ocean Mission, Ministry of Earth Sciences, India (# MoES/ PAM/DOM/196/2023 (E15710)), and the Anusandhan National Research Foundation Prime Minister Early Career Research Grant (ANRF-PMECRG)India(#ANRF/ECRG/2024/002069/ENS).

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Carullo, A., Parvis, M.: An ultrasonic sensor for distance measurement in automo-tive applications. IEEE Sensors Journal **1**(2), 143–147 (2001)
2. Zhmud, V.A., Kondratiev, N.O., Kuznetsov, K.A., Trubin, V.G., Dimitrov, L.V.: Application of ultrasonic sensor for measuring distances in robotics. In: Journal of Physics: Conference Series. vol. 1015, p. 032189. IOP Publishing (2018)
3. Bai, Y.W., Shen, L.S., Li, Z.H.: Design and implementation of an embedded home surveillance system by use of multiple ultrasonic sensors. IEEE Transactions on Consumer Electronics **56**(1), 119–124 (2010)
4. Peña, E., Legaspi, M.G.: Uart: A hardware communication protocol understanding universal asynchronous receiver/transmitter. Visit Analog **54**(4), 1–5 (2020)
5. Wheat, D.: Arduino software. In: Arduino Internals, pp. 89–97. Springer (2011)
6. Mukaro, R., Carelse, X.F.: A serial communication program for accessing a microcontroller-based data-acquisition system. Computers & Geosciences **23**(9), 1027–1032 (1997)
7. Dawoud, D.S., Dawoud, P.: Serial Communication Protocols and Standards. River Publishers (2022)
8. Laddha, N.R., Thakare, A.P.: A review on serial communication by uart. Interna-tional Journal of Advanced Research in Computer Science and Software Engineer-ing **3**(1) (2013)
9. Sharma, P., Kumar, A., Kumar, N.: Analysis of uart communication protocol. In: 2022 International Conference on Edge Computing and Applications (ICECAA).pp. 323–328. IEEE (2022)
10. DFRobot: Underwater ultrasonic obstacle avoidance sensor (3 m) (2026), available online: https://wiki.dfrobot.com/Underwater_Ultrasonic_Obstacle_Avoidance_Sensor_3m_SKU_SEN0598 (accessed January 13, 2026)
11. Toa, M., Whitehead, A.: Ultrasonic sensing basics. Texas Instruments Technical Report pp. 53–75 (2020)
12. Bardis, N.: Coding of checksum components for increasing the control reliability of data transmission for military applications. WSEAS Transactions on Commu-nications **7**(11), 1122–1131 (2008)
13. Lewin, M.: All about xor. Overload Journal pp. 14–19 (2012)