International Journal Software Engineering and Computer Science (IJSECS)
5 (3), 2025, 918-937
Published Online December 2025 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v5i3.5202.

RESEARCH ARTICLE Open Access

IoT-Based Prototype System for Automated Measurement of Human Height, Body Weight, and Temperature Using Sensors

Christian P. Colombini *
Information Systems Study Program, Faculty of Computer Technology, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.
Corresponding Email: colombinitian@yahoo.com.

Frencis M. Sarimole
Information Systems Study Program, Faculty of Computer Technology, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.
Email: matheosfrancis.s@gmail.com.

Received: July 29, 2025; Accepted: September 5, 2025; Published: December 1, 2025.

Abstract: This study presents the development of an IoT-based prototype for automatic measurement of body height, weight, and temperature using Arduino Uno microcontroller. The research responds to operational challenges observed at Puskesmas Cengkareng Timur, where conventional manual measurement tools contribute to process inefficiencies and frequent data recording errors during patient examinations. The prototype integrates three sensor components: HC-SR04 ultrasonic sensor for height measurement, Load Cell with HX711 amplifier module for weight detection, and MLX90614 infrared sensor for contactless temperature monitoring. Sensor outputs are processed through Arduino Uno and displayed via I2C LCD interface. Performance evaluation through laboratory calibration and field trials indicates measurement accuracy within 0.15% error margin across all parameters. Implementation of this device at the health center demonstrates potential for improving service delivery efficiency and reducing patient queue duration during preliminary health screening.

Keywords: Internet of Things; Arduino Uno; Biometric Measurement; Healthcare Automation; Ultrasonic Sensor; Infrared Sensor.

1. Introduction

Health is a very important part of human life that needs to be cared for and managed well. In modern times, the growth of information and communication technology has greatly influenced many aspects, including systems for delivering health care. One of the new technologies is the use of the Internet of Things (IoT) in health service systems as it allows things or devices to be connected and talk to each other. This offers real-time data collection and analysis capabilities [1]. At Puskesmas Kelurahan Cengkareng Timur, still mostly manual measurements are done for body height, weight, and temperature. Because these traditional methods often result in inaccuracies and inefficiencies in recording data, they then lead to poor quality of healthcare services offered to patients. The use of manual tools creates many chances for human error starting from the first measurement up until the last entry of data. Healthcare workers have to read measurement values by hand, write them down on patient records, and keep checking that everything is accurate all through this process. During busy hours when there are lots of patients increasing substantially, these manual processes become more problematic because staff members are under pressure trying to make sure everything is right. The problems seen at the health center go beyond just simple measurement mistakes. Manual recording systems give room for data loss, unreadable handwriting, and mistakes in writing down information from one system into another record-keeping system. Also, regular calibration and maintenance are needed for conventional measurement tools but they often do not have standard protocols to ensure accuracy remains consistent between different operators. These limitations directly affect the quality of care patients receive since healthcare providers rely on precise biometric data for diagnosis treatment planning as well as monitoring health conditions.

In view of these operational challenges, it is imperative to have a solution that increases both the efficiency and accuracy of the measurement of the key health parameters. Microcontroller and sensor technologies have continued to advance, thus creating great opportunities in this area. Arduino-based platforms are used for many automation applications because they are reliable and cheap; therefore, these platforms would be appropriate for the measurements needed in healthcare systems. Moreover, since Arduino technology is open-source, it gives one the leeway to customize according to institutional needs. This research is intended to create a prototype device that automatically measures body height, weight, and temperature utilizing IoT technology with an Arduino Uno microcontroller. The design of the prototype includes three main sensors: HC-SR04 ultrasonic for height measurement, Load Cell with HX711 module for weight measurement, and MLX90614 temperature sensor which measures body temperature without contact [2]. With this kind of technology, measurements can be done automatically which reduces possible errors in recording and increases the accuracy of data gotten [3]. The choice of certain sensor technologies was based on careful thought about measurement needs and practical limitations. The HC-SR04 ultrasonic sensor works on principles that reflect sound waves; it allows distance measurements without contact, thus making it appropriate for height measurements. Load Cell technology together with the HX711 amplifier module allows accurate weight measurement based on how strain gauges work. On the other hand, MLX90614 infrared sensors allow temperature measurement without contact physically; this is very important regarding infection control as well as patient comfort. Each of these sensors has its unique benefits that cater to certain measurement problems faced clinically.

Integration of these sensors into a single system offers both technical opportunities and challenges. The Arduino Uno microcontroller is used as the main processing unit, which controls the operation of sensors, processes measurement data, and controls output displays. Real-time data processing takes place in the microcontroller architecture, with results displayed on an I2C LCD interface for immediate viewing by healthcare personnel. The design of the system emphasizes synchronization between multiple sensors to achieve simultaneous measurements as this would reduce the total time taken to process one patient. Apart from specifications related to technology, this research also attempts to highlight some aspects of practical implementation that are important for an operation in a community health center. The prototype needs to perform well under real clinical conditions since factors like room temperature, humidity, and electromagnetic interference can affect sensor performance. Durability and ease of maintenance are important because healthcare facilities want equipment that can be used every day without much technical help. The design of the user interface is also important because healthcare workers with different levels of technical knowledge need to use the device easily. The expected results from this study include positive contributions toward improving healthcare delivery at Puskesmas Kelurahan Cengkareng Timur while potentially serving as a model for similar systems developed in other healthcare facilities. With the implementation of the proposed device, it is hoped that patient waiting time will be reduced and basic healthcare services will become more optimized. The automation of routine biometric measurements allows healthcare workers more time for patient interaction and clinical assessment rather than manual data recording tasks.

The study also discusses broader implications for the acceptance of healthcare technologies in places with limited resources. Community health centers usually have financial limitations that restrict their access to expensive commercial medical tools. Solutions based on Arduino provide affordable options that keep up with acceptable standards of accuracy yet are within financial reach. The proposed system’s modular design allows for future upgrades or changes as institutional needs change or new sensor technologies emerge. The laboratory calibration procedures and field testing validation methodology at the actual health center location ensure that the prototype works well under real-world operating conditions. Accuracy in measurement will be validated by comparing it against standard reference instruments to quantify precision and identify any systematic errors that need correction. Several problem formulations based on the above background guide this study for designing and constructing a prototype: first, accurate body height measurement using HC-SR04 ultrasonic sensor with Arduino Uno microcontroller; second, accurate body weight measurement through Load Cell sensor using Arduino Uno microcontroller; third, accurate body temperature measurement via MLX90614 sensor integrated into an Arduino Uno microcontroller; fourth synchronized and efficient integration of HC-SR04, Load Cell, and MLX90614 sensors within one Arduino Uno prototype circuit to improve measurement accuracy for height, weight, and body temperature parameters. These research objectives define specific technical goals related to practical healthcare delivery issues that have an immediate impact on patient experience and quality of service at the community health center.

2. Related Work

Recently, the use of microcontroller-based automated measurement systems has become a hot topic in research, especially for applications in healthcare and monitoring. Several studies have reported on the use of platforms based on Arduino integrated with different configurations of sensors to tackle measurement problems in various fields. The Arduino microcontroller, particularly the Arduino Uno and ATMega series, has been widely used in many automation projects due to its open-source architecture, low cost, and ease of programming [6][7]. These platforms provide enough processing power for real-time data acquisition while being accessible to researchers and practitioners working under constrained resource conditions. The combination of several sensors with Arduino boards makes it possible to develop advanced measurement systems that can carry out complicated tasks which previously required expensive commercial equipment.

There have been several studies that successfully implemented Arduino-based measurement systems in different contexts. Ginting *et al*. (2023) created an automated sorting system for Medan oranges based on circumference measurement using ATMega2560 microcontroller, demonstrating the ability of microcontroller platforms to perform precise dimensional measurements in agricultural applications [1]. Their study pointed out the role of sensor calibration and synchronization in obtaining accurate classification results. Likewise, Charaan *et al*. (2023) realized an IoT-based flood detection and prevention system using Arduino Uno with WiFi module; this shows how well Arduino platforms may be integrated with communication technologies for real-time monitoring and alert systems [2]. The study stressed the reliability of sensors based on Arduino in environmental monitoring applications where continuous operation and data transmission are vital requirements. Romadan *et al*. (2024) opened more IoT applications by making a soil moisture monitoring system for chili plants using NodeMCU ESP8266 with fuzzy logic methodology that integrates Blynk and ThingSpeak platforms for data visualization plus remote monitoring; this work showed how boards compatible with Arduino could be mixed with cloud-based services to create easy access toward agricultural monitoring solutions demonstrating scalability potential within such systems [3].

Ultrasonic measurements and motion detection have been widely used in distance and motion sensing for non-contact measurement. Nalla *et al*., (2025), discussed the detection of moving objects by ultrasonic radar technology [4]. They described the basic principles of operation and considerations for accuracy when using ultrasonic sensors in dynamic environments, providing insights that can help improve signal processing and noise reduction techniques for applications requiring accurate distance measurements. Sumampouw *et al*. (2022) described a prototype system for automatic regulation of DC fan speed using PIR sensors, ultrasonic sensors, and DHT11 temperature sensors based on an Arduino Uno and NodeMCU [8]. This demonstrates the feasibility of integrating various types of sensors within one microcontroller system. Their research dealt with synchronization problems as well as power management considerations when operating multiple sensors at the same time; this directly relates to technical requirements for multi-parameter measurement devices.

In the context of healthcare measurement applications, several researchers have explored Arduino-based solutions for biometric data collection. Ludya *et al*. (2023) created a height and weight measuring device to detect stunting in children aged 2-5 years, which included entertainment features to enhance user experience during the measuring process [5]. Their study underlined how critical it is to design medical devices with the user at the center, especially when young children feeling anxious about health assessments are the target users. This research revealed practical issues such as device stability, optimization of measurement range, and methods for displaying results that are appropriate in healthcare settings. Nurlette and Wijaya (2018) built a device that measures height and ideal body weight based on Arduino; their focus was on integrating measurement sensors with calculation algorithms to determine body mass indeks [10]. The prototype proved that physical measurement could be combined with computational analysis in order to give feedback on health assessments immediately. This study also discussed calibration procedures needed to achieve accuracy comparable to standard medical instruments used in clinical practice.

Infrared thermal sensors have gained particular interest in recent healthcare applications for temperature measurement, particularly due to the increased demand for non-contact thermometry. An accuracy analysis comparison study of the MLX90614 IR sensor system and ultrasonic sensors based on Arduino with standard thermometers was conducted by Inayah (2021) [11]. This provided data on the measurement precision and reliability of such systems under various environmental conditions. The research also identified factors that affect the performance of infrared temperature sensors, which included variations in ambient temperature, distance from measurement, and surface emissivity characteristics. These findings are very important for ensuring reliable non-contact temperature measurement implemented in clinical environments since accuracy here has direct implications on diagnostic decisions. Putro *et al*. (2024) developed an automatic contactless thermometer based on Arduino Uno as a solution to the problem of hygienic means of temperature measurement in healthcare facilities [18]. Their work also described considerations related to user interface design and response time optimization that would ensure practical usability in high-volume screening scenarios.

Weight measurement systems using load cell technology have also been investigated in Arduino-based implementations. Asri *et al.* (2024) designed an automatic weighing machine based on Arduino, examining the integration of load cell sensors with HX711 amplifier modules to achieve precise weight measurements [13]. Their research addressed calibration methodologies, load distribution effects, and long-term stability considerations relevant to weight measurement applications. The study provided practical insights into mechanical design requirements that ensure accurate force transmission from the weighing platform to the load cell sensor, minimizing measurement errors caused by structural deflection or uneven load distribution.

Combining IoT features with Arduino-based measurement systems has created new opportunities for remote monitoring and data management in healthcare applications. Gupta *et al.* (2020) designed an integrated healthcare monitoring device for obese adults using Internet of Things technology that merged several biometric sensors with cloud connectivity for continuous health tracking [16]. Their system showed how Arduino platforms can work as gateways to send health data over the internet so healthcare providers can check on patients' conditions without being there in person. The study discussed issues related to data security, choosing a communication protocol, and reducing power consumption in battery-operated devices. Rizal *et al.* (2023), gave theoretical frameworks and implementation strategies for IoT concepts which can be used as a guideline for architecture design, communication protocols, and data management approaches that apply to healthcare monitoring systems [9]. It has been noted in their work that scalability and interoperability are important when designing IoT-enabled medical devices since such devices may need to integrate with existing healthcare information systems.

Programming and software development aspects of Arduino-based systems are well documented in technical literature. Kadir (2017, 2018) gave much guidance on Arduino programming, integrating sensors, and developing applications using Processing among other platforms as well as Android applications via App Inventor [6][12][14]. These resources give practical implementation strategies for acquiring sensor data, processing signals, and developing user interfaces related to the construction of measurement devices. Wicaksono (2019) described interfacing sensors with detailed explanations applicable both to hardware connection schemes and how software is programmed for an application on Arduino [7]. Blum (2020) had advanced tools of Arduino discussed along with engineering practices that included optimization strategies, debugging methodologies, and best practices toward robust embedded systems development [17]. This programming resource is very critical in implementing complex algorithms required by multi-parameter measurement devices concerning sensor fusion and data processing. Sathish Kumar *et al.* (2016) showed the use of Arduino Uno in IoT-based smart systems garbage alert mechanism which gave basic knowledge about sensor integration and alert system design that can be applied to healthcare monitoring contexts [15].

Though there has been great improvement shown in the past study, a few gaps still exist in creating integrated biometric measurement systems for primary healthcare facilities. Most of the studies done before focused on single-parameter measurements or dealt with specific application domains outside clinical healthcare settings. There has been limited research regarding the sensors for height, weight, and temperature being integrated simultaneously on one Arduino-based platform meant for community health center operations. Moreover, very few studies have covered the real problems involved in deploying such systems to resource-constrained healthcare environments where technical support may be limited and equipment needs to endure continuous daily use. This research will try to fill these gaps by developing a prototype that integrates multiple measurement modalities while taking into account operational requirements and constraints typical of community health centers in developing regions.

3. Research Method and Equipment Preparation

The design methodology employed in the research follows several sequential steps illustrated through the flowchart diagram displayed in Figure 1 below.

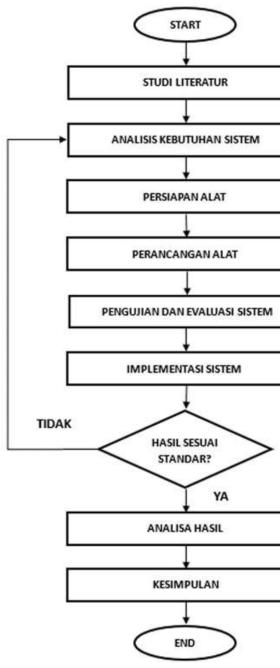


Figure 1. Research Roadmap

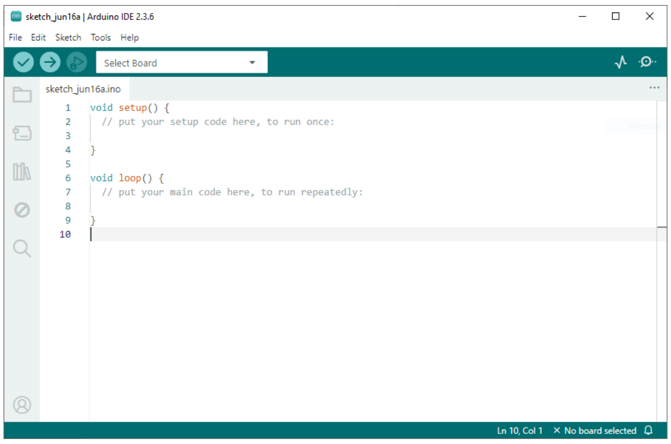
The research adopts a structured and iterative methodological approach, beginning with the Literature Study phase to gather relevant information and establish foundational understanding. Following that, System Requirements Analysis identifies the specifications and functionalities needed for the prototype. Once requirements are clearly defined, the process advances to Equipment Preparation and Tool Design, where necessary hardware and software are assembled and the prototype design is developed. The subsequent critical phase involves System Testing and Evaluation to verify functionality and performance. When testing results meet established standards, the system proceeds to Implementation. However, if results fall short of expectations, the process loops back to System Requirements Analysis for refinement and improvement (iterative cycle). After successful implementation and standard compliance, Result Analysis examines collected data, culminating in Conclusions drawn from the entire research sequence [4]. The research methodology begins with designing an automated measurement prototype for height, weight, and body temperature based on Internet of Things (IoT) technology using Arduino Uno microcontroller. The development process starts with sensor selection and testing, specifically the HC-SR04 ultrasonic sensor for height measurement, Load Cell with HX711 module for weight measurement, and MLX90614 sensor for non-contact body temperature measurement. After sensor selection, hardware integration with Arduino Uno takes place, where sensor data undergoes processing and displays on LCD I2C [5]. Subsequently, the prototype undergoes field testing at Puskesmas Kelurahan Cengkareng Timur to evaluate measurement accuracy and efficiency. Testing data receives thorough analysis to determine accuracy levels and effectiveness in enhancing healthcare services. Research outcomes aim to deliver innovative solutions for optimizing basic health measurement processes and reducing patient waiting times.

3.1 Equipment Preparation

The chapter discusses hardware and software preparation required for designing the prototype Automated Height, Weight, and Body Temperature Measurement Device Using Sensors Based on Internet of Things with Arduino Uno. Below are the software and hardware utilized throughout the research.

3.1.1 Arduino IDE

Arduino IDE (Integrated Development Environment) serves as an integrated development platform specifically designed for programming Arduino microcontrollers. The software provides developers with a complete workspace for writing code, compiling, and uploading to Arduino boards easily, featuring a text editor, message area, console, and toolbar with basic functions. Arduino IDE advantages include an intuitive user interface, extensive global community support offering various libraries and code examples, plus cross-platform compatibility across Windows, macOS, and Linux [6]. Commonly used libraries on Arduino IDE vary depending on project requirements, though some fundamental ones include Wire.h for I2C communication and SoftwareSerial.h for additional serial communication. The platform proves essential for every Arduino-based project development, ranging from simple LED control to complex automation systems.

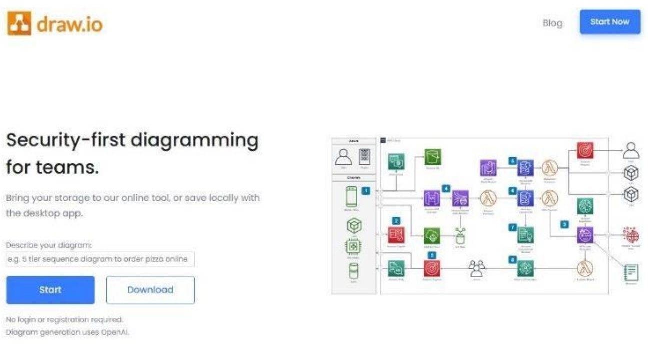

sketch_jun16a | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Select Board
sketch_jun16a.ino
Ln 10, Col 1
No board selected

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 2. Arduino IDE

3.1.2 Draw.io

Draw.io operates as a web-based application used for creating various diagram types such as flowcharts, system diagrams, electronic circuits, process mapping, hardware schematics, and interface designs. The application runs directly from browsers without installation requirements, while supporting integration with cloud storage platforms like Google Drive, OneDrive, and Dropbox. Draw.io advantages include an intuitive and user-friendly interface, free availability without feature restrictions, export support in multiple formats (PNG, JPG, SVG, PDF), plus an extensive collection of symbols and templates, including Arduino hardware and common sensor symbols. Draw.io proves valuable for documenting and designing systems for Arduino Uno R3-based projects, such as automated height, weight, and body temperature measurement prototypes using HC-SR04 sensors (ultrasonic distance sensor), MLX90614 (infrared temperature sensor), and Load Cell with HX711 module (weight sensor). For such applications, Draw.io facilitates creating electronic circuit schematics (wiring diagrams) and system block diagrams that visually explain workflow and inter-hardware relationships. Primary benefits of using Draw.io in Arduino projects include enhanced technical documentation, simplified debugging processes, improved team communication, and professional supporting material for final reports or theses explaining system architecture in a structured manner.



Security-first diagramming for teams.

Bring your storage to our online tool, or save locally with the desktop app.

Describe your diagram:
e.g. 5 tier sequence diagram to order pizza online

No login or registration required.
Diagram generation uses OpenAI.

Figure 3. Draw.io

3.1.3 Arduino Uno R3

Arduino Uno R3 represents one of the most popular microcontroller boards based on ATmega328P (Advanced Technology megaAVR 328 Pico), widely adopted in electronics and open-source programming communities. Arduino Uno R3 features 14 digital input/output (I/O) pins, where 6 can function as PWM (Pulse Width Modulation) outputs, plus 6 analog input pins enabling analog sensor readings [7]. Additionally, the board includes a USB (Universal Serial Bus) type-B port for programming and computer data communication, an external power jack for power supply, and an ICSP (In-Circuit Serial Programming) header for direct programming. Arduino Uno R3 possesses 32 KB flash memory capacity (0.5 KB used by bootloader), 2 KB SRAM (Static Random Access Memory), and 1 KB EEPROM (Electrically Erasable Programmable Read-Only Memory). The processor operates at speeds up to 16 MHz (Megahertz) using a crystal oscillator, sufficient for various mid-level microcontroller applications. Arduino Uno R3 advantages compared to previous versions or other types like Arduino Nano and Arduino Mega include system stability, standard form factor compatible with various shields (additional modules), plus substantial community support and abundant documentation. Arduino Uno R3 employs the ATmega16U2 chip (Advanced Technology megaAVR 16 USB-to-Serial) as a USB-to-Serial converter, replacing the FTDI chip (Future Technology Devices International) in older versions, delivering better data transfer speeds and communication stability. Various official and third-party libraries support programming such as LiquidCrystal for LCD displays, Servo for servo motors, Wire for I2C (Inter-Integrated Circuit) communication, SPI (Serial Peripheral Interface), SoftwareSerial for additional serial communication, Adafruit_MLX90614 for infrared temperature sensors, HX711 for load cell reader modules, and NewPing for ultrasonic sensors like HC-SR04 [8]. Arduino Uno R3 finds applications across multiple fields including home automation systems (smart home), robotics projects, automated measurement tools, environmental monitoring systems, and simple healthcare devices such as automated temperature and weight measurement tools. Primary advantages of Arduino Uno R3 include ease of learning, flexibility in electronic prototype development, plus affordable cost, making it highly suitable for educational purposes, research, and simple industrial applications.

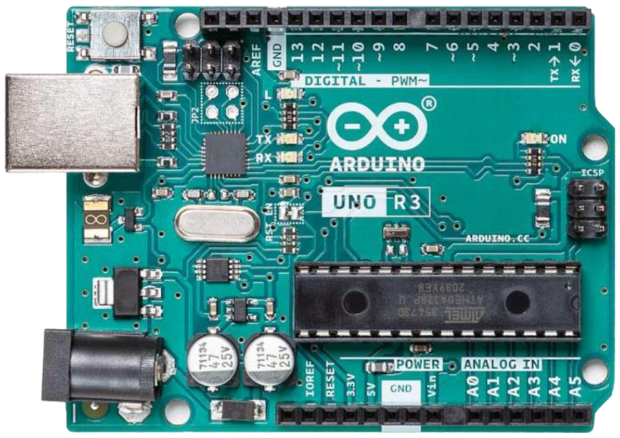


Figure 4. Arduino Uno R3

3.1.4 HC-SR04 Sensor

HC-SR04 operates as an ultrasonic sensor module designed to measure distance and detect object presence by utilizing sound wave principles. The sensor works by emitting ultrasonic waves from the Trig pin, then measuring the time required for those waves to bounce back after hitting an object and being received by the Echo pin [9]. Its primary advantage lies in delivering relatively accurate and fast non-contact distance measurements, making it ideal for applications where physical contact proves undesirable or impractical. Advantages include very low cost, easy integration with microcontrollers like Arduino, and adequate performance across various environmental conditions with a measurement range from 2 cm to 400 cm. Libraries commonly used for HC-SR04 include NewPing.h or often implemented with simple manual code using the pulseIn() function. Accuracy reaches approximately ±3mm, though variations may occur depending on environmental conditions and implementation. Applications are widely found in robotics projects for obstacle avoidance, liquid level measurement systems in containers, and automation applications requiring distance detection.

Figure 5. HC-SR04 Sensor

3.1.5 Load Cell Sensor & HX711 Module

Load Cell functions as a strain gauge-based load sensor that reads resistance changes when pressure is applied, while the HX711 module serves as a high-precision 24-bit analog-to-digital converter (ADC) specifically designed to read very small millivolt analog signals from load cells, amplify them, and convert them into digital data readable by microcontrollers [10]. The combination proves common in digital scales due to sensitivity and accuracy. Benefits include the ability to read loads down to grams with high accuracy (±0.1%–1%) provided calibration is precise. Advantages include high precision levels and low power consumption, while the drawback involves requiring meticulous calibration. The HX711.h library simplifies Load Cell & HX711 Module integration on Arduino. Applications prove vital in automated digital scale projects.

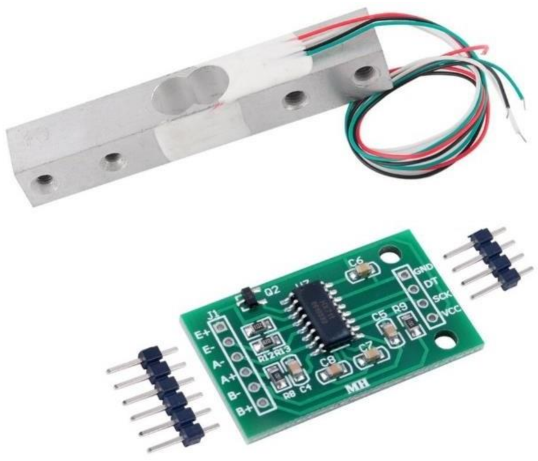


Figure 6. Load Cell Sensor and HX711 Module

3.1.6 MLX90614 Sensor

MLX90614 operates as a non-contact infrared thermometer sensor designed to measure object temperature without physical touch [11]. The sensor works by detecting infrared radiation emitted by objects, then performing internal calculations to produce ambient temperature and object temperature values. Its primary advantage lies in measuring temperature from a distance, proving highly useful in situations where physical contact is unsafe for hot or moving objects, impractical, or risks contamination, such as measuring human body temperature during pandemics. Advantages include high accuracy, typically ±0.2°C for human body temperature in medical ranges, and rapid response [12]. Libraries commonly used for MLX90614 include Adafruit_MLX90614.h or SparkFunMLX90614.h. Accuracy proves excellent, with measurement resolution of 0.02°C and factory calibration accuracy up to ±0.5°C for temperature ranges from -10 to 80°C, making it an appropriate choice for medical and industrial applications. Applications are common in non-contact medical thermometers, industrial temperature control systems, and electronic device temperature monitoring projects.[13]

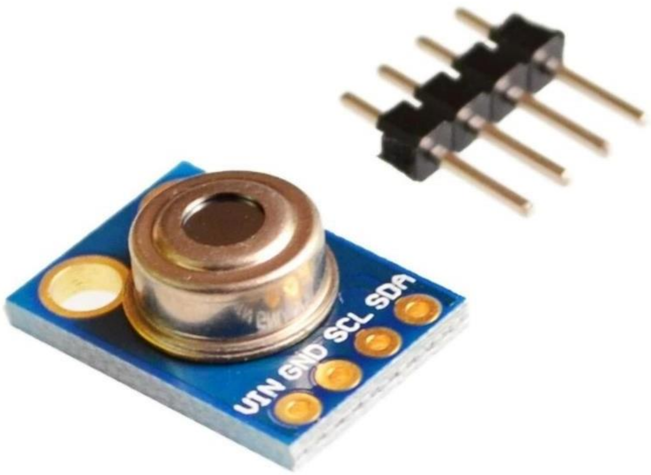


Figure 7. MLX90614 Sensor

3.1.7 LCD I2C

LCD I2C represents a Liquid Crystal Display (LCD) module equipped with an I2C (Inter-Integrated Circuit) controller chip, such as PCF8574, enabling communication between LCD and microcontroller using only two data pins (SDA and SCL) plus power and ground pins.[14] Its primary advantage involves drastically simplifying cable connections required between LCD and microcontroller, saving numerous input/output (I/O) pins that can be allocated for other functions in complex projects. Advantages include ease of use requiring minimal wiring, availability in 16x2 or 20x4 character sizes, and the ability to display clear real-time text information with controllable backlight. Libraries commonly used for LCD I2C include LiquidCrystal_I2C.h or NewliquidCrystal.h. Usage proves widespread in Arduino projects for displaying sensor data, making it a common and necessary part in many automation and measurement projects.

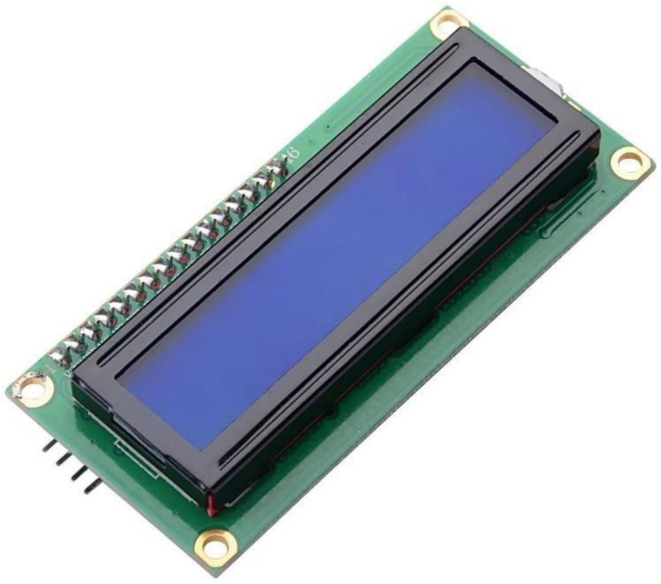


Figure 8. LCD I2C

3.1.8 Breadboard

Breadboard serves as a solderless (non-permanent) prototype circuit board used for assembling and testing electronic circuits temporarily before permanent design on PCB (Printed Circuit Board). Breadboards feature small holes connected horizontally and vertically internally using metal tracks, allowing electronic parts like resistors, capacitors, sensors, and microcontrollers such as Arduino Uno R3 to be mounted and connected with jumper wires without soldering [6]. Primary breadboard advantages include flexibility in building prototype circuits quickly, reusability multiple times, cost savings, plus safety for beginners as soldering is unnecessary. In Arduino Uno R3 prototype projects, breadboards prove beneficial for testing sensors like HC-SR04 (ultrasonic sensor), MLX90614 (infrared temperature sensor), and load modules (load cell + HX711) by facilitating assembly and wire arrangement between parts before designing permanent devices.

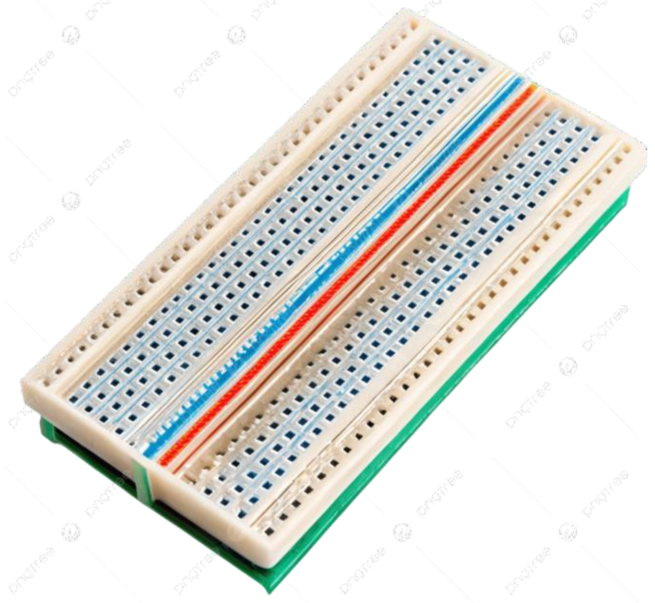


Figure 9. Breadboard

3.1.9 Jumper Wires

Jumper Wires are single cables with connectors at both ends, functioning to connect electronic parts to each other on breadboards, prototype boards, or directly to microcontroller pins without soldering [7]. Their primary advantage involves facilitating prototype creation and circuit testing quickly and easily, allowing designers and developers to try various hardware configurations without permanent modifications. Advantages include high flexibility, availability in various types such as male-to-male, male-to-female, female-to-female for different connection needs, and very affordable pricing. Usage forms the foundation in every electronics development and learning process with Arduino, serving as a basic tool for making connections between various sensors, modules, and microcontrollers.

Figure 10. Jumper Wires

4. Result and Discussion

4.1 Results

4.1.1 Circuit Diagram

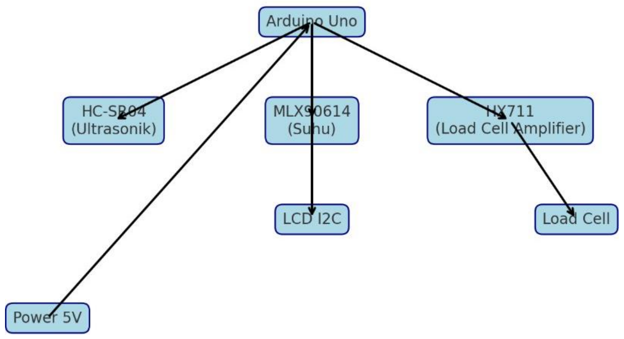


Figure 11. Health Measurement Device Circuit Diagram

4.1.2 Prototype Assembly for Height Measurement Tool (HC-SR04 Ultrasonic Sensor)

1) Prepare Main Hardware: Arduino Uno R3, Breadboard, HC-SR04 Ultrasonic Sensor, Male-to-Male Jumper Wires. Place the Arduino Uno and breadboard on your workspace. Identify the pins on the HC-SR04 Ultrasonic Sensor: VCC, Trig, Echo, and GND.
2) Connect the VCC pin on the HC-SR04 sensor to the positive power rail (typically marked with a red line or +) on the breadboard. Then, connect the GND pin on the HC-SR04 sensor to the negative ground rail (typically marked with a blue line or -) on the breadboard.
3) Now, connect the 5V pin on the Arduino Uno to the positive power rail (red) on the breadboard. Then, connect one of the GND pins on the Arduino Uno to the negative ground rail (blue) on the breadboard. This will supply power to the entire breadboard and connected sensors.
4) Connect the Trig pin on the HC-SR04 sensor to digital pin 9 on the Arduino Uno. Then, connect the Echo pin on the HC-SR04 sensor to digital pin 10 on the Arduino Uno. These are the pins that Arduino will use to send and receive ultrasonic signals. That completes all connections for the HC-SR04 ultrasonic sensor for height measurement.

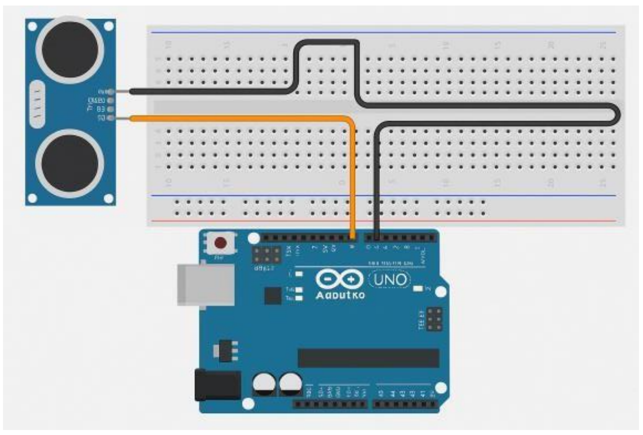


Figure 12. Height Measurement Prototype Assembly

4.1.3 Prototype Assembly for Body Temperature Detection Tool (MLX90614 Infrared Temperature Sensor)

1) Required Hardware: Arduino Uno R3 (already connected to breadboard), Breadboard (already connected to Arduino), MLX90614 Infrared Temperature Sensor, Male-to-Male Jumper Wires. Prepare the MLX90614 Sensor: Take your MLX90614 sensor. Identify its pins: VCC, GND, SDA, and SCL.
2) Connect the VCC pin on the MLX90614 sensor to the positive power rail (red) on the breadboard that is already connected to Arduino's 5V. Then, connect the GND pin on the MLX90614 sensor to the negative ground rail (blue) on the breadboard that is also already connected to Arduino's GND.
3) The MLX90614 sensor uses I2C communication protocol. Connect the SDA (Serial Data) pin on the MLX90614 sensor to pin A4 (SDA) on the Arduino Uno. Then, connect the SCL (Serial Clock) pin on the MLX90614 sensor to pin A5 (SCL) on the Arduino Uno [15].

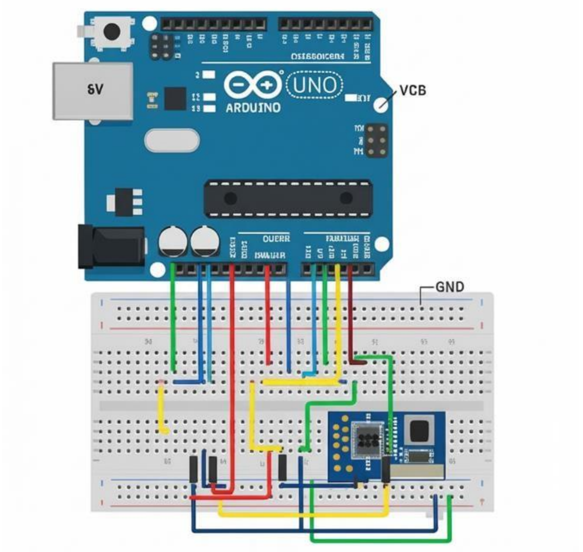


Figure 13. Body Temperature Detection Prototype Assembly

4.1.4 Prototype Assembly for Weight Measurement Tool (Load Cell and HX711)

1) Main Hardware: Arduino Uno R3 (already connected to breadboard), Breadboard (already connected to Arduino), Load Cell (Weight Sensor), HX711 Load Cell Amplifier Module, Male-to-Male Jumper Wires. Prepare Hardware: Take your Load Cell and HX711 Module. Identify the pins on the HX711: VCC, GND, DT, and SCK. Also observe the cables on the Load Cell (typically there are 4 cables with different colors, such as red, black, white, green/blue).
2) The cables on the load cell will be connected to the HX711 module. Cable colors may vary, but generally there are 4 cables:
   - Connect the Excitation+ (E+) cable from the load cell to the E+ pin on the HX711 module.
   - Connect the Excitation- (E-) cable from the load cell to the E- pin on the HX711 module.
   - Connect the Output+ (O+) cable from the load cell to the A+ pin on the HX711 module.
   - Connect the Output- (O-) cable from the load cell to the A- pin on the HX711 module.
     (If your load cell has 5 cables, typically there is one additional ground cable that may not be used or connected to E-.)
3) Now, we will connect the HX711 module to the Arduino Uno. Connect the VCC pin on the HX711 module to the positive power rail (red) on the breadboard (connected to Arduino's 5V). Connect the GND pin on the HX711 module to the negative ground rail (blue) on the breadboard (connected to Arduino's GND). Connect the DT (Data) pin on the HX711 module to digital pin 2 on the Arduino Uno. Connect the SCK (Clock) pin on the HX711 module to digital pin 3 on the Arduino Uno. Final. You have successfully connected all sensors to the Arduino Uno and breadboard [16].

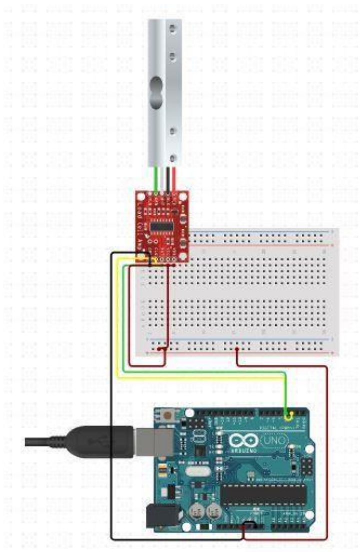


Figure 14. Weight Measurement Tool Prototype Assembly

Below is the overall device circuit after installation:

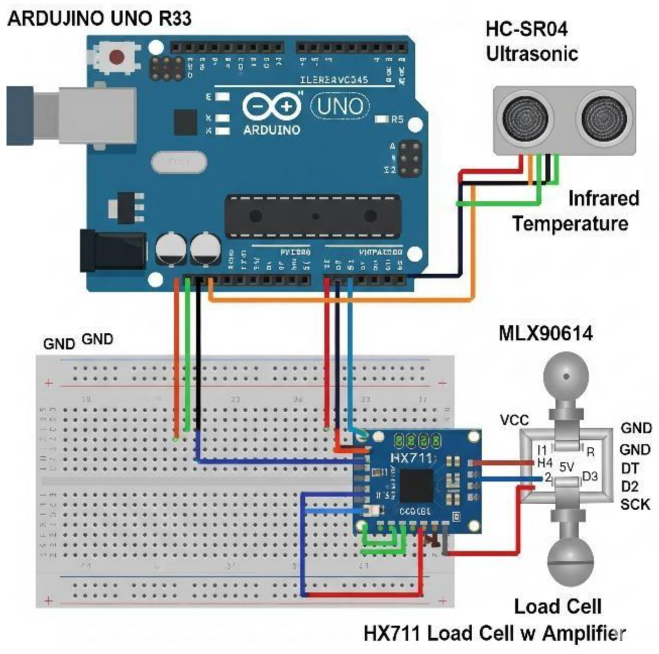
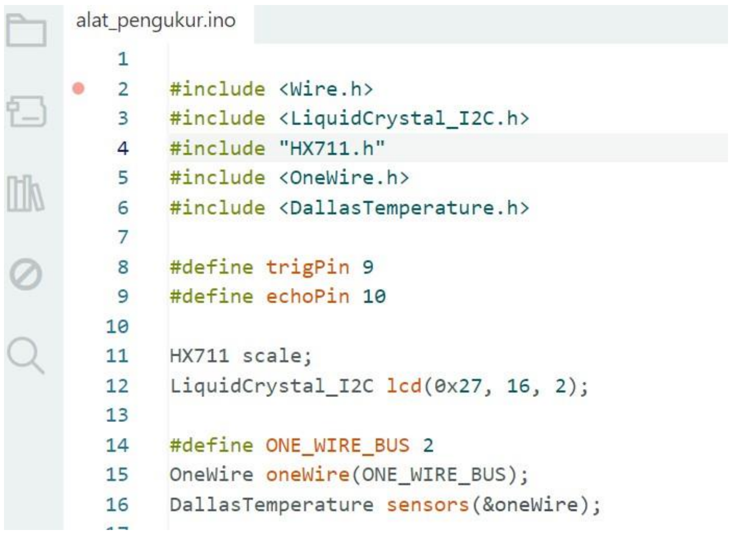


Figure 15. Complete Prototype Assembly for Weight, Height, and Body Temperature Measurement Tool

4.1.5 Programming the Measurement Tool

After all sensors are physically connected, the next step is programming your Arduino Uno. This involves:

1) Installing Libraries: You need to install appropriate libraries for each sensor in your Arduino IDE (for example, Adafruit MLX90614 for temperature sensor and HX711 for weight sensor). You can find them through the Sketch > Include Library > Manage Libraries... menu in Arduino IDE.
2) Writing Code (Sketch): You will write code (sketch) in Arduino IDE to read data from each sensor, perform necessary calculations (for example, converting distance to height, weight calibration), and display results (for example, to Serial Monitor or LCD screen if you add one).
   - For Height (HC-SR04): Code will measure sound reflection time and convert it to distance.
   - For Body Temperature (MLX90614): Code will read temperature value from sensor.
   - For Weight (Load Cell + HX711): Code will read value from HX711 and calibrate it to obtain accurate weight.
3) Calibration: Specifically for the weight sensor, calibration is a crucial step. You need to place a known load (for example, 1 kg) on top of the load cell and adjust the calibration factor in your code until the reading is accurate.



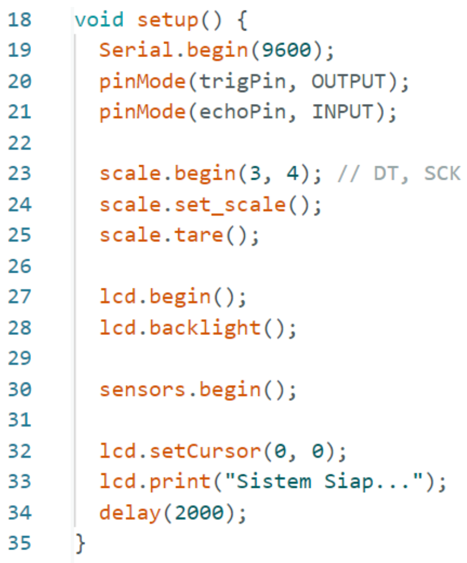
alat_pengukur.ino

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "HX711.h"
#include <OneWire.h>
#include <DallasTemperature.h>

#define trigPin 9
#define echoPin 10

HX711 scale;
LiquidCrystal_I2C lcd(0x27, 16, 2);

#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
```

Figure 16. Code Implementation Part 1

Explanation:

- #include <Wire.h> Used for I2C communication between devices, such as I2C LCD.
- #include <LiquidCrystal_I2C.h> Library for controlling 16x2 LCD screen that uses I2C communication.
- #include "HX711.h" Used for HX711 weight sensor module typically connected to load cell.
- #include <OneWire.h> Library for OneWire communication protocol, used by temperature sensors like MLX90614.
- #include <DallasTemperature.h> Additional library to facilitate temperature reading from MLX90614 sensor through OneWire.
- trigPin and echoPin are pins used to connect HC-SR04 ultrasonic sensor. trigPin (9): used to send ultrasonic signal. echoPin (10): receives reflection signal to calculate distance (used as height measurement).
- HX711 scale; Creates scale object to read weight value from HX711 sensor.
- LiquidCrystal_I2C lcd(0x27, 16, 2); Creates LCD object with address 0x27, size 16 columns × 2 rows. I2C address may differ depending on module.
- #define ONE_WIRE_BUS 2 Sets pin D2 on Arduino for communication with MLX90614 temperature sensor.
- OneWire oneWire(ONE_WIRE_BUS); Initializes OneWire protocol on that pin.
- DallasTemperature sensors(&oneWire); Creates sensors object to facilitate temperature reading from MLX90614 sensor connected through OneWire.

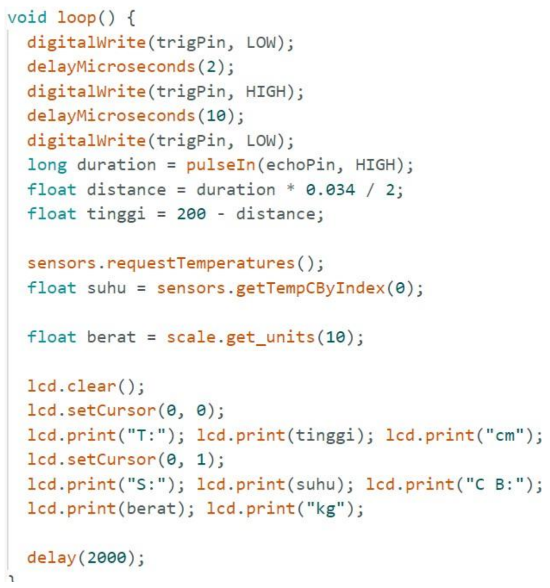


```
void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  scale.begin(3, 4); // DT, SCK
  scale.set_scale();
  scale.tare();

  lcd.begin();
  lcd.backlight();

  sensors.begin();

  lcd.setCursor(0, 0);
  lcd.print("Sistem Siap...");
  delay(2000);
}
```

Figure 17. Code Implementation Part 2

Explanation:

In the setup() function section, the system begins by initializing serial communication using Serial.begin(9600);, which allows Arduino to send data to the serial monitor at 9600 bps speed. Next, two pins connected to the HC-SR04 ultrasonic sensor, namely trigPin and echoPin, are set as output and input respectively using pinMode(). The weight sensor uses the HX711 module, and is initialized with scale.begin(3, 4); indicating that the DT pin is connected to Arduino pin 3 and the SCK pin to pin 4. Then, scale.set_scale(); is used to set the weight calibration factor (default value, needs adjustment), and scale.tare(); will set the initial value (zero) as reference—ignoring initial load weight, for example base weight. Next, the I2C LCD is activated with lcd.begin(); and the LCD backlight is turned on with lcd.backlight();, allowing the display to be clearly visible. The MLX90614 temperature sensor is activated through sensors.begin(); to start communication with OneWire protocol. Finally, the system displays the message "System Ready..." on the first line of the LCD with lcd.setCursor(0, 0); to set the initial writing position, followed by lcd.print("System Ready...");. A delay of 2 seconds (delay(2000);) is given so users have time to read the message before the system starts measuring.



```
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  float distance = duration * 0.034 / 2;
  float tinggi = 200 - distance;

  sensors.requestTemperatures();
  float suhu = sensors.getTempCByIndex(0);

  float berat = scale.get_units(10);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("T:"); lcd.print(tinggi); lcd.print("cm");
  lcd.setCursor(0, 1);
  lcd.print("S:"); lcd.print(suhu); lcd.print("C B:");
  lcd.print(berat); lcd.print("kg");

  delay(2000);
}
```

Figure 18. Code Implementation Part 3

Explanation:
In the loop() function, the Arduino system will continuously run a series of commands to read and display data from three different sensors. First, distance measurement is performed using the HC-SR04 ultrasonic sensor. The sensor works by sending ultrasonic waves through the trigPin and waiting for the reflection to return to the echoPin. The commands digitalWrite(trigPin, LOW); and HIGH along with microsecond delays (delayMicroseconds) are used to create trigger pulses. After that, the pulseIn(echoPin, HIGH); function calculates the time needed for the pulse to return after being reflected by an object (user's body). This time is then multiplied by the constant 0.034 and divided by two to get distance in cm. The height value (tinggi) is calculated with the formula 200 - distance, which assumes that the sensor height from the floor is 200 cm and is reduced by the distance to the user's head. Next, the system reads body temperature using the MLX90614 sensor. The sensors.requestTemperatures(); function instructs the sensor to perform temperature reading, then the result is obtained with getTempCByIndex(0);, which returns the temperature value in degrees Celsius from the first connected sensor. Then, weight reading is performed using a load cell sensor connected through the HX711 module. The scale.get_units(10); function will read weight 10 times and return the average value in kilogram units. After all data is obtained, the LCD will be cleared first (lcd.clear();) so new data doesn't overlap with old data. Writing starts from the first line (setCursor(0, 0)) to display height (T:), then moves to the second line (setCursor(0, 1)) to display body temperature (S:) and weight (C B: is an abbreviation for "Celsius Body"). Each data is followed by its unit such as cm, C B:, and kg. Finally, delay(2000); provides a 2-second pause before the measurement process is repeated, so users have enough time to read the results on the LCD.

Below is the system workflow diagram for the human height, body temperature, and weight detection tool:

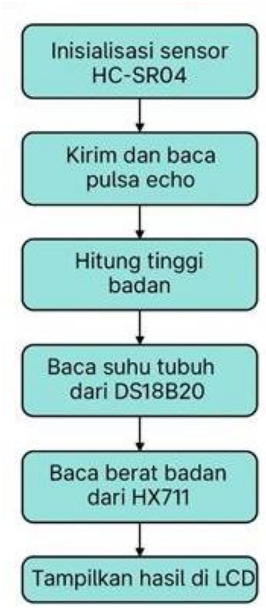

Inisialisasi sensor HC-SR04
Kirim dan baca pulsa echo
Hitung tinggi badan
Baca suhu tubuh dari DS18B20
Baca berat badan dari HX711
Tampilkan hasil di LCD

Figure 19. System Workflow Diagram

Explanation:
The height, body temperature, and weight measurement device system is designed using Arduino microcontroller that integrates several sensors to perform automatic measurements. When the device is turned on, the system will run an initialization process to ensure all parts such as HC-SR04 ultrasonic sensor, MLX90614 temperature sensor, HX711 weight sensor connected to load cell, and I2C LCD module function properly.[17] After initialization is complete, the system starts the measurement process. The ultrasonic sensor will measure the distance between the user's head and sensor to calculate height based on the difference from a fixed length (for example 200 cm) with the detected distance. Next, the MLX90614 temperature sensor will read the user's body temperature with high accuracy and send that data to the microcontroller. At the same time, the HX711 weight sensor will read the value from the load cell and convert it to body weight in kilogram units. All measurement results are then displayed in real-time on the I2C LCD screen, with a display that includes height (in cm), body temperature (in °C), and weight (in kg). The system provides approximately a 2-second pause before repeating the reading process so the displayed data is stable and easy to read by users.[18] With the automated system, the measurement process becomes faster, more accurate, and efficient without needing separate conventional tools.

4.2 Discussion

Testing of the height sensor with measuring tape was conducted to measure the accuracy level of the distance sensor which will be compared with manual tape measurements, weight measurements with sensors will be compared with scales, and temperature measurements with sensors will be compared with thermometers.

4.2.1 HC-SR04 Sensor Testing Results

Initial testing was carried out by comparing measurement data from the HC-SR04 sensor with manual measurements. Samples were taken from 10 different individuals, and each individual was measured 2 times. The HC-SR04 sensor reads the distance from the sensor to the top point of the individual being measured. Distance H2 is the sensor placement height (200 cm) minus the sensor distance to the individual. Testing results are shown in Table 1 and Table 2.

Table 1. First Testing Results on Height Sensor

No.	H1 Height (cm)	Distance Read by Height Sensor (cm)	H2 Sensor Height Measurement (cm)	H1 - H2 Error	% Error
1	177.8	22.0	178.0	-0.2	0.11
2	183.4	16.6	183.4	0.0	0.00
3	192.2	8.0	192.0	0.2	0.10
4	168.8	31.2	168.8	0.0	0.00
5	179.9	20.0	180.0	-0.1	0.06
6	193.8	7.2	192.8	1.0	0.52
7	155.4	44.6	155.4	0.0	0.00

8	190.9	10.0	190.0	0.9	0.47
9	155.7	44.3	155.7	0.0	0.00
10	160.7	39.3	160.7	0.0	0.00

Average % error: 0.13.

Table 2. Second Testing Results on Height Sensor

No.	H1 Height (cm)	Distance Read by Height Sensor (cm)	H2 Sensor Height Measurement (cm)	H1 - H2 Error	% Error
1	177.8	22.2	177.8	0.0	0.00
2	183.4	16.0	184.0	-0.6	0.33
3	192.2	8.6	191.4	0.8	0.42
4	168.8	31.2	168.8	0.0	0.00
5	179.9	20.2	179.8	0.1	0.06
6	193.8	7.0	193.0	0.8	0.41
7	155.4	44.6	155.4	0.0	0.00
8	190.9	9.1	190.9	0.0	0.00
9	155.7	44.3	155.7	0.0	0.00
10	160.7	39.3	160.7	0.0	0.00

Average % error: 0.12.

4.2.2 Load Cell Sensor Testing Results

Load cell sensor testing was conducted to assess the sensor's accuracy level, which will later be compared with results from manual digital scales. The testing process involves comparison between weight measured by the load cell sensor and weight detected by digital scales. Samples were taken from 10 different individuals, where each person was measured twice. Testing results are shown in Table 3 and Table 4.

Table 3. First Testing Results on Load Cell Sensor

No.	H1 Weight (kg)	H2 Sensor Weight Measurement (kg)	H1-H2 Error	% Error
1	72.2	72.0	0.2	0.28
2	63.2	63.2	0.0	0.00
3	53.7	53.8	-0.1	0.19
4	89.6	89.2	0.4	0.45
5	52.0	52.1	-0.1	0.19
6	72.4	72.4	0.0	0.00
7	63.3	63.3	0.0	0.00
8	71.1	71.0	0.1	0.14
9	73.8	73.9	-0.1	0.14
10	57.5	57.5	0.0	0.00

Average % error: 0.14.

Table 4. Second Testing Results on Load Cell Sensor

No.	H1 Weight (kg)	H2 Sensor Weight Measurement (kg)	H1 - H2 Error	% Error
1	72.2	72.1	0.1	0.14
2	63.2	63.4	-0.2	0.32
3	53.7	53.7	0.0	0.00
4	89.6	89.4	0.2	0.22
5	52.0	52.1	-0.1	0.19
6	72.4	72.3	0.1	0.14
7	63.3	63.3	0.0	0.00
8	71.1	71.1	0.0	0.00
9	73.8	73.9	-0.1	0.14
10	57.5	57.5	0.0	0.00

Average % error: 0.11.

4.2.3 Load Cell Sensor Voltage Testing Results

Load cell sensor voltage testing was conducted by measuring voltage on the weight sensor in on and off states. Load cell sensor voltage comes from Arduino's 5V Pin. Testing results are shown in Table 5.

Table 5. Load Cell Sensor Voltage Testing Results

No.	Load Cell Sensor Condition	Voltage (V)	
		Input Pin	Output Pin
1	ON State	4.85	4.31
2	OFF State	0	0

4.2.4 Ultrasonic Sensor Voltage Testing Results

Ultrasonic sensor voltage testing was conducted by measuring voltage on the height sensor in off and on states. Ultrasonic sensor voltage comes from Arduino's 5V Pin. Testing results are shown in Table 6.

Table 6. Ultrasonic Sensor Voltage Testing Results

No.	Ultrasonic Sensor Condition	Voltage (V)	
		Input Pin	Output Pin
1	ON State	4.93	1.28
2	OFF State	0	0

4.2.5 Maximum Reading Results of Ultrasonic Sensor

Testing was conducted to measure the maximum capability of the ultrasonic sensor that can still be read. Testing results are shown in Table 7.

Table 7. Maximum Reading Testing Results of Ultrasonic Sensor

Sensor Maximum Reading
202 cm

4.2.6 MLX90614 Sensor Testing Results

MLX90614 sensor testing was conducted to assess the sensor's accuracy level, which will later be compared with results from manual thermometers. The testing process involves comparison between temperature measured by the MLX90614 sensor and temperature detected by thermometers. Samples were taken from 10 different individuals, where each person was measured twice. Testing results are shown in Table 8 and Table 9.

Table 8. First Testing Results on MLX90614 Sensor

No.	H1 Body Temperature (°C)	H2 Sensor Temperature Measurement (°C)	H1 - H2 Error	% Error
1	36.2	36.1	0.1	0.28
2	37.2	37.3	-0.1	0.27
3	37.2	37.2	0.0	0.00
4	36.9	36.9	0.0	0.00
5	36.9	37.0	-0.1	0.27
6	36.5	36.4	0.1	0.27
7	37.3	37.3	0.0	0.00
8	37.4	37.4	0.0	0.00
9	37.3	37.3	0.0	0.00
10	36.3	36.2	0.1	0.28

Average % error: 0.14.

Table 9. Second Testing Results on MLX90614 Sensor

No.	H1 Body Temperature (°C)	H2 Sensor Temperature Measurement (°C)	H1 - H2 Error	% Error
1	36.2	36.2	0.0	0.00
2	37.2	37.2	0.0	0.00
3	37.2	37.3	-0.1	0.27
4	36.9	36.9	0.0	0.00
5	36.9	36.8	0.1	0.27
6	36.5	36.5	0.0	0.00
7	37.3	37.3	0.0	0.00
8	37.4	37.4	0.0	0.00
9	37.3	37.3	0.0	0.00
10	36.3	36.5	-0.2	0.55

Average % error: 0.11.

4.2.7 Overall System Testing Results

Overall testing aims to evaluate existing software to ensure whether calculations in determining categories and requirements for weight, body temperature, and height are consistent with manual calculations. The testing process was conducted by measuring once for 10 different users. Testing results can be seen in Table 4.10.

Table 10. Overall System Testing Results

No.	Manual			Sensor		
	Height (cm)	Weight (kg)	Body Temp (°C)	Height (cm)	Weight (kg)	Body Temp (°C)
1	177.9	72.2	36.3	177.8	72.2	36.2
2	183.3	63.2	37.2	183.4	63.3	37.2
3	192.2	53.7	37.2	191.9	53.7	37.3
4	168.8	89.6	36.9	168.8	89.6	36.8
5	179.9	52.0	36.9	179.8	52.1	36.8
6	193.8	72.4	36.5	193.8	72.4	36.5
7	155.4	63.6	37.3	155.5	63.7	37.4
8	190.9	71.1	37.4	190.9	71.1	37.3
9	155.7	73.8	37.3	155.7	73.8	37.2
10	160.7	57.5	36.3	160.8	57.6	36.3

From the test results in the previous section, some important things can be said about how well and accurately the automated health measurement device works. This device uses IoT-based sensors with an Arduino Uno microcontroller. The HC-SR04 ultrasonic sensor showed great performance in measuring height, with an average error rate of 0.13% for the first test and 0.12% for the second test. These numbers mean that the sensor is very consistent and reliable when used for height measurements without contact. The small margin of error shows that this sensor can replace manual measuring tapes in healthcare facilities, especially during fast screening processes. Also, the Load Cell sensor with the HX711 module proved very accurate in weight measurement, achieving average error rates of 0.14% and 0.11% in the first and second tests respectively. The ability of this sensor to maintain such low error margins across different weight ranges (from 52.0 kg to 89.6 kg) demonstrates its suitability for clinical applications where precise weight measurements are essential for medical assessments and treatment planning. Another example is that MLX90614 infrared temperature sensor performed very well too; it had average error rates of 0.14% and 0.11% in consecutive tests! The non-contact measurement capability of this sensor proves particularly valuable in modern healthcare settings especially when hygiene requirements and infection control protocols are considered! The accuracy of this sensor across normal body temperature ranges (36.2°C to 37.4°C) also confirms its reliability for fever screening as well as general health monitoring! Results from voltage testing confirmed stable power supply to all sensors: Load Cell sensor got a 4.85V input with a 4.31V output while ultrasonic received a 4.93V input with a 1.28V output during operation; these voltage levels fall within acceptable operating ranges hence ensuring consistent performance from these sensors over prolonged usage periods. Testing on the whole system showed that all three sensors worked together well because readings were almost exactly like manual ones for all measurements taken .The Arduino Uno microcontroller managed data collection from many sensors at once very well, showing results on an I2C LCD screen right away. It was able to finish full measurement cycles in less than two seconds which makes it great for busy healthcare places where fast patient processing is needed!

The prototype has effectively solved some problems that traditional manual measurement methods face, such as being time-consuming, having possible human error, and requiring multiple separate devices. By bringing height, weight, and temperature measurements into one automated system, this device makes the patient intake process easier and cuts down waiting times very much. The IoT-based setup also gives chances for future improvements, like data logging, cloud connection, and working with electronic health record systems. But there are some things to consider for future enhancements. Environmental factors such as ambient temperature, humidity, and air currents may affect sensor readings, particularly for the ultrasonic and infrared sensors. Implementing environmental compensation algorithms could further enhance measurement accuracy. Also, the current prototype needs users to get in the right position for accurate readings; therefore future versions can have automatic guidance or different sensor arrays to help the user experience better. This research proves that it is possible and practical to use Arduino-based IoT sensors for automatic health measurements in a community health center setting. The very high accuracy levels gotten from all three measurement parameters confirm that this technical approach is valid and back up a wider application in healthcare facilities that want to modernize their patient intake process while keeping measurement reliability and cost-effectiveness.

5. Conclusion and Recommendations

With this research, an IoT-based automated prototype measuring height, weight, and body temperature using an Arduino Uno microcontroller has been created. This prototype can help improve the efficiency and accuracy of health measurements at the Cengkareng Timur Community Health Center. The prototype uses several sensors: an HC-SR04 ultrasonic sensor, a Load Cell with HX711 module, and an MLX90614 temperature sensor that could perform automatic measurements with good accuracy levels. The test results showed that this device could reduce recording errors which are often found in manual tools and accelerate the measurement process; thus optimizing basic healthcare services and reducing patient waiting time.

The next development of the prototype should be directed at adding more comprehensive connectivity features such as integration with mobile applications or health data management systems so that access to these data as well as their analysis will become easier for medical personnel. Future research may consider additional sensors or new technologies that could improve device functionality and trials in various other community health center locations to get more diverse data on effectiveness and acceptance by the public of this device.

References

[1] Ginting, Y. N., Abdillah, J., & Sarimole, F. M. (2023). Rancang bangun alat sortir jeruk medan berdasarkan luas lingkar berbasis mikrokontroler ATMega2560. *Sistem Komputer dan Teknologi Intelegensi Artifisial (SIKOMTIA)*, *1*(3), 192–198. https://doi.org/10.59039/sikomtia.v1i3.18

[2] Charaan, R. M. D., Shobana, J., Krishnamoorthy, P., Princy, B. A., Abinaya, R. J., & Murugesan, K. (2023). Enhancement of IoT based flood detection and prevention using Arduino UNO with WiFi module. *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2290–2294. https://doi.org/10.1109/ICACCS57279.2023.10112673

[3] Romadan, D. P., Arinal, V., Sarimole, F. M., & Tundo, T. (2024). Prototipe sistem monitoring kelembapan tanah pada tanaman cabai berbasis Internet of Things dengan metode fuzzy logic menggunakan NodeMCU Esp8266, Blynk dan Thingspeak. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *5*(1), 130–140. https://doi.org/10.57152/malcom.v5i1.1600

[4] Nalla, S., Rakshitha, K., Kumar, K. S., & Samad, M. (2025). Detection of moving objects with ultrasonic radar. *2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)*, 1251–1255. https://doi.org/10.1109/ISACC65211.2025.10969357

[5] Ludya, M., Herlambang, Y., & Yunidar, D. (2023). Produk alat ukur tinggi dan berat badan pendeteksi stunting dengan fitur hiburan untuk anak usia 2-5 tahun. *Productum: Jurnal Desain Produk (Pengetahuan dan Perancangan Produk)*, *6*(1), 51–62. https://doi.org/10.24821/productum.v6i1.7685

[6] Kadir, A. (2018). *Arduino & sensor*. Penerbit Andi.

[7] Wicaksono, M. F. (2019). *Aplikasi Arduino dan sensor*. Informatika Bandung.

[8] Sumampouw, G., Saputra, R. D., Sandy, M., Hidayat, A. M., & Utomo, R. M. (2022). Prototype sistem pengaturan kecepatan kipas DC otomatis menggunakan sensor PIR, sensor ultrasonik, sensor DHT11 berbasis mikrokontroler Arduino Uno dan Node MCU.

[9] Rizal, M., Sondakh, D. E., & Ashari, I. F. (2023). *Konsep dan implementasi Internet of Things*. Yayasan Kita Menulis.

[10] Nurlette, D., & Wijaya, T. K. (2018). Perancangan alat pengukur tinggi dan berat badan ideal berbasis Arduino. *Sigma Teknika*, *1*(2), 172–184.

[11] Inayah, I. (2021). Analisis akurasi sistem sensor IR MLX90614 dan sensor ultrasonik berbasis Arduino terhadap termometer standar. *Jurnal Fisika Unand*, *10*(4), 428–434. https://doi.org/10.25077/jfu.10.4.428-434.2021

[12] Kadir, A. (2017a). *Pemrograman Arduino dan Processing*. PT Elex Media Komputindo.

[13] Asri, S., et al. (2024). Arduino-based automatic weighing machine design. *IOP Conference Series: Earth and Environmental Science*. https://doi.org/10.1088/1755-1315/1381/1/012028

[14] Kadir, A. (2017b). *Pemograman Arduino & Android menggunakan App Inventor*. PT Elex Media Komputindo.

[15] Sathish Kumar, N., Vuayalakshmi, B., Prarthana, R. J., & Shankar, A. (2016). IOT based smart garbage alert system using Arduino UNO. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 1028–1034. https://doi.org/10.1109/TENCON.2016.7848162

[16] Gupta, D., Parikh, A., & Swarnalatha, R. (2020). Integrated healthcare monitoring device for obese adults using internet of things (IoT). *International Journal of Electrical and Computer Engineering*, *10*(2), 1239–1247. https://doi.org/10.11591/ijece.v10i1.pp1239-1247

[17] Blum, J. (2020). *Exploring Arduino tools and techniques for engineering wizardry*. John Wiley & Sons, Inc.

[18] Putro, C., Anindyasari, F., Aprilia, S., & Prasojo, I. (2024). Rancang bangun automatic contactless thermometer berbasis Arduino Uno. *Jurnal Mahasiswa Ilmu Kesehatan*, *2*(2), 97–107. https://doi.org/10.59841/jumkes.v2i2.2669