

# OPTICAL DISTANCE MEASUREMENT FOR EMBEDDED PROGRAMMING AND VISUALIZATION IN EDUCATIONAL CONTEXTS

Maik Rosenberger<sup>a\*</sup>, Mirco Andy Eilhauer<sup>a</sup>, Andrei Golomoz<sup>a</sup>, Raik Illmann<sup>a</sup>, Martin Richter<sup>a</sup>, Richard Fütterer<sup>a</sup>, Gunther Notni<sup>a</sup>

<sup>a</sup> Department of Quality Assurance and Industrial Image Processing, Ilmenau, Germany

\* maik.rosenberger@tu-ilmenau.de

**Abstract** – Nowadays, a multitude of sensors collect and process enormous amounts of measurement data. Depending on the complexity, these tasks are usually performed by microcontrollers, which read out the sensor data, pre-process it, and then transmit it to a PC, for example. Communication between the microcontroller and PC is often complex, but of central importance. Equally important is the subsequent processing and interpretation of the acquired measurement data. Using the example of an optical distance measurement, a hardware module was developed that can establish an interface to a measuring computer via a microcontroller. In this system, care was taken to ensure that the embedded programming focused on pure C code, and that visualization is script-based in higher-level software. In this way, students learn how to use an embedded platform and an optical sensor, as well as how to further process data measurement on a control computer

**Keywords:** optical ranging, control charting, analog interface, data transfer

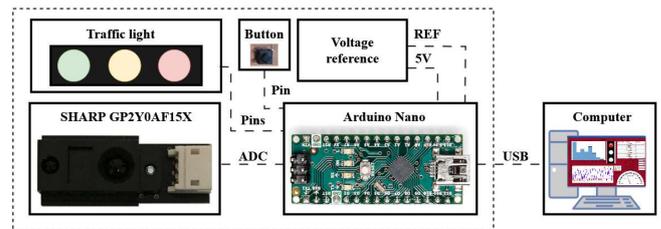
## 1. INTRODUCTION

This research presents an innovative and practical approach to teaching basic programming skills in the C programming language using an optical sensor. The focus is on the development of a learning system specifically designed for use in higher education to provide students with a structured introduction to hardware-based programming and sensor read out without prior knowledge. By combining an Arduino microcontroller with an optical distance sensor and evaluating the measured data using MATLAB, a complete measurement chain from data acquisition to visualization is realized. The methodological focus is on the integration of sensors, data processing, and real-time visualization, thereby creating a deeper understanding of technical interrelationships.

## 2. HARDWARE

The system in figure 1, which is located on a specially developed circuit board, consists of a modular hardware platform based on an Arduino Nano board [1] connected to a SHARP GP2Y0AF15X distance sensor. This measures the distance to an object in a range of 15 to 150 mm and outputs a corresponding voltage in the range of 2.3 V to 2.9 V, which is digitized using the Arduino's integrated 10-bit ADC [2]. A special feature of the system is the external voltage reference, which allows maximum utilization of the ADC's dynamic range, thereby increasing the accuracy of the measurement. Serial data transmission takes place via a USB connection

(USART), with the PC acting not only as a display unit but also as a control element.



**Figure 1** This image shows the hardware setup, which was illustrated with a system diagram showing how the Arduino Nano is connected to the distance sensor, the traffic light LED display, the push button, and to a computer via USB for data acquisition and analysis [3][4][5].

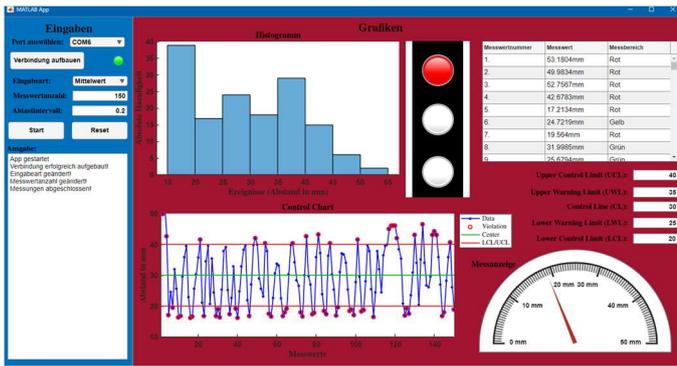
A central element of the methodological implementation is the use of so-called control charts, which graphically display and monitor the status of the measurement process. These charts use historical comparison data to show whether a measured value lies within the predefined control limits [6], [7], [8]. Based on this information, the PC sends corresponding signals to the Arduino, which then provides visual feedback via a colored LED traffic light (red, yellow, green). Control is either via user input on the PC or via a button on the setup itself. This simple yet effective feedback generates direct feedback on the quality of the measurement process.

## 3. METHODS

The microcontroller is programmed using the C compiler "Code Vision AVR," which has an integrated code generator (CodeWizardAVR). This allows standard functions such as ADC initialization and serial communication to be configured efficiently. The program execution logic is implemented in an endless loop, with the microcontroller continuously waiting for input from the PC. A special feature is the implementation of a nested switch-case structure, which allows for the simple implementation of various commands and feedback [9]. For example, specific measurements or LED states can be activated by sending character strings. The omission of interrupts in the current implementation simplifies control but imposes limitations on response time, which will be discussed in the outlook.

On the software side, MATLAB is used to further process the data received from the microcontroller. The MATLAB code enables the creation of control charts, the graphical representation of the measured value distribution in histograms, and real-time monitoring through an interactive app layout (see figure 2) implemented in MATLAB's App Designer. The key program commands for serial communication, data cleaning, and graphical output are used [10]. Particular attention is paid to data integrity, which is

why the buffers are regularly emptied, and the presence of new data is checked before reading.



**Figure 2** This image shows the implementation of the interactive layout with histogram, control charts, a gauge for the current measurement and buttons.

Another key issue is the conversion of the ADC raw data into real distance values [11]. For this purpose, a calibration was performed using a series of measured value pairs (ADC value and measured distance). A functional relationship between the digital measured value and the real distances was then established using a polynomial fitting method (polyfit) [12], [13]. A sixth-degree polynomial proved to be optimal, as it offers both high accuracy and acceptable computing time. The quality of the fit was evaluated using the coefficient of determination  $R^2$  [12], [14], with a value close to 1 indicating very good agreement. This modeling forms the basis for the correct representation of the measurements in MATLAB app layout.

The app layout itself allows for the presentation of all measured values form and the visualization of the current measured value on a speedometer. In addition, the parameters for the number of measurements, measurement intervals, and control limits can be adjusted via input fields [15], [16]. The status of the measurement series is displayed in an additional information field. These functions enable user-friendly interaction with the system and provide a clear representation of the entire measurement process.

#### 4. RESULTS AND OUTLOOK

The components developed within this project – both hardware and software – were successfully used in a student internship. Implementation took place under the close supervision and active participation of the students, which promoted not only technical understanding but also teamwork and problem-solving skills. The system is used in the course Computer Aided Quality (CAQ) and Sensor Signal Processing at the Technische Universität Ilmenau. The goal is to make abstract teaching content tangible through practical examples and interactive experiments.

In summary, this work demonstrates a functional and didactically valuable system for teaching technical content, which is versatile in its application thanks to its modular design and open programming structure. The combination of simple hardware, standardized programming languages, and powerful evaluation software creates an effective learning

environment. Looking ahead, the control mechanisms will be further improved using interruptions to ensure real-time capability, and the system architecture will be extended to include additional sensors and control components.

The results presented in this paper underscore the potential of an integrated hardware-software learning environment for transforming practical C programming instruction and pave the way for future advances in technical education and quality assurance.

#### REFERENCES

- [1] *Arduino.cc*, <https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf>. Zugegriffen 22. Dezember 2024.
- [2] *Mouser.de*, [https://www.mouser.de/datasheet/2/365/SHARP\\_GP2Y0AF15X-2487211.pdf](https://www.mouser.de/datasheet/2/365/SHARP_GP2Y0AF15X-2487211.pdf). Zugegriffen 22. Dezember 2024.
- [3] Flowchart maker & online diagram software“. *Diagrams.net*, <https://app.diagrams.net/>. Zugegriffen 22. Dezember 2024.
- [4] *Conrad.com*, <https://asset.conrad.com/media10/isa/160267/c1-/de/838243793PI01/image.jpg?x=400&y=400&format=jpg&ex=400&ey=400&align=center>. Zugegriffen 22. Dezember 2024.
- [5] *Digikey.com*, [https://mm.digikey.com/Volume0/opasdata/d220001/medias/images/321/MFG\\_GP2Y0AF15X.jpg](https://mm.digikey.com/Volume0/opasdata/d220001/medias/images/321/MFG_GP2Y0AF15X.jpg). Zugegriffen 22. Dezember 2024.
- [6] MATLAB App designer. (o. D.). MATLAB & Simulink. <https://de.mathworks.com/help/matlab/app-designer.html>
- [7] Ring, Christina, u. a. „Das Bestimmtheitsmaß  $R^2$  bei linearen Regressionsmodellen mit und ohne Intercept - die Tücken der Statistikprogramme“. *WiSt. Wirtschaftswissenschaftliches Studium*, Bd. 35, Nr. 11, 2006, S. 607–612, <https://doi.org/10.15358/0340-1650-2006-11-607>.
- [8] *Asq.org*, <https://asq.org/quality-resources/control-chart?srsltid=AfmBOor-6kOlzmVkPyilewplfm45QDqiGejqwwc6R2vMoEo386dvryDK>. Zugegriffen 22. Dezember 2024.
- [9] InfoTech, H. P. „HP InfoTech - CodeVisionAVR C Compiler“. *Hpinfotech.ro*, <http://www.hpinfotech.ro/>. Zugegriffen 22. Dezember 2024
- [10] *Mathworks.com*, <https://de.mathworks.com/help/stats/control-charts.html>. Zugegriffen 22. Dezember 2024.
- [11] General Description. (o. J.). High-precision buffered voltage reference. Abgerufen 8. Januar 2025, von Microchip.com
- [12] Papula, Lothar. *Mathematical formulary*. Vol. 4. Springer Fachmedien Wiesbaden, 2017. ,pp. 76,446-448
- [13] Bharucha-Reid, Albert T., and Masilamani Sambandham. *Random polynomials: Probability and mathematical statistics: a series of monographs and textbooks*. Academic Press, 2014.,pp. 1-2
- [14] ŠKREB, S. The mean value. *Hrvatski geografski glasnik*, 1931, Vol. 3, No. 1, pp. 233-241
- [15] Entwickeln von Apps mithilfe von App Designer - MATLAB & Simulink - MathWorks Deutschland. (o. D.). <https://de.mathworks.com/products/matlab/app-designer.html>
- [16] *A Beginner's Guide to Control Charts - the W. Edwards Deming Institute*. <https://deming.org/a-beginners-guide-to-control-charts/>. Zugegriffen 22. Dezember 2024