ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research

Vol. 10, Issue, 08, pp. 39358-39361, August, 2020

https://doi.org/10.37118/ijdr.19788.08.2020

RESEARCH ARTICLE OPEN ACCESS

DEVELOPMENT OF COMMUNICATION BETWEEN ESP32 AND ARDUINO PLATFORMS, USING A LOGIC LEVEL AND VOLTAGE CORRECTION SYSTEM

Humberto da C. Frizzera*[1], Fabiano M. Tavares[1], George F. dos Santos[1], Gentil V. Barbosa[1] and Humberto X. de Araújo[1]

[1]PPGMCS - Graduate Program in Computational Modeling of Systems - Federal University of Tocantins (UFT) Quadra 109 Norte, Avenida NS-15, ALCNO-14 - North Master Plan, CEP, TO, 77001-090 - Palmas - TO – Brazil

ARTICLE INFO

Article History:

Received 19th May 2020

Received in revised form

17th June 2020

Accepted 21st July 2020

Published online 30th August 2020

Key Words:

ESP32, Arduino, Microcontrollers, Communication, Level Shifter, Protocol I2C.

****Corresponding author:***

Humberto da C. Frizzera,

ABSTRACT

This article is the development of communication between the ESP32 and the Arduino, using the system of correction of logical level and tension. One is about simple an electronic solution so that it has communication in a duplex way between the cited equipment, without causing damages between the plates and making possible that the Arduino is used natively together with its sensors and without losing the rich possibilities of also using the ESP32 with its interfaces Wi-Fi and Bluetooth. For the development, beyond the ESP32- Wroom and the Arduino Nano, they had also been used level shifter, for the intermediation of the communication between the ESP32 and the Arduino, and protocol of communication I2C.

Citation: ***Humberto da C. Frizzera, Fabiano M. Tavares, George F. dos Santos, Gentil V. Barbosa and Humberto X. de Araújo.*** "Development of communication between esp32 and arduino platforms, using a logic level and voltage correction system*", International Journal of Development Research*, 10, (08), 39358-39361.

INTRODUCTION

The Arduino prototyping platform was designed in Italy in 2005 to enable an integrated architecture that brings many advantages when compared to other existing platforms on the market, the largest of which is the ease of use and according to (McRoberts, 2018), it allows people who are not from technical areas can learn the basics to create their electronics projects in a relatively short period.The IoT prototype developer community has frequently used so-called "Single Board Computers" or Microcontrollers, which is the integrated hardware and software solution to collect sensor data in different areas of knowledge, such as sensors electric current, water flow, thermal, among others (De Oliveira et al. 2017) (Noraini Azmi et al. 2018).Due to a huge community spread around the world that, professionally or not, use Arduino and share projects and prototypes in addition to source codes for different applications for this platform, Arduino has gained a lot of visibility. This popularity has generated a very intense market niche for very low-cost accessories for it, from LED lamps to highly complex sensors. Sometime later, other platforms emerged with the same purpose as Arduino, such as Espressif ESP8266, a platform of Chinese origin and superior to Arduino with 10 times the processing speed and access to Wi-Fi already integrated. The new solutions as an example of ESP8266 have enabled the advent of the long-awaited connectivity of things called IoT - Internet Of Things (Yu *et. al*, 2020) (Barbon *et. al*, 2016). According to (Oliveira, 2017), these new platforms allow direct access to the Internet using low energy consumption rates and low coding complexity. According to (Wortmann, 2015), it is almost impossible not to come across the term IoT. The world today has seen a tremendous increase in interest on the Internet of Things. Companies began to introduce several IoT-based products and services. In essence, innovation on the Internet of Things is characterized by the combination of physical and digital components to create new products and enable new business models. Thanks to increasingly efficient energy management, broadband communication, reliable memory, and advances in microprocessor technologies, it has become possible to digitize the functions and key features of products from the industrial era.With the popularity of tools for Arduino, Espressif Systems designed its platform to bring almost total compatibility between ESP and Arduino making countless low-cost sensors and peripheral boards originally developed for Arduino work, both at the hardware level and software, on ESP platform. This has made ESP very attractive for those who already used Arduino and with the advantages of having native communication with Wi-Fi, which is not found in Arduino's. Automatically, for decades, automation has been part of the daily life of human beings, whether helping their profession, locomotion, entertainment, or comfort. Currently, there are old solutions that partially solve our new problems, and that with small adjustments can enhance and bring complete solutions. The development of communication between the two microcontrollers, Arduino and ESP32, becomes important mainly in the existence of already developed projects that have to upgrade needs to improve, adapt or customize their application, bringing improvements such as Internet connectivity, low cost, speed and efficiency.

With this in mind, this work deals with the development of a simple electronic solution so that there is communication in duplex mode between the equipment, avoiding an overload between the microcontrollers without losing its numerous functionalities, such as Arduino with its sensors and ESP with its interfaces Wi-Fi and Bluetooth. For intermediating the communication between the microcontrollers, the level shifter and the I2C protocol were used for the interaction between them.

Resources Used

In addition to the Arduino and ESP32 microcontrollers, a voltage level correction circuit, level shifter, was used, with the possibility of transmission in both directions, bringing the feasibility of hardware connection between the different platforms through the software communication protocol. I2C which is Inter-Integrated Circuit.The level shifter is a hardware used to perform conversions between voltages from a lower voltage range to a higher voltage range and vice versa. The circuit in question has interconnected transistors implemented as main devices operable within the main voltage range (Huang et al. 2015). A level shifter device converts one voltage level to another. Voltage level changes are used to interface between several circuit blocks that operate with different supply voltages (Gosatwar and Ghodeswar, 2015).I2C is a serial bus protocol created by Philips, the I2C bus is popular because it is simple to use. It is used to allow the fastest device to communicate with slower devices without data loss. The I2C bus controller interfaces with the master and slave devices (Kumari and Gayathri, 2017).The functioning of the I2C Protocol is basically the interaction between elements following a master / slave hierarchy. Second (Nguyen and Dugenske, 2018), the device performing the master function, has the ability to send and request information from the device performing the slave function in the communication structure, where they answer the aforementioned requests.

Development of the Solution

To resolve the communication between the two devices, a level shifter voltage leveling circuit was used, for the union between the ESP32 and Arduino platforms, and the I2C communication protocol to send commands between them.

The level shifter has the first and second power connections 3.3V and 5V connected in the voltage ranges that supply power to the different platforms, namely, 5 volts on Arduino and 3.3 volts on ESP32. A core voltage fixing element, implemented in the 2N7000 transistor, has a limit voltage greater than or equal to the difference between the 3.3 signal and 5V signal voltage range and the main voltage range is configured to avoid overloading the transistors with voltages beyond the core voltage range. The input for the level shifter must be within the voltage range of the core. The output signal of the level shifter has a high level in the high voltage of the voltage range of 3.3V and 5V and a low level in approximately a threshold voltage above the low voltage level of the core voltage range (Inoue et al. 2017).

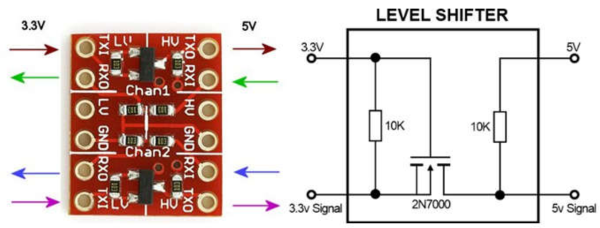

Figure 1 (a): Level shifter board - Figure1(b): Basic layout

In short, the 3.3V (Low Voltage) and 5V (High Voltage) connections are used to connect the supply voltages of the ESP32 and Arduino platforms respectively. The 5V Signal and 3.3V Signal connections are used to connect the I2C communication ports of both microcontrollers. This makes it possible for the level shifter circuit to identify input voltages and switch output voltages in both directions.The communication scheme between Arduino and ESP32 consists of using the I2C communication protocol, which describes the operation of a serial communication bus that uses only two wires. As a result of the reduced amount of wires for communication, the number of pins used by the elements is also reduced, in addition to promoting a greater organization of the set when we have many devices interacting with each other, (Kumari, et al. 2017). The protocol in question also enables the communicability between several devices, since it works with the logical addressing paradigm between them.At I2C, data travels through the Serial Data (SDA) and Clock Serial (SCL) ports, with SCL being the port that sends clock synchronize pulses and SDA the port that effectively travels with data in both directions.

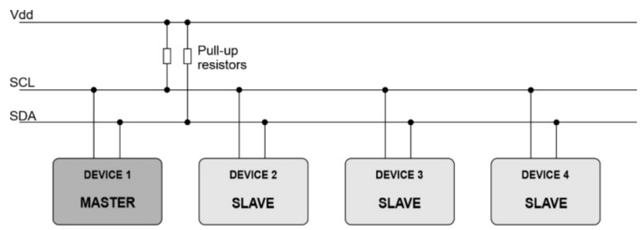

Figure 2: Communication and transmission of the I2C protocol

The vast majority of modern prototyping platforms such as Arduino and ESP32 have native support for communication through the I2C protocol. In the Arduino model Nano, the communication pins reserved for the I2C protocol are A4 and A5, being the first responsible for the data transmission and the second for the clock synchronize pulses. Similarly, in ESP32, where the pins responsible for this communication are 2 and 15, with pin 15 being responsible for data transmission.

Circuit Assembly: The communication between Arduino and ESP32 would happen naturally when connecting pins A4 (Arduino) to 15 (ESP32) and A5 (Arduino) to 2 (ESP32), but this connection is made impossible by the different voltage bands between the platforms, namely, 5 volts on Arduino and 3.3 volts on ESP32. As previously mentioned, it is necessary to use an intermediate circuit to match the communication between both platforms.

Figure 3 shows the basic circuit assembly demonstration.

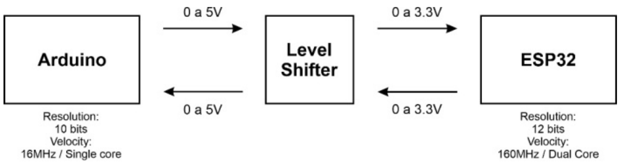

Figure 3. Demonstration of the circuit assembly.

I2C communication takes place digitally, ensuring only two logic voltage levels as shown in table 1 below.

Table 1. Logical voltage levels on the platforms used

Arduino		ESP32	
Lowlevel	High Level	Lowlevel	High Level
0 Volts / Low	5 Volts / High	0 Volts / Low	3.3 Volts / High

Implementation of the Solution

There are cases of prototypes with legacy systems, where they do not have access to the internet and need to communicate with the outside world, making an upgrade necessary. For example, a prototype contains the Arduino microcontroller, but external communication needs to be carried out, in these cases, it is necessary to use another microcontroller for internet communication, we denote the ESP32, which has Wi-Fi and Bluetooth interfaces. For communication between microcontrollers, it is necessary to use the level shifter in the voltage range and filtering of the logic level, because they work in different ranges, both in the logical level and in the voltage range.The use of the approached circuit model would fit, for example, in the use of sensing that may overburden the microcontroller used. Typical occurrences would be programmatic approaches that use a large processing flow or even interruptions by hardware, which, in the latter case, could bring instability in the attention span of the microcontroller to other processes. This would, of course, require the use of more than one control module. The example in figure 4 illustrates the prototyping of the hardware communication development, using a logic and voltage level correction system.

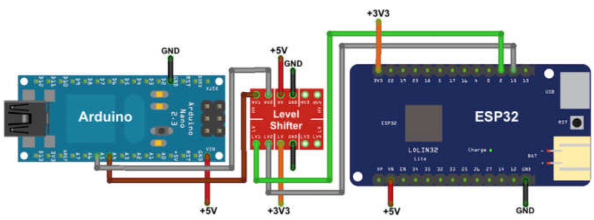

Figure 4. Scheme and prototyping of the proposed solution

This solution has as main functionality, avoiding overload and instability in the microcontroller when using some sensors, where they use the microcontrollers to the maximum, making it necessary to pay greater attention to the hardware, for example, in the use of a water flow sensor that requires attention of the microcontroller through hardware interruptions.When using a sensor where it interrupts the Arduino with the pulses to capture the data, the Arduino, in addition to receiving the data captured from the sensor, is responsible for carrying out the necessary data processing and calculations, making it essential to use another microcontroller for avoid instability and incorrect data capture, where the ESP32 will be responsible for connectivity to the internet, that is, the Arduino captures the sensor data, passes it through the ESP32 where it will transmit this data to the internet, thus making the Internet of Things communication. In this example, the ESP32 microcontroller would be more available for attention to continuous communication with the internet. Other applications and/or sensors that need more than one platform, need more attention from the microcontroller, because they use larger processing or work in a redundant way, making it necessary to use one more microcontroller, since they communicate with each other.As the data collection is performed through a single microcontroller, the other microcontroller may be responsible for other tasks, such as the communication of data with the external environment. The model also has the benefit of the sensor working natively with its microcontroller, namely the Arduino.

Conclusion

In this work, an important solution was presented for communication between ESP32 and Arduino platforms, using a logic level and voltage correction system, avoiding an overload between microcontrollers without losing its numerous functionalities, such as Arduino with its sensors and ESP with its Wi-Fi interfaces and Bluetooth.This communication solution between these microcontrollers becomes very useful, mainly in the existence of projects already developed with the ESP32 or with the Arduino, where it is necessary to use more digital ports and/or analog inputs to perform the upgrade of your application, bringing improvements such as Internet connectivity, low cost, speed and efficiency.The method covered in this article is essential for the system to bring all the possibilities listed. The presented approach can still be useful in countless cases and situations, such as corrections of logic level and/or voltage range between countless models of microcontrollers, bringing possibilities of applications of low financial cost, high capacity of connectivity and in data processing.

REFERENCES

Barbon, G., Margolis, M., Palumbo, F., Raimondi, F., &Weldin, N. 2016. “Taking Arduino to the Internet of Things: The ASIP programming model” Computer CommunicationsVolumes 89–90, 1 September, Pages 128-140.

De Oliveira, Fábio Franco; Villaça, Rodolfo da Silva; Santos, Celso Alberto Saibel; Colistete Junior, Roberto. “EstGeoMag: Integrando soluções de Hardware, Software e Internet das Coisas na medição de grandezas Geomagnéticas.” In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE (SEMISH), 44, 2017, São Paulo. Anais do XLIV Seminário Integrado de Software e Hardware. Porto Alegre: Sociedade Brasileira de Computação,Jul. 2017.

Huang, Tien-Chien, and Ruey-Bin Sheen. "Level shifter for high density integrated circuits." U.S. Patent No. 9,197,199. 24 Nov. 2015.

Inoue, H., Matsuzaki, T., Nagatsuka, S., Ishizu, T., &Onuki, T. "Semiconductor device including buffer circuit and level shifter circuit, and electronic device including the same." U.S. Patent No. 9,755,643. 5 Sep. 2017.

Kumari R. S. S. and C. Gayathri, 2017. "Interfacing of MEMS motion sensor with FPGA using I2C protocol," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, pp. 1-5.

Kumari, R. Shantha Selva, and C. Gayathri. "Interfacing of MEMS motion sensor with FPGA using I2C protocol." 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS). IEEE, 2017.

Lei Yu, BabarNazir, YinlingWang. "Intelligent power monitoring of building equipment based on Internet of Things technology."Computer Communications, Volume 157, 1 May 2020, Pages 76-84.

McRoberts, Michael. "Arduino básico". Novatec Editora, 2018, pp. 12-15.

Nguyen, V., & Dugenske, A. (2018). An I2C based architecture for monitoring legacy manufacturing equipment. Manufacturing Letters, 15, 67–70.

Noraini Azmi, SukhairiSudin, Latifah MunirahKamarudin, Ammar Zakaria, Retnam Visvanathan, Goh Chew Cheik, Syed Muhammad Mamduh Syed Zakaria, Khudhur Abdullah Alfarhan, R Badlishah Ahmad. 2018 IOP Conf. Ser.: Mater. Sci. Eng. 318 012051. "Design and Development of Multi-Transceiver Lorafi Board consisting LoRa and ESP8266-Wifi Communication Module".

Oliveira, Sérgio. "Internet das coisas com ESP8266, Arduino e Raspberry PI". NovatecEditora, 2017, pp. 60.

PritiGosatwar and UjwalaGhodeswar, "Design of voltage level shifter for multi-supply voltage design," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, 2016, pp. 0853-0857.

Wortmann, F., Flüchter, K. 2015. "Internet of Things". Bus Inf Syst Eng 57, 221–224.

****