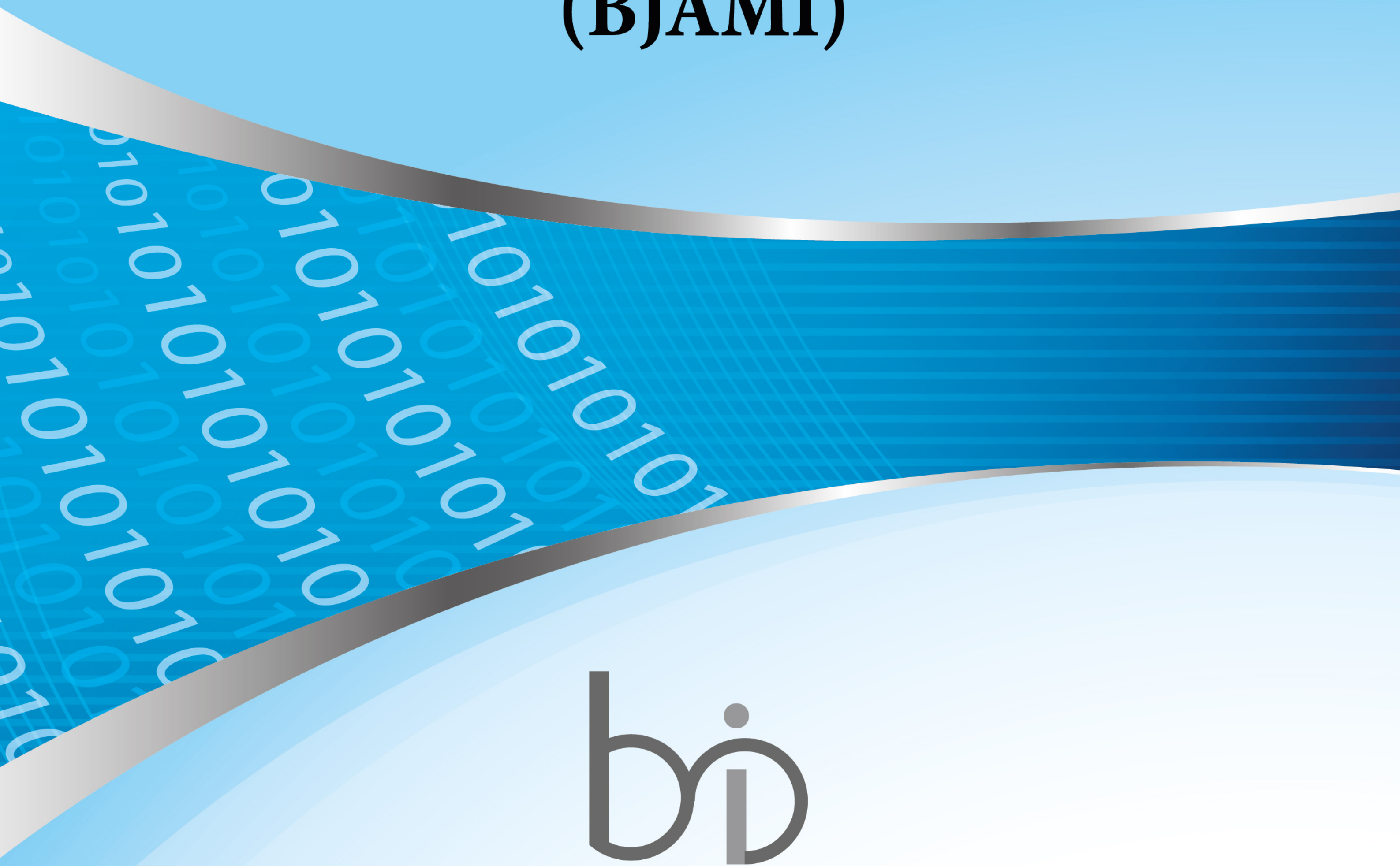
GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

ISSN 2545-4803 on line

BALKAN JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS (BJAMI)



YEAR 2020

VOLUME III, Number 1

GOCE DELCEV UNIVERSITY - STIP, REPUBLIC OF NORTH MACEDONIA
FACULTY OF COMPUTER SCIENCE

ISSN 2545-4803 on line

BALKAN JOURNAL OF APPLIED MATHEMATICS AND INFORMATICS

BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS

(BJAMI)

YEAR 2020 **VOLUME III, Number 1**

AIMS AND SCOPE:
BJAMI publishes original research articles in the areas of applied mathematics and informatics.

Topics:
1. Computer science;
2. Computer and software engineering;
3. Information technology;
4. Computer security;
5. Electrical engineering;
6. Telecommunication;
7. Mathematics and its applications;
8. Articles of interdisciplinary of computer and information sciences with education, economics, environmental, health, and engineering.

Managing editor
Biljana Zlatanovska Ph.D.

Editor in chief
Zoran Zdravev Ph.D.

Lectoure
Snezana Kirova

Technical editor
Slave Dimitrov

Address of the editorial office
Goce Delcev University – Stip
Faculty of philology
Krste Misirkov 10-A
PO box 201, 2000 Štip,
Republic of North Macedonia

BALKAN JOURNAL
OF APPLIED MATHEMATICS AND INFORMATICS (BJAMI), Vol 3

ISSN 2545-4803 on line
Vol. 3, No. 1, Year 2020

EDITORIAL BOARD

Adelina Plamenova Aleksieva-Petrova, Technical University – Sofia,
Faculty of Computer Systems and Control, Sofia, Bulgaria
Lyudmila Stoyanova, Technical University - Sofia , Faculty of computer systems and control,
Department – Programming and computer technologies, Bulgaria
Zlatko Georgiev Varbanov, Department of Mathematics and Informatics,
Veliko Tarnovo University, Bulgaria
Snezana Scepanovic, Faculty for Information Technology,
University “Mediterranean”, Podgorica, Montenegro
Daniela Veleva Minkovska, Faculty of Computer Systems and Technologies,
Technical University, Sofia, Bulgaria
Stefka Hristova Bouyuklieva, Department of Algebra and Geometry,
Faculty of Mathematics and Informatics, Veliko Tarnovo University, Bulgaria
Vesselin Velichkov, University of Luxembourg, Faculty of Sciences,
Technology and Communication (FSTC), Luxembourg
Isabel Maria Baltazar Simões de Carvalho, Instituto Superior Técnico,
Technical University of Lisbon, Portugal
Predrag S. Stanimirović, University of Niš, Faculty of Sciences and Mathematics,
Department of Mathematics and Informatics, Niš, Serbia
Shcherbacov Victor, Institute of Mathematics and Computer Science,
Academy of Sciences of Moldova, Moldova
Pedro Ricardo Morais Inácio, Department of Computer Science,
Universidade da Beira Interior, Portugal
Sanja Panovska, GFZ German Research Centre for Geosciences, Germany
Georgi Tuparov, Technical University of Sofia Bulgaria
Dijana Karuovic, Tehnical Faculty “Mihajlo Pupin”, Zrenjanin, Serbia
Ivanka Georgieva, South-West University, Blagoevgrad, Bulgaria
Georgi Stojanov, Computer Science, Mathematics, and Environmental Science Department
The American University of Paris, France
Iliya Guerguiev Bouyukliev, Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences, Bulgaria
Riste Škrekovski, FAMNIT, University of Primorska, Koper, Slovenia
Stela Zhelezova, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Bulgaria
Katerina Taskova, Computational Biology and Data Mining Group,
Faculty of Biology, Johannes Gutenberg-Universität Mainz (JGU), Mainz, Germany.
Dragana Glušac, Tehnical Faculty “Mihajlo Pupin”, Zrenjanin, Serbia
Cveta Martinovska-Bande, Faculty of Computer Science, UGD, Republic of North Macedonia
Blagoj Delipetrov, Faculty of Computer Science, UGD, Republic of North Macedonia
Zoran Zdravev, Faculty of Computer Science, UGD, Republic of North Macedonia
Aleksandra Mileva, Faculty of Computer Science, UGD, Republic of North Macedonia
Igor Stojanovik, Faculty of Computer Science, UGD, Republic of North Macedonia
Saso Koceski, Faculty of Computer Science, UGD, Republic of North Macedonia
Natasa Koceska, Faculty of Computer Science, UGD, Republic of North Macedonia
Aleksandar Krstev, Faculty of Computer Science, UGD, Republic of North Macedonia
Biljana Zlatanovska, Faculty of Computer Science, UGD, Republic of North Macedonia
Natasa Stojkovik, Faculty of Computer Science, UGD, Republic of North Macedonia
Done Stojanov, Faculty of Computer Science, UGD, Republic of North Macedonia
Limonka Koceva Lazarova, Faculty of Computer Science, UGD, Republic of North Macedonia
Tatjana Atanasova Pacemska, Faculty of Electrical Engineering, UGD, Republic of North Macedonia

CONTENT

UDC: 628.38.049.77:004.38]:621.371.31.037.33

MEASUREMENT AND VISUALIZATION OF ANALOG SIGNALS WITH A MICROCOMPUTER CONNECTION

GOCE STEFANOV, VASILIJA SARAC, BILJANA CHITKUSHEVA DIMITROVSKA

Abstract. In this paper, a microcomputer connection scheme is discussed and implemented. The analog signals are connected on the first microcomputer and then these signals are sent with serial transmissions to the second microcomputer. The values of analog signals from the second microcomputer with I2C bus are sent on the display. TwoAtmega328 controllers based on Arduino Nano board are used for the implementation of serial transmissions. The designed solution is practically implemented, and the results are verified.

Keywords: microcontroller, I2C bus, serial bus, LCD

1. Introduction

Microcomputers are intelligent electronic components that have a number of advantages over discrete electronic components. Their main advantage is the packing density of the chip itself. It is the result of the development of microelectronics and enables over one million discrete electronic elements to be embedded on a surface of 1 cm^2. Their second advantage, which distinguishes them from discrete electronic components, is their application flexibility. The latter implies the ability to run different applications with the same network hardware, and with software changes [1].

This not only reduces the time needed to design an electronic device, but it also gives the electronic circuit designer flexibility in exploring new applications with a newly implemented hardware solution [2], [3].

Since the advent of the first microcomputers until today, they have experienced an increase in the processing speed and the ability to increase their connectivity with a larger number of input-output devices. Today 8-bit, 16-bit, 32-bit, and 64-bit microcomputers from different manufacturers can be found on the market: Microchip, ARM, and Motorola.

In this paper, a solution for measurement and visualization on two sensor signals with the microcomputer connection on two controllers is shown. Atmega 328P microcomputer [2], [3] is used.

2. Measurement and Visualization of Sensor Signal with the Microcomputer Connection on Two Controllers

This paper presents a solution for measurement and visualization of data from a voltage sensor and current sensor. The two sensor readings are character coded by the first microcomputer and then transmitted as a long character string via standard serial communication to the second microcomputer. The latter splits and decodes the string stream and displays the variables on a 20×4 LCD display [4], [5]. Figure 1 shows the block circuit for microcomputer connections. In this, the microcomputer connection the following circuits are used:

-two microcomputers
-analog to digital convertor -voltage sensor
-current sensor
-expander
-LCD display

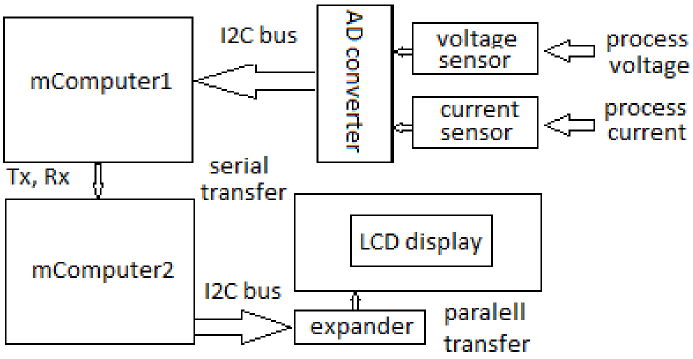


Figure 1.*Block diagram on microcomputer connections*

Figure 1 shows that the circuit is based on two microcomputers. The first microcomputer through I2C interface is connected with the analog to digital converter. The analog to digital convertor is connected to the voltage and the current sensor. Also, *T*x and *R*x pin of the first microcomputer is connected with *R*x, *T*x pin on the second microcomputer. The second microcomputer through I2C interface is connected to a LCD display. So, the data on the voltage and the current from process controlling parameters from sensors through the AD convertor and the two microcomputers are visual presented on the LCD display.

2.1 Hardware Part of the Microcomputer Connection Circuit

Figure 2 shows the real microcomputer circuit. In this microcomputer connection, the following circuits are used:

a.) two microcomputers Atmega328P in Arduino Nano platform
b.) 4 channel analog to digital convertor ADS1115
c.) voltage sensor
d.) current sensor
e.) expander
f.) 20x4 LCD display

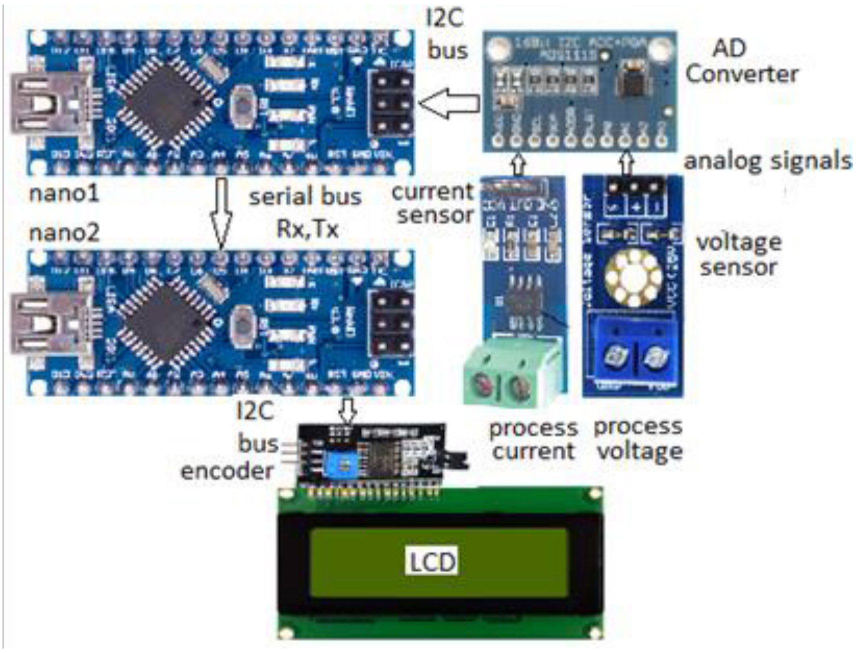


Figure 2.*Real microcomputer circuit*

Figure 2 shows that the circuit is based on two microcomputers Atmega 328Pembedded in an Arduino Nano platform. The first microcomputer with I2C interface is connected with an analog to digital converter ADS1115. ADS1115 analog to digital convertor is connected to the voltage and the current sensor. Also, Tx and Rx pin of the first microcomputer is connected with Rx, Tx pin on the second microcomputer. The second microcomputer with I2C interface is connected to the LCD 20x4 display. So, the data on the voltage and the current from the process controlling parameters from sensors through ADS1115, the two microcomputers are visual presented on the LCD display.

a.) Microcomputer

The microcomputer Atmega 328P in an Arduino Nano platform is selected. Figure 3 shows a pin connection on Arduino Nano [4].

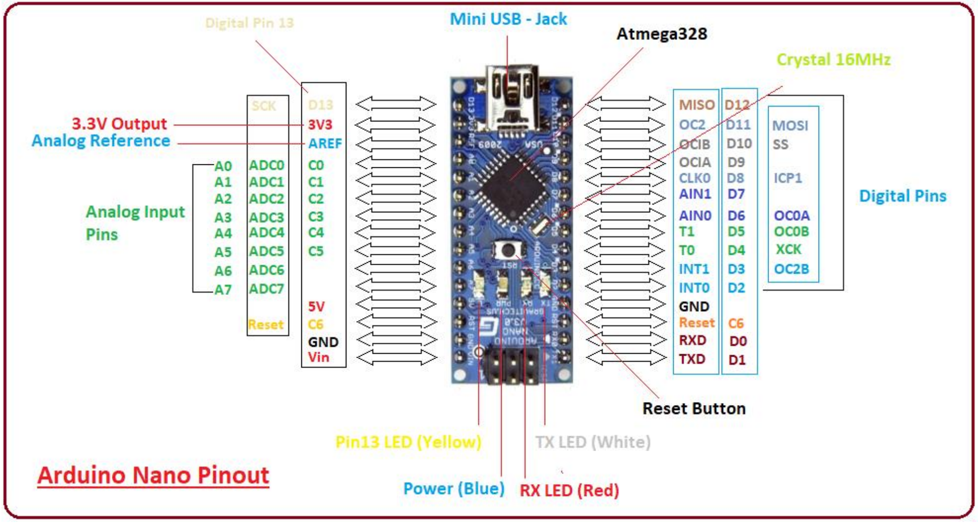


Figure 3. *Pin connection on Arduino Nano*

Arduino Nano V3.0 is a microcontroller board based on Atmega328. Arduino Nano is a very small board that can be placed quite easily in a breadboard. It is normally used in projects where the weight of electronics components is a crucial issue, e.g. quad copter, omnothopter etc. It has 14 digital pins that can be used as output as well as input. Among these 14 digital pins, 6 are pwm pins. It has 8 analog pins and 1 UART serial port (*T*x/*R*x). It has a 16MHz crystal oscillator.

b.) Analog to digital convertor ADS1115

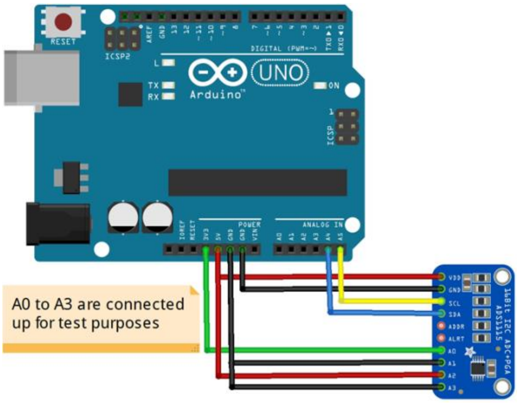
The ADS1115 analog-to-digital converter provides high 16-bit precision at 860 samples/second over I2C. The chip can be configured as 4 single-ended input channels, or two differential channels. As a nice bonus, it even includes a programmable gain amplifier, up to x16, to help boost up smaller single/differential signals to the full range. We choose this ADC because it can run from 2V to 5V power/logic, can measure a large range of signals and it is very easy to use. It is a great general-purpose 16-bit converter [5]. Figure 4 shows the electronic board on this AD converter.

Figure 4.*Electronic board ADS1115*

The ADS11x5 chips have a base 7-bit I2C address of 0x48 (1001000) and a clever addressing scheme that allows four different addresses to use just one address pin (named ADR for ADdRess). To program the address, connect the address pin in this way:

- 0x48 (1001000) ADR - GND
- 0x49 (1001001) ADR - VDD
- 0x4A (1001010) ADR - SDA
- 0x4B (1001011) ADR – SCL

Figure 5 shows a typical connection on ADS1115 and Arduino Uno controller.



A0 to A3 are connected up for test purposes

Figure 5. *Typical connection on ADS1115 and Arduino Uno*

c.) *Voltage sensor*

This module is based on the principle of a resistive voltage divider design. It can make the red terminal connector input voltage up to 5 times smaller. Input voltages are up to 5V. The voltage detection module input voltage is not greater than 5Vx5=25V, (if using 3.3V systems, the input voltage is not greater than 3.3Vx5=16.5V). AVR chips have 10-bit AD, so this module simulates a resolution of 0.00489V (5V/1023). So, the minimum voltage of the input voltage detection module is 0.00489Vx5=0.02445V. Figure 6 shows this voltage sensor.

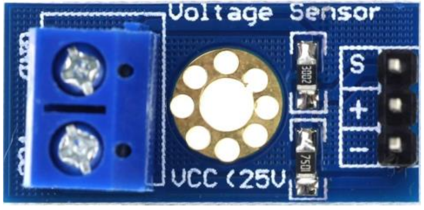


Figure 6.*Voltage sensor*

d.) *Current sensor*

The current sensor chip ACS712ELC-5A is used. The module can measure plus or minus 5 amps, corresponding to the analog output 66mV/A. There is no detection of the current through, and the output voltage is VCC/2. Figure 7 shows this current sensor.

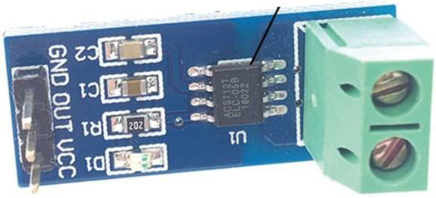


Figure 7.*Current sensor*

e.) *Expander*

The expander circuit is used for the connection on a microcomputer through I2C bus with the LCD display. By this circuit, serial I2C data converts to parallel data. Figure 8 shows this expander circuit.

Figure 8.*Expander circuit*

f.) *LCD display*

The LCD 20x4 display is used for visualization on date values on the voltage and the current. The LCD is connection with the expander circuit by parallel date port. Figure 9 shows the LCD 20x4 display.

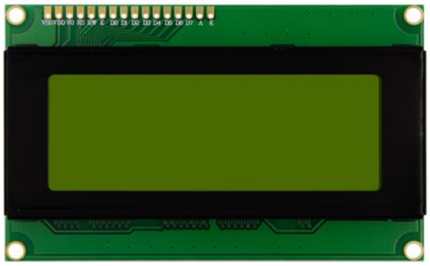


Figure 9.*LCD 20x4 display*

2.1 Software Part of the Microcomputer Connection Circuit

In the microcomputer connection serial I2C and serial Rx, Tx bus is used. In this part, the principles of these connections will be shown.

a.) I2C interface

I2C communications have become the *de facto* method of communicating between microcontrollers, microcomputers and a variety of integrated circuits and sensors. I2C is a serial protocol used on a low-speed 2-wire interface [6]. It was originally developed by Phillips in 1982 to allow integrated circuits within television receivers to communicate with one another. I2C is used with microcontrollers such as the Arduino and with microcomputers such as the Raspberry Pi. Many displays and sensors interface with their host controller using I2C, Figure 10.

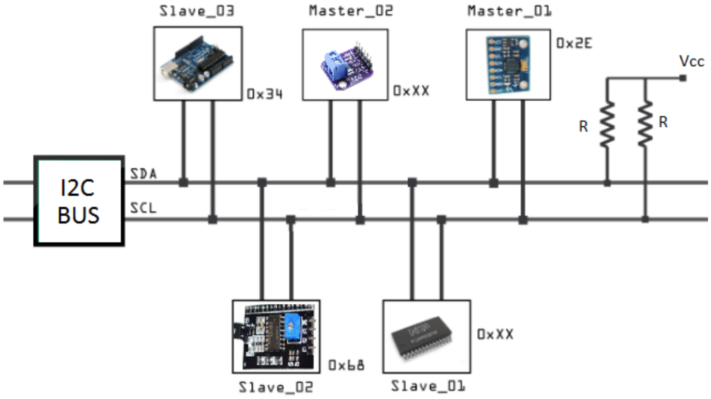


Figure 10.*I2C communication*

I2C can only be used over short distances; after all, it was originally meant to communicate between integrated circuits on the same printed circuit board. The maximum distance of a reliable transmission decreases as the speed increases; at the slowest speed (100 Kbaud or a clock rate of 100 KHz) the maximum distance is about a meter.

The I2C communication can be easily implemented in many electronic designs that require communication between a master device and multiple slave devices or even multiple master devices. The easy implementation comes with the fact that only two wires are required for communication between up to almost 128 (112) devices when using 7 bits addressing and up to almost 1024 (1008) devices when using 10 bits addressing.

The data signal is transferred in sequences of 8 bits. So, after a special start condition occurs, there comes the first 8 bits sequence which indicates the address of the slave to which the data is being sent. After each 8 bits sequence follows a bit called Acknowledge. After the first Acknowledge bit, in most cases, there comes another addressing sequence, but this time for the internal registers of the slave device. After the addressing sequences, follows the data sequences as many until the data is completely sent and it ends with a special stop condition, Figure 11.

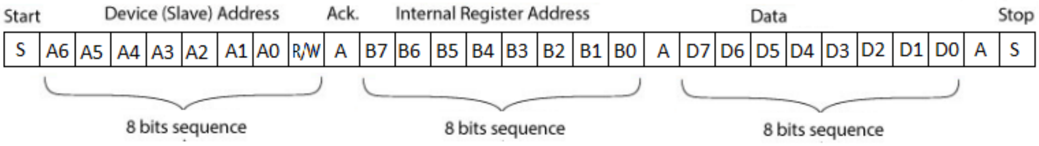


Figure 11.*Date transfer with I2C protocol*

Let us take even a closer look at these events. The start condition occurs when the data line drops low, while the clock line is still high. After this, the clock starts and each data bit is transferred during each clock pulse. The device addressing sequence starts with the most significant bit (MSB) first and ends with the least significant bit (LSB) and it is actually composed of 7 bits because the 8th bit is used for indicating whether the master will write to the slave (logic low) or read from it (logic high), Figure 12.

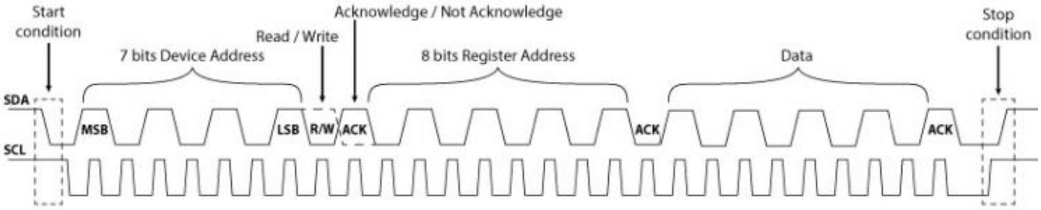


Figure 12.*The impulse sequence for I2C communications*

The next bit, AKC/ NACK, is used by the slave device to indicate whether it has successfully received the previous sequence of bits. So, at this time, the master device hands the control of the SDA line over to the slave device and, if the slave device has successfully received the previous sequence, it will pull the SDA line down to the condition called Acknowledge. If the slave does not pull the SDA line down, the condition is called Not Acknowledge, and it means that it did not successfully receive the previous sequence, which can be caused by several reasons. For example, the slave might be busy, might not understand the received data or command, cannot receive any more data and so on. In such a case, the master device decides how it will proceed as shown in Figure 13.

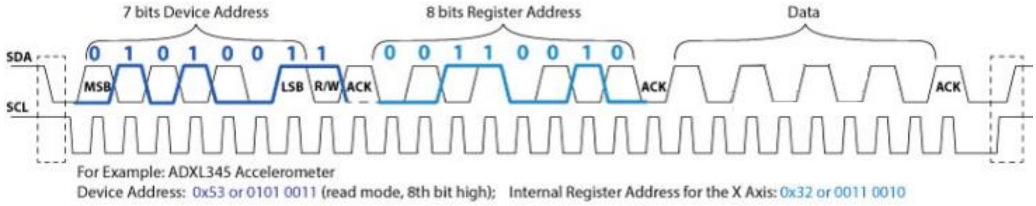


Figure 13. *The impulse sequence in case of a busy slave*

Next is the internal registers addressing. The internal registers are locations in the slave's memory containing various information or data. For example, the ADX345 Accelerometer has a unique device address and additional internal registers addresses for the X, Y and Z-axis. So, if we want to read the data of the X-axis, first we need to send the device address and then the particular internal register address for the X-axis. These addresses can be found from the datasheet of the sensor. After addressing, the data transfer sequences begin from either the master or the slave depending on the selected mode at the R/W bit. After the data is completely sent, the transfer will end with a stop condition that occurs when the SDA line goes from low to high while the SCL line is high.

b.) UART Serial communication

Figure 14 shows standard UART communications between two microcomputers.

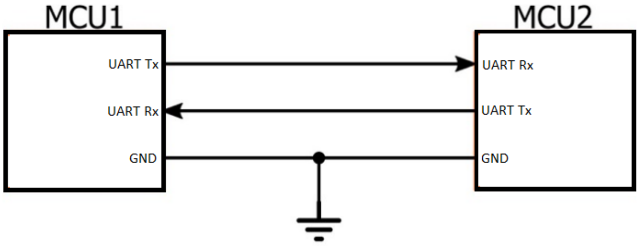


Figure 14. *UART communications between two microcomputers*

UART stands for Universal Asynchronous Reception and Transmission is a simple communication protocol that allows the Arduino to communicate with serial devices. The UART system communicates with digital pin 0 (*R*x), digital pin 1 (*T*x), and with another computer via the USB port [7]. This peripheral, found on all Arduino boards, allows the Arduino to directly communicate with a computer thanks to the fact that the Arduino has an onboard USB-to-Serial converter. Therefore, programs written in Windows, Mac, or Linux OS can be used with an Arduino connected to a USB port as if it was a serial port (serial port communication is trivial compared to USB communication).

c.) Software

All Arduino boards support software in micro C language. This means that the program is compatible with all controllers in the ARM family. Here are some typical software instructions for initializing individual devices in the specific task of this paper.

Include library:

```
#include <Wire.h> // Library for I2C communication
#include <LiquidCrystal_I2C.h> // Library for LCD
#include <Adafruit_ADS1115.h> // Library for AD convertor circuit
#include <LiquidCrystal_I2C.h> // Library for LCD
// Wiring: SDA pin is connected to A4 and SCL pin to A5.
// Connect to LCD via I2C, default address 0x27 (A0-A2 not jumpered)
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27,20, 4); // Change to (0x27,20,4) for 20x4 LCD.
```

Initialize LCD:

```
void setup () {
// Initiate the LCD:
lcd.init ();
lcd.backlight ();
delay (100);
Serial.begin (9600);
lcd.setCursor (2, 0);
lcd.print ("GEES-electronic");
lcd.setCursor (0, 1);
lcd.print ("microcontroll-I2C");
lcd.setCursor (1, 2);
lcd.print ("voltage - current ");
lcd.setCursor (6, 3);
lcd.print ("V");
lcd.setCursor (16, 3);
```

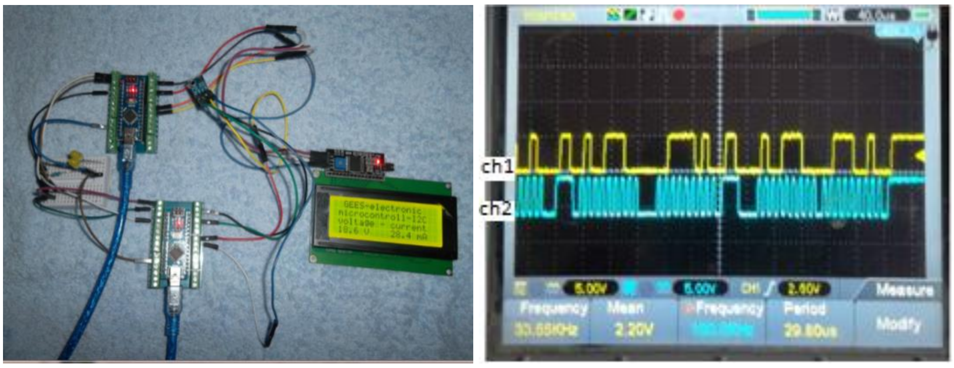
```
 lcd.print ("mA");
 pinMode (ledpin, OUTPUT);
```

Read sensors values:

```
void loop (void)
{
int16_t adc0, adc1;
adc0 = ads.readADC_SingleEnded(0);
adc1 = ads.readADC_SingleEnded(1);
```

3. Practical Implementation of the Microcomputer Connection

A practically realized connection of the two microcomputers with an integrated I2C and UART interface and embedded code is shown in Figure 15a, and Figure 15b shows an experimentally obtained waveform on pin SCL –channel1 and pin SDA-channel2 at I2C bus on the second controller.



a.) b.)

Figure 15.*A practically realized connection of the two microcomputers with an integrated I2C and UART interface: a.) prototype electronic board, b.) waveform pin SCL-chanel1 and pin SDA-channel 2 for vertical 5v/div, horizontal 40µS/div.*

Figure 15b shows that the clock speed SCL frequency (channel 1) is 33.55 kHz and the data speed SDA frequency (channel 2) is 100 kHz.

4. Conclusion

In this paper, the microcomputer connection scheme is designed and implemented. The analysis of the used hardware components as well as the theoretical description of the UART and I2C serial protocols are made. The analog signals are connected in an AD converter circuit and then are sent to the first microcomputer. These signals are sent with serial transmissions to the second microcomputer. The values on analog signals from the second microcomputer with I2C bus are transmitted on the LCD display. Two Atmega328 controllers, based on Arduino Nano board, are used for the implementation of serial transmissions. Based on the solution, an electronic prototype is practically implemented, and the results are verified.

References

[1] Marcovitz A. B., Introduction to Logic and Computer Design, McGraw-Hill, New York, 2008.
[2] Atmel AVR, (2018) 48A/PA/88A/PA/168A/PA/328/P, Microchip Technology Incorporated, All Rights Reserved.
[3] Atmel ATmega, (2014), 640/V-1280/V-1281/V-2560/V-2561/V, Atmel Corporation, 1600 Technology Drive, San Jose, CA 95110 USA.
[4] https://www.theengineeringprojects.com/2018/06/introduction-to-
[5] arduino-nano.html.
[6] http://arduinolearning.com/code/ads1115-analog-to-digital-converter-and-arduino.php
[7] https://maker.pro/arduino/tutorial/common-communication-peripherals-on-the-arduino-uart-i2c-and-spi
[8] https://www.allaboutcircuits.com/technical articles/back-to-basics-the-universal-asynchronous-receivertransmitter-uart/

Goce Stefanov
University of Goce Delcev Stip
Faculty of Electrical Engineering
R. N. Macedonia
goce.stefanov@ugd.edu.mk

Vasilija Sarac
University of Goce Delcev Stip
Faculty of Electrical Engineering
R. N. Macedonia
vasilija.sarac@ugd.edu.mk

Biljana Citkuseva Dimitrova
University of Goce Delcev Stip
Faculty of Electrical Engineering
22 October bb, 2420 Radovis
R. N. Macedonia
biljana.citkuseva@ugd.edu.mk