

# Implementation of Microcontroller-Based Memristive Chaotic Circuit

S.C. YENER<sup>a,\*</sup>, C. BARBAROS<sup>b</sup>, R. MUTLU<sup>b</sup> AND E. KARAKULAK<sup>b</sup>

<sup>a</sup>Sakarya University, Electrical and Electronics Engineering Department, Sakarya, Turkey

<sup>b</sup>Namık Kemal University, Electronics and Communication Eng. Dept., Çorlu, Tekirdağ, Turkey

In 1971, Leon Chua theoretically postulated that the memristor is the fourth fundamental circuit element, besides the three well-known circuit elements; namely, resistor, capacitor and inductor. For a long time, because of lack of a simple and practical realization, memristor remained just a theoretical element and rarely appeared in the literature. In 2008, a research team from HP laboratories declared that they had found a physical implementation based on thin films, behaving as a memristor. Memristor can offer new opportunities in circuit design due to its nonlinear behavior and memory. Nevertheless, since a cheap and reliable practical implementation of memristor is yet unavailable on the market, the design of such a realization, which mimics memristor behavior, is vital from the point of view of real-world circuit design. In this paper, a new microcontroller-based memristive chaotic circuit is proposed. Presented design has been implemented using an Arduino Mega board, which solves numerically the dynamics of the memristor-based chaotic system using Runge-Kutta method. It sends the chaotic signals to the outputs of the circuit, using digital-to-analog converters. Chaotic dynamics and the strange attractors are obtained from the circuit using both, the computer simulations and the lab experiments. Considering both simulation and experimental results, it is shown that the proposed circuit mimics well the dynamics of the memristive chaotic system.

DOI: [10.12693/APhysPolA.132.1058](https://doi.org/10.12693/APhysPolA.132.1058)

PACS/topics: 84.30.-R, 84.30.BV, 05.45.-A

## 1. Introduction

Leon Chua theoretically postulated, in 1971, that there should have been one more fundamental circuit element, besides the three well-known circuit elements; namely, resistor, capacitor and inductor. He called the missing circuit element, memristor, and predicted its properties [1]. For decades, it remained just as a theoretical element and appeared in the literature rarely. A team from HP laboratories announced the fabrication of a physical implementation, behaving as a memristor [2]. Only a handful of organizations or companies have memristors since it does not have a simple and practical realization. That is why reports on theoretical analysis and simulations of memristors are more common than its experimental studies. Since 1971, emulators are designed to show how a memristor would behave. They can present the behavior of memristor, to a certain extent, in various applications.

Modeling of electronic components and systems and their implementations are crucial for achieving the accurately working practical systems. Various electronic systems and their realizations [3–6] and specifically memristor emulators and SPICE macromodels, presenting memristor-like behavior, can be found in the literature [7–17]. SPICE models are beneficial for modeling characteristics of the memristor, but they have no alternatives in practical realizations.

As soon as the memristor was found [2], its chaotic circuit studies appeared in literature [18]. Some of them have circuit implementations [14, 15, 18]. It is hard to realize memristor-based analog chaotic circuit applications, since there is no memristor available commercially [14, 15, 18].

In this study, an alternative path is taken, to design a memristive chaotic circuit and, instead of using an emulator in analog chaotic circuit, a microcontroller based memristive chaotic circuit is designed. An Arduino Mega 2560 R3 microcontroller board is used as the microcontroller. Two digital to analog DAC0808 converters are connected to the microcontroller. A cubic flux-charge relationship is chosen for the Memristor characteristic. The state-space equations of the Chua circuit, given in [18] are solved in the microcontroller using Runge-Kutta method. Two of the selected state variables are sent to DACs. The analog signals are easily observable using an oscilloscope.

The paper is organized as follows. In the second section, the cubic flux-charge characteristic of the memristor is presented, the dynamic model of the Chua's circuit is given, and its dynamics are studied with MATLAB™ simulations. In the third section, the circuit schema is given and Proteous™ simulations of microcontroller-based memristive chaotic circuit is made. The fourth section includes the experimental results. The paper is concluded with fifth section.

## 2. Memristor cubic modelling and memristor-based Chua's circuit

Memristor can be defined by two types of nonlinear constitutive parameters which are called memristance

\*corresponding author; e-mail: [syener@sakarya.edu.tr](mailto:syener@sakarya.edu.tr)

$M(q)$  and memductance  $W(\varphi)$ , respectively, and which are defined by:

$$M(q) = d\varphi(q)/dq, \quad (1)$$

$$W(\varphi) = dq(\varphi)/d\varphi. \quad (2)$$

The memristor designed in this work is a flux-controlled memristor, described by the Eq. (2). The relation between the terminal voltage  $v(t)$  and the terminal current  $i(t)$  of the memristor is obtained by:

$$i(t) = dq/dt = (dq/d\varphi)(d\varphi/dt) = (dq/d\varphi)v(t) = W(\varphi(t))v(t). \quad (3)$$

Nonlinear resistor in Chua's circuit is defined by Zhong with cubic nonlinearity. It has been revealed that all features of the circuit are captured correctly by this definition [19]. The  $q(t)$ - $\varphi(t)$  constitutive law of memristor with cubic nonlinearity is used for implementation of chaotic circuits [18]. Memristor-based Chua's circuit used in this study is shown in Fig. 1. The cubic polynomial definition of memristor is defined as follows:

$$q(\varphi) = \alpha\varphi + \beta\varphi^3. \quad (4)$$

Thus, the memductance function is given by:

$$W(\varphi) = dq/d\varphi = \alpha + 3\beta\varphi^2. \quad (5)$$

The following set of equations is written for the rescaled system.

$$\begin{aligned} d\varphi(t)/dt &= v_{C_1}(t)/\delta, \\ di_L(t)/dt &= v_{C_2}(t)/L, \\ dv_{C_1}(t)/dt &= 1/C_1((v_{C_2}(t) - v_{C_1}(t))/R) \\ &\quad - (\alpha + 3\beta\varphi^2(t))v_{C_1}(t), \\ dv_{C_2}(t)/dt &= 1/C_2((v_{C_1}(t) - v_{C_2}(t))/R) \\ &\quad - i_L(t). \end{aligned} \quad (6)$$

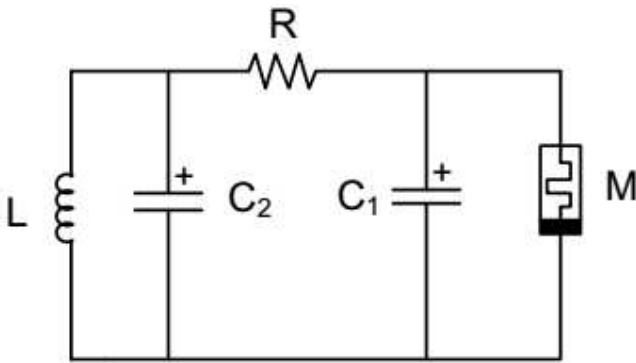


Fig. 1. M-B Chua's circuit.

Circuit parameters in Fig. 1 and cubic polynomial memristor parameters are selected as  $C_1 = 6.8$  nF,  $C_2 = 68$  nF,  $L = 18$  mH,  $R = 2.00$  k $\Omega$ . Then, scaling factors and initial conditions are  $\alpha = -0.601 \times 10^{-3}$ ,  $\beta = 21.2 \times 10^{-6}$ ,  $\varphi(0) = 0$ ,  $i_L(0) = 0$ ,  $v_{C_1}(0) = 0.01$ ,  $v_{C_2}(0) = 0.01$ . Time domain waveforms and chaotic phase portraits are obtained using MATLAB. However, due to space considerations only phase portraits are shown in Fig. 2.

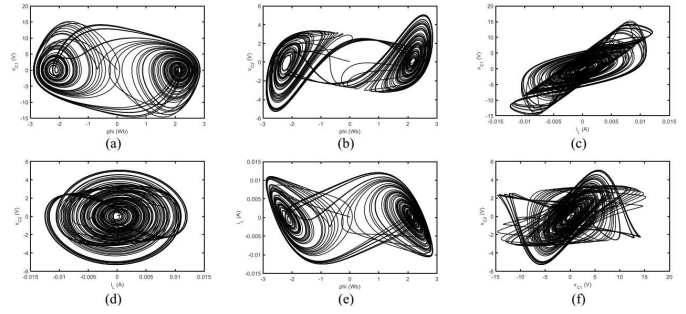


Fig. 2. 2D Phase portraits from the MATLAB simulation, (a)  $\varphi$ - $v_{C_1}$ , (b)  $\varphi$ - $v_{C_2}$ , (c)  $i_L$ - $v_{C_1}$ , (d)  $i_L$ - $v_{C_2}$ , (e)  $\varphi$ - $i_L$ , (f)  $v_{C_1}$ - $v_{C_2}$ .

### 3. Computer simulations of the chaotic memristor circuit

In this section, using Proteus<sup>TM</sup>, computer simulations of the microcontroller-based memristive chaotic circuit have been performed. Circuit schematic of the system is shown in Fig. 3. Two ADC0808 analog-to-digital converters are connected to digital output ports of Arduino Mega 2560 R3 microcontroller board. The outputs of ADC0808 DACs are connected to LM741 operational amplifiers to obtain the desired range of the output voltages. The system has also a switch, shown in Fig. 3, used for changing the desired outputs. The chaotic memristive circuit equations are solved using Range Kutta method in Arduino Mega 2560 R3 program. The Arduino Mega 2560 R3 is able to send only two of the state variables at once to the ADC0808 outputs. The program is written to be able to send two of the selected state variables, solved by Runge-Kutta method, to DAC outputs by pushing the switch. Resistance  $R$  is the tuning parameter for the system and it is very important for creation of a proper chaotic behavior. After simulations, it is seen that results are in good agreement with MATLAB results and that Proteus<sup>TM</sup> is able to simulate well the memristive chaotic circuit waveforms. Due to space considerations of the paper, only experimentally obtained signals are shown in the following section.

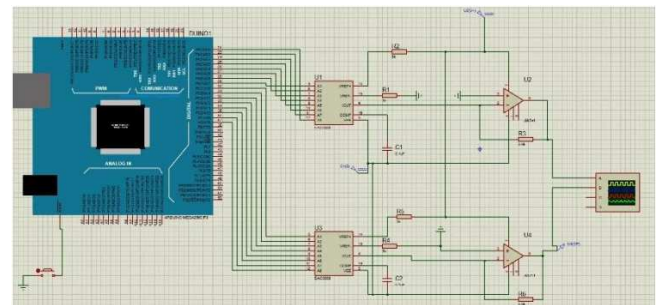


Fig. 3. Circuit schematic obtained using Proteus.

### 4. Experimental results

In this section experimental results of the microcontroller-based memristive chaotic circuit are

obtained and results are shown. A photograph of the circuit, used for the lab experiments, is shown in Fig. 4.

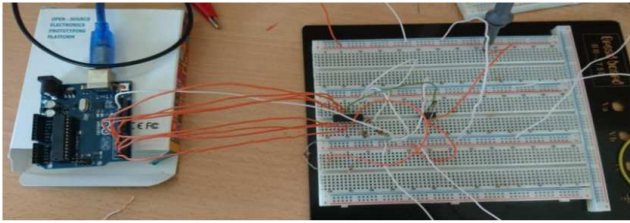


Fig. 4. The experimental circuit and measurement setup.

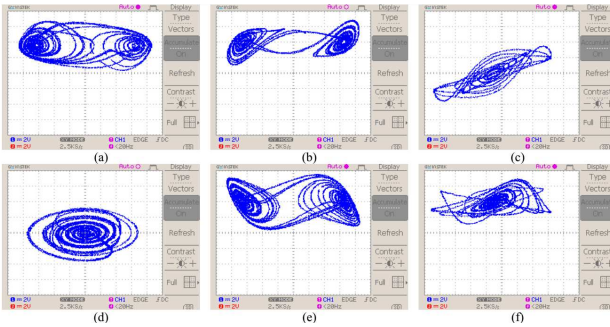


Fig. 5. Experimental results of 2D phase portraits, (a)  $\varphi-v_{C1}$ , (b)  $\varphi-v_{C2}$ , (c)  $i_L-v_{C1}$ , (d)  $i_L-v_{C2}$ , (e)  $\varphi-i_L$ , (f)  $v_{C1}-v_{C2}$ .

The Arduino Mega 2560 R3 microcontroller board is programmed and run to calculate the state variables numerically and to send the selected two of them to DAC outputs in real time. Resistance  $R$  shown in Fig. 1 is still used as the tuning parameter. It is defined similar to MATLAB and Proteus simulation. The waveforms are acquired by a 60 MHz digital oscilloscope, two waveforms simultaneously. Time domain characteristics and corresponding phase portrait waveforms are shown in Fig. 5. The microcontroller-based system is clearly able to demonstrate chaotic behaviour and is able to reproduce well the Matlab and Proteus simulations.

The hysteresis loop characteristics of the circuit is shown in Fig. 6. As can be seen, the time domain characteristics and phase portraits, obtained from the real circuit, are in very good agreement with corresponding MATLAB and Proteus simulation plots.

## 5. Conclusions

The design, as a proper practical implementation, representing the memristive behavior, is of great importance from the perspective of real-world circuit design, since no commercially available memristor exists yet. In this study, a microcontroller-based memristive chaotic circuit is presented. Beyond the prior works, our design is more practical and implementable using only an Arduino microcontroller board, two DACs and a switch. With

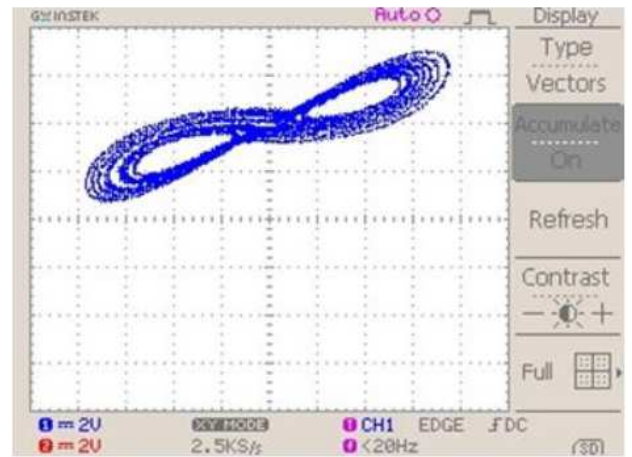


Fig. 6. Hysteresis loop characteristics of memristor.

its analog outputs, it is able to produce chaotic output signals and performs well. It allows easy examination of the memristive chaos, although it is made of off-the-shelf components. By proper modifications of its program and by adding potentiometers or a key-pad, the examination of the effect of the parameters on the chaos can be made easier. This design and its application in memristive chaotic circuits provide more practical perspective to memristor-based chaotic circuits. The operation speed of presented circuit is limited by the speed of the used microcontroller. However, the performance of the circuit is still better than those of the similar circuits, given in literature. In addition, it is easier to build and the circuit can be designed much quicker than an analog-based memristive circuit.

## References

- [1] L.O. Chua, *IEEE Trans. Circuit Theory.* **18**, 507 (1971).
- [2] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, *Nature* **453**, 80 (2008).
- [3] D. Karayel, V. Yegin, *Acta Phys. Pol. A* **130**, 272 (2016).
- [4] T. Yener, S.C. Okumus, S. Zeytin, *Acta Phys. Pol. A* **127**, 917 (2015).
- [5] C.Z. Zulkifli, H.N. Hassan, W. Ismail, S.N. Semunab, *Acta Phys. Pol. A* **128**, B-86 (2015).
- [6] R. Tuntas, *Acta Phys. Pol. A* **128**, B-78 (2015).
- [7] S. Benderli, T.A. Wey, *Electron. Lett.* **45**, 377 (2009).
- [8] Á. Rak, G. Cserey, *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **29**, 632 (2010).
- [9] S. Shin, K. Kim, S.M. Kang, *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **29**, 590 (2010).
- [10] D. Batas, H. Fiedler, *IEEE Trans. Nanotechnol.* **10**, 250 (2011).
- [11] Y. Chen, X. Wang, in: *2009 IEEE/ACM Int. Symp. Nanoscale Archit. NANOARCH*, 2009, p. 7.

- [12] Yu Zhang, X. Zhang, J. Yu, in: *2009 Int. Conf. Commun. Circuits Syst.*, 2009, p. 928.
- [13] R. Mutlu, E. Karakulak, in: *Natl. Conf. Electr. Electron. Comput. Eng.*, IEEE, 2012, p. 380.
- [14] Ş.Ç. Yener, H.H. Kuntman, *Radioengin.* **23**, 1140 (2014).
- [15] Ş.Ç. Yener, A. Uygur, H.H. Kuntman, *Analog Integr. Circuits Signal Process.* **89**, 719 (2016).
- [16] H. Kim, M.P. Sah, C. Yang, S. Cho, L.O. Chua, *IEEE Trans. Circuits Syst.* **59**, 2422 (2012).
- [17] S.C. Yener, R. Mutlu, T. Yener, H.H. Kuntman, in: *2017 Electr. Electron. Comput. Sci. Biomed. Eng. Meet.*, IEEE, 2017, p. 1.
- [18] B. Muthuswamy, *Int. J. Bifurc. Chaos.* **20**, 1335 (2010).
- [19] G.Q. Zhong, *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **41**, 934 (1994).