International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# Design and Implementation of a Controlled Interface Circuit for Autonomous Remote-Controlled Car

**Ahmed A. Al-Moadhen[1], Haider G. Kamil[2], Ali R. Khayeat[3]**

**Submitted**: 05/09/2022 **Accepted**: 20/12/2022

**Abstract**
This paper has designed and implemented an electronic circuit and algorithm for interfacing an autonomous remote-controlled car (ARCC) with the computer. This interfacing prepares the conditions to study and analyse the controlled object and endow it with the ability to be smarter and intelligent when performing its duties. The ARCC will be guided through the commands issued by the computer. It is manually controlled by the operator directing the car to the destination. The interface circuit consists of an Arduino microcontroller, switch, and connections. The designed circuit's validation has been tested through simulation and practically on the real system. The simulation test was done under a C# software environment, while the practical test was done on the real car. The experiments describe the car's movement from the initial position to the destination position. These experiments showed that the average system response time is: (346.4 ms) when the ARCC moves front, (360.9 ms) when the ARCC moves back, (1393.23ms) when the ARCC turns left and (935.77ms) when the ARCC turns right. A graphical user interface (GUI) has been designed and implemented to facilitate controlling activities related to ARCC. The GUI is a control panel used to set the parameters of the ARCC environment to prepare it to receive commands.

## 1. Introduction

The human dream is to design and implement a mobile-controlled object that can perform its tasks intelligently and automatically. The object with this capability can be assigned to do many applications such as glass cleaning, underwater investigation, aerospace explorations, and many others. Autonomy and diversity are fundamentals in the field of robotics systems. The former has a pivotal role in a system that interacts autonomously with its environment. While the latter plays a critical role in a system that deals with various external tasks, i.e., autonomous controlled mobile objects face various tasks due to a diversity of environment and human needs.

In this paper, we are interested in an autonomous robot which is represented as an autonomous remote-controlled car. This car will be called Autonomous Remote-Controlled Car (ARCC) throughout the paper. This ARCC can be used in many domains. The domain may be in the industry to carry processed materials, domestic service to support humans, and planetary exploration to navigate and discover distant planets. This idea will open the door to control other types of mobile objects such as flying objects, e.g., quadcopters or drones, and underwater objects such as autonomous underwater vehicles (AUVs).A key aspect of this paper is that it will open the door to endow the ARCC with several important techniques for moving, turning, planning, acting, obstacle detection, and obstacle avoidance and obstacle destruction. Controlling objects with little human involvement is a major area of interest within robotics.

However, a major problem with this kind of application is the lack of a general algorithm to run the interface circuit. Along with the good growth in electronics industries and software development, algorithms play an important role in managing system hardware. Therefore, this paper will fill this gap between the system software and hardware. This study investigates the usefulness of controlling a mobile object by collecting important statistics. The statistic is the system response time, illustrated in experiments and results (as seen in Section 5).

The outlined of the paper is as follows: Section 2 is reviewing the literature related to this study. Section 3 is explained the specification and design of the interface circuit. The necessary algorithm to run the proposed system is presented in Section 4. Section 5 is addressed how the information passes through the system. Then Section 6 is reported the experiments and results. The conclusion is presented in section 7.

## 2. Literature Review

The principle of controlling an object remotely has been investigated by a large and growing number of literatures. More recently, attention has focused on controlling, particularly robotics systems. A considerable amount of literature has been published on designing a framework to deal with moving and flying objects such as cars and quadrotors respectively (Huang, 2022; Manzoor & Wu, 2015). A study by (Simmons, 1994) involved layering

[1]*Department of Electrical and Electronics Engineering, College of Engineering - University of Kerbala, Karbala 56000, Iraq. ahmedh1333@uokerbala.edu.iq*
[2]*Department of Computer Engineering Techniques, AlSafwa University College, Karbala 56000, Iraq.*
[3]*Department of Computer Science, College of Computer Science and Information Technology, University of Kerbala, Karbala 56000, Iraq.*

reactive behaviours of the developed complex robot system into components which operate deliberately. The structured control method of reactive behaviours can lead to exceptional situations and the deliberative components can handle normal situations.

The researchers in (Bohren et al., 2011) explained that building a broad robotic platform with autonomous capability is a real objective for roboticists, however, the recent generation of autonomous systems is beyond shortly of this objective. They argued that a possible action toward wide ability is by creating a considerable set of explicit, practical robot behaviours that can achieve certain plans given a sufficiently defined, constrained application scope. Work on a formal framework of an autonomous vehicle as a combination of consistent control cycles, with a periodic sense, plan and act cycle was undertaken by (Py et al., 2010). The author in (Ambros-Ingerson & Steel, 1988) attempted to support replanning in dynamic environments where unexpected events can occur in addition to providing a framework with the ability to support Interleaving of Planning and Execution (IPE) to bring about the intended effects of failing actions.

In 2009, (Brenner & Nebel, 2009) integrated Planning, Execution, and Monitoring, leading to a new principled approach to Continual Planning. They proposed an algorithm which deliberately defers parts of the planning operation to final stages in an agent's plan, act, and monitor cycle and then automatically defines when to switch back to revise a part of the executed plan. The researchers in (Zhang, 2022) presented an AI based technology-assisted industrial for designing and transforming the system and gave the framework of intelligent industrial design. Then implementing the industrial outcome design and transformation system. In a recent survey by (Ingrand & Ghallab, 2017), numerous achievements in formatting the operations of deliberation related to the robotic system were suggested. The planned action is a prominent research point in robotic systems. This point is inspired by the essential abilities required by autonomous mobile robots encountering domains, missions, and interactions. Applications in space exploration, domestic service and personal robotic systems rely significantly on additional advancement and evolution of robot deliberation capabilities. Another survey by (Hawes, 2011) presented arguments for developing a rationale management framework for a smart system. Based on past studies, the survey determined the required components of such a frame, mainly concentrating on procedures that can develop goals and preferred goals to evolve the focus of goal-directed behaviour. The result presented in paper (McGann et al., 2008) showed that onboard planning and execution within its proposed framework can gracefully handle uncertainties in the subsea domain within the computational capacity available to their Autonomous Underwater Vehicles (AUV's).

An investigation by (Choi & Amir, 2009) suggested an algorithm that integrates logical and motion planners. Their planner is developed to deal with objects by robots. Earlier work used hierarchical high-level and low-level motion planner to solve this problem. They used manual encoding between both layers. This was one of the technical toughness of the problem. The survey's authors in (Wolfe et al., 2010), the authors proved that task-level planning can be successfully extended to the kinematic level to generate robust and high-quality plans. The authors in (Tu et al., 2022) designed and implemented an intelligent garbage bin to help people. This garbage bin supports humans to arrange and make domestic rubbish and control garbage surplus from influencing the living environment. The study in x proposed the development of a photovoltaic module simulator. The simulation can run under regular function according to the diverse electrical variables selected and indicates multiple benefits of being low price, small-sized, and easy to implement. The study in (Tang et al., 2012) proposed the development of a photovoltaic module simulator. The simulation can run under regular function according to diverse electrical parameters selected and indicates multiple benefits of being low price, small-sized, and easy to implement.

The survey in (Youssef et al., 2022) exhibited how sociable robots can affect humans' lives and established different capacities for a robot to interact socially. In addition, they highlighted the consequences of technology and the investigation of social robotics and handled the human side of the human-robot interaction. Finally, they supplied a multi viewpoint illustration of the future of social robotics. However, the (César-Tondreau et al., 2022) used a method to compare a traditional motion planner which uses the dynamic window approach (DWA) to specify travel costs based on the geography slope local to the robot's current position.

**Specifying and Designing the Interface Circuit**

To date, various methods and techniques have been developed and introduced to control mobile vehicles (as described in Section 2). Figure 1 illustrates the block diagram of the proposed interfacing circuit which is situated between the computer and the ARCC. This circuit is developed through this paper and becomes one of its main contributions (the other contributors will be presented in Section 4). The design of this circuit is based on using (Arduino) microcontroller (*Arduino*, n.d.), switches (IC 4066) (*IC Switch 4066*, n.d.). and the Radio Control Transmitters and Receiver (RC). Section 4 will describe the algorithm which controls this circuit. The Arduino type is UNO. Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is intended for anyone making interactive projects. The 4066 switch contains linearization circuitry which lowers the "on" resistance and increases the switch

linearity, more details of this switch can be found at (*IC Switch 4066*, n.d.).

The RC component consists of two joysticks, the one at the right is used for turning the ARCC left and right, while the one at the left is used for directing the ARCC front and back. The RC transmitting part sends data to the RC receiving part by generating a modulated radio frequency carrier while the receiver is tuned to detect the transmitter's carrier frequency (Beginners' Guide, n.d.).

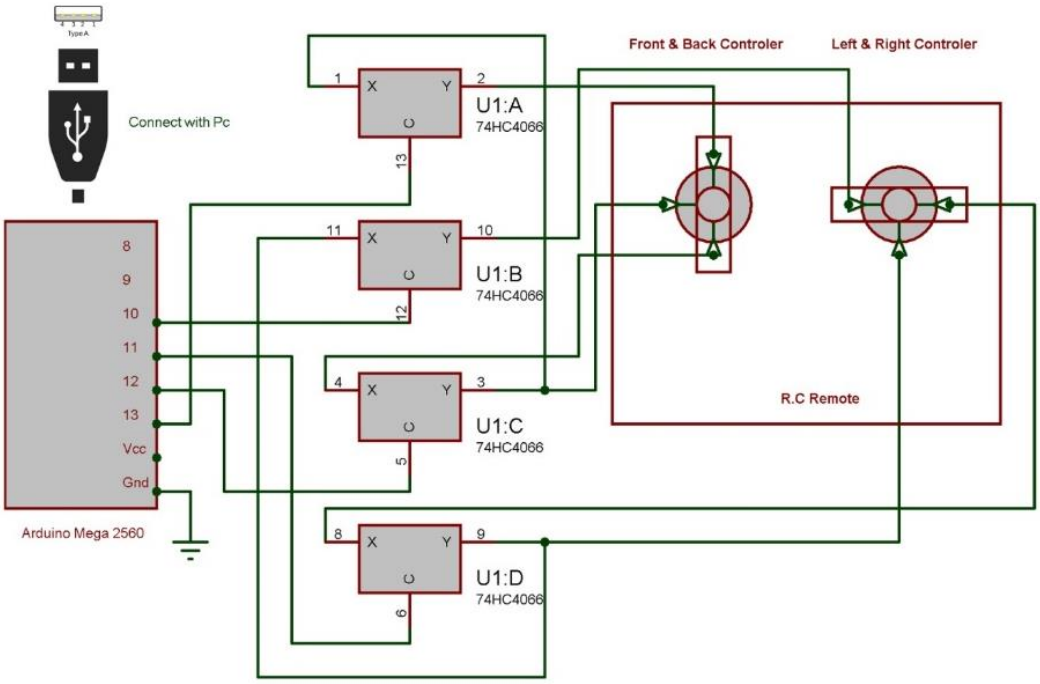


**Figure 1:** Block Diagram of Interfacing Circuit

## 3. Algorithm for Running the Interfacing Circuit

The circuit presented in Section 3 cannot be worked without the necessary software which runs the different parts of that circuit. This algorithm is prepared according to Figure 1 and synthesized using the necessary steps to control the ARCC. The algorithm passes through a sequence of stages which are (see Figure 2):

1. Setting up the ARCC Environment (GUI)
2. Generating environment Objects (Classes)
3. Loading Details (Images)
4. Transformation Handling (angle2vector, distance)
5. Drawing Canvas
6. Keyboard Handling
7. Click Handling to start the task
8. Timer handling to record task time.
9. Initialization (frame, …)
10. Registration of Handlers
11. Get Things Ready to Move

The first step of this algorithm describes the environmental setting to assign values for the environment dimensions (width and height). The setting also sets the environment time. In the second step, the algorithm defines classes for every object in the environment. This paper's main objects are: car, the interface circuit and computer. In the third step, the main details of the environment's objects are loaded into the control module in the computer. Step four deals with the information transformation from angle into vector. In step fifth, the environment canvas is drawn.

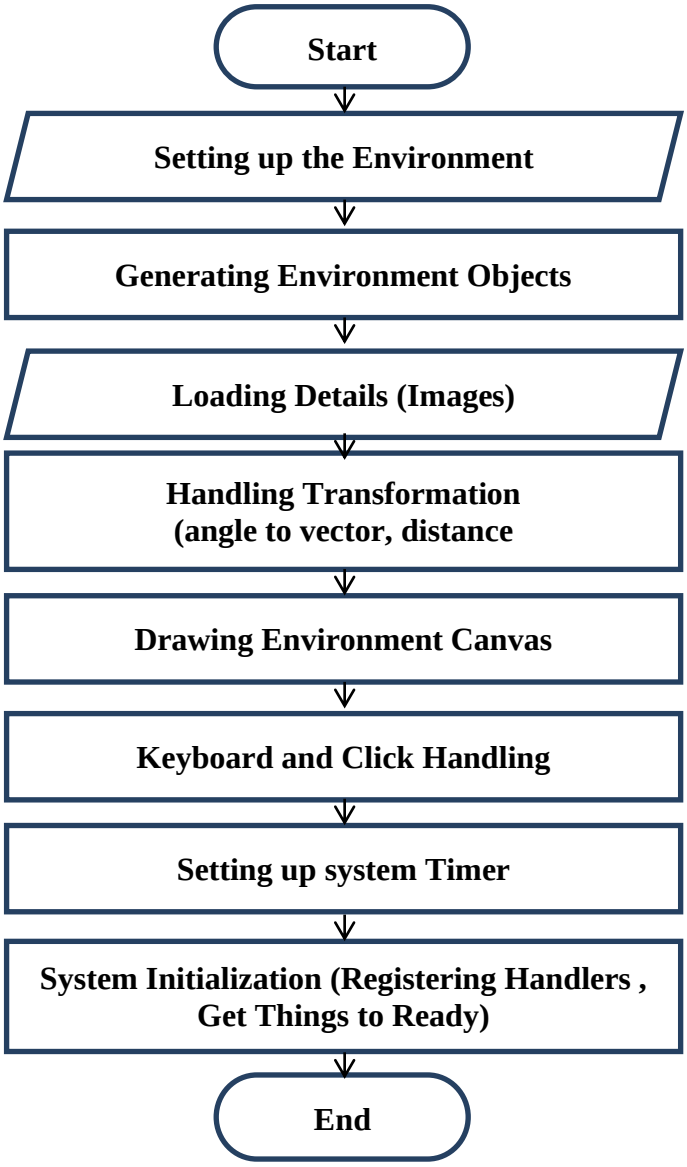


**Figure 2:** Control Flowchart

Steps sixth, seventh, and eighth are dealing with handling keyboard (left, right, up, and down), click, and timer activities. In step nine, the canvas created in step five will be initialized. Then the handlers will be registered in step ten. After all these steps, the system (ARCC) will be ready to receive the orders and perform them. In the initialization stage, the communication port between the computer and the circuit is identified, then the car angle and position are specified. The main algorithm role is initializing the interface circuit between the computer and ARCC.

## 4. The Information Flow Through Interface Circuit

The information inside the system presented in this paper passes through a sequence of steps starting from the system initial state of motor movement as explained in Figure 3. In the first, the system should be initialized with suitable parameter values. These parameters include ARCC position, orientation, Arduino pin activation, and serial port selection. Then the system is ready to receive an order. When the operator issues a command, then the interface circuit checks the commands database for the commands details such as: command type and parameters. Then the interface circuit loads the appropriate procedure for that command and orders the controller to drive the ARCC motors.

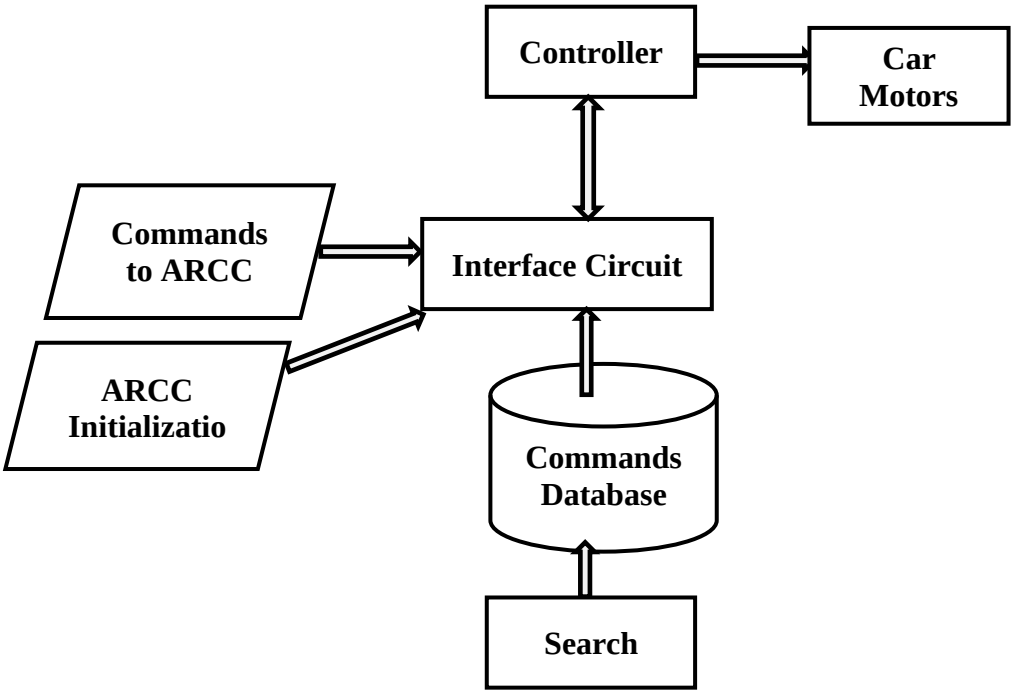


**Figure 3:** ARCC Information Flow

## 5. Experiments and Results

In this section, the ARCC is controlled by using the circuit in Figure 1 and driven by the algorithm in Figure 2. The main components (parts) of this experiment are:

1. Small car which is provided by two motors; one is used to move the car front and back while the other is used to direct the car left and right.
2. Arduino Mega 2560.
3. Computer with Windows 7 operating system.
4. Microsoft Visual Studio 13 – C# environment.
5. IC Switch 4066
6. Test board
7. Wire connections
8. Arduino Software

Figure 4 shows the physical interface circuit which is used to control the car by using a computer. The Arduino is loaded by a procedure (the necessary software) to be the bridge between the computer and ARCC. The procedure is described by the algorithm that explained in Section 4, and written in C# language to be more understandable by the Arduino. Figure 5 depicts the testing ARCC.

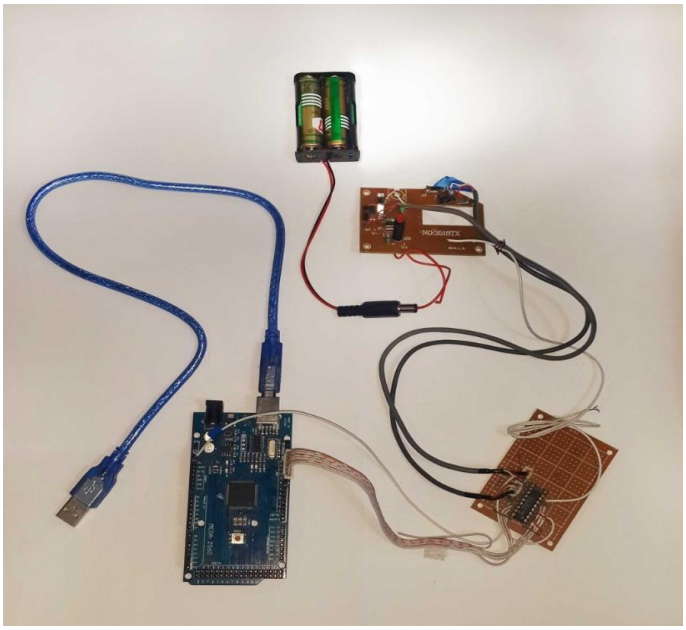

**Figure 4:** Interface Circuit

**Figure 5:** ARCC

ARCC Environment Setup

At the first, Figure 6 shows the necessary code that is used to setup the Arduino pins (10, 11, 12, 13) which are used to control ARCC movement and directions.

```
int data;
void setup(){
  Serial.begin(9600);
  pinMode(13,OUTPUT);pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);pinMode(10,OUTPUT);
  pinMode(9,OUTPUT);pinMode(8,OUTPUT);}
void loop(){
  if(Serial.available()){
    data = Serial.read();
    switch(data){
      case 'a':gitalWrite(13,HIGH);    break;
      case 'b':digitalWrite(13,LOW);   break;
      case 'c':digitalWrite(12,HIGH);  break;
      case 'd':digitalWrite(12,LOW);   break;
      case 'e':digitalWrite(11,HIGH);  break;
      case 'f':digitalWrite(11,LOW);   break;
      case 'g':digitalWrite(10,HIGH);  break;
      case 'h':digitalWrite(10,LOW);   break;
      case 'i':digitalWrite(9,HIGH);   break;
      case 'j':digitalWrite(9,LOW);    break;
      case 'k':digitalWrite(8,HIGH);   break;
      case 'l':digitalWrite(8,LOW);    break;
      }}}
```

**Figure 6:** Excerpt Code for Setting Up Arduino Pins

The letters (a, c, e, and g) are used to open the circuit ports while letters (b, d, f, and h) are used to close the circuit ports. The C# programming language under Visual Studio environment is used in this paper. Therefore, the following functions are used to control ARCC activities (opening port, move front, move back, turn left, and turn right).

1. serialPort1.Open()
2. public void go_front()
3. public void stop_front()
4. public void go_back()
5. public void stop_back()
6. public void go_right()
7. public void stop_right()
8. public void go_left()
9. public void stop_left()

The next step is to set the keyboard keys (up, down, left, and right). The following functions are used to do that setting.

1. Form1_KeyDown(object sender, KeyEventArgs key)
2. Form1_KeyUp(object sender, KeyEventArgs Up)

Figure 7 explains the GUI used to move the ARCC by keyboard.

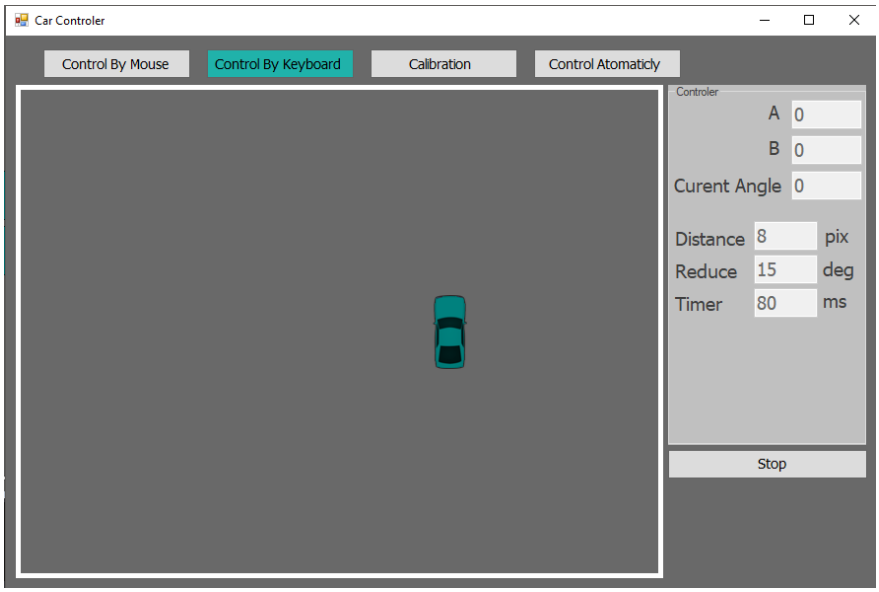


**Figure 7:** ARCC Controlling GUI

The Controller GUI

The graphical user interface (GUI) of this system is depicted in Figure 8. From this figure, the user can choose to control the car manually (by mouse or keyboard) or automatically. In case the user selects a manual control, the button (up, bottom, left, right) is used to guide the car. Controlling the ARCC automatically is out of the scope of this paper.

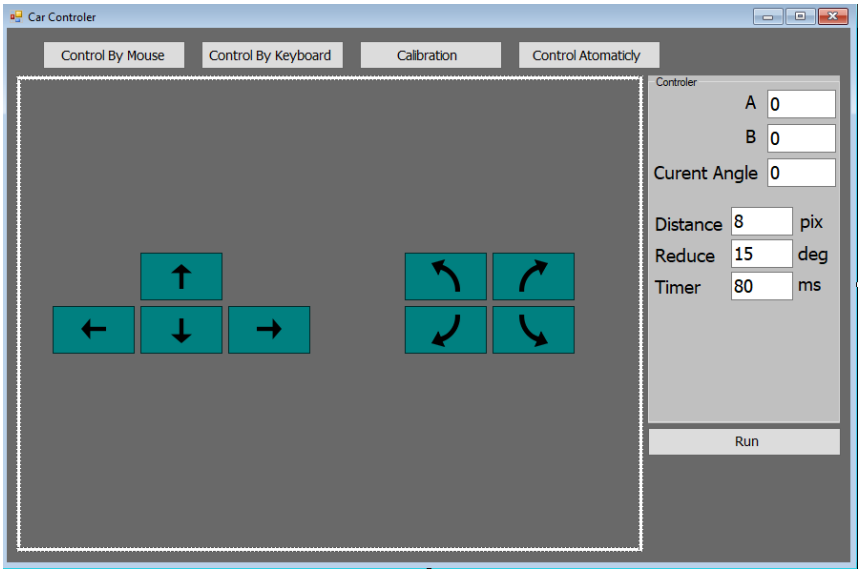


**Figure 8:** ARCC GUI

System Response Time

Response time is the number of milliseconds that the system takes from the moment an operator initiates an activity by pressing the direction arrow to move the ARCC until it reaches the destination. The basic model of ARCC response time and working time is outlined in Figure 9. While Figure 8 shows the detail model of system response time and working time. The response time is tested in four different movement scenarios. These scenarios are moving the ARCC front, back, left and right. The movement distance is 1 meter and the experiment is repeated for 30 times.

Figure 9 shows the response time calculation in four different situations (front, back, left, and right). The average response time is as follows: front = 346.4 ms, back = 360.9 ms, left = 1393.23 ms, and right = 935.77 ms. Then Figure 10, Figure 11, Figure 12, Figure 13, Figure 14 Figure 15 present the response time for these four movements. The overall results are used to calibrate the response time of the ARCC and make its movement smooth and controlled.

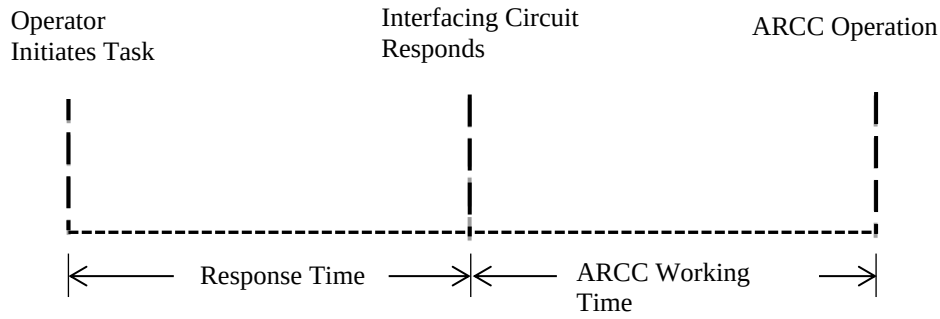


**Figure 1:** Basic Model of Response time and ARCC Working Time

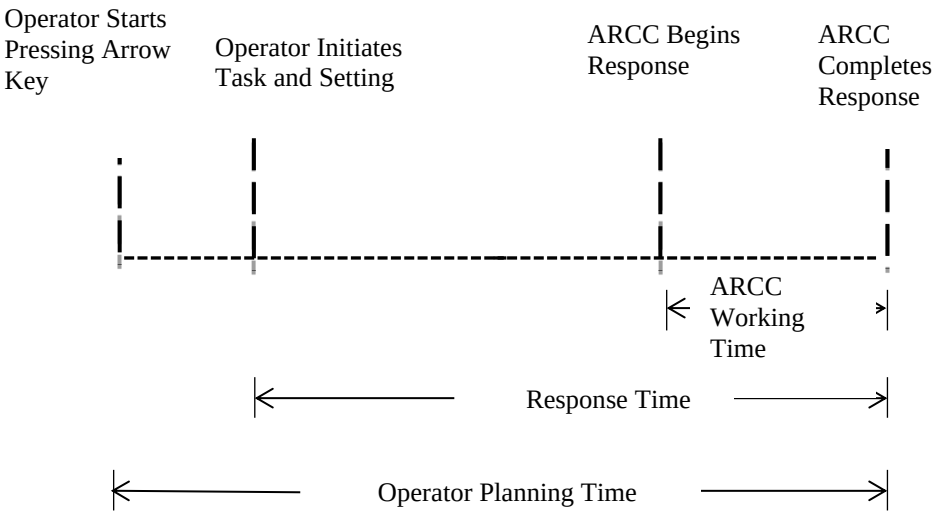


**Figure 2:** More Detailed Model of Response Time and ARCC Working Time

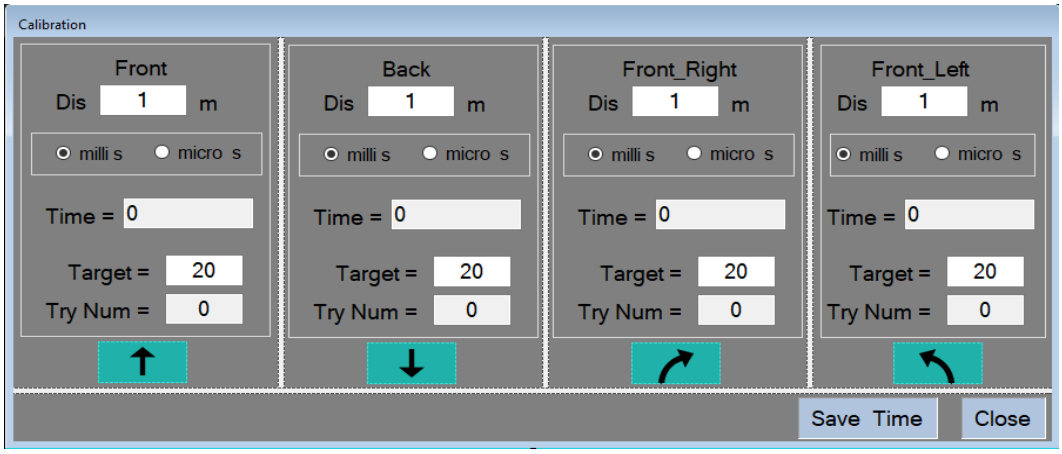


**Figure 11:** Response Time Calculation

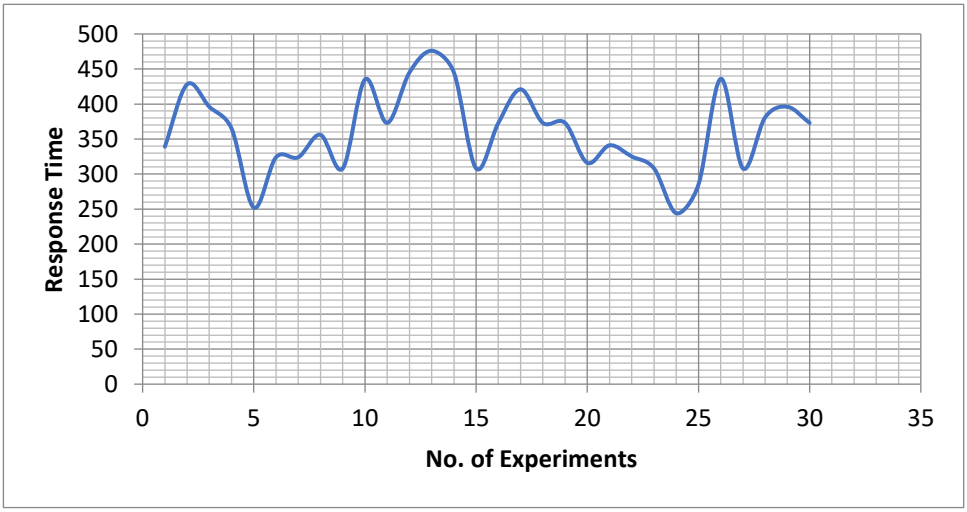


**Figure 12:** Response Time for Front Movement

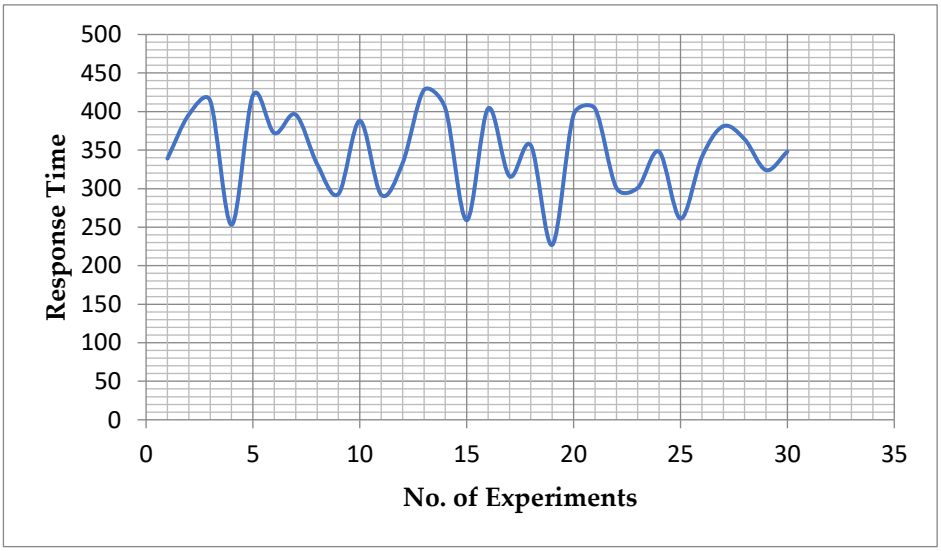


**Figure 13:** Response Time for Back Movement

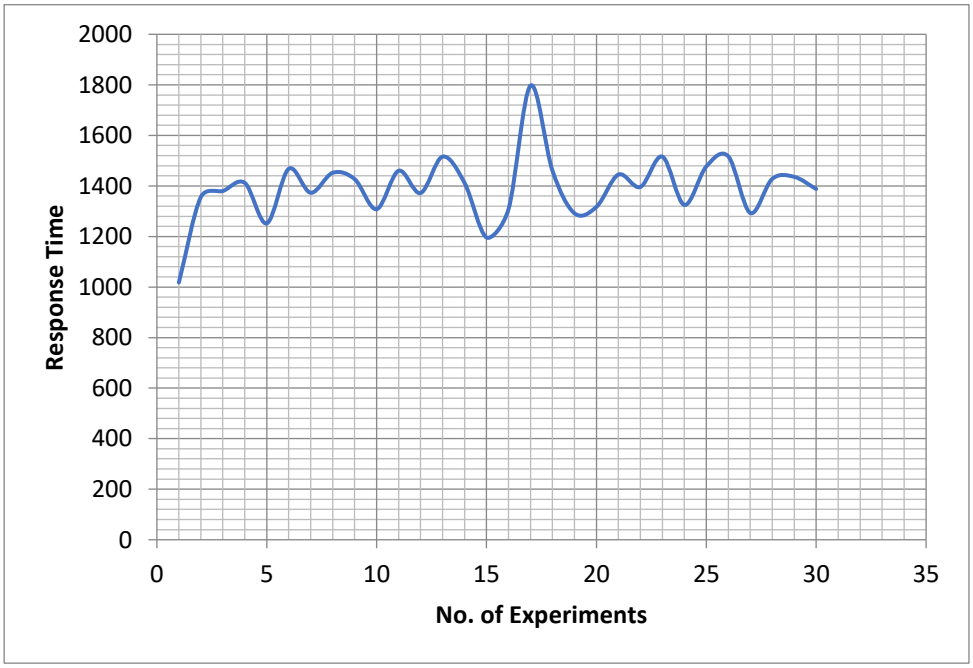


**Figure 14:** Response Time for Left Movement

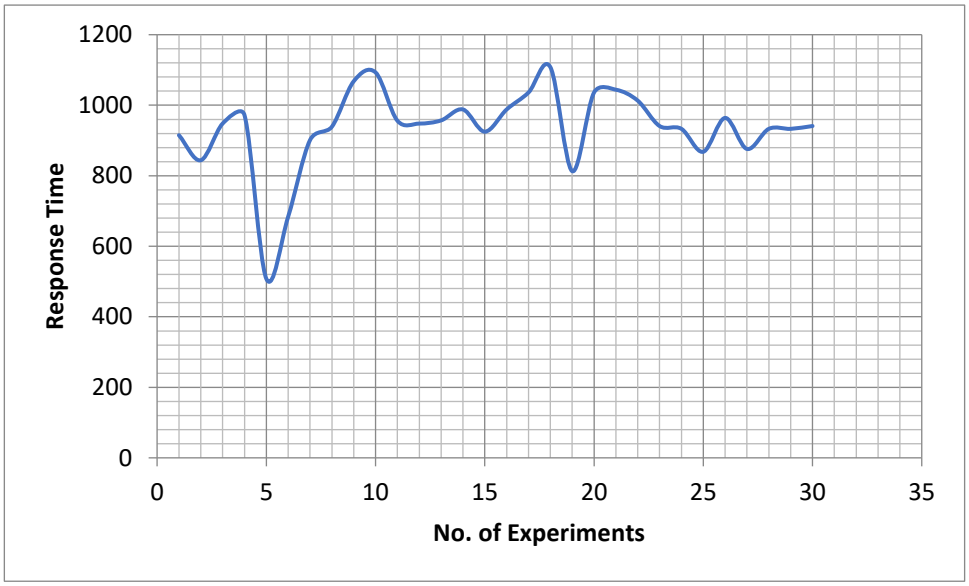


**Figure 15:** Response Time for Right Movement

## 6. Conclusions

The main goal of the current study was to determine the importance of interfacing mobile objects with a computer. Therefore, the aim was to design and implement the interface circuit between the autonomous remote-controlled car and a computer to control the ARCC activities. This study also presents how: the circuit connections are linked; the algorithm is run through the connections and the information is flowed in the system.

The results of this investigation show that it is possible to analyse and study external mobile objects' performance by interfacing them to a computer. When the data is collected, another software application can analyse and conclude from these results. Therefore, this paper proposed a new way to control the remote-controlled car and one can start from this point to build a more complex controlled system by adding more facilities to this system. More research is needed to upgrade the system to be more reliable and dependable by incorporating more advanced techniques of monitoring, observation, goal reasoning, and learning.

## References

[1] Ambros-Ingerson, J., & Steel, S. (1988). Integrating Planning, Execution and Monitoring. In J. Allen, J. Hendler, & A. Tate (Eds.), *Proceedings of the Seventh National Conference on Artificial Intelligence AAAI88* (Vol. 1, pp. 83–88). Kaufmann.

[2] *Arduino*. (n.d.). Retrieved July 31, 2022, from https://www.arduino.cc/

[3] *Beginners' Guide*. (n.d.). Retrieved July 31, 2022, from https://rcplanes.online/guide1.htm

[4] Bohren, J., Rusu, R. B., Jones, E. G., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mösenlechner, L., Meeussen, W., & Holzer, S. (2011). Towards autonomous robotic butlers: leasons learned with PR2. *2011 IEEE International Conference on Robotics and Automation*, 5568–5575. https://doi.org/10.1109/ICRA.2011.5980058

[5] Brenner, M., & Nebel, B. (2009). Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems*, *19*(3), 297–331. https://doi.org/10.1007/s10458-009-9081-1

[6] César-Tondreau, B., Warnell, G., Kochersberger, K., & Waytowich, N. R. (2022). Towards Fully Autonomous Negative Obstacle Traversal via Imitation Learning Based Control. *Robotics*, *11*(4), 67. https://doi.org/10.3390/robotics11040067

[7] Choi, J., & Amir, E. (2009). Combining Planning and Motion Planning. *2009 IEEE International Conference on Robotics and Automation*, 238–244. https://doi.org/doi: 10.1109/ROBOT.2009.5152872

[8] Hawes, N. (2011). A survey of motivation frameworks for intelligent systems. *Artificial Intelligence*, *175*(5–6), 1020–1036. https://doi.org/10.1016/j.artint.2011.02.002

[9] Huang, X. (2022). Personalized Travel Route Recommendation Model of Intelligent Service Robot Using Deep Learning in Big Data Environment. *Journal of Robotics*, *2022*, 7778592. https://doi.org/10.1155/2022/7778592

[10] *IC Switch 4066*. (n.d.). Retrieved July 31, 2022, from http://www.electroschematics.com/wp-content/uploads/2011/04/74VHC4066.pdf

[11] Ingrand, F., & Ghallab, M. (2017). Deliberation for autonomous robots: A survey. *Artificial Intelligence*, *247*, 10–44.

[12] Manzoor, M. F., & Wu, Q. (2015). Control and Obstacle Avoidance of Wheeled Mobile Robot. *Proceedings - 7th International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN 2015*, *0*(2), 235–240. https://doi.org/10.1109/CICSyN.2015.48

[13] McGann, C., Py, F., Rajan, K., Thomas, H., Henthorn, R., & McEwen, R. (2008). A Deliberative Architecture for AUV Control. *IEEE International Conference on Robotics and Automation*, 1049–1054.

[14] Py, F., Rajan, K., & McGann, C. (2010). A systematic agent framework for situated autonomous systems. *AAMAS '10: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, 583–590.

[15] Simmons, R. G. (1994). Structured control for autonomous robots. *IEEE Transactions on Robotics and*

*Automation*, *10*(1), 34–43. https://doi.org/10.1109/70.285583

[16] Tang, K.-H., Chao, K.-H., Chao, Y.-W., & Chen, J.-P. (2012). Design and Implementation of a Simulator for Photovoltaic Modules. *International Journal of Photoenergy*, *2012*, 368931. https://doi.org/10.1155/2012/368931

[17] Tu, L.-F., Peng, Q., & Deng, R. (2022). Design and Implementation of Intelligent Control Garbage Bin. *Journal of Electrical and Computer Engineering*, *2022*, 7306548. https://doi.org/10.1155/2022/7306548

[18] Wolfe, J., Marthi, B., & Russell, S. (2010). Combined Task and Motion Planning for Mobile Manipulation. *Proceedings of the International Conference on Automated Planning and Scheduling*, 254–257.

[19] Youssef, K., Said, S., Alkork, S., & Beyrouthy, T. (2022). A Survey on Recent Advances in Social Robotics. *Robotics*, *11*(4), 75. https://doi.org/10.3390/robotics11040075

[20] Zhang, F. (2022). Design and Implementation of Industrial Design and Transformation System Based on Artificial Intelligence Technology. *Mathematical Problems in Engineering*, *2022*, 9342691. https://doi.org/10.1155/2022/9342691