Can Bus Communication Demonstration Tool for Education

Ahmed Al-Areqi and Tamás Szakács

May 19, 2021

CAN BUS COMMUNICATION DEMONSTRATION TOOL FOR EDUCATION

Ahmed Al-Areqi
Institute of Mechatronics and Vehicle Technology
Bánki Donát Faculty of Mechanical and Safety Engineering
Óbuda University
Budapest, Hungary
Ahmed773403@stud.uni-obuda.hu

Tamás Szakács
Institute of Mechatronics and Vehicle Technology
Bánki Donát Faculty of Mechanical and Safety Engineering
Óbuda University
Budapest, Hungary
szakacs.tamas@bgk.uni-obuda.hu

***Abstract*—Intelligent vehicles is a fast developing area. For the internal combustion engines are 150 years of dead-end road of vehicle technology – which 150 years are still counting – the greatest development in vehicle industry can now be seen in self driving technology. The demand on intelligent solutions is the fastest area in this industry. Intelligent solutions are requiring reliable communication channels. The most widely used vehicle communication is the Controller Area Network (CAN). The CAN (bus) is a communication protocol between the different control units in the car, where the controllers (computers) can exchange information directly (without a direct link unit between them) and it is a protocol based on the principle of messages. Each message has its own structure consisting of Message-ID, message start, message body, and end of the message. CAN messages do not contain the address of the sender or the addressee, but the content of the message itself contains an additional identifier for the information transmitted like RPM, temperature, etc. In this paper, we will propose an implementation of an education tool to demonstrate the principle of a CAN bus protocol. The demonstration tool comes with sensors, actuators, LCD, LED, input, and output functions. This tool will simplify the idea of how the CAN bus sends messages in the data bus with the priority of the order. Also, it will support the theoretical knowledge of the CAN bus by a practical demonstration of it.**

***Index Terms*—Controller Area Network (CAN), Electronic Control Units (ECU), MCP2515, Esp8266, Servo Motor, Liquid Crystal Display(LCD), Arduino Uno, Demonstration Tool**

I. Introduction

In self-driving technology, an embedded system is implemented based on a high-performance CPU, low-power consumption, and high-speed communication protocol. Field Programmable Gate Array (FPGA) is considered the most functional CPU with high computational capabilities and low power consumption. It is the technology of the autonomous driving vehicle due to its high computational capability [1]. On the other side of self-driving technology, the CAN bus is the most powerful and efficient communication protocol for autonomous vehicle systems [2]. Imagine that the car is like a human body. The Control Area Network (CAN bus) is the nervous system, which facilitating communication among all parts of the body. Similarly, nodes or electronic control units (ECU) are connected via the can bus, which acts as a central networking system. In an automotive, can be the airbags, the engine control unit, the audio system or any part of the vehicle. A modern car can have up to 70 ECUs [3], this is where the car standard comes in handy as it allows ECU to communicate with each other without complex dedicated wiring in between. In turn, this allows for several features to be added via software alone like an electronic gear box control, so can allow any ECU to communicate with entire system without causing an overload to the controller computer. The CAN bus chip, which we have chosen is MCP2515, this chip is a stand-alone Controller Area Network and this designed to make the applications simple, so this application need interfacing with the Controller Area Network. This chip consist of the control logic and registers, The CAN module and The SPI protocol block. The MCP2515 CAN Bus Controller is an easy Module, and this type give assistance to CAN bus version 2.0B, this can be hired with communication, which around 1Mbps, so sometimes a double CAN bus module is helpful to design a complete communication system [4]. The aim of this paper is to simplify the idea of the communication system inside a vehicle by implementing an education tool. This tool will demonstrate how different components contact each other by CAN bus protocol. The paper will be presented as follows, Section-II describes the components which are used in the system and the comparison of different types of communication protocol. Section-III presents the hardware design of the tool with the diagram. Section-IV shows the flowcharts of the tool with an explanation of the principle of work. Section V concludes the paper with the summarization of the work. Section VI finally states some suggestions for developing the tool.

II. Methodology

We have used in our project these components: Esp8266 MCU, Arduino UNO, Arduino Nano MCP 2515, CAN BUS, Servo Motor and LCD. And we have talked about SPI BUS and I2C BUS.All these components which are mentioned will be place in one demonstration tools. This will make an educational environment that contains both hardware and software. The students can learn how to program or reprogram the CAN massages in ECU and how to add or delete CAN's nodes. This demonstration tools will assist students to develop of computational thinking which is very important in programming. In this paper [5], the author used many visual programming applications and educational environments to test how is important to integrate interactive education tools to develop the computational thinking for students in Hungary.

A. Esp8266

The Esp8266 is a open source platform, low-power, and highly integrated Wi-Fi micro-controller designed by Espressif Systems. It is very low price compared to its performance and capabilities and it is available in several types that differ in its features and the number of its terminals. It operates at wide temperature from (-40 C) to (+25C). Esp8266 specification: Operating voltage is 3.3V, Current consumption is from 10uA to 170mA, Flash memory attached 16MB max (512K normal), Processor Speed is from 80MHZ to 160MHZ, RAM is 32K+80K, GPIOs is 17 (multiplexed with other functions, Analog to digital 1 input with 1024 step (10bit) resolution, Wi-Fi is 802.11 support. See Fig. 1. Esp8266 is an Arduino like device. It's core component is ESP8266. Its pins could be Programmable. Its power could be can delivered through micro-usb port. What makes this micro-controller preferable is that It can be worked through multiple programming environments and different programming languages [6].

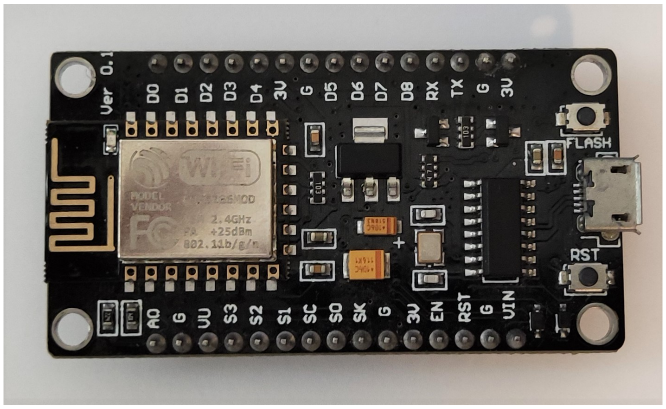
Ver 0.1
D0 D1 D2 D3 D4 3V G D5 D6 D7 D8 RX TX G 3V
FLASH
RST
A0 G VU S3 S2 S1 SC S0 SK G 3V EN RST G VIN
Wi-Fi

Fig. 1. Esp8266 micro-controller.

B. Arduino UNO

Arduino is considered as one of the most popular microcontrollers which have an open-source platform for building and developing electronics project. Arduino Uno is categorized as a part of the Arduino series which supported with ATmage 328 microcontroller. This board also is suitable for education and training purposes because it has IDE (integrated development environment) and it is easy to write and edit the software through a USB cable. Arduino board can be implemented in communication projects in both wire-connection by (SPI, CAN, I2C) or wireless- connection by (Wi-Fi, Bluetooth, FM). Also, this board can be integrated into controlling processes in homes, industries, or hospitals because it can deal with different analog signals such as (heat, light, sound, pressure, humidity) and to control the output for instance (motor, light, sound). Arduino Uno has 14 digital pins and 6 of them are supported PWM signals as an output. It also has 6 analog pins that are used to receive an analog signal with a range of voltage (0 - 5) DC Volt. In addition, these analog pins can be used as digital pins. This board is integrated with 16 MHz a ceramic oscillator. Finally, this board can be powered through a USB port or power port with a range of DC voltage (0 - 18) Volt [7]. See Fig. 2 .

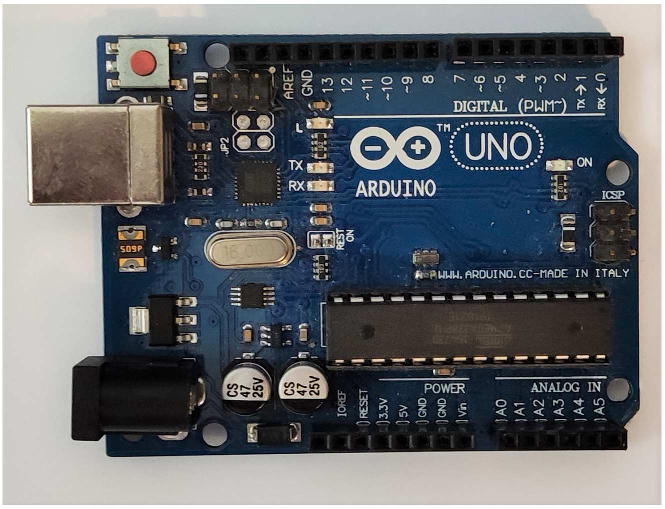
AREF GND 13 12 ~11 ~10 ~9 8 7 ~6 ~5 4 ~3 2 TX→1 RX←0
DIGITAL (PWM~)
UNO
ARDUINO
ON
ICSP
TX
RX
L
WWW.ARDUINO.CC-MADE IN ITALY
POWER
IOREF RESET 3.3V 5V GND GND Vin
ANALOG IN
A0 A1 A2 A3 A4 A5

Fig. 2. Arduino UNO Micro-controller.

C. MCP2515

MCP2515 is a stand-alone controller area network chip. It has features: bus speed until 1Mb/s, bus length around 1000 meter, Standard and remote frames and extended data, one-shot mode, interruption output, SPI interface, and 5V operation. This chip has a high-speed CAN transceiver. The protocol connection between the can module and the micro-controller unite is through SPI protocol. So, it is easy to interface with any micro-controller having an SPI interface. It has characteristic of connection speed of Up to 115200 UART baud rate (default 9600), Up to 1Mb/s CAN-BUS baud rate, TX and RX led indicator, 4pin Grove connector, 3.3V working voltage, Easy-to-use Arduino library, and small size [8].See Fig. 3.

D. SPI BUS

Serial Peripheral Interface is a synchronous data protocol that controllers use to communicate with one or more peripherals or with another controller with high-speed data transmission over short distances. One of the main advantages of SPI is that it provides higher data transfer rate than UART and I2C and is used when a high-speed serial connection is needed. Data could be sent to more than one receiver , but it suffers from having more parties than UART and I2C [9]. See Fig. 4

E. CAN BUS

In the CAN bus, a CPU decides what the incoming message means and what messages will be transferred Sensors,

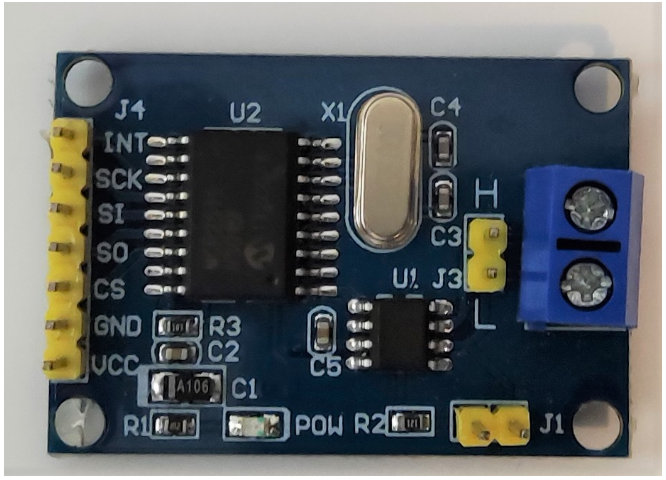

Fig. 3. MCP 2515 CAN BUS module.

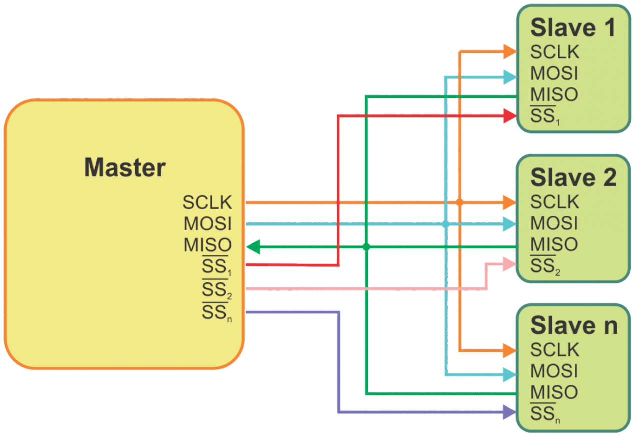

Fig. 4. SPI BUS diagram [10]

controllers, and motors connecting to the CPU The CAN controller receives the bits via the carrier and converts them to the processing unit when the message is completed. The CPU also sends messages to the protocol controller to publish them when the carrier is empty. The transmission and reception device converts data from the CAN Bus level to the level of understanding of the protocol controller upon receipt and converts the data from the protocol controller level to the CAN Bus level at the time of transmission Each node is able to send or receive but not at the same time, each message carries an identifier consisting of either 11 bits or 29 bits, the priority of the message is defined according to this identifier. CAN bus has only two wires which are High and Low for bi-directional information exchange in all the system components as shown above. See Fig. 5. Typically, the communication speed for CAN ranges from 50 Kbps to 1Mbps and the distance can range from 40 meters at 1Mbps to 1000 meters at 50kpbs [11].

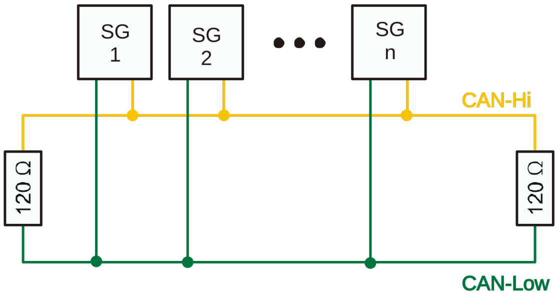

Fig. 5. CAN bus electrical sample topology [12].

F. I2C BUS

High-speed and bi-directional serial communication system for connection with terminals as in the case of SPI and UART. This protocol is slower than SPI, but it uses fewer lines for communication, so it is possible to choose between one of the following four speeds: Standard - 100 kbps Fast - 400 kbps Double speed - 1Mbps High speed - 3.33 Mbps.The main feature that distinguishes I2C from UART and SPI is that the I2C bus can support multiple master and sub slave devices on the same bus as the ability to support multiple devices in one bus allows us to reduce the number of external pins on the controller, which reduces the cost and size of the device. All connected devices can operate either master or slave. In addition, these devices can switch between these two modes. Also, each connected device has its own unique address and its predefined address, and therefore the main device can determine the sub device that will contact it [13].See Fig. 6.

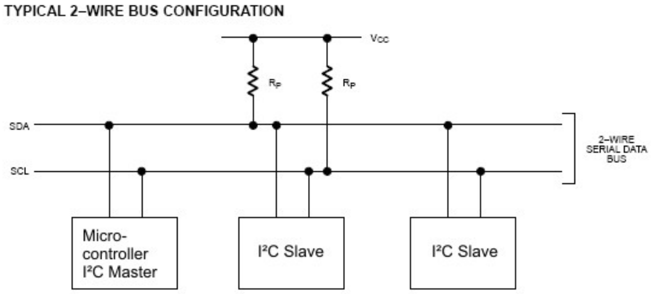

Fig. 6. I2C bus electrical sample topology [14]

G. Servo Motor

Servo motor is a close loop control circuit of dc motor control. It is a DC motor equipped with an electronic circuit to control the direction and rotation of the motor shaft, and also equipped with a set of gears. It has features of fast response to reach a velocity of the rated value as soon as the engine is connected to electricity and stops as soon as the power is disconnected. The relationship between voltage and speed is a linear relationship which means as the voltage increase the speed of the motor increase. This motor circuit is usually controlled by an operational amplifier [15]. See Fig. 7

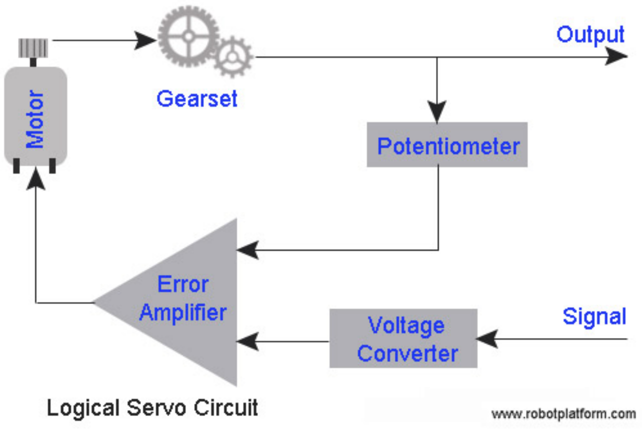

Fig. 7. Circuit diagram of servo motor [16].

H. LCD

Liquid Crystal Display is the output part of the system where the parameters of the controllable parts are displayed. The LCD screen needs 6 digital pins from the Arduino as digital outputs. It also requires 5V power and a GND ground outlet. R / W and VSS are connected to the ground port. As for VCC, it is connected to the 5-volt port, and RS is connected it to port number 12 on the Arduino. See Fig. 8

Fig. 8. LCD display.

I. Comparison of CAN over SPI and I2C

Since we have already learnt how to use SPI with Arduino and I2C with Arduino, let us compare the features of SPI and I2C with CAN controller chip with the MCP2515 CAN module Bus [17] . See TABLE. I

TABLE I
COMPARISON OF CAN OVER SPI I2C

Parameter	SPI	I2C	CAN
Speed	3 Mbps to 10 Mbps	Stander:100Kbps Fast:400Kbps V-Fast:3.4Mbps	10KBps to 1MBps upon length of wire
Type	Synchronous	Synchronous	Asynchronous
Number of wires	MISO(n),MOSI(n), SCLK(n),SS(n)	SDA SCL	CAN-H CAN-L
Duplex	Full Duplex	Half Duplex	Half Duplex

III. SYSTEM ASSEMBLY METHOD

In the system, we implement a small prototype for the vehicle information system which includes the following: input (Digital, Analog), output (Digital, Analog), and interface by LCD. The system includes 4 Nodes and each node has different characteristics. Node1 has the interface part of the system LCD and the digital output LEDs connect to Arduino Nano with the MCP2515 CAN module Bus. The node1 is set as receiver. Node2 is set as a sender and has three switches connect to Arduino UNO with the MCP2515 CAN module Bus. Each switch, that is a digital input unit, sends a different message to the CAN bus and not received any message. Node3 contains the analog input unit potentiometer connects to Arduino Nano with the MCP2515 CAN module Bus. The node 3 set as sender and it sends data to the CAN bus. Finally, node 4 is set as a receiver and it has the responsibility to control the servo- motor by Pulse Width Modulation (PWM) technique which is analog write. The servomotor connects to the ESP8266 micro-controller chip with the MCP2515 CAN module Bus. See Fig. 9

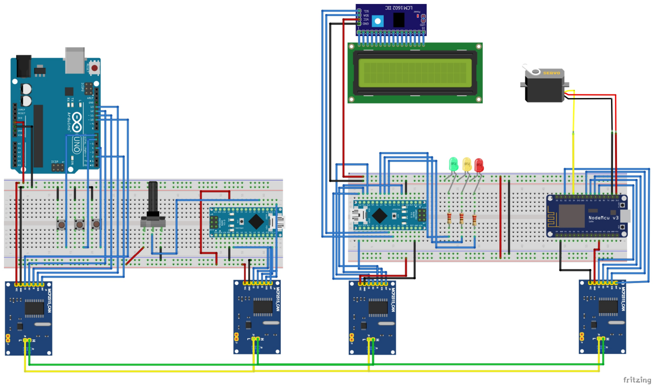

Fig. 9. Circuit diagram of the system .

IV. EXPERIMENT WITH HARDWARE

The system is a demonstration prototype for educational propose. We implement the components of a complete system such as (Digital Input, Digital Output, Analog Input, Analog Output, interface) to make the idea of the prototype clear for the student who will use it. At the moment of turning on the system, all the control units (nodes) supplied by 5v Dc source. Nodes, that are connected to CAN-BUS, start to send or receive the message to the CAN bus. Node1 starts to receive the data from the CAN bus. It is implemented to receive data only from node2 in the address of (0x036) and from node3 in the address of (0x0F6). The address (0x036) holds the data of the control unit of the servo motor in node4. Also, the address (0x0F6) holds the data of the digital states for LEDs that connect to node1. The node1 starts to receive the data form address (0x036) and display the servo position in the LCD display. Also, it receives the data form address (0x0F6) to control the LEDs state. On the other hand, node2 is implemented as the sender. It reads data from the potentiometer (control unit) by Analog to Digital Converter (ADC) and changes data to CAN message then send it to CAN bus with (0x036) address. Node3 is implemented as a sender, but in the digital not in the analog. Node3 reads the switches state and changes data to CAN message then send it to CAN bus with (0x0F6) address. Finally, node4 receives all the messages in the CAN bus and only accept data from (0x036), which holds the data of the control position, and translates the message by Digital to Analog in Pulse Width Modulation signal to the servo motor. See Fig. 14

- Node1 represents algorithm of stating the position of the servo motor on LCD and controlling the LEDs based on CAN's received signal. See Fig. 10
- Node2 represents algorithm of reading the position of servo motor from potentiometer and sending the value through CAN message. See Fig. 11
- Node3 represents algorithm of reading a digital signal from push buttons and sending their signals through CAN message. See Fig. 12
- Node4 represents algorithm of controlling the servo motor based on CAN's received signal. See Fig. 13

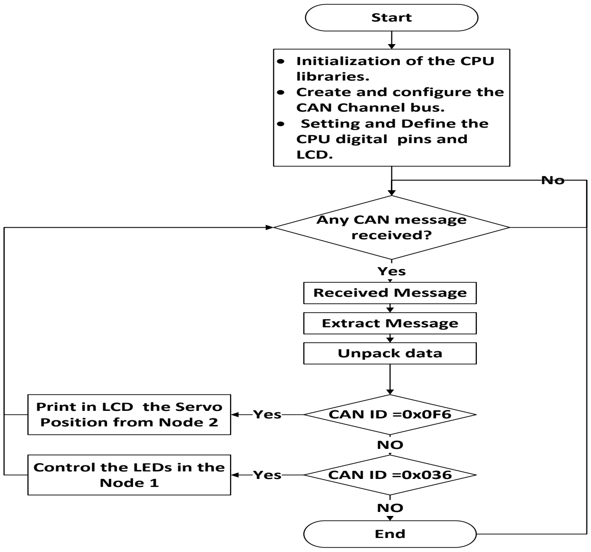
Start
• Initialization of the CPU libraries.
• Create and configure the CAN Channel bus.
• Setting and Define the CPU digital pins and LCD.
No
Any CAN message received?
Yes
Received Message
Extract Message
Unpack data
Print in LCD the Servo Position from Node 2
Yes
CAN ID =0x0F6
NO
Control the LEDs in the Node 1
Yes
CAN ID =0x036
NO
End

Fig. 10. Flow-Chart of node1

V. Conclusion

In conclusion, the CAN bus is the most commonly used in modern technology. It is the fastest and the most efficient protocol compares to other communication protocols such as (SPI, I2C, Serial). Also, it can be implemented with multiple micro-controllers or different voltages levels. Our demonstration tool will help the student to get a clear idea about how CAN bus works, the control system components (input, output, control, interface), building a network of many nodes, and dealing with multiple micro-controllers.

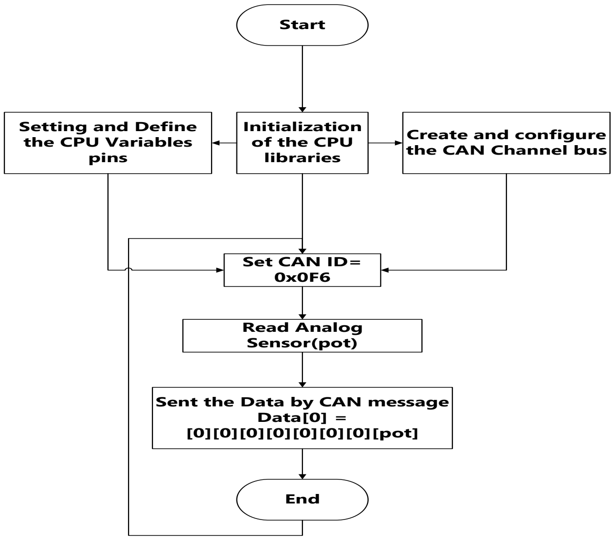
Start
Setting and Define the CPU Variables pins
Initialization of the CPU libraries
Create and configure the CAN Channel bus
Set CAN ID= 0x0F6
Read Analog Sensor(pot)
Sent the Data by CAN message Data[0] = [0][0][0][0][0][0][0][pot]
End

Fig. 11. Flow-Chart of node2

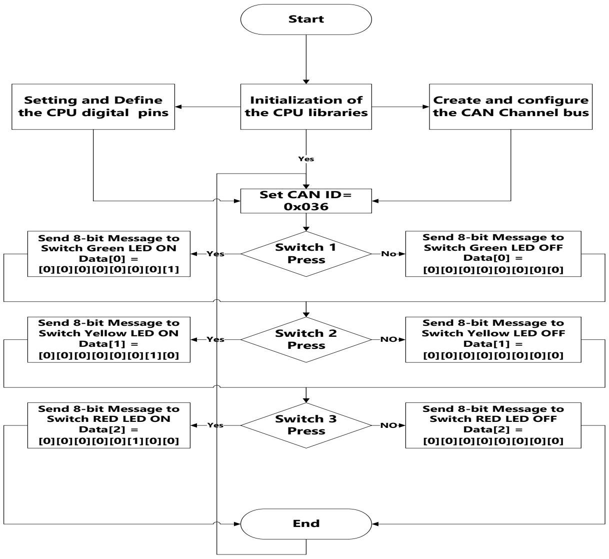
Start
Setting and Define the CPU digital pins
Initialization of the CPU libraries
Create and configure the CAN Channel bus
Yes
Set CAN ID= 0x036
Send 8-bit Message to Switch Green LED ON Data[0] = [0][0][0][0][0][0][0][1]
Yes
Switch 1 Press
No
Send 8-bit Message to Switch Green LED OFF Data[0] = [0][0][0][0][0][0][0][0]
Send 8-bit Message to Switch Yellow LED ON Data[1] = [0][0][0][0][0][0][1][0]
Yes
Switch 2 Press
NO
Send 8-bit Message to Switch Yellow LED OFF Data[1] = [0][0][0][0][0][0][0][0]
Send 8-bit Message to Switch RED LED ON Data[2] = [0][0][0][0][0][1][0][0]
Yes
Switch 3 Press
NO
Send 8-bit Message to Switch RED LED OFF Data[2] = [0][0][0][0][0][0][0][0]
End

Fig. 12. Flow-Chart of node3

VI. Future work

The future work is to implement this demonstration tool in a real car and see the response. Also, we will build a MATLAB software for the system to analysis the stability and the efficiency of the system and we will connect the system of the car to the Cloud and GPRS system.

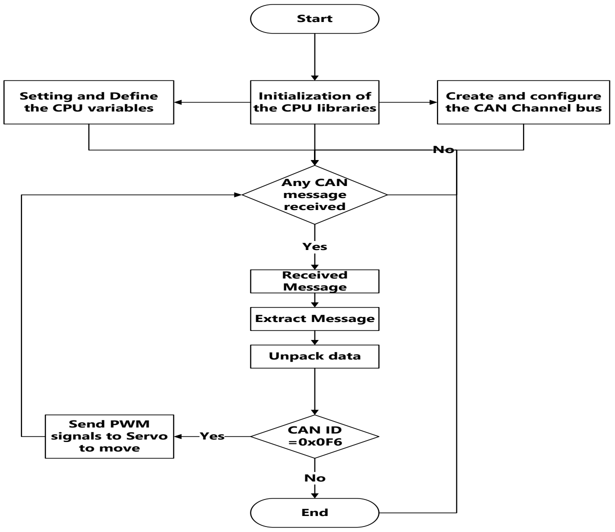
Start
Setting and Define the CPU variables
Initialization of the CPU libraries
Create and configure the CAN Channel bus
No
Any CAN message received
Yes
Received Message
Extract Message
Unpack data
Send PWM signals to Servo to move
Yes
CAN ID =0x0F6
No
End

Fig. 13. Flow-Chart of node4

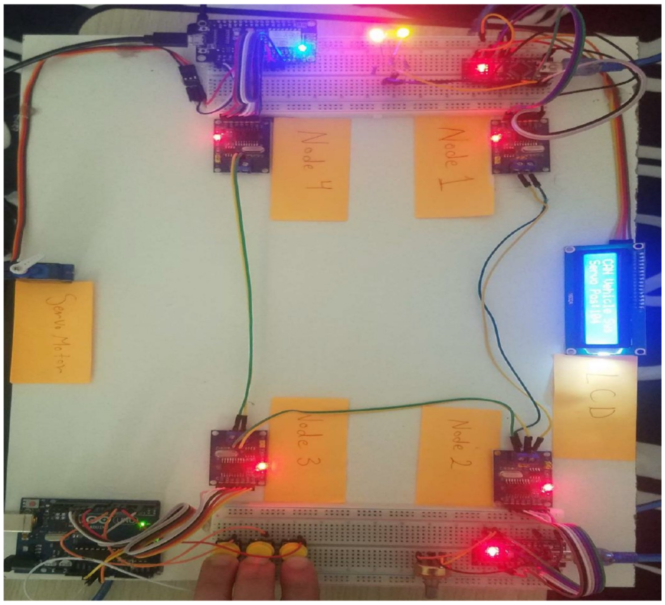

Fig. 14. Hardware design of the system.

References

[1] Reda, A., Bouzid, A. and Vásárhelyi, J., 2020. Model Predictive Control for Automated Vehicle Steering. Acta Polytechnica Hungarica, 17(7), pp.163-182.

[2] Elshaer, A., Elrakaiby, M. and Harb, M., 2018. Autonomous Car Implementation Based on CAN Bus Protocol for IoT Applications. 2018 13th International Conference on Computer Engineering and Systems (ICCES).

[3] ”CAN Bus Explained - A Simple Intro (2021)”, CSS Electronics, 2021. [Online]. Available: https://www.csselectronics.com/screen/page/simple-intro-to-can-bus/language/en.

[4] Stand-Alone CAN Controller with SPI Interface, 1st ed. Microchip Technology Inc., 2021.

[5] ”Placing Event-Action-based Visual Programming in the Process of Computer Science Education”, Acta Polytechnica Hungarica, vol. 16, no. 2, 2019. Available: 10.12700/aph.16.2.2019.2.3 [Accessed 10 February 2021].

[6] Y. S. Parihar, ”Internet of Things and Nodemcu,” Journal of Emerging Technologies and Innovative Research , vol. 6, no. 6, p. 1085, 2019 June.

[7] H. R. Hatem, ”control of pulse width modulation on direction and speed of DC motor using arduino,” Journal of Engineering and Applied Science , vol. 14, no. 5, pp. 1493-1497, March 2019.

[8] D. S. A. T. Harshal Hemane1, ”Interfacing CAN Protocol With PIC Microcontroller,” International Research Journal of Engineering and Technology (IRJET) , vol. 5, no. 9, pp. 402-405, Sep 2018.

[9] Interfacing SPIEEPROM with PIC16F, 1st ed. pantech solutions.

[10] En.wikipedia.org. 2021. Serial Peripheral Interface - Wikipedia. [online] Available at: ¡https://en.wikipedia.org/wiki/Serial-Peripheral-Interface¿ [Accessed 6 April 2021].

[11] D. D. V. N. Mr. Sushil K. Jadhav1, ”IIOT AND CAN BASED INDUSTRIAL PARAMETERS MONITORING AND CONTROL,” International Research Journal of Engineering and Technology (IRJET) , vol. 06, no. 01, pp. 756 - 760, Jan 2019.

[12] Commons.wikimedia.org. 2021. File:CAN-Bus Elektrische Zweidrahtleitung.svg - Wikimedia Commons. [online] Available at: ¡https://commons.wikimedia.org/wiki/File:CAN-Bus-Elektrische-Zweidrahtleitung.svg¿ [Accessed 6 April 2021].

[13] N. Semiconductors, ”2C-bus specification and user manual,” NXP Semiconductors, 4 April 2014.

[14] Avrhelp.mcselec.com. 2021. BASCOM-AVR. [online] Available at: ¡https://avrhelp.mcselec.com/index.html?using-the-i2c-protocol.html¿.

[15] S. C. Santosh Mohan Rajkumar, ”Development of Embedded Speed Control System for DC Servo Motor using Wireless Communication,” National Institute of Technology, Silchar, December 2019.

[16] Robotplatform.com. 2021. Robot Platform — Knowledge — How Servo Works?. ¡http://www.robotplatform.com/knowledge/servo/what-makes-a-servo.html¿

[17] ”Electronic Communication Protocols Basics and Types with Functionality”, ElProCus - Electronic Projects for Engineering Students. [Online]. Available: https://www.elprocus.com/communication-protocols/.