Title: Arduino UNO R4 WiFi CAN Bus protocol design tutorial	**Course**: Learn Arduino UNO R4 WiFi Free Online Course
Website: https://pro.tips	**Date**: 03/29/2025
Course designer: Pro.Tips Website	

1. Document overview

The purpose of this document is to demonstrate designing CAN bus protocol from requirements that can be easily implemented on any platform and any programming language, for this example case we are going to use Arduino UNO R4 WiFi.

The tutorial will follow the following steps:

1. Requirements gathering
2. Requirements analysis
3. CAN protocol design

2. Requirements

For this project we are going to use a Digital IO Board device that communicates with Main Controller over CAN communication bus.

Below is the list of system requirements:

1. The Digital IO Board shall read 5 digital inputs and control setting & clearing of 5 digital outputs.
2. The Digital IO Board shall send inputs status to main controller at minimum of 100Hz frequency.
3. The Digital IO Board shall receive outputs status from the Main Controller and set/clear the 5 digital outputs according to status received.
4. The Digital IO Board shall maintain communication heartbeat with Main Controller at minimum of 4Hz frequency.
5. The Digital IO Board shall clear all outputs if communication is lost with Main Controller.

3. Requirements analysis

From requirements we understand that the system is comprised of two devices: Digital IO Board and Main Controller.

For our system we are going to use Arduino UNO R4 WiFi for Digital IO Device and a desktop computer with PCAN-USB behaving as Main Controller Device.

The Arduino UNO R4 WiFi will read digital pins D0 to D4 as inputs and make use of digital pins D5 to D9 as digital outputs.

We will pick CAN communication as connection between Arduino UNO R4 WiFi and Main Controller devices to communicate IO status between each other while maintaining continues connection alive between them.

4. CAN Protocol design

This section describes CAN messages meeting the requirements described from the requirements section.

4.1 CAN 29 Bit Extended ID Definition

8 bits (D) Device #

8 bits (M) Message #

13 (X) Unused bits

D	D	D	D	D	D	D	D	M	M	M	M	M	M	M	M	X	X	X	X	X	X	X	X	X	X	X	X	X
29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

4.1.1 Device

The following are list of device #'s:

0x00 Main Controller (Desktop)

0x01 Digital IO Device (Arduino UNO R4 WiFi)

4.1.2 Message

This refers to message # associated to specific CAN packet.

Message #	**Definition**
1	Heartbeat
2	Read digital inputs status
3	Set digital outputs status

The messages are described in detail in section 4.2 below.

4.2 CAN Messages

4.2.1 Heartbeat message

Message #: 0x01

Source: Main Controller (Desktop)

The Main Controller (Desktop) sends heartbeat message to Digital IO Device (Arduino) every 250ms. This message shall be used by Digital IO Device (Arduino) to determine communication alive status with Main Controller.

Byte	Bits	Data item description
1-8		Unused

4.2.2 Read Digital Inputs Status

Message #: 0x02

Source: Digital IO Device (Arduino)

The Digital IO Device (Arduino) transmits digital inputs status to Main Controller (Desktop).

Byte	Bits	Data item description
1	0-4	Inputs status BIT0: Arduino D0 input status BIT1: Arduino D1 input status BIT2: Arduino D2 input status BIT3: Arduino D3 input status BIT4: Arduino D4 input status
1	5-7	Unused
2-8		Unused

4.2.2 Set Digital Outputs Status

Message #: 0x03

Source: Main Controller (Desktop)

The Main Controller (Desktop) transmits digital outputs status to Digital IO Device (Arduino).

Byte	Bits	Data item description
1	0-4	Outputs status BIT0: Arduino D5 output status BIT1: Arduino D6 output status BIT2: Arduino D7 output status BIT3: Arduino D8 output status BIT4: Arduino D9 output status
1	5-7	Unused
2-8		Unused