



IT 

## **Modifica “Name” di tag RFID tramite comunicazione seriale pc-plc Wago**

I.	Introduzione .....	1
II.	Cablaggio .....	3
III.	Immagine di processo.....	5
IV.	Protocollo di comunicazione del reader (RAQC) .....	7
V.	Configurazione plc con software WAGO-I/O-CHECK.....	9
VI.	Configurazione pc.....	11
VII.	Interfaccia Gimatic.....	12

EN 

## **Modify “Name” field of RFID tag using serial communication between pc-Wago plc**

I.	Introduction.....	18
II.	Wiring .....	20
III.	Process image .....	22
IV.	Reader’s communication protocol (RAQC).....	24
V.	Plc configuration using software WAGO-I/O-CHECK.....	26
VI.	PC configuration .....	28
VII.	Gimatic interface .....	29

**Version: A**

**Last update: 13/12/2022**

## I. Introduzione

In elettronica e telecomunicazioni, l'identificazione a radiofrequenza (in inglese Radio-Frequency IDentification, acronimo RFID) è una tecnologia di riconoscimento, validazione e/o memorizzazione automatica di informazioni a breve distanza. Essa si basa sulla memorizzazione di dati in particolari dispositivi elettronici passivi chiamati tag, capaci di rispondere a chiamate di prossimità da parte di dispositivi attivi, sia fissi che portatili, chiamati reader o lettori. L'identificazione e lo scambio di informazioni tra questi due elementi avvengono a radiofrequenza. Nello specifico, un sistema RFID è solitamente costituito da tre elementi fondamentali:

- uno o più tag RFID (detto anche transponder);
- un apparecchio di lettura e/o scrittura (reader);
- un sistema informativo di gestione dei dati per il trasferimento dei dati da e verso i lettori.

L'elemento principale che caratterizza un sistema RFID è il tag RFID, il quale è costituito da un microchip che contiene dati in una memoria (tra cui un numero univoco universale scritto nel silicio e non modificabile), un'antenna ed un supporto fisico che tiene insieme il chip e l'antenna chiamato "substrato". Il suo funzionamento è basato sul principio fisico dell'induzione elettromagnetica: l'antenna del tag riceve dal reader un segnale elettrico che viene trasformato in energia elettrica per alimentare il microchip ed inviare la risposta al comando ricevuto.

Gimatic fornisce soluzioni per la movimentazione automatica dei componenti. Per fare questo, solitamente, un robot è utilizzato in combinazione con diversi EOATs (End Of Arm Tools), ciascuno dedicato ad una specifica operazione. In un'applicazione simile, il polso del robot può essere dotato di un'unità di lettura (reader) e di qualsiasi EOAT equipaggiato con un tag RFID. La memoria del tag contiene:

- dati generici: nome e descrizione dell'EOAT, codice identificativo, massa ed ingombri
- proprietà di massa e geometriche: coordinate del centro di massa, momento d'inerzia
- proprietà geometriche: parametri di calibrazione
- distinta: lista di tool disponibili con descrizione, edizione e quantità

In questa applicazione sono stati utilizzati i seguenti prodotti Gimatic:

CODICE PRODOTTO	DESCRIZIONE	LINK AL SITO + MANUALE UTENTE	IMMAGINE
RAQC	reader RFID	<a href="https://shop.gimatic.com/it/raqc">https://shop.gimatic.com/it/raqc</a>	
RBQC	tag RFID	<a href="https://shop.gimatic.com/it/rbqc">https://shop.gimatic.com/it/rbqc</a>	
RQCBOX	Interfaccia di comunicazione con RAQC	<a href="https://shop.gimatic.com/it/rqcbox">https://shop.gimatic.com/it/rqcbox</a>	

Al posto dei prodotti sopra mostrati, inoltre, è possibile utilizzare anche prodotti compatibili come ad esempio

- reader: RRAQC, CRAQC, RAQC-C e versioni con I/O di tipo npn come RAQCN, RRAQCN, CRAQCN
- tag: RRBQC, CRBQC

Ciascun reader dovrà essere opportunamente abbinato al suo tag in base alle forme ed ingombri meccanici. Per maggiori informazioni di utilizzo e di compatibilità si rimanda alla documentazione tecnica (IST-RQC, IST-KIT RFID, IST-RQCBOX e RFID-TOOLS AND MANUALS) scaricabile dallo shop e dal link <https://shop.gimatic.com/Product/GetExtraFile?id=2636> .

La presente applicazione vuole essere una dimostrazione di fattibilità tecnica in modo da proporre una possibile soluzione al problema legato al riconoscimento dei tag. A causa del limitato numero di pin presenti sui reader RAQC (pari ad 8), è possibile distinguere fino a 255 dispositivi differenti assegnando a ciascuno nel relativo campo "ID" (IDentificatore utente) un numero progressivo da 1 a 255. Nei seguenti capitoli verranno quindi presentate l'architettura hardware e l'interfaccia software realizzate nella presente dimostrazione (il codice sorgente è disponibile su richiesta per approfondimenti o per ulteriori espansioni).

Implementando una comunicazione seriale con il reader RAQC (utilizzando per comodità di interfaccia una RQCBOX, ma non sarebbe indispensabile), è possibile assegnare un nome univoco a ciascun RBQC scrivendolo nella memoria interna al microchip. Questa operazione consente di oltrepassare il limite dei 255 dispositivi identificabili poiché il reader può leggere e scrivere nella memoria del tag che ha capacità di circa 4kB.

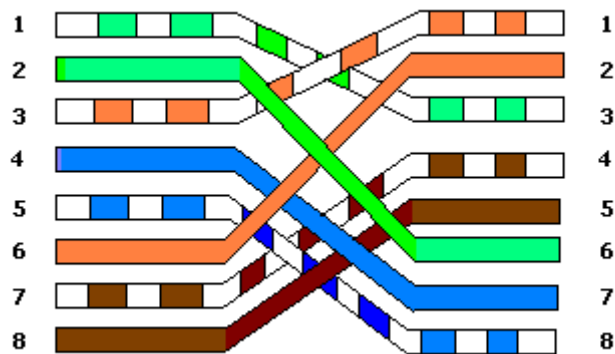
Per l'implementazione della comunicazione seriale, si è deciso di utilizzare un plc Wago che comunica con il controllore master (in questo caso un PC) utilizzando il protocollo Modbus TCP. Tramite l'apposita interfaccia realizzata su PC, è possibile verificare la presenza del tag e leggerne oppure scriverne il campo Name.

## II. Cablaggio

La configurazione minima dei moduli Wago utilizzati per il corretto funzionamento è:

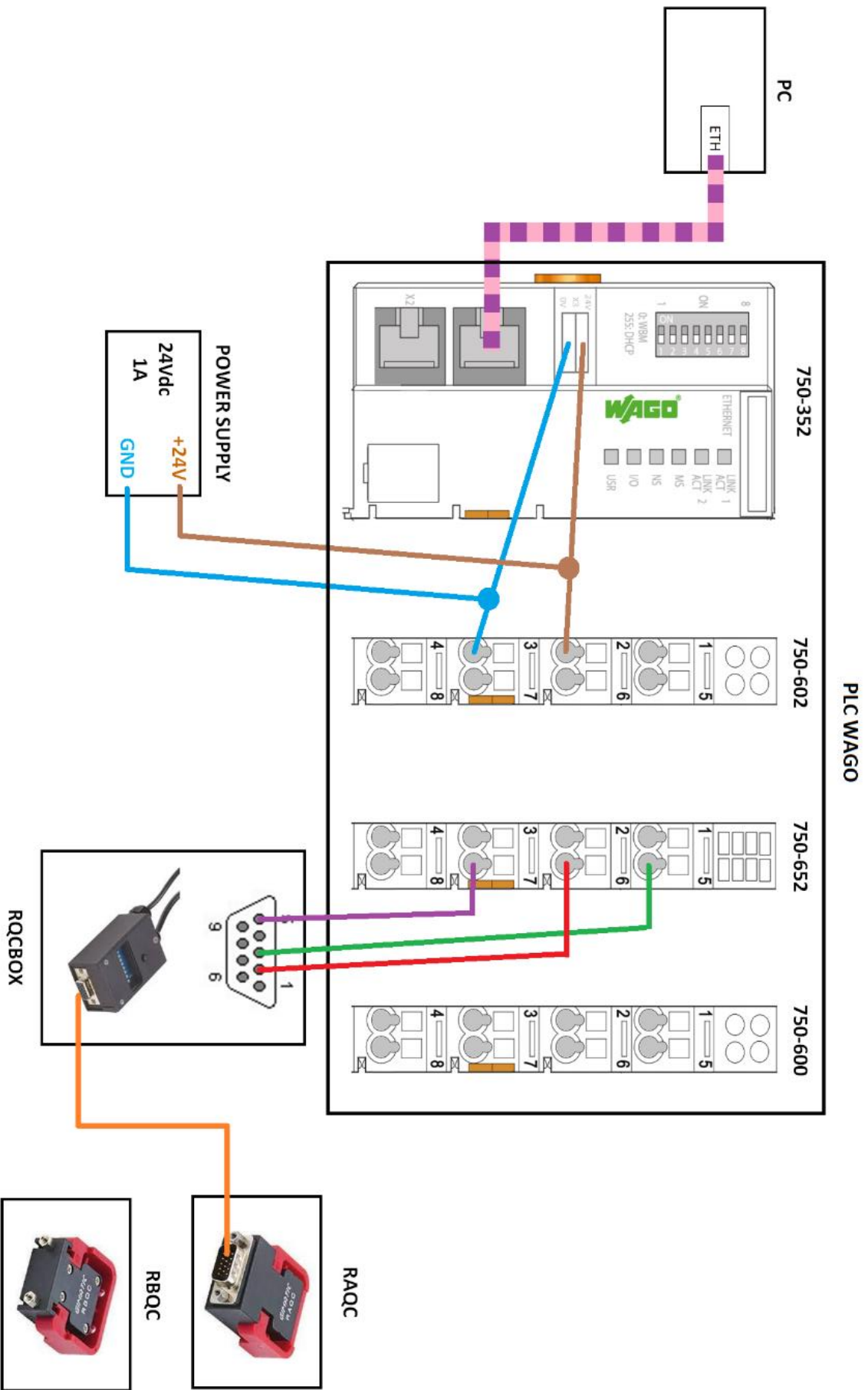
- 1- Testata Wago Fieldbus Ethernet (cod. 750-352)
- 2- Modulo Wago alimentazione (cod. 750-602)
- 3- Modulo Wago interfaccia seriale RS-232 (cod. 750-652)
- 4- Modulo Wago terminatore (cod. 750-600)

Il collegamento tra plc e pc deve essere fatto con cavo Ethernet RJ45 incrociato (null modem) come mostrato in figura sottostante:



Per alimentare il plc, è necessario utilizzare un alimentatore stabilizzato a 24Vdc 1A.

L'ordinamento ed il cablaggio dei moduli va effettuato nel seguente modo:



### III. Immagine di processo

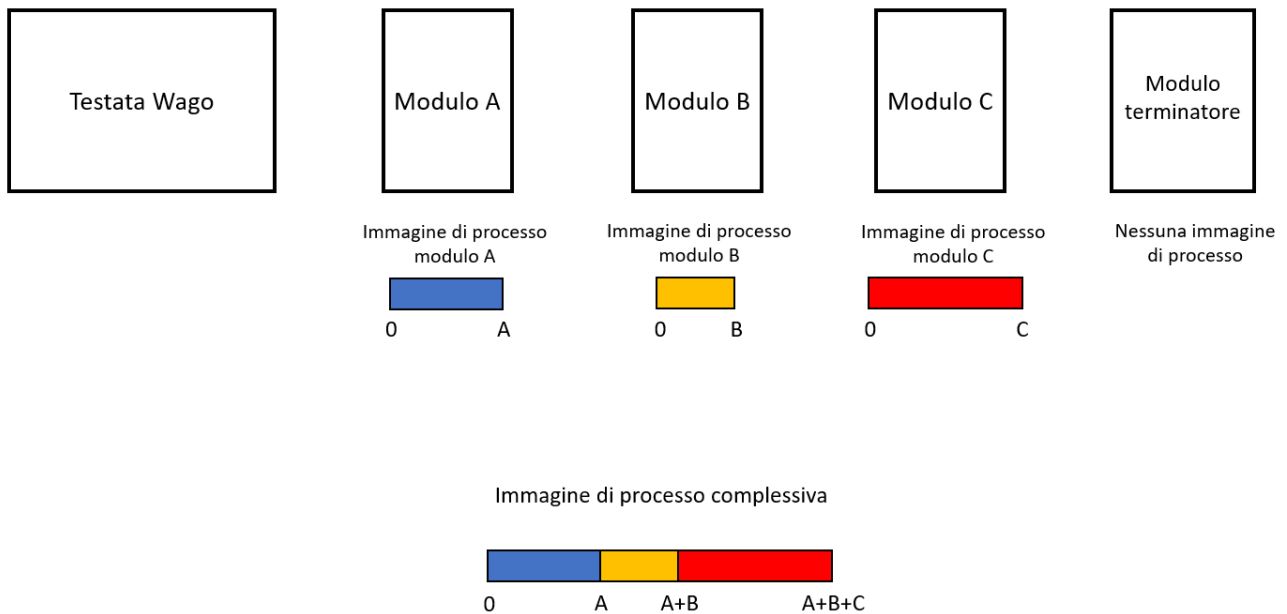
La comunicazione tra pc e plc avviene mediante scritture e/o letture di un'apposita zona di memoria interna al plc chiamata immagine di processo e la rappresentazione dei dati dei moduli I/O all'interno dell'immagine di processo dipende dall'accoppiatore/controller Fieldbus utilizzato. In questa dimostrazione, usando l'accoppiatore Wago Ethernet 750-352, l'immagine di processo è composta da 24+24 byte così suddivisi:

Input Data		Output Data	
S0	Status byte 0	C0	Control byte 0
S1	Status byte 1	C1	Control byte 1
D0	Data byte 0	D0	Data byte 0
D1	Data byte 1	D1	Data byte 1
D2	Data byte 2	D2	Data byte 2
...	...	...	...
D20	Data byte 20	D20	Data byte 20
D21	Data byte 21	D21	Data byte 21

I dati che devono essere inviati e ricevuti saranno immagazzinati nei byte Data Input e Data Output (D0 ... D21) ed il flusso dei dati viene gestito attraverso i byte Status S0-S1 e Control C0-C1. I byte ricevuti dal plc sono contenuti nell'area di memoria Input Data, mentre i byte da trasmettere si trovano nell' Output Data. Per ulteriori dettagli della struttura dei byte Status e Control si rimanda al manuale Wago 750-652 (link: <https://www.wago.com/it/sistemi-i-o/interfaccia-seriale-rs-232-485/p/750-652>).

Si presti attenzione che, qualora si voglia implementare la soluzione adottata nel presente documento, la dimensione e la composizione dell'immagine di processo dipendono da come è composta la serie dei moduli collegati alla testata: in questo caso, il modulo seriale 750-652 è il primo della serie (non vanno considerati i moduli che non hanno immagini di processo, come il modulo alimentazione 750-602 ed il modulo terminatore 750-600) quindi per accedere alla sua immagine di processo è necessario fare letture/scritture partendo dal byte 0; se, invece, sono presenti altri moduli collegati prima del modulo seriale, bisogna tenere conto delle dimensioni delle loro memorie. Pertanto, per accedere all'immagine di processo dell'n-esimo modulo, si dovranno fare letture/scritture partendo dal byte OFFSET+0, dove OFFSET è proprio la somma di tutte le immagini di memoria degli n-1 moduli che lo precedono; l'i-esimo byte dell'immagine di processo dell'n-esimo modulo, quindi, nell'immagine di processo complessiva si troverà al byte OFFSET+i.

La figura seguente aiuta a capire come è strutturata l'immagine di processo complessiva data dalla connessione di più moduli in cascata ad un'unica testata Wago (i moduli A, B e C hanno immagini di processo rispettivamente di dimensioni A, B, e C bytes).



Utilizzando il modulo seriale 750-652, se si vogliono fare letture o scritture che richiedono più byte rispetto ai 22 dell'immagine di processo, come nel caso della scrittura e della lettura del campo Name del tag, è necessario ripetere più volte l'operazione mantenendo in memoria le comunicazioni precedenti per poi concatenarle ed analizzarle solo alla fine (se, ad esempio, una scrittura richiede 58 byte, il pc dovrà effettuare 3 scritture consecutive di 22, 22 e 14 byte).

L'utilizzo delle immagini di processo è tipico dei plc Wago: è infatti possibile, modificando opportunamente il livello fisico del collegamento tra pc e plc (connessione elettrica e protocollo di comunicazione di basso livello), utilizzare un'altra testata standard Wago serie 750 mantenendo inalterata la parte di comunicazione ad alto livello (ovvero i pacchetti che devono essere scambiati tra reader e tag). In questa applicazione, viene utilizzata una testata Ethernet con protocollo MODBUS/TCP, ma in modo analogo è possibile riprodurre le stesse funzionalità con testate Wago Ethernet TPC/IP, Canopen, RS-232, RS-485 o altro.

## IV. Protocollo di comunicazione del reader (RAQC)

Il reader RAQC è un dispositivo in grado di fare letture e scritture RFID e può essere integrato in applicazioni che necessitano della tecnologia RFID con frequenza di lavoro 13.56 MHz. Il dispositivo comunica con un sistema host (PC o PLC) mediante una linea seriale con standard RS232 e funge da tramite attraverso una serie di comandi tra quest'ultimo ed il tag presente nell'area d'influenza dell'antenna. Il protocollo usato per la comunicazione è di tipo master/slave e, attraverso la linea seriale, è possibile configurare i parametri di comunicazione / funzionamento del dispositivo ed aggiornarne il firmware. Infine, il dispositivo RAQC è in grado di gestire degli I/O digitali sia in versione PNP che NPN.

Il protocollo master/slave prevede che il dispositivo slave, dopo aver ricevuto un messaggio a lui indirizzato dal dispositivo master, trasmetta un messaggio di risposta dopo un tempo minimo di circa 10 ms. Il tag viene configurato di default con i seguenti parametri: indirizzo 255, baud rate 19200, 8 bit di dati, nessuna parità e 1 bit di stop. Questi parametri possono tuttavia essere modificati tramite appositi registri.

Per semplificare le spiegazioni dei comandi verranno utilizzate le seguenti convenzioni generali:

SOH	Carattere 01h (0x01)
STX	Carattere 02h (0x02)
ETX	Carattere 03h (0x03)
EOT	Carattere 04h (0x04)
ENQ	Carattere 05h (0x05)
ACK	Carattere 06h (0x06)
NAK	Carattere 15h (0x15)
SYN	Carattere 16h (0x16)
CR	Carattere 0Dh (0x0D)
'0' ... '9'	Caratteri 30h ... 39h (0x30 ... 0x39)
'A' ... 'F'	Caratteri 41h ... 46h (0x41 ... 0x46)
< ... >	Caratteri 30h ... 39h (0x30 ... 0x39), 41h ... 46h (0x41 ... 0x46)
< bcc >	Checksum

La struttura generale di un messaggio è la seguente:

**SOH <add h> <add l> ... <bcc> CR**

dove SOH è il carattere d'inizio, <add h> e <add l> sono i caratteri corrispondenti all'indirizzo, <bcc> è il carattere di controllo o checksum e CR è il carattere finale del pacchetto.

L'indirizzo del dispositivo è espresso da un byte (0 ... 255 in decimale, 0x00 ... 0xFF in esadecimale, in questo esempio viene usato l'indirizzo 255) che viene trasformato in due caratteri ASCII: il primo carattere ASCII <add h> corrisponde alla codifica in ASCII del nibble alto del byte, mentre il secondo carattere ASCII <add l> corrisponde alla codifica in ASCII del nibble basso del byte. Esempio: 255 → 0xFF → 'F' 'F'. Questa regola vale anche per qualsiasi valore espresso da un byte.

Ad esempio, il comando 'richiesta dati' è così composto: SOH <add h> <add l> ENQ <bcc> CR, quindi il messaggio da trasmettere ad un dispositivo con indirizzo 1 è: SOH '0' '1' ENQ <bcc> CR (in esadecimale: 0x01, 0x30, 0x31, 0x05, <bcc=0x05>, 0x0D).



La codifica dei comandi di lettura e scrittura dati dal tag è 'E' '2' ed 'E' '3' ed in particolare, i comandi di lettura e scrittura del campo Name che sono stati implementati sono:

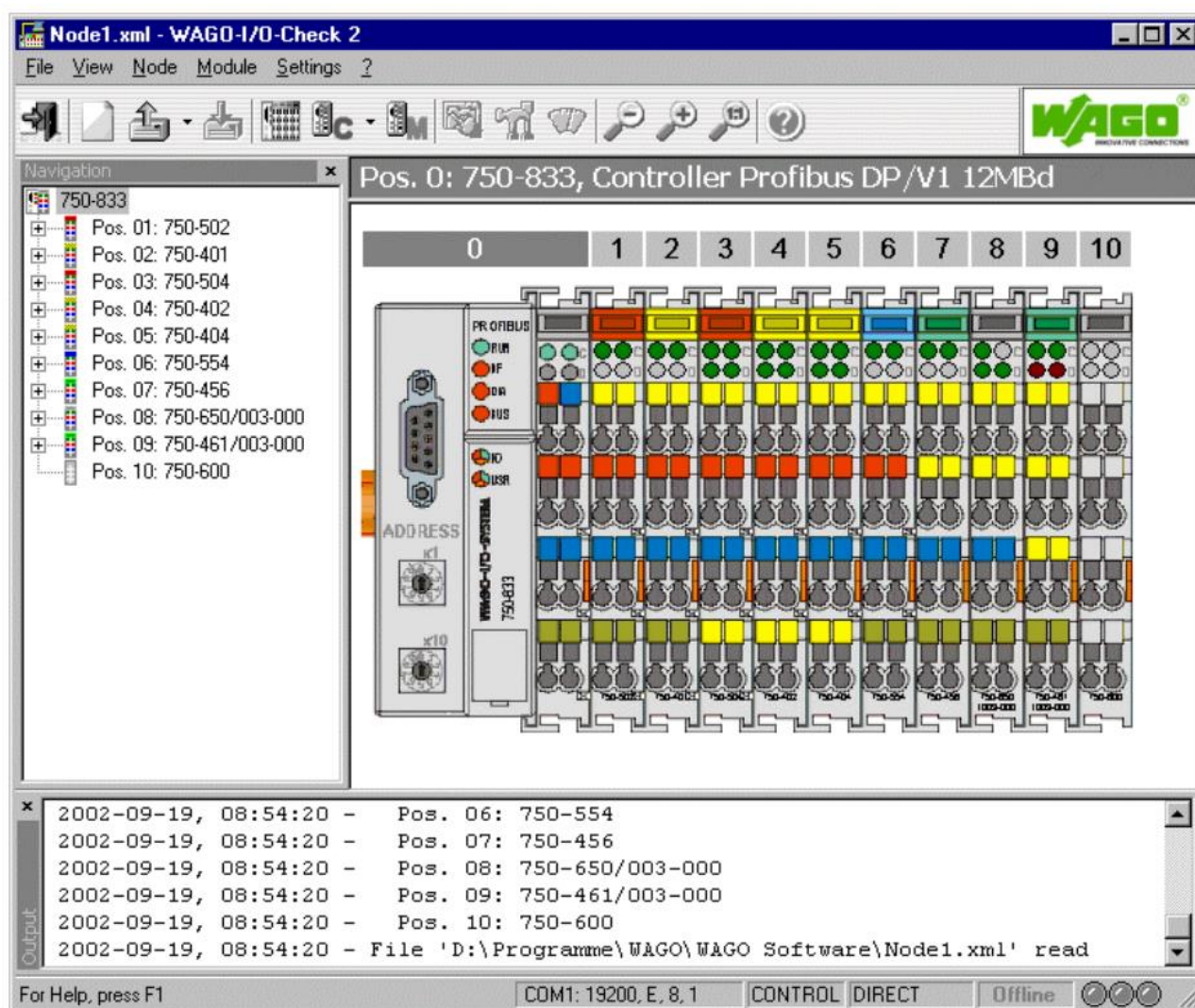
- Lettura Name: SOH 'F' 'F' STX 'E' '2' '0' '0' '0' '0' '0' '1' '3' '0' '0' '1' '0' '0' '0' '0' '0' '0' '1' '4' ETX <bcc=0x71> CR
- Scrittura Name: SOH 'F' 'F' STX 'E' '3' '0' '0' '0' '0' '0' '1' '3' '0' '0' '1' '0' '0' '0' '0' '0' '0' '1' '4' <40 byte contenenti il nome codificato secondo specifiche> ETX <bcc> CR

## V. Configurazione plc con software WAGO-I/O-CHECK

L'interfaccia WAGO-I/O-CHECK (disponibile con cod. 759-920) è un applicativo Windows che consente all'utente di interagire e rappresentare graficamente un nodo Wago 750 direttamente collegato al bus di campo (Fieldbus). WAGO-I/O-CHECK, inoltre, è in grado di leggere la configurazione di ciascun nodo connesso all'accoppiatore e visualizzare tutti i nodi sullo schermo del PC.


Le immagini presenti in questo capitolo sono dimostrative e non corrispondono all'applicazione presentata.

L'interfaccia WAGO-I/O-CHECK si presenta come nella figura seguente:



Partendo da sinistra, vengono mostrati l'accoppiatore/controller Fieldbus (ovvero la testata 750-352) ed in successione tutti i moduli a lui collegati, ovvero, nella presente soluzione, il modulo alimentazione (750-602), il modulo seriale (750-652) ed il modulo terminatore (750-600).



Dopo aver selezionato un modulo, tramite i bottoni e  è possibile modificare i parametri e leggere l'immagine di processo (ovviamente, è possibile editare solo i parametri modificabili e leggere l'immagine di processo dei soli moduli che la possiedono).

Selezionando il modulo seriale 750-652, è possibile impostare i seguenti parametri che saranno poi utilizzati come default ad ogni successiva accensione:

Parameter	Value
COM-Port	COM1
Baudrate	115200
Parität	None
Datenbits	8
Stopbits	1
Timeout (ms)	500

Premendo l'apposito bottone per leggere l'immagine di processo, invece, apparirà una finestra simile a questa, in cui è possibile leggere i valori dei registri Status, Control e Data sia dei campi Input che Output.

Pos. : 750-650/003-000		
RS 232 C Interface (Adjustable)		
Byte	Output	Input
CT/ST	0x00	0x00
D0	0x00	0x00
D1	0x00	0x00
D2	0x00	0x00

Per qualsiasi ulteriore dettaglio si rimanda al sito ufficiale Wago ed al manuale d'uso scaricabile al link:

<https://www.wago.com/it/software/wago-i-o-check/p/759-920#downloads>

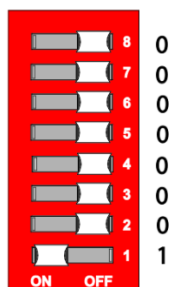
## VI. Configurazione pc

**Prerequisito:** per avviare l'applicazione Gimatic, il pc deve avere sistema operativo Windows.

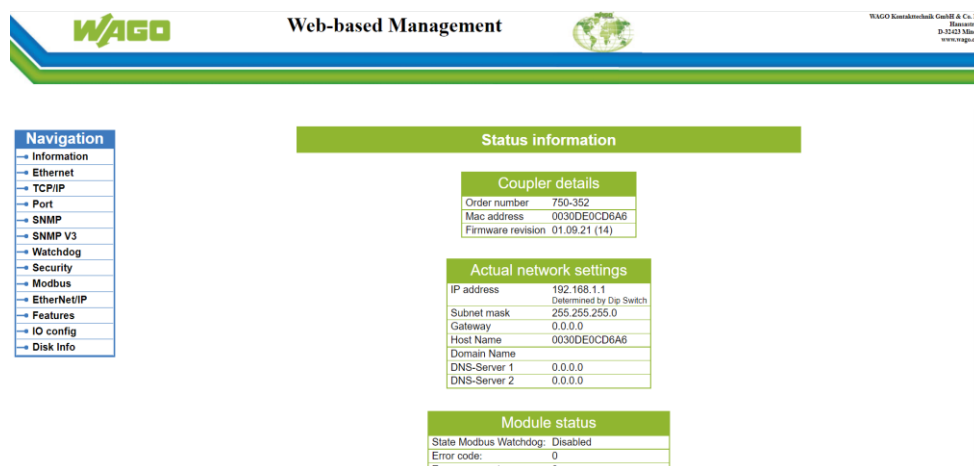
Una volta verificati i cablaggi, procedere con la configurazione del pc per la comunicazione con il plc impostando l'indirizzo IP statico:

- Indirizzo ip: "192.168.1.X" (con  $1 < X < 255$ , in questo esempio  $X=2$ )
- Subnet mask: "255.255.255.0"

Verificare la correttezza delle impostazioni digitando nella barra di ricerca del browser del pc l'indirizzo ip del plc (l'ultimo byte di questo indirizzo è modificabile attraverso i dip-switch presenti sulla testata, per semplicità "192.168.1.1" impostandoli come in figura).

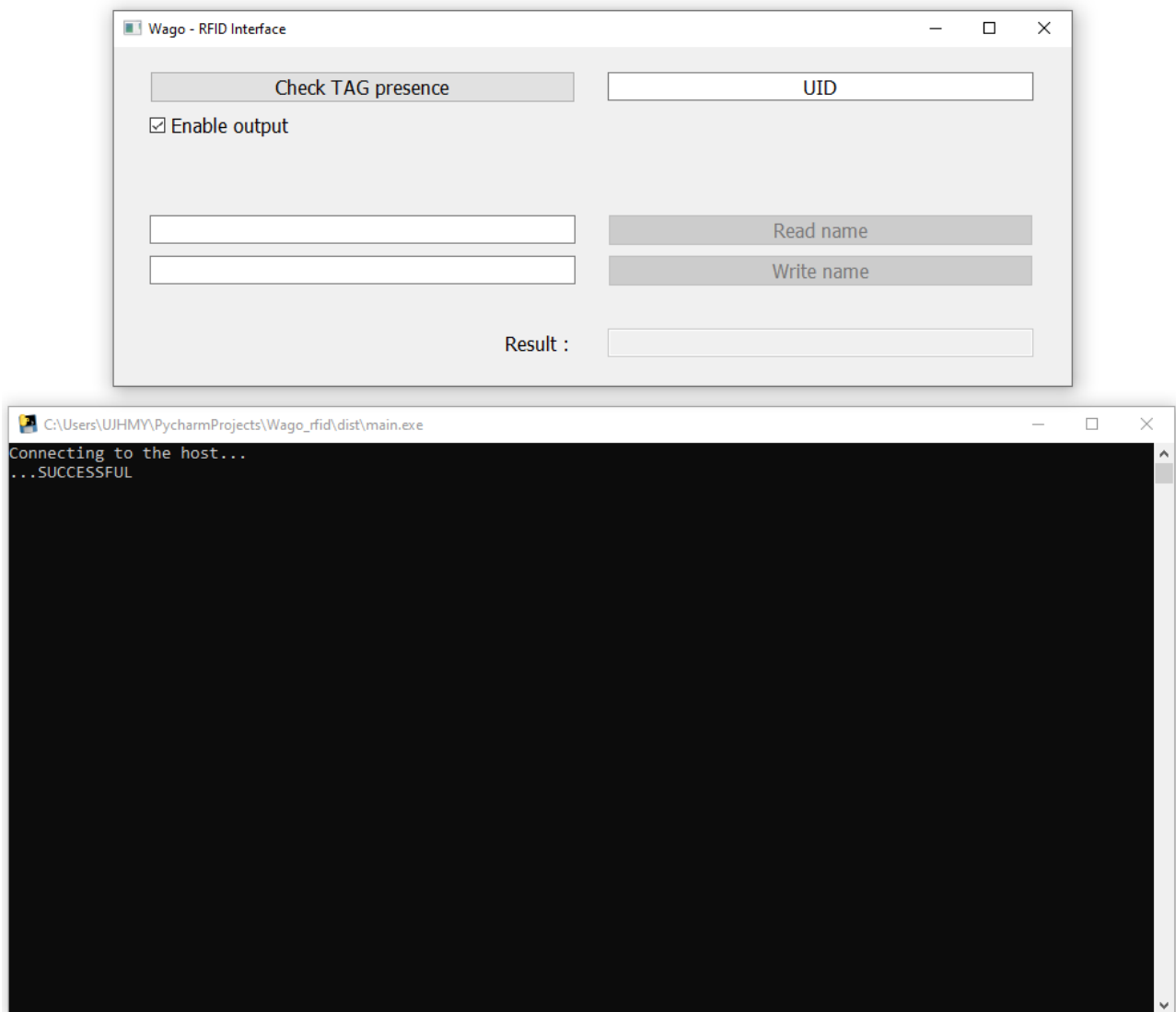


A questo punto, se tutto è corretto, dovrebbe aprirsi una pagina simile a questa:



## VII. Interfaccia Gimatic

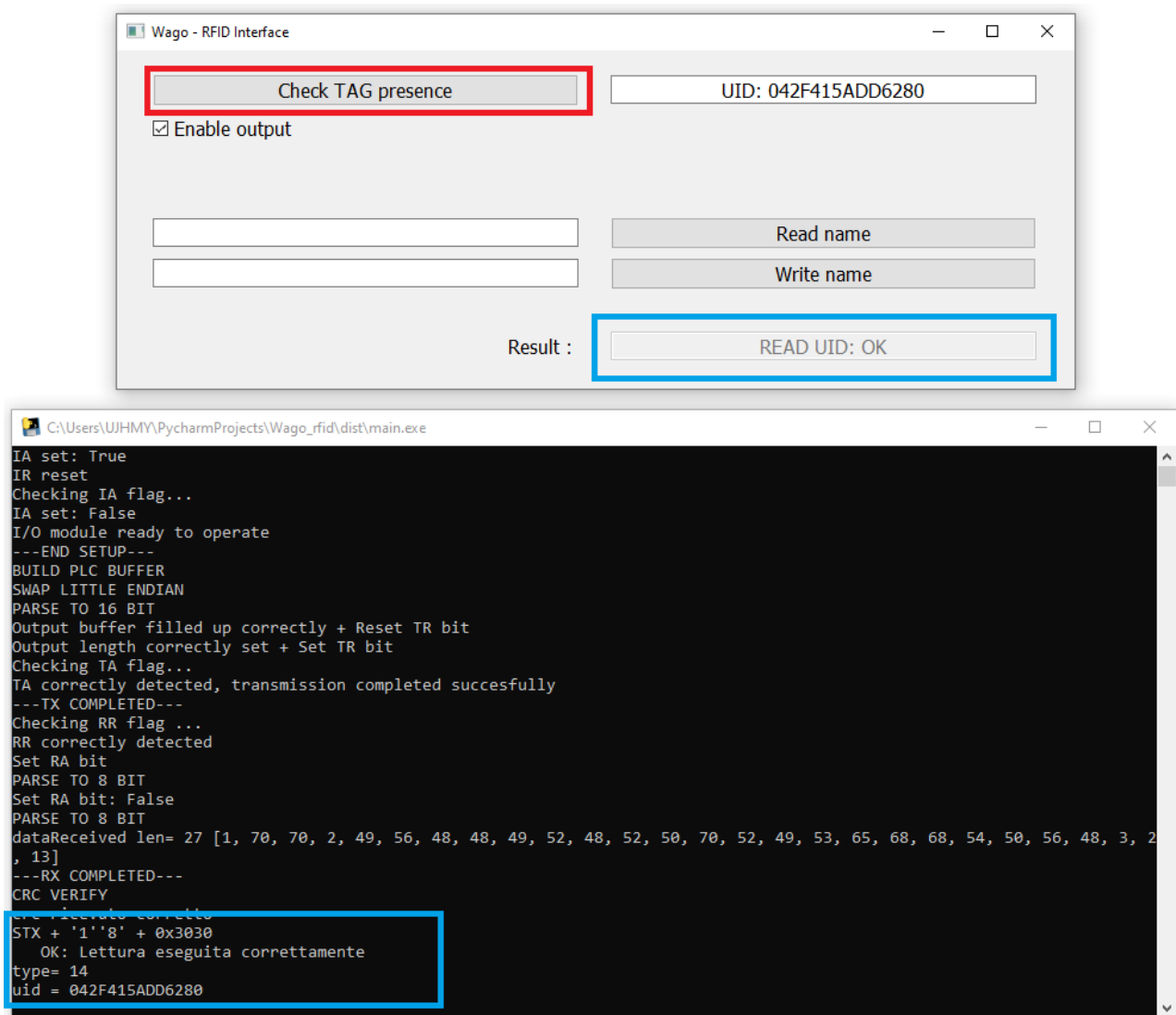
All'avvio, l'interfaccia grafica ed il relativo prompt dei comandi si presentano nel seguente modo:



Attraverso i relativi pulsanti, è possibile:

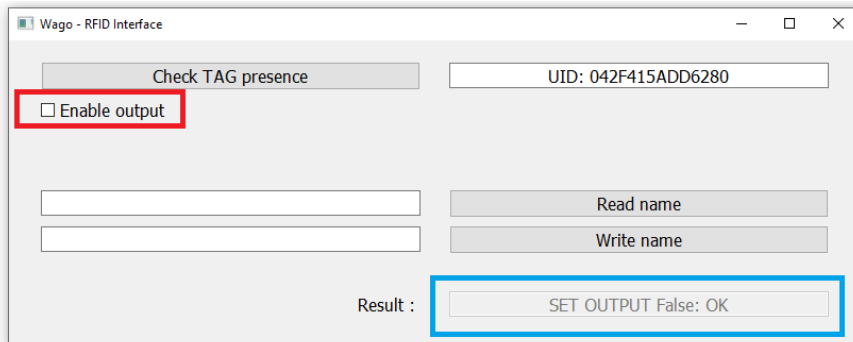
- 1- verificare la presenza di un tag leggendone il suo UID (Unic IDentifier). Se presente, viene visualizzato nella relativa casella "UID" e vengono abilitati i due pulsanti "Read name" e "Write name".
- 2- attivare/disattivare le uscite dell'RAQC spuntando la casella "Enable output"
- 3- leggere e scrivere il nome del tag RFID
- 4- verificare l'esito dell'ultima operazione fatta nella casella "Result"

In seguito, vengono riportate alcune schermate alla fine dell'esecuzione di ciascun comando:

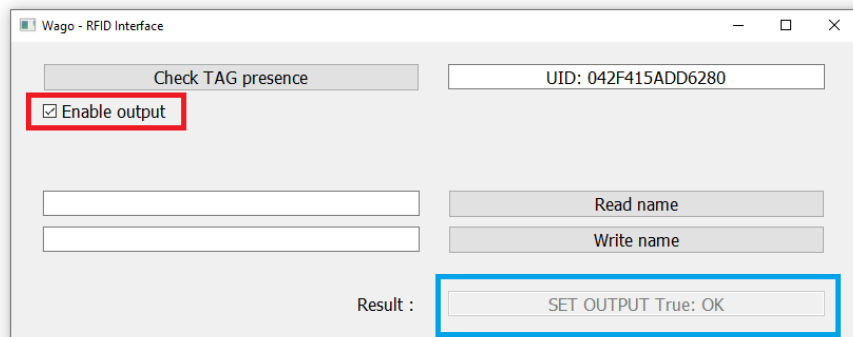


Alla pressione del pulsante "Check tag presence", se il tag (interno all'RBQC) è presente e correttamente funzionante, nell'interfaccia verrà riportato nel campo "Result" il messaggio *READ UID: OK*, mentre nel prompt verranno visualizzati sia il tipo che l'UID del tag.

Per attivare oppure disattivare le uscite spuntando la casella "Enable output", si otterranno queste due schermate (l'esito positivo, oltre al campo "Result", è visualizzato anche nel prompt attraverso il messaggio di ACK ricevuto):

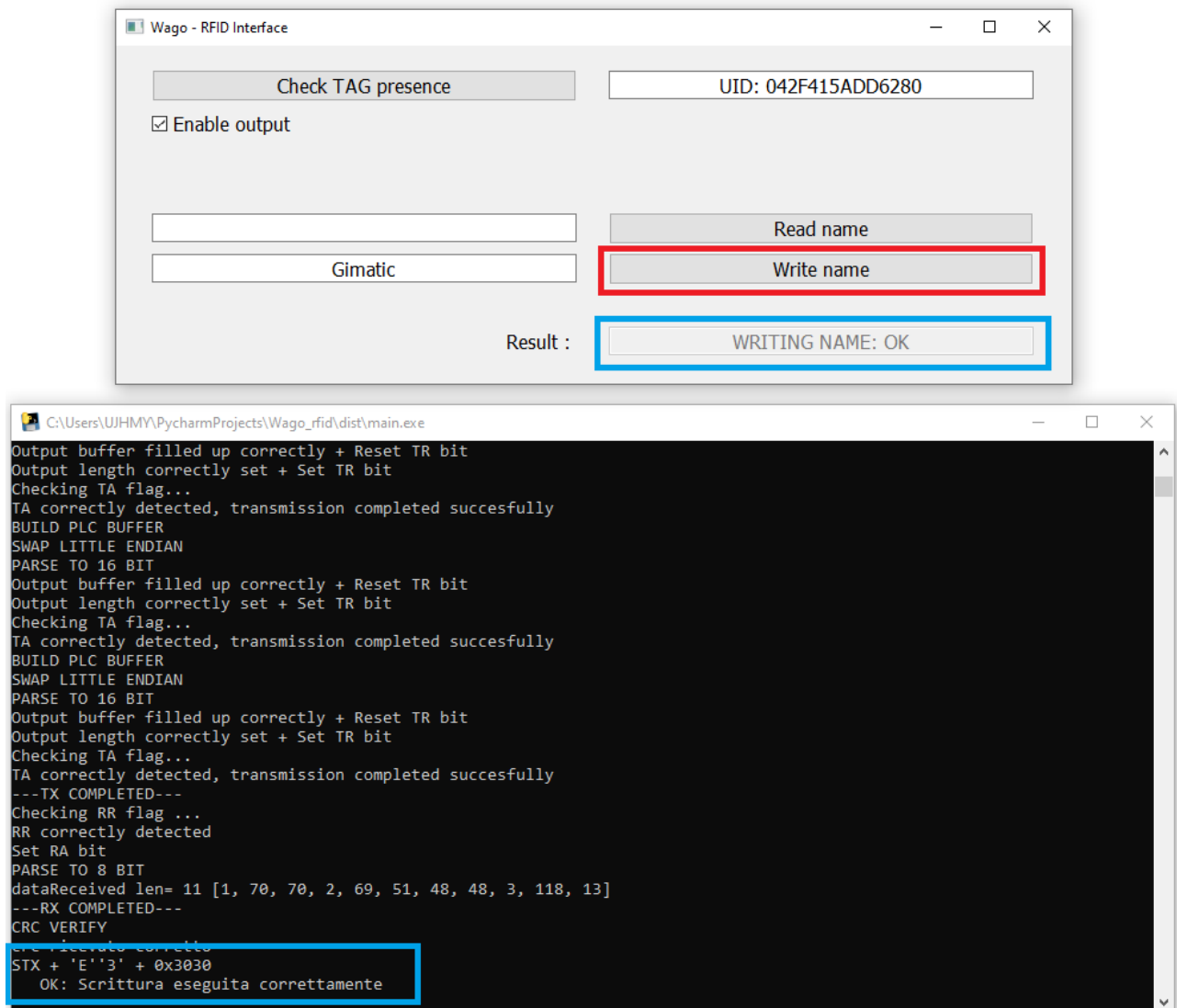


```
C:\Users\UJHMY\PycharmProjects\Wago_rfid\dist\main.exe
uscite disattivate
CRC CALCULATE
buffer to send: len=23 [1, 70, 70, 2, 50, 70, 70, 70, 55, 56, 49, 48, 48, 48, 48, 48, 49, 48, 49, 3, 122, 13]
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
---TX COMPLETED---
Checking RR flag ...
RR correctly detected
Set RA bit
PARSE TO 8 BIT
Set RA bit: False
PARSE TO 8 BIT
dataReceived len= 6 [1, 70, 70, 6, 7, 13]
---RX COMPLETED---
CRC VERIFY
CRC ricevuto corretto
ACK
```



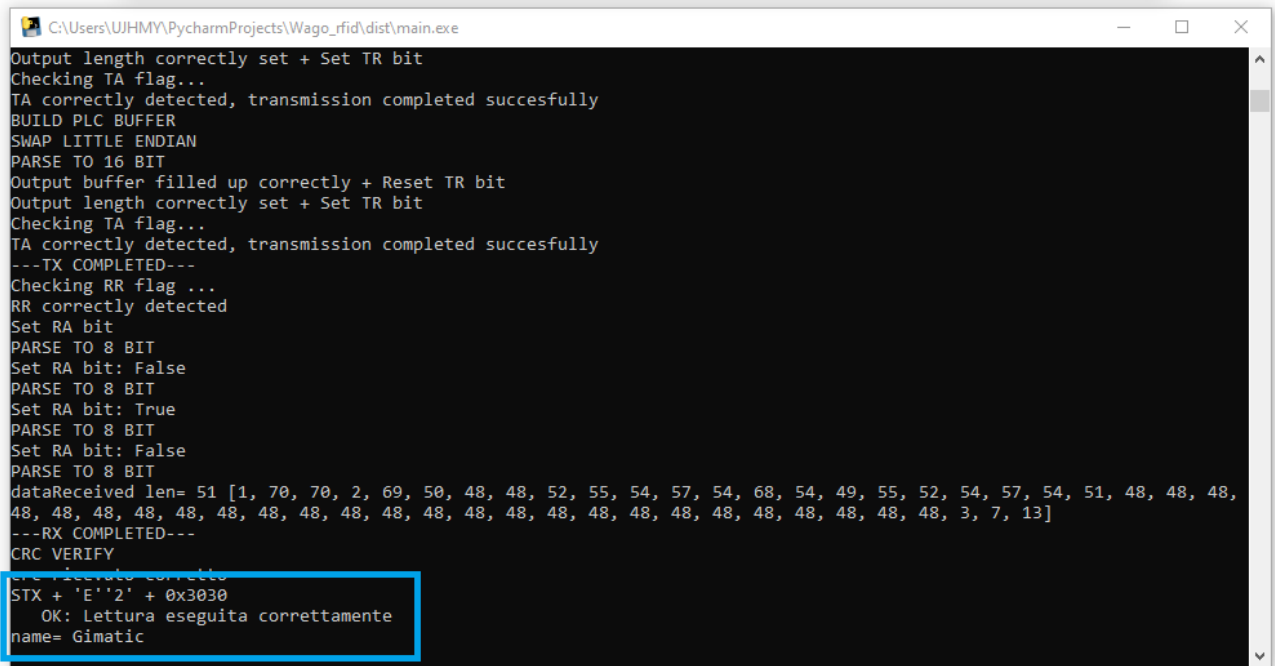
```
C:\Users\UJHMY\PycharmProjects\Wago_rfid\dist\main.exe
uscite attivate
CRC CALCULATE
buffer to send: len=23 [1, 70, 70, 2, 50, 70, 70, 70, 55, 56, 49, 48, 48, 48, 48, 48, 49, 48, 48, 3, 123, 13]
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
---TX COMPLETED---
Checking RR flag ...
RR correctly detected
Set RA bit
PARSE TO 8 BIT
Set RA bit: False
PARSE TO 8 BIT
dataReceived len= 6 [1, 70, 70, 6, 7, 13]
---RX COMPLETED---
CRC VERIFY
CRC ricevuto corretto
ACK
```

Quando viene premuto il comando “Write name”, l’esito positivo dell’operazione di scrittura è visualizzato in questo modo:

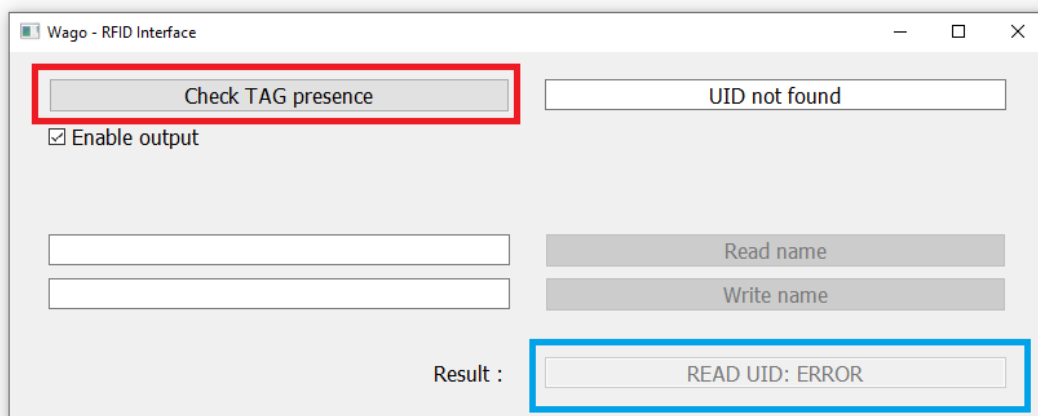


In modo simile, quando verrà eseguita una lettura tramite il comando “Read name”, l’esito positivo verrà mostrato in questo modo:





16



```
C:\Users\UJHMY\PycharmProjects\Wago_rfid\dist\main.exe
Error in setting IR bit
IR set
Checking IA flag...
IA set: True
IR reset
Checking IA flag...
IA set: False
I/O module ready to operate
---END SETUP---
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
---TX COMPLETED---
Checking RR flag ...
RR correctly detected
Set RA bit
PARSE TO 8 BIT
Set RA bit: False
PARSE TO 8 BIT
dataReceived len= 11 [1, 70, 70, 2, 49, 56, 48, 49, 3, 8, 13]
---RX COMPLETED---
CRC VERIFY
CRC ricevuto corretto
STX + '1'8' + 0x3031
  Errore: TAG non presente
```

## I. Introduction

In electronics and telecommunications, radiofrequency identification (Radio-Frequency Identification, acronym RFID) is a technology for recognition, validation and automatic memorization of information in short distance. It is based on storing data in particular electronic passive devices called tag, able to reply to query by other electronic active devices, fixed or portable, called reader. Identification and information exchange between these two devices takes place by radio frequency. A RFID system is usually composed by three elements:

- One or more RFID tag (also called transponder)
- One device that can read and/or write (reader)
- A storing system to manage data to be transferred to/from the reader

The main element in a RFID system is the tag, which consists of a microchip that stores data in its internal memory (including a unique universal number written on silicon and not modifiable), an antenna and a physical support that holds the chip and antenna together called "substrate". Its operation is based on the physical principle of electromagnetic induction: tag's antenna receives from the reader an electrical signal that is transformed into electricity to power the microchip and send back the reply to the received command.

Gimatic provides solutions for automatic component handling. To do this, usually, a robot is used in combination with several EOATs (End Of Arm Tools), each dedicated to a specific operation. In a similar application, the robot's wrist can be equipped with a reading unit (reader) and any EOAT equipped with a RFID tag. This memory contains:

- Generic data: name and description of the EOAT, identifier code, mass and volume
- Geometric and mass properties: center of mass's coordinates, moment of inertia
- Geometric properties: calibration parameters
- Bill of material: list of available tools with description, version and quantity

In this application the following Gimatic products have been used:

PRODUCT CODE	DESCRIPTION	LINK TO SHOP + USER MANUAL	PICTURE
RAQC	reader RFID	<a href="https://shop.gimatic.com/en/raqc">https://shop.gimatic.com/en/raqc</a>	
RBQC	tag RFID	<a href="https://shop.gimatic.com/en/rbqc">https://shop.gimatic.com/en/rbqc</a>	
RQCBOX	Communication interface with RAQC	<a href="https://shop.gimatic.com/en/rqcbox">https://shop.gimatic.com/en/rqcbox</a>	

Instead of the products above shown, is also possible to use these compatible devices:

- reader: RRAQC, CRAQC, RAQC-C and variants with npn I/O like RAQCN, RRAQCN, CRAQCN
- tag: RRBQC, CRBQC

Each reader must be suitably matched to its tag according to the shapes and mechanical dimensions.

More usage and compatibility information are provided in technical documents (IST-RQC, IST-KIT RFID, IST-RQCBOX e RFID-TOOLS AND MANUALS) downloadable from the Gimatic shop and from the link <https://shop.gimatic.com/Product/GetExtraFile?id=2636> .

This application is intended to be a demonstration of technical feasibility to propose a possible solution to the problem related to the recognition of tags. Due to the limited number of pins on RAQC reader (equal to 8), it is possible to distinguish up to 255 different devices by assigning them in the field "ID" (user IDentifier) a progressive number from 1 to 255. In the next chapters, the hardware architecture and software interface developed in this demonstration will then be presented (source code is available on request for further study or further expansion).

Implementing a serial communication with RAQC reader (using for convenience of interface a RQCBOX, but it would not be indispensable), is possible to give a unique name to each tag RBQC by writing it in the internal microchip's memory. This operation allows to exceed the limit of 255 identifiable devices because the reader can read and write in the tag memory that has a capacity of about 4kb.

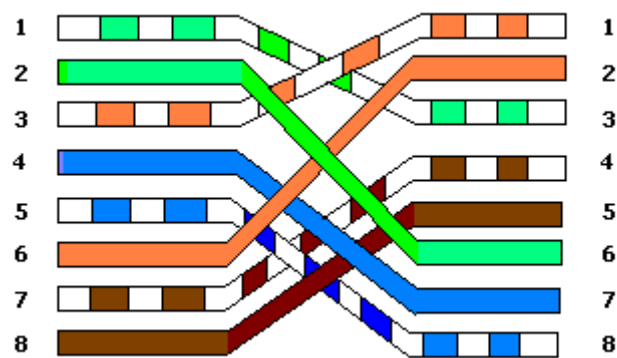
To implement the serial communication, a Wago plc is used to communicate with the master controller (in this case a PC) using the Modbus TCP protocol. Through the appropriate Gimatic interface created on PC, user can check the presence of the tag and read or write its Name.

## II. Wiring

The minimum configuration of Wago module used to implement this application is:

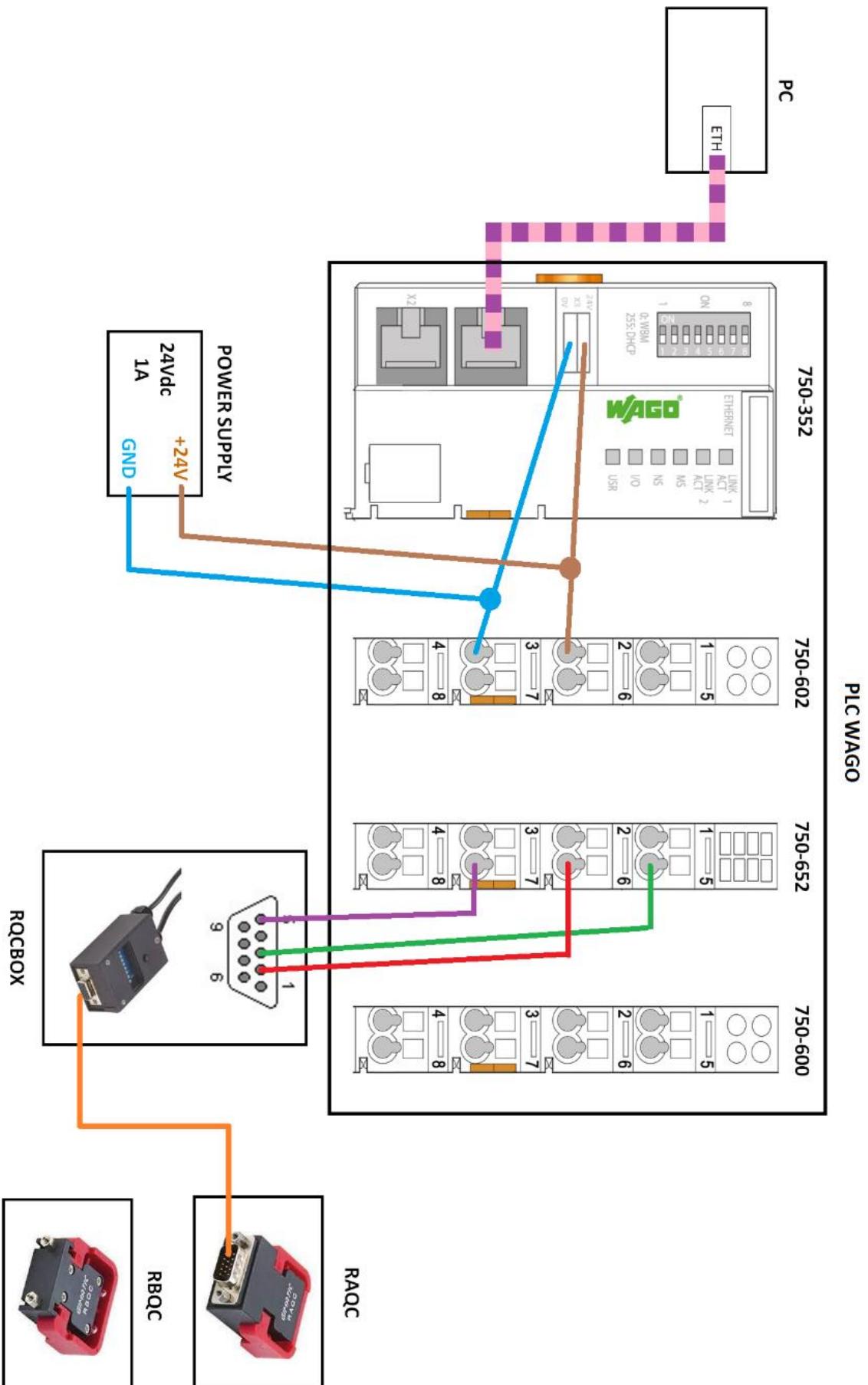
- 1- Wago Fieldbus Ethernet header (code 750-352)
- 2- Wago power supplier module (code 750-602)
- 3- Wago serial interface RS-232 module (code 750-652)
- 4- Wago termination module (code 750-600)

The wiring between pc and plc needs a crossed Ethernet cable (null modem) as shown in the following figure:



Wago plc needs a 24Vdc 1A stabilized power supply.

The connection order and wiring of the module is shown below:



### III. Process image

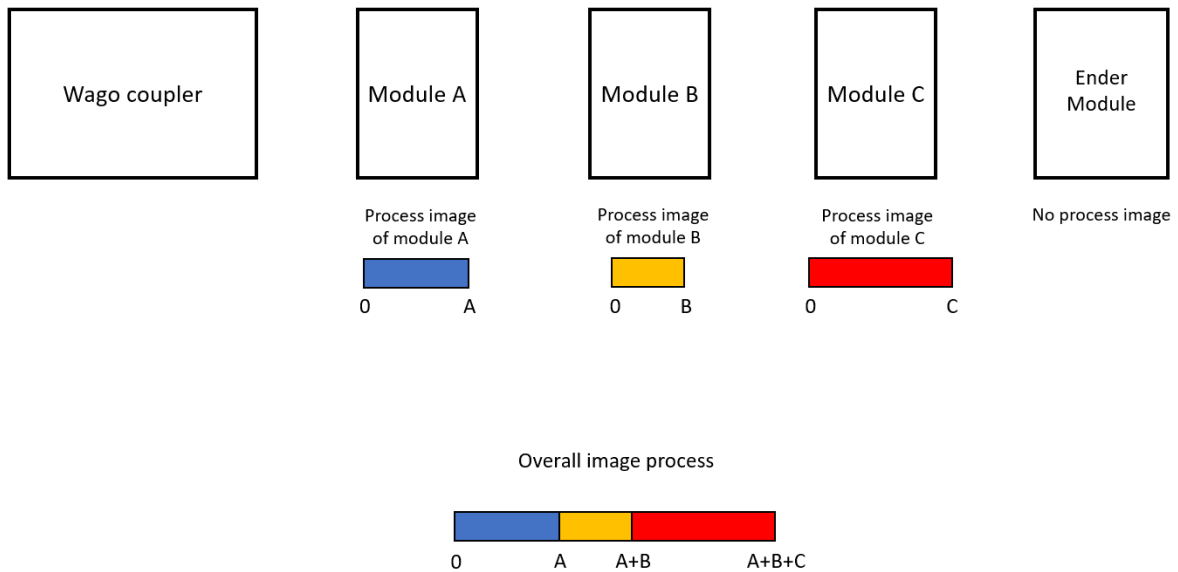
The communication between pc and plc takes place through writing and reading in a specific memory area inside the plc called process image and the representation of the I/O module data within process images depends on Fieldbus coupler/controller used. In this demo, using Wago Ethernet 750-352 coupler, the process image consists of 24+24 bytes divided as follow:

Input Data		Output Data	
S0	Status byte 0	C0	Control byte 0
S1	Status byte 1	C1	Control byte 1
D0	Data byte 0	D0	Data byte 0
D1	Data byte 1	D1	Data byte 1
D2	Data byte 2	D2	Data byte 2
...	...	...	...
D20	Data byte 20	D20	Data byte 20
D21	Data byte 21	D21	Data byte 21

Data to be sent and received are stored in bytes named Data Input and Data Output (D0 ... D21) and data flow is managed through Status S0-S1 e Control C0-C1 bytes. Bytes received by plc are stored in the Input Data memory area while bytes to be sent are stored in Output Data. For further details of the structure of the Status and Control bytes, refer to the Wago 750-652 manual (link: <https://www.wago.com/global/i-o-systems/rs-232-485-serial-interface/p/750-652> ).

Please note that if you want to reproduce the solution adopted in this document, the size and composition of the process image depend on how the series of modules connected to the coupler is composed: in this demo, the serial module 750-652 is the first of the series (should not be considered modules that don't have process image, as power supplier module 750-602 and ending module 750-600) so to access its process image it is necessary to make reads / writes starting from byte 0; however, if other modules are connected before the serial module, their memories must be taken into account. Therefore, to access the n-th module's process image, reads/writes must be done starting from byte OFFSET+0, where OFFSET is the sum of all n-1 modules' process image that precede it; i-th byte of the n-th module's process image, finally, in the overall process image will be the byte OFFSET+i.

The following figure helps to understand how the overall process image is structured given by the connection of several modules in cascade to a single Wago coupler (Modules A, B, and C have process image length of respectively A, B and C bytes).



Using serial module 750-652, if you need reads or writes that require more bytes than the 22 of the process image, as in the case of reading and writing the Name field of the tag, it is necessary to repeat the operation several times, keeping the previous communications in memory and then concatenating and analyzing them only at the end (for example, if a writing requires 58 byte, pc has to do 3 consecutive writings of 22, 22 and 14 bytes).

The use of process images is typical of Wago plc: it is possible, by suitably modifying the physical layer of the connection between pc and plc (electrical connection and low-level communication protocol), to use another standard Wago 750 series coupler while maintaining the high-level communication part (ie the packets that must be exchanged between reader and tag) remains unchanged. In this application, an Ethernet coupler with MODBUS/TCP protocol is used, but in a similar way you can reproduce the same functionalities using Wago couplers Ethernet TPC/IP, Canopen, RS-232, RS-485 o other ones.



## IV. Reader's communication protocol (RAQC)

The RAQC reader is a device capable of making RFID reads and writes and can be integrated in equipment that requires RFID technology with a 13.56 MHz working frequency. This device communicates with a host system (PC or PLC) through a serial line with RS232 standard and acts as a link through a series of commands between the latter and the tag present in the area of influence of the antenna. The protocol used to communicate is master/slave and, through the serial line, it is possible to configure reader's communication and functional parameters and to upload the firmware. Finally, RAQC is able to manage digital I/O both PNP and NPN versions.

Master/slave protocol requires that the slave device, after receiving a message from the master, sends a response after a minimum time of about 10ms. Tags are configured with these default parameters: address 255, baud rate 19200, 8 bits data, parity none and 1 stop bit. These parameters could also be modified by some special registers.

In order to simplify the explanations of the commands, the following general conventions will be used:

SOH	Character 01h (0x01)
STX	Character 02h (0x02)
ETX	Character 03h (0x03)
EOT	Character 04h (0x04)
ENQ	Character 05h (0x05)
ACK	Character 06h (0x06)
NAK	Character 15h (0x15)
SYN	Character 16h (0x16)
CR	Character 0Dh (0x0D)
'0' ... '9'	Characters 30h ... 39h (0x30 ... 0x39)
'A' ... 'F'	Characters 41h ... 46h (0x41 ... 0x46)
< ... >	Characters 30h ... 39h (0x30 ... 0x39), 41h ... 46h (0x41 ... 0x46)
< bcc >	Checksum

The general structure of a message is as follows:

**SOH <add h> <add l> ... <bcc> CR**

where SOH is the initial character, <add h> and <add l> are the characters corresponding to the address, <bcc> is the control character or checksum and CR is the packet's final character.

The slave address is coded using a byte (0 ... 255 in decimal format, 0x00 ... 0xFF in hexadecimal format, in this example address 255 is used) that is transformed into two ASCII characters: the first one <add h> corresponds to the ASCII encoding of the high nibble of the byte, while the second one <add l> corresponds to the ASCII encoding of the low nibble of the byte. Example: 255 → 0xFF → 'F' 'F'. This method is always used for any other byte.

'Data enquiry' command, for example, is composed as follow: SOH <add h> <add l> ENQ <bcc> CR, so the message sent to a slave with address 1 is SOH '0' '1' ENQ <bcc> CR (in hexadecimal format: 0x01, 0x30, 0x31, 0x05, <bcc=0x05>, 0x0D).

The packets sent to the reader to read and write tag's data are coded 'E' '2' and 'E' '3'; in particular, read and write of the Name fields implemented in the software interface are:

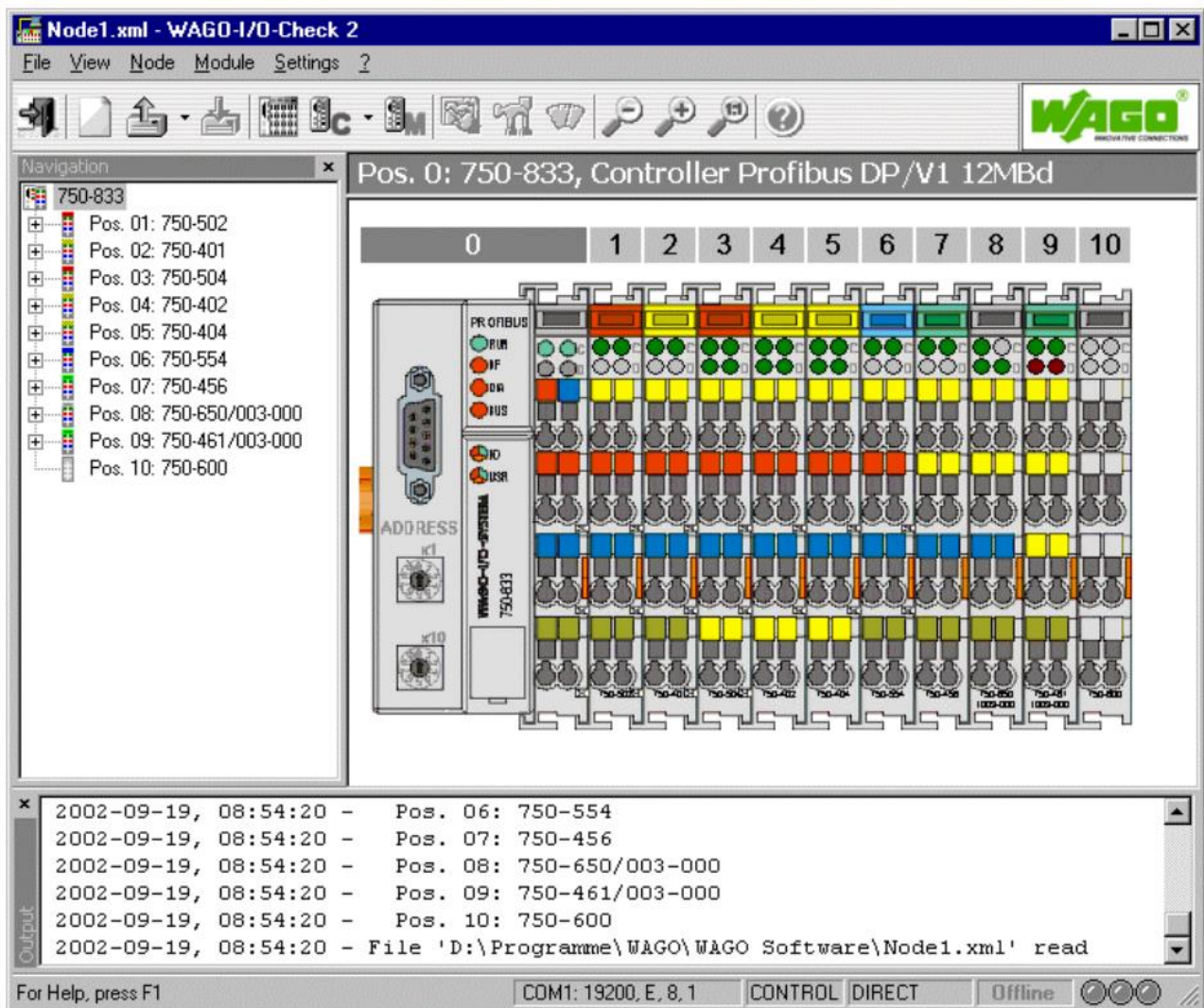
- Reading Name: SOH 'F' 'F' STX 'E' '2' '0' '0' '0' '0' '0' '1' '3' '0' '0' '1' '0' '0' '0' '0' '0' '0' '1' '4' ETX <bcc=0x71> CR
- Writing Name: SOH 'F' 'F' STX 'E' '3' '0' '0' '0' '0' '0' '1' '3' '0' '0' '1' '0' '0' '0' '0' '0' '0' '1' '4' <40 byte containing the name coded according to specs> ETX <bcc> CR

## V. Plc configuration using software WAGO-I/O-CHECK

WAGO-I/O-CHECK interface (order code 759-920) is a Windows app that allows a user to interact and graphically represents a Wago 750 node connected to the Fieldbus. WAGO-I/O-CHECK can also read the configuration data of each node directly connected to the coupler and displays all the nodes.

The images in this chapter are demonstrative only and do not correspond to the application presented.

The WAGO-I/O-CHECK interface is shown in the following figure:



Starting from the left, the interface shows the Fieldbus header (the coupler 750-352) and all the modules connected to the coupler, ie power supplier (750-602), serial module (750-652) and ending module (750-600).



Selecting a module, using buttons and is possible to modify its parameters and to read its process image (obviously, it is possible to edit only modifiable parameters and to read the process image of the modules that own it).

Selecting serial module 750-652, it is possible to set the following parameters that will be used as default at each subsequent reset:

Parameter	Value
COM-Port	COM1
Baudrate	115200
Parität	None
Datenbits	8
Stopbits	1
Timeout (ms)	500

Pressing the button to read the process image, a window like this one is shown in which user can read values stored in Status, Control and Data registers, both Input and Output.

Pos. : 750-650/003-000		
RS 232 C Interface (Adjustable)		
Byte	Output	Input
CT/ST	0x00	0x00
D0	0x00	0x00
D1	0x00	0x00
D2	0x00	0x00

For every further detail, please check official Wago website and their user manual following the link:

<https://www.wago.com/global/software/wago-i-o-check/p/759-920#downloads>

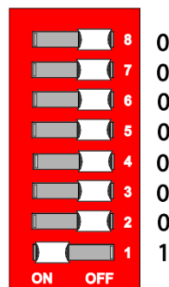
## VI. PC configuration

**Prerequisite:** to use Gimatic application, the PC must have a Windows operating system.

Once wirings have been verified, proceed over configuring the pc to communicate with plc setting a static IP:

- Ip address: "192.168.1.X" (constrain  $1 < X < 255$ , in this example  $X=2$ )
- Subnet mask: "255.255.255.0"

Check the correctness of the settings by typing in the search bar of the pc browser the ip address of the plc (the last byte of this address can be changed through the dip-switches on the plc header, for simplicity "192.168.1.1" setting them as shown in the figure):



If everything is correct, it would be opened a page like this:

**WAGO** Web-based Management

Navigation:

- Information
- Ethernet
- TCP/IP
- Port
- SNMP
- SNMP V3
- Watchdog
- Security
- Modbus
- EtherNet/IP
- Features
- IO config
- Disk Info

**Status information**

**Coupler details**

Order number	750-352
Mac address	0030DE0CD6A6
Firmware revision	01.09.21 (14)

**Actual network settings**

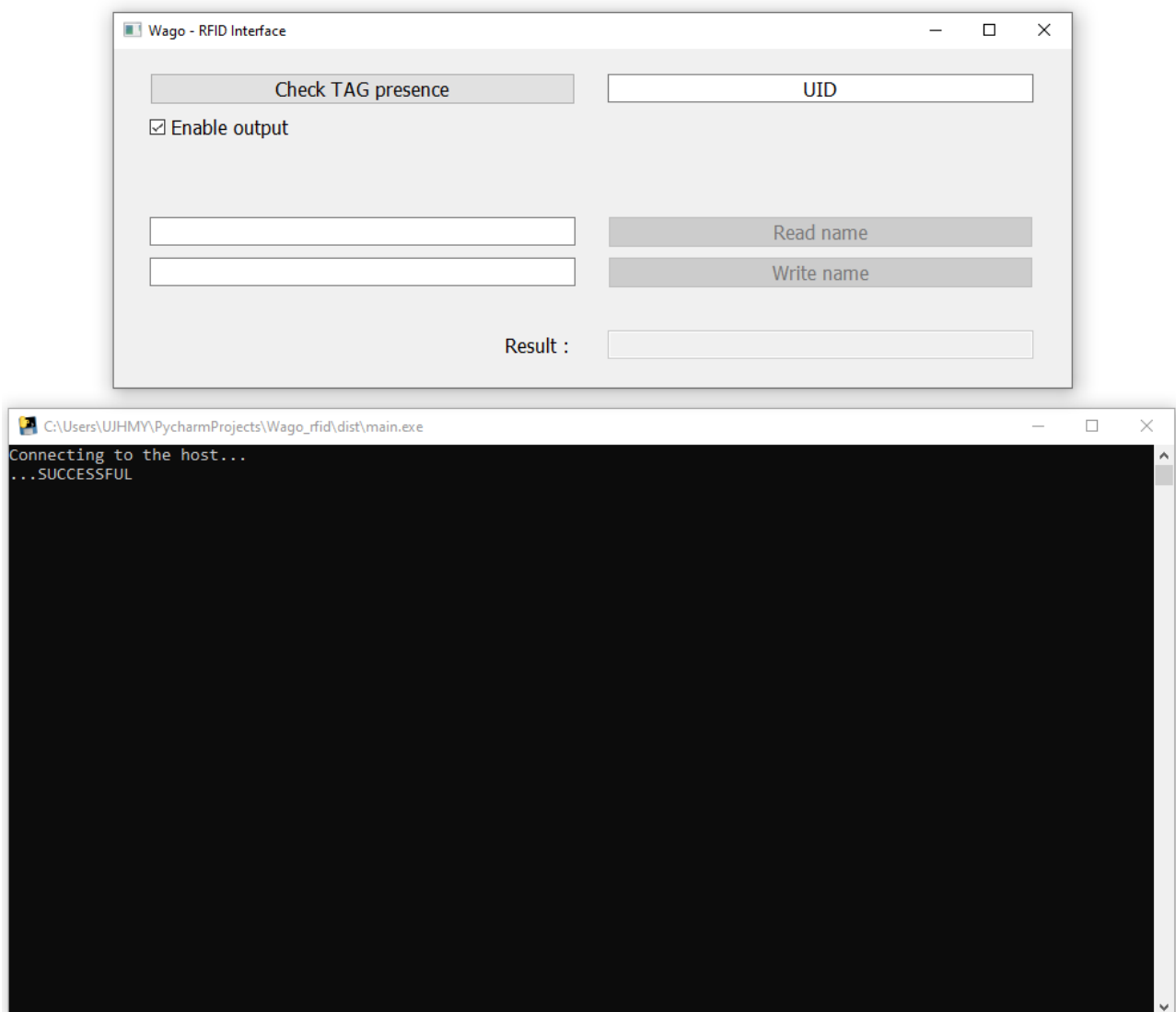
IP address	192.168.1.1
	Determined by Dip Switch
Subnet mask	255.255.255.0
Gateway	0.0.0.0
Host Name	0030DE0CD6A6
Domain Name	
DNS-Server 1	0.0.0.0
DNS-Server 2	0.0.0.0

**Module status**

State Modbus Watchdog	Disabled
Error code	0
Error description	0

## VII. Gimatic interface

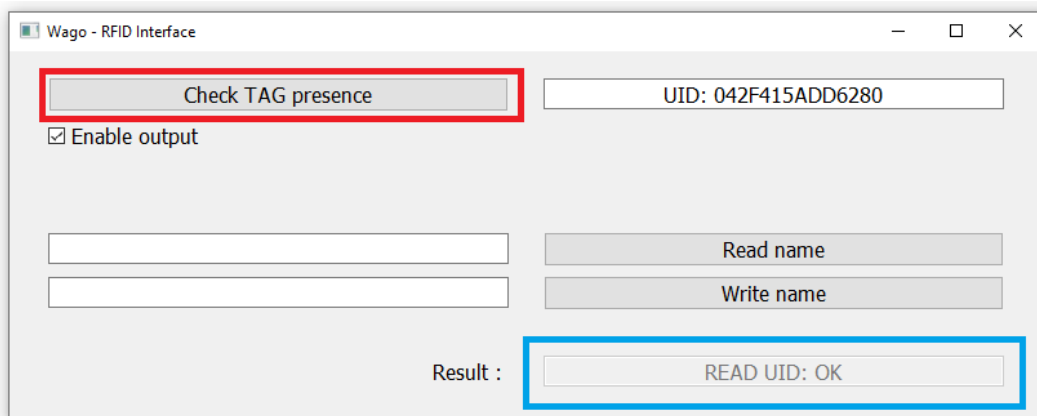
At startup, the GUI and its command prompt look like this:



According to button pressed, user can:

- 1- Check tag presence reading its UID (Unic Identifier). If present, it is shown in the relative box "UID" and the two buttons "Read name" and "Write name" became available.
- 2- Enable/disable RAQC outputs by ticking "Enable output" checkbox
- 3- Read and write RFID tag's name
- 4- Check the result of the latest operation done in "Result" box

Below are some screenshots captured at the end of each command execution:



```

C:\Users\UJHMY\PycharmProjects\Wago_rfid\dist\main.exe
IA set: True
IR reset
Checking IA flag...
IA set: False
I/O module ready to operate
---END SETUP---
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
---TX COMPLETED---
Checking RR flag ...
RR correctly detected
Set RA bit
PARSE TO 8 BIT
Set RA bit: False
PARSE TO 8 BIT
dataReceived len= 27 [1, 70, 70, 2, 49, 56, 48, 48, 49, 52, 48, 52, 50, 70, 52, 49, 53, 65, 68, 68, 54, 50, 56, 48, 3, 2, 13]
---RX COMPLETED---
CRC VERIFY
CRC verified correctly
STX + '1'8' + 0x3030
OK: Lettura eseguita correttamente
type= 14
uid = 042F415ADD6280

```

Pressing “Check tag presence” button, if the tag (RBQC) is around the reader and properly working, in the “Result” box will be shown the message *READ UID: OK*, while in command prompt will be shown both the type and the tag UID.

To enable or disable outputs by checking the box "Enable output", you will get these two screenshots (the successful outcome, in addition to the field "Result", is also displayed in the command prompt through the ACK message):

Wago - RFID Interface

Check TAG presence

UID: 042F415ADD6280

☐ Enable output

Read name

Write name

Result : SET OUTPUT False: OK

```

C:\Users\UJHMY\PycharmProjects\Wago_rfid\dist\main.exe
uscite disattivate
CRC CALCULATE
buffer to send: len=23 [1, 70, 70, 2, 50, 70, 70, 70, 55, 56, 49, 48, 48, 48, 48, 48, 49, 48, 49, 3, 122, 13]
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
---TX COMPLETED---
Checking RR flag ...
RR correctly detected
Set RA bit
PARSE TO 8 BIT
Set RA bit: False
PARSE TO 8 BIT
dataReceived len= 6 [1, 70, 70, 6, 7, 13]
---RX COMPLETED---
CRC VERIFY
CRC ricevuto corretto
ACK

```

Wago - RFID Interface

Check TAG presence

UID: 042F415ADD6280

☒ Enable output

Read name

Write name

Result : SET OUTPUT True: OK

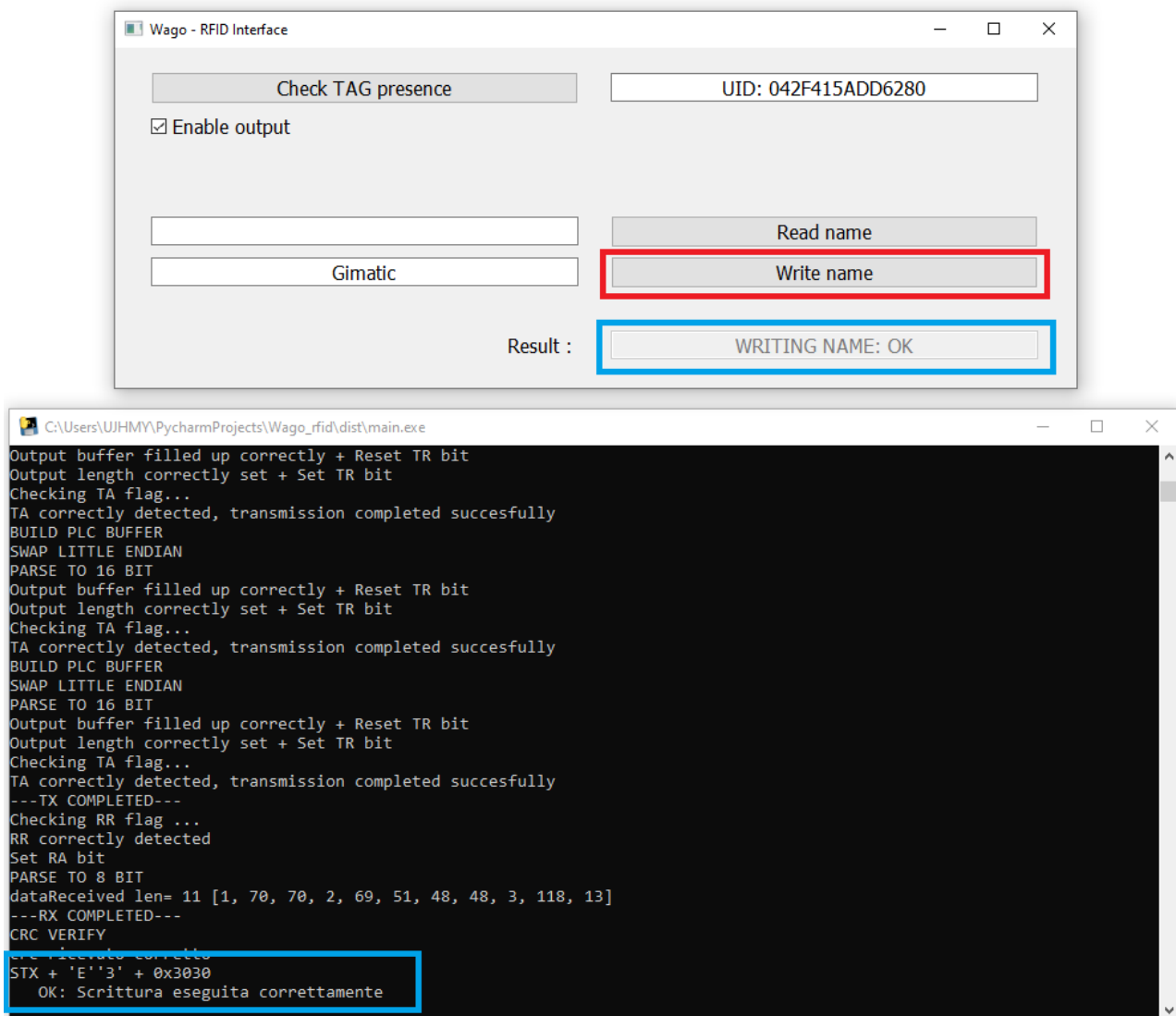
```

C:\Users\UJHMY\PycharmProjects\Wago_rfid\dist\main.exe
uscite attivate
CRC CALCULATE
buffer to send: len=23 [1, 70, 70, 2, 50, 70, 70, 70, 55, 56, 49, 48, 48, 48, 48, 48, 49, 48, 48, 3, 123, 13]
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
---TX COMPLETED---
Checking RR flag ...
RR correctly detected
Set RA bit
PARSE TO 8 BIT
Set RA bit: False
PARSE TO 8 BIT
dataReceived len= 6 [1, 70, 70, 6, 7, 13]
---RX COMPLETED---
CRC VERIFY
CRC ricevuto corretto
ACK

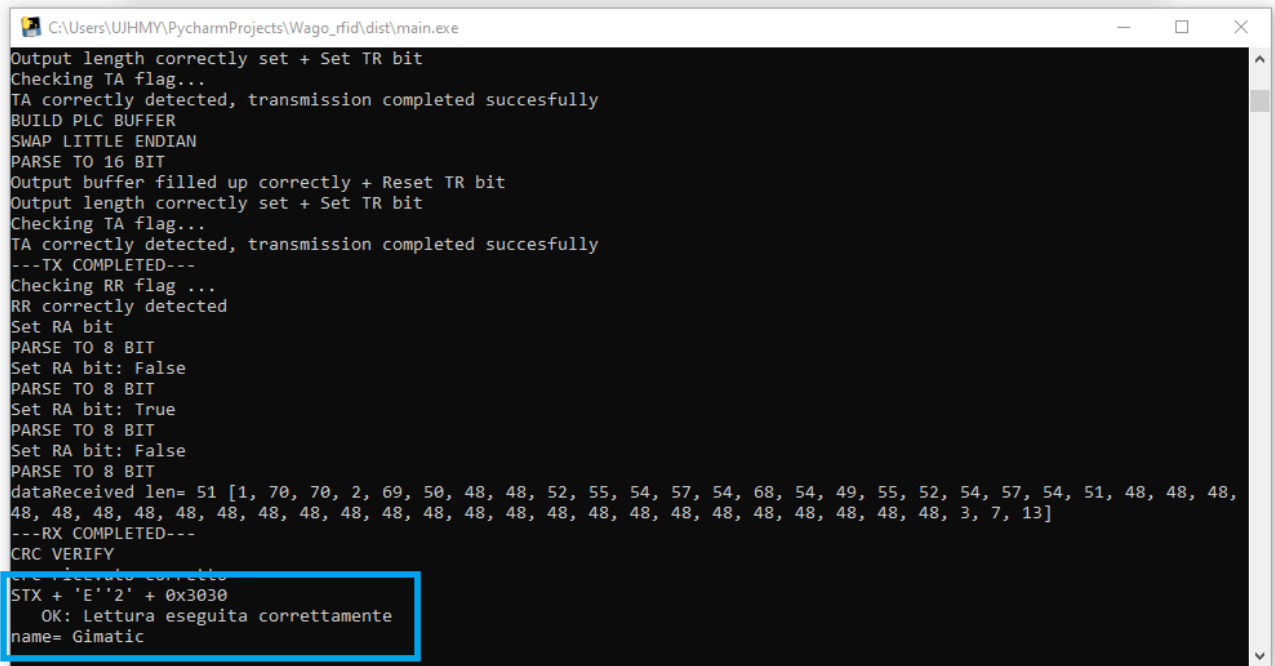
```



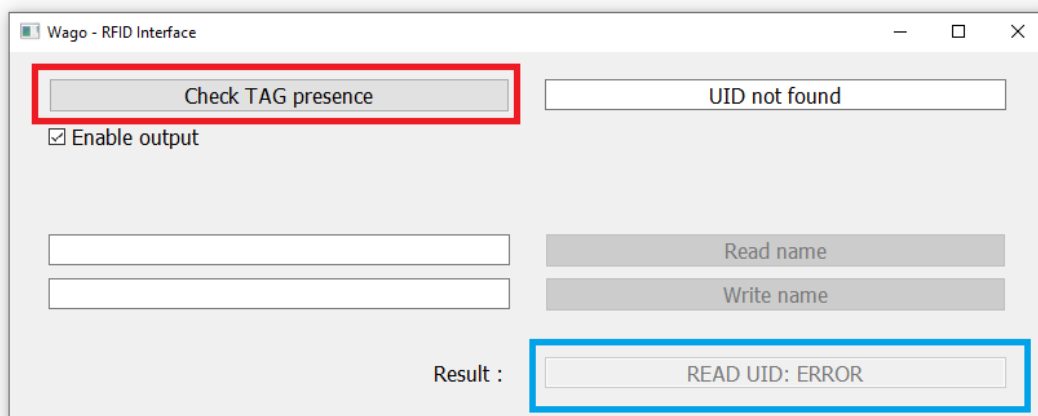
Pressing “Write name” button, the successful writing operation is displayed as follows:



Similarly, pressing “Read name” button, the successful reading operation is displayed as follows:



33



```
C:\Users\UJHMY\PycharmProjects\Wago_rfid\dist\main.exe
Error in setting IR bit
IR set
Checking IA flag...
IA set: True
IR reset
Checking IA flag...
IA set: False
I/O module ready to operate
---END SETUP---
BUILD PLC BUFFER
SWAP LITTLE ENDIAN
PARSE TO 16 BIT
Output buffer filled up correctly + Reset TR bit
Output length correctly set + Set TR bit
Checking TA flag...
TA correctly detected, transmission completed succesfully
---TX COMPLETED---
Checking RR flag ...
RR correctly detected
Set RA bit
PARSE TO 8 BIT
Set RA bit: False
PARSE TO 8 BIT
dataReceived len= 11 [1, 70, 70, 2, 49, 56, 48, 49, 3, 8, 13]
---RX COMPLETED---
CRC VERIFY
CRC ricevuto corretto
STX + '1'8' + 0x3031
  Errore: TAG non presente
```