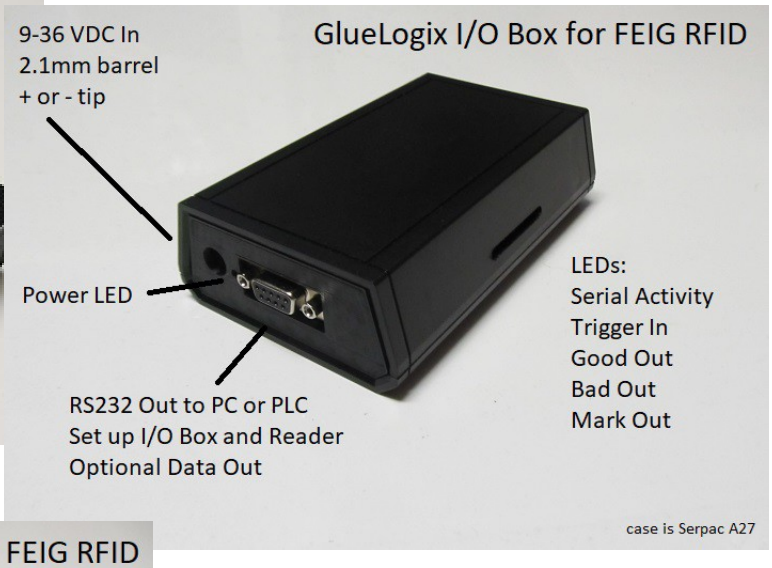
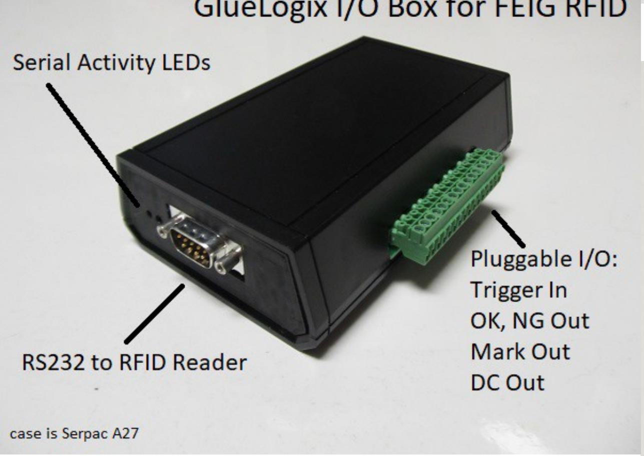
GlueLogix IO Box for FEIG RFID Readers
User Guide
Larry Martin, GlueLogix Inc.
Larry@GlueLogix.com
USA 919.342.0201
Rev A, 01Jul2019

1. Theory of Operations

The GlueLogix IO Box for FEIG RFID Readers provides a convenient interface between the data driven world of RFID and the I/O driven world of automation. The IO Box is meant to connect between a PLC or other automation controller and an RFID reader. In response to a trigger, the IO Box sends a serial command, parses the reply, and sets OK or NG outputs as appropriate. The Host side data connection is optional and may be left disconnected after nonvolatile setup.

2. Outside View



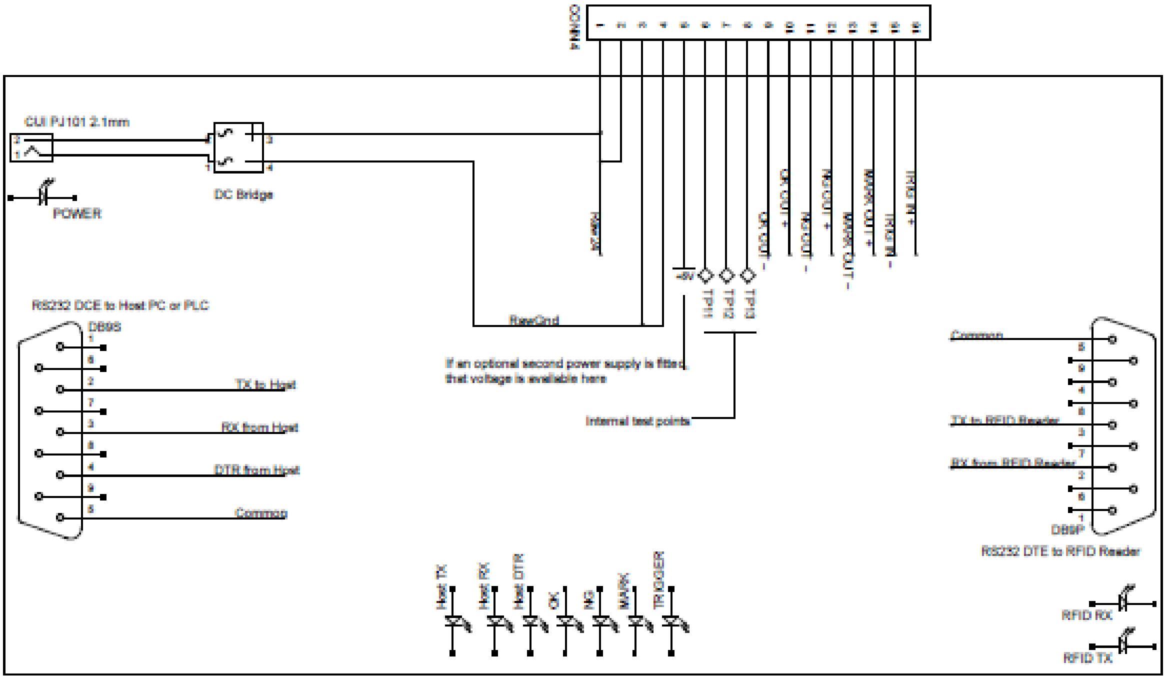


2.1. Case

The case is a Serpac A27 with 3d printed end plates.

http://www.serpac.com/userprints/A27.pdf

3. Connections



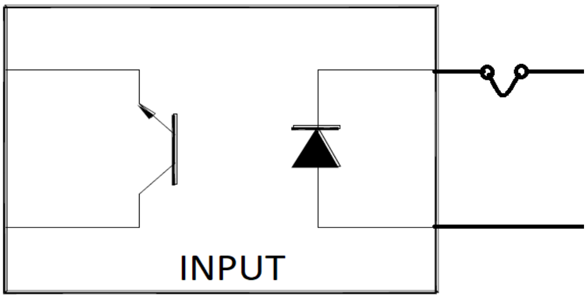
Connectors, connector pins and LED orders in the figure above match the shipped hardware.

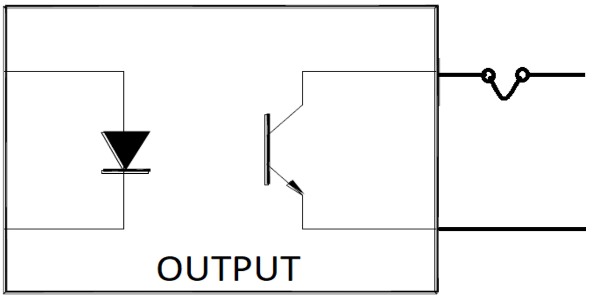
3.1. I/O Circuits

Socketed Lite-On LTV817 chips are used for I/O. Each I/O point has an optional current limiter in line, shown as a wire loop in the diagram. The board has a pair of pin sockets. By default, a wire is installed in the pin sockets for no current limiting. If you open the cover, you can easily replace those wire segments with through hole resistors.

External logic Asserts the input by passing current through the LED half of the LTV817.

The IO Box Asserts its output by forward biasing the transistor half of the LTV817, allowing current to flow in external logic.





The I/O scheme of the IO Box is more like a relay or SSR interface than standard NPN or PNP logic, so it can easily interface with either.

3.2. Auxiliary Power Supply

On request, GlueLogix can install a secondary internal power supply and expose that voltage on the I/O Header. This feature can be useful for powering certain FEIG RFID readers that require 5VDC from the 24V more common in automated systems.

3.3. Trigger

The IO Box trigger is coupled to the Trigger LED, last on the right. The LED is ON when current is flowing through the trigger circuit. By default, the IO Box performs its trigger actions on the dark edge, i.e., the event when the Trigger LED goes from light to dark. That sense can be reversed by setting the InvertGap flag, value 0x10 in location 0x0C.

The active edge of the trigger causes the I/O Box to send its configured command and begin processing replies and retries.

The inactive trigger edge (dark to light by default) causes the status outputs and LEDs from the previous cycle to be cleared. Therefore, the IO Box can be run as a read window device, in which the status outputs are managed by handshake:

- Host asserts trigger,
- IO Box sends commands and sets status,
- Host releases trigger,
- IO Box releases status lines.

Alternately, the IO Box can be run with a pulsed trigger, in which the status LEDs persist until the next cycle, by simply making the trigger pulse the inactive sense (by default, asserted, current flowing, LED On).

Trigger pulses should be at least 1 mS duration.

3.4. Power

Power must be 9 to 36 VDC. The barrel connector next to the host serial port is the main power input. It is a 2.1x5.5mm standard power connector. Voltage can be fed either plus tip or minus tip.

Input voltage passes through a diode bridge that rectifies its polarity. The rectified voltage can be fed out to sensors, RFID reader and other targets on the I/O Connector. Power can also be fed in on the I/O Connector, but it **must be correct polarity** or the IO Box may be damaged.

3.5. Communications

The Host and Target ports are both DB9 RS232.

The Target port is a male DB9P pinned as Data Terminal Equipment (DTE). On the Target side, only 3 pins are active:

Pin	**Function**
2	RX From Target
3	TX To Target
5	Common

The Host port is a female DB9S pinned as Data Communications Equipment (DCE). On the Host side, the DTR port is also active:

Pin	**Function**
2	TX To Host
3	RX From Host
4	DTR From Host
5	Common

Default communication settings are 115200 bits per second, No Parity, 8 Data, 1 Stop. Alternate settings may be requested at order time, or changed by serial command after delivery.

When a command is sent that changes the communication settings, that command is replied to at the old settings. Settings are immediately changed to the new settings, so the next command must be sent at the new baudrate. All settings are Volatile until saved to memory by the command **Q04!<enter>**, see the **Serial Commands** section.

One of the functions of the IO Box, shared with all GlueLogix TCU devices, is to act as a hub for serial port communications. Depending on signals and settings, the Host side serial port of the IO Box can talk to the IO Box itself or to the downstream RFID Reader.

The DTR signal is the primary means of switching communication targets. When DTR is asserted (positive voltage), the Host port bypasses the IO Box and talks directly to the downstream RFID reader. Most PC software will set DTR by default when opening a serial port, so this arrangement allows standard software for any given RFID reader to talk through the IO Box as if it weren't there. This state, called Bypass mode, is very useful for reader setup.

When DTR is released (negative voltage), the Host port talks directly to the IO Box microprocessor. This state is called Command mode.

Secondary means of switching, unique to the I/O Box, are:

- RS232 Break (TX line asserted for longer than a word time) in Bypass Mode forces a switch to Command Mode.
- Reception of a Trigger in Bypass mode forces a switch to Command Mode.
- Sending the byPass command P00!<enter> in Command Mode switches to Bypass Mode. Note: The opcode B is in the hexadecimal range and is reserved for direct memory access.

3.6. Programming

Firmware may be updated by use of a PropPlug adapter from Parallax Inc, 32201-ND at Digikey:

https://www.parallax.com/product/32201

The programming pins are located at the left end of the LED opening. The PropPlug attaches with component side down, toward the circuit board.

Factory default parameters will be restored during programming.

4. Serial Commands

Except where noted, IO Box commands are ASCII, case sensitive, and may be terminated with any combination of **<CR>** or **<LF>** characters, shown by **<enter>**. IO Box replies always end with the **<CRLF>** combination.

4.1. Direct Commands

The Direct Commands comprise an opcode, two character value, and a check byte.

4.1.1. Direct Command Opcodes

The opcodes 0-9 and A-F (case sensitive) provide direct write access to the first 16 locations of user memory, which are all configuration registers.

Other opcodes used by the I/O Box are:

Q Query. Examples:

Query Status Registers: **Q00!<enter>**

Query Target Serial Buffers: **Q01!<enter>**

Return to Factory Defaults: **Q03!<enter>**

Save Setup to NVM: **Q04!<enter>**

There are undocumented internal checks at Q02!, Q05! and Q06!. Users should avoid them.

Other Hex Values: dumps configuration memory: **Q0F!<enter>**

O Output, triggers the I/O Reply since there is only one 32 bit I/O array: **O00!<enter>**

V Version: **V00!<enter>**

R Read one time as if a trigger has arrived: **R00!<enter>**

r Release the read state or "unRead". Increments the internal trigger count (aka "gap count")

P Switch to byPass Mode, connecting the Host and Target ports directly: **P00!<enter>**

p Bypass or reactivate the Trigger input, remaining in command mode:

Bypass or pause trigger: **p01!<enter>**

Reactivate or unpause trigger: **p00!<enter>**

L Light Test
Turn off all lights: **L00!<enter>**
Turn on the Green light or OK Output: **L01!<enter>**
Turn on the Yellow light or NG Output: **L02!<enter>**
Turn on the Red light or Mark Output: **L04!<enter>**

G Reset the Gap Count: **G00!<enter>**

4.1.2. Direct Command Check Byte

The Direct Command includes a check byte, the binary XOR of the Opcode byte and the two Value bytes. Unlike the rest of the protocol, the Check Byte must be transmitted in binary.

If an exclamation point ('!', "bang", ASCII 0x21) is sent in this space, the XOR function is overridden. All example commands in this document use the '!' as an XOR override.

4.2. Hex Records

Hex Records are used to read or write any location in memory. They are not case sensitive.

The overall length of any serial message is limited to 128 bytes (0x80). Therefore, the data length of hex WRITE records is limited to 128 bytes minus the 11 byte length of the header, or 117 bytes (0x75)

4.2.1. Hex Record Prefixes

A READ Hex Record starts with a semicolon (';'). Any content after the data length will be ignored.

A WRITE Hex Record starts with a full colon (':') like a classic Intel Hex Record, https://en.wikipedia.org/wiki/Intel_HEX

4.2.2. Hex Record Addressing

Hex Records for the IO Box have a 4 byte address represented by 8 characters in ASCII-HEX range 0-9 and A-F. Hex Records are addressed relative to the first variable, not absolute, so that hex records can be used across firmware versions as long as the key variables detailed here are maintained in the same pattern by GlueLogix.

4.2.3. Hex Record Data Length

Data length is a single byte represented by two ASCII-HEX characters.

4.2.4. Hex Record Data

Data is represented by an unbroken ASCII-HEX string, double the data length because each data byte is represented by two characters.

Data is both directions is terminated by <CRLF> (0x0D,0x0A), which also ends the Hex record.

4.3. Example

The following illustrates writing the first 16 configuration registers by direct command and reading them back out by Hex Record:

```
Commands:
000!
100!
200!
300!
400!
500!
600!
700!
802!
901!
A00!
BC9!
C44!
D28!
E80!
F00!
;0000000010

Final Reply:
;000000001000000000000000000020100c944288000
```

4.4. Memory Map

Direct Memory commands and Hex Records access the same configuration memory space:

4.4.1. Configuration Registers

```
    Byte LLParams[kLLParamsSize] 'LLP_Bank                          '00..1F
```

The 32 configuration registers map as follows. Unused locations are maintained for similarity to other GlueLogix designs.

Address	Function
0x00..0x03	Unused

Address	Function
0x03	LSB (rightmost) byte of the Tag ID in the RFID reader reply, override program defaults.
0x04	Number of bytes to report for Tag ID, override program defaults.
0x05..0x0A	Unused
0x0B	Configuration Flag B, used bits as follows: 0x01 Pause (set) or Unpause (clear) automatic actions, see **p00!<enter>** 0x02 Command Mode, see **P00!<enter>**
0x0C	Configuration Flag C, used bits as follows: 0x10 Invert Gap, reverse the sense of the Trigger input 0x20 FEIG Advanced Protocol
0x0D..0x16	Unused
0x17	Serial Baud and Flags, bits as follows: 0x01..0x08 Low Nibble is Baud Rate, as follows: 1 => 9600 2 => 14400 3 => 19200 4 => 38400 5 => 57600 6 => 115200 0x10 Even Parity 0x20 Odd Parity Data is always 8 bits with 1 Stop bit. 115200 bps N81 encodes as 0x06
0x17..0x1C	Unused
0x1D	Target Flags, bits as follows: 0x01..0x08 Unused 0x10..0x80 RFID Retries A value of 0x30 configures up to 3 retries of each Trigger command before the NG bit is set.
0x1E, 0x1F	Unused

4.5. *Trigger Command*

```
Byte bTTBCommandLength                                              '20
Byte TargetTxBuf[kTargetTxBufSize]                                  '21..9F
```

This Hex record sets the Trigger Command to FEIG Inventory on Antenna 2. The first data byte is the command length, which will be written to address 0x20, immediately after the last Configuration Register. The last two bytes are the FEIG CRC.

```
:000000200B0A02000AFFB00110027426
```

4.5.1. Other

Further memory locations are undocumented and best left alone.

5. Terms and Abbreviations

$ Alternate prefix for Base 16 or Hexadecimal numbers, see also "**0x**".

0x Prefix for Base 16 or Hexadecimal numbers, from C language.

https://en.wikipedia.org/wiki/Hexadecimal

aka Also Known As

bps, baud, baudrate Speed of RS232 serial link in Bits Per Second, including all data and framing bits

CR, <CR> Carriage Return, the UNIX line end character, the ASCII character 13 (0x0D)

DTR Data Terminal Ready, a standard RS232 signal

LF, <LF> Line Feed, the ASCII character 10 (0x0A). <CRLF> is the Microsoft line end.

mS Abbreviation for Millisecond.

NVM, NVMEM Non Volatile Memory

RS232 Standard communications protocol: https://en.wikipedia.org/wiki/Serial_port

VDC Volts Direct Current, or Volts DC