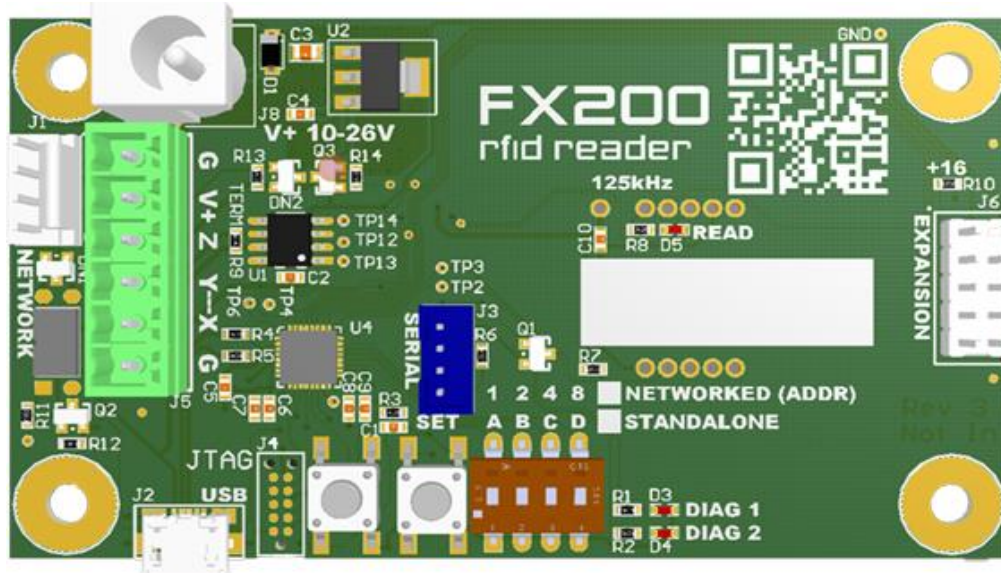


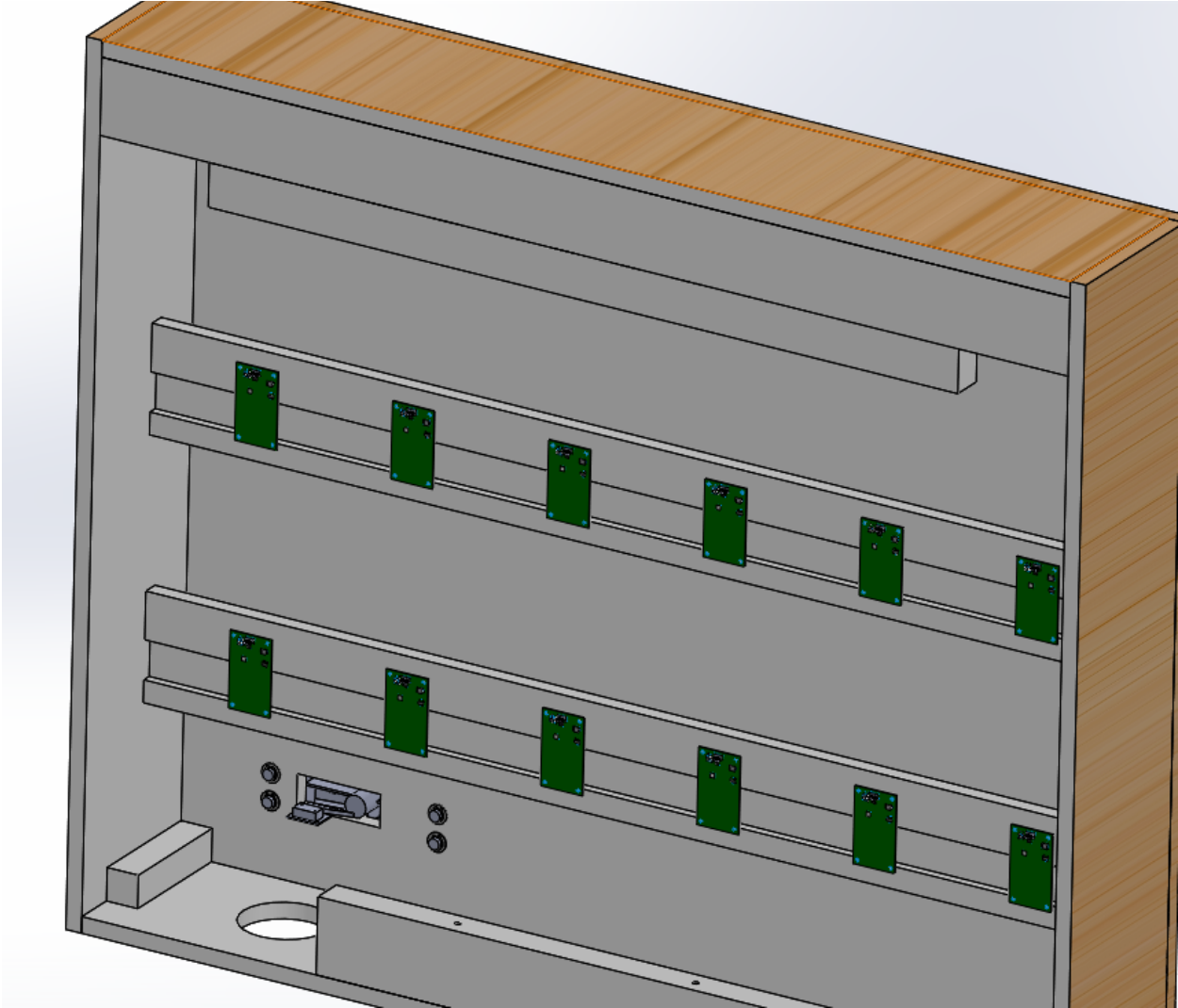
FX200 RFID Sensor Datasheet



October 15, 2018

Table of Contents

Overview	1
Diagnostic LEDS	1
DIAG1	1
DIAG2	1
READ	1
Set Switch.....	1
ABCD Switch.....	1
Configuration	2
Expansion Setting	2
Wiring	3
FX200s are wired in parallel.....	4
Using with FX Controllers	5
Instantiation	5
Available Library Functions.....	5
Read Modes	6
FX200 and FX450 Example	7
Using with other Controllers	8
Sending Commands.....	8
Receiving Replies.....	8
Mechanical Layout	9
Mounting Examples.....	10



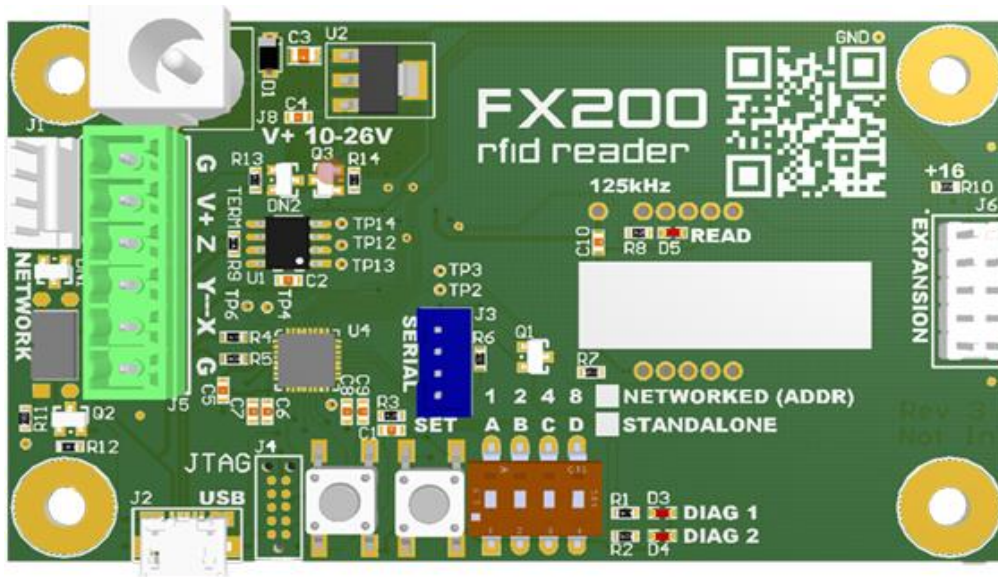
..... 10

Contact 12

References 12

Overview

The FX200 is an RFID sensor capable of reading 125kHz RFID tags. The sensors use RS485 for network communication, allowing for long wires (thousands of feet!) to be used.



Diagnostic LEDS

DIAG1

This LED blips off when the receiver resets, normally on.

DIAG2

This LED will blip on when a command is successfully received.

READ

This LED blips on when a tag is detected.

Set Switch






This switch is used to set the FX200 address after a change.

ABCD Switch

This is the address selection switch. See section **Configuration**

Configuration

Every FX200 needs to have its own address, starting from address 0. For instance, if you have 8 readers, you would use addresses 0 through 7. Set the FX200 switches to the addresses below. Note that if you change the address while the device is powered you must press the SET button to register the new settings.

Switch Setting	Add.	Switch Setting	Add.	Switch Setting	Add.	Switch Setting	Add.
	0		8	 + Expansion	16	 + Expansion	24
	1		9	 + Expansion	17	 + Expansion	25
	2		10	 + Expansion	18	 + Expansion	26
	3		11	 + Expansion	19	 + Expansion	27
	4		12	 + Expansion	20	 + Expansion	28
	5		13	 + Expansion	21	 + Expansion	29
	6		14	 + Expansion	22	 + Expansion	30
	7		15	 + Expansion	23	 + Expansion	31

Expansion Setting

To set addresses 16 through 31, the expansion option must be enabled by bridging pins 1 and 7 at J6.



Wiring

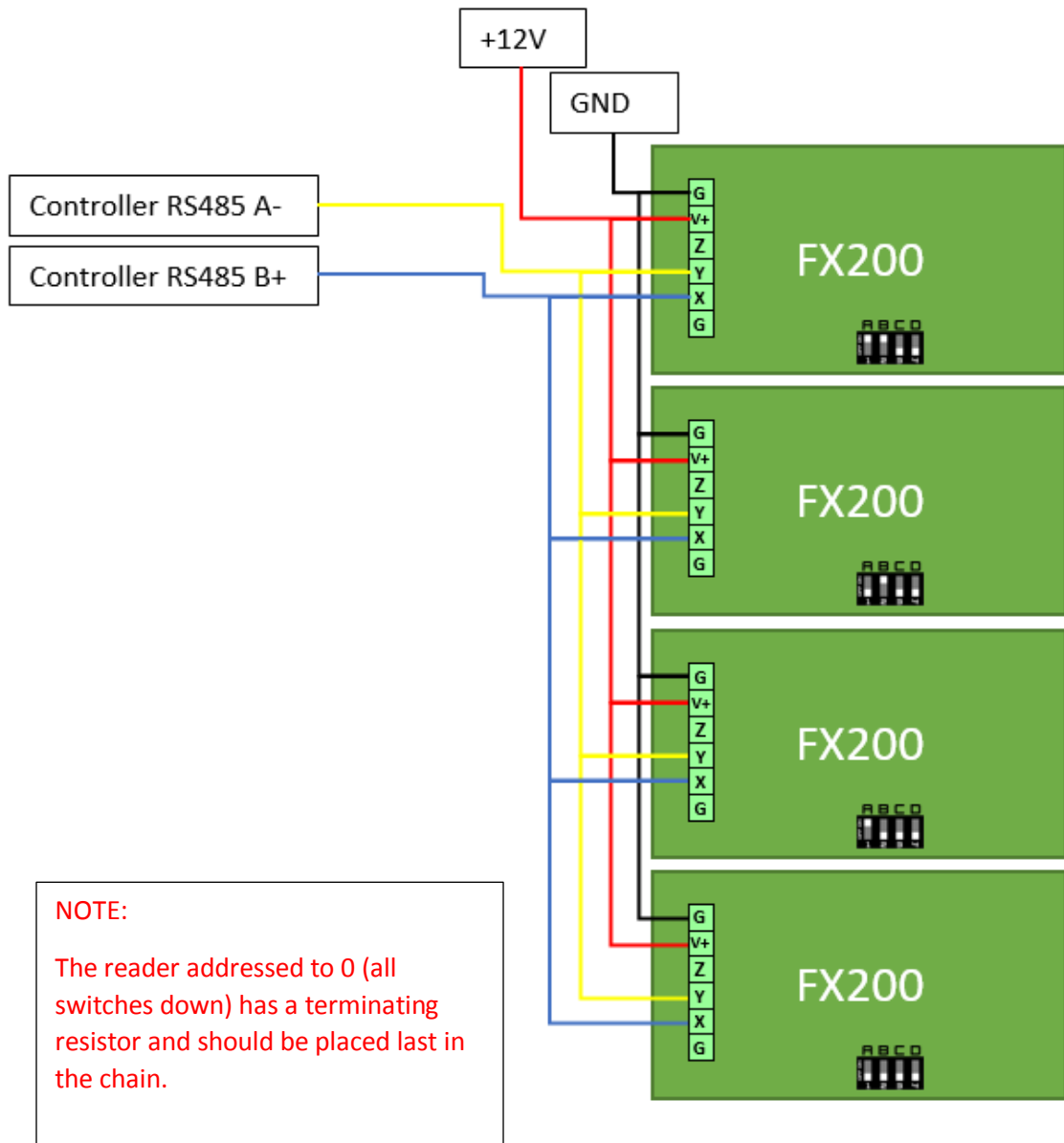
For the RFID based games, the FX200 is used. The FX200 is a networkable (not Ethernet network) reader which can be placed far distances (>100ft) around a room for optimal placement. The connection to the readers is located INSIDE the BAC. This connection and wiring are done for you when you purchase readers with the BAC from Escape Room Techs. If you are modifying or piecing together a system, please consult Escape Room Techs for the procedure to wire the readers into the BAC.

When placing your readers around the room, keep the reader with address 0 (all switches down) at the end of the line. It has a special hardware configuration and will work best at the end.

Some uses may wish to use CAT5/6 wiring for their RFID readers. In this case, wire like this:

CAT5/6 or other 4x Twisted Pair Wiring	Signal	BAC/FX450 Connection	FX200 Connection
Orange Pair (both wires)	V+ (12V to 24V)	V+	V+
Green Solid	RS-485 A-	A	Y
Green Strip	RS-485 B+	B	X
Brown Pair (both wires)	Ground	G	G
Blue Pair	Not used	-	-

FX200s are wired in parallel.



Using with FX Controllers

An FX200 library is included with the Arduino packages for the FX350 and FX450 controllers. This library makes using the FX200 network simple and easy.

Add the following include to use the library:

```
#include "fx200.h"
```

Instantiation

The following is an example configuration for 3 FX200s reading in EVENODD mode.

```
// Number of readers
const int numberOfReaders = 3;

// Our FX200 configuration
FX200Configuration config
{
    /* Read Style:*/ RFIDReadStyle::EVENODD,
    /* RS485 Pin:*/ RS485_ENABLE,
    /* RS485 Uart:*/ RS485_SERIAL,
    /* Read Time:*/ 400,
    /* Expire Time:*/ 2000
};

// Instantiate FX200 object with number of readers and read configuration
FX200<numberOfReaders> fx200(config);
```

Available Library Functions

void scan(byte index = 0);

Scans the RFID network for new data. In EVENODD and SEQUENTIAL mode, index may be omitted as the library controls which reader to scan and when. In IMMEDIATE mode, index is used to indicate which RFID reader to scan.

void setSolvedTag(byte index, const char* data);

Sets the solved tag for reader at address index using the string data pointed to by data. This is useful for restoring solved tag data read from storage (i.e. EEPROM).

bool isSolved();

Returns true if all the readers match their respective solve tag, otherwise false.

bool hasNewData(int index = -1);

Indicates if data of the FX200 at address index has changed. Calling this function resets the data flag, as does calling getTag(). If index is omitted, all tags are checked for new data.

bool learn();

Reads tag data of all readers. If all readers have a tag, the data is stored and this function returns true. Otherwise this function returns false.

void init();

Call to initialize the FX200 object. This function sets up Serial communication, sets the readTime and expireTime of each FX200, and clears any data stored by each FX200. If DEBUG_RFID is defined, this function also gets and prints the version of each reader.

String getTag(byte index);

Returns a string. Either "NONE" for no tag, "EXPIRED" to indicate expired data, or a tag ID if a tag is present.

Read Modes

RFIDReadStyle::EVENODD

Switches between scanning all even addressed and odd addressed FX200s at an interval of readTime ms (set in FX200Configuration). This is a fast read mode that will work for most configurations. If interference is encountered from adjacent readers, IMMEDIATE mode should be used instead.

RFIDReadStyle::SEQUENTIAL

Scans each reader one at a time in sequence every readTime ms (set in FX200Configuration). This mode is relatively slow compared to the other modes available.

RFIDReadStyle::IMMEDIATE

In this mode the developer has control over which readers are scanned and when. This is useful for avoiding interference when readers are very close together and helps with keeping overall read time low.

FX200 and FX450 Example

```
#include "fx200.h"
#include "fx450.h"

// Number of readers
const int numberOfReaders = 1;

// Our FX200 configuration
FX200Configuration config
{
  /* Read Style:*/ RFIDReadStyle::EVENODD,
  /* RS485 Pin:*/ RS485_ENABLE,
  /* RS485 Uart:*/ &Serial5,
  /* Read Time:*/ 400,
  /* Expire Time:*/ 2000
};

FX200<numberOfReaders> fx200(config);

void setup()
{
  Serial.begin(115200);

  // Initialize our object
  fx200.init();
}

void loop()
{
  if (digitalRead(INPUT0) == HIGH) learn(); // Learn the tags when INPUT0 is
high

  fx200.scan(); // Command the RFID network to scan, should be called in a
loop with minimal blocking

  if (fx200.hasNewData()) // Do we have new data?
  {
    // Iterate over all the readers and see what they have
    for (int i = 0; i < numberOfReaders; i++)
    {
      Serial.print(i); // Print the index of the reader
      Serial.print(": ");
      Serial.println(fx200.getTag(i)); // Print the tag data
    }
  }

  // If FX200 tags are learned, and they match, solve the game
  if (fx200.isSolved())
  {
    Serial.println("Solved");
  }
}
```

Using with other Controllers

To communicate with the FX200 network your own device must have RS485 capability. The following table is a list of all available commands.

Command	Example Command	Example Reply	Description
REQRFID	###255, 0, REQRFID###	###0, 255, :[TAGID]!!! ###0, 255, :NONE!!! ###0, 255, :EXPIRED!!!	This is the simplest request and form of operation. When sent, the reader will attempt to read a tag for readtime milliseconds then return the tag id, NONE, or EXPIRED
TRGXXXX	###255, 1, TRG400###	###1, 255, Read in: 400!!!	Set the reader to trigger a read sequence in XXXX milliseconds. Use 0 (zero) to trigger read immediately
READRFID	###255, 2, READRFID###	###2, 255, :[TAGID]!!! ###2, 255, :NONE!!! ###2, 255, :EXPIRED!!!	Read the tag data last read by the RFID Reader. Use after tags have been triggered to read (TRG command).
VERSION	###255, 3, VERSION###	###3, 255, May 09 2018:08!!!	Returns the date-time stamp of when the FX200 was programmed
TIMEXXXX	###255, 4, TIME1000###	###4, 255, Duration: 1000!!!	The time in milliseconds the reader will attempt to read a tag. Defaults 400ms. Max 9999ms. Min 200ms. Sets readtime.
EXPIREXXX	###255, 5, EXPIRE500###	###5, 255, Expire: 500!!!	The time in ms that the reader will host the last read value. (300-9999ms) Default 1000 second, min is readtime + 50 or 300 (whichever's greater)

Sending Commands

Commands sent to the FX200 must have the following format. See table above for examples.

###255, [address], [command]###

Start token – This is the start of the command, always "###"

Source (always 255) – This is the source of the command, 255 is the master address.

[address] – Address of the FX200 set in **Configuration**

[command] – The command from the table above

End token – This is the end of the command, always "###"

Receiving Replies

###[address], 255, [reply]!!!

Start token – This is the start of the command, always "###"

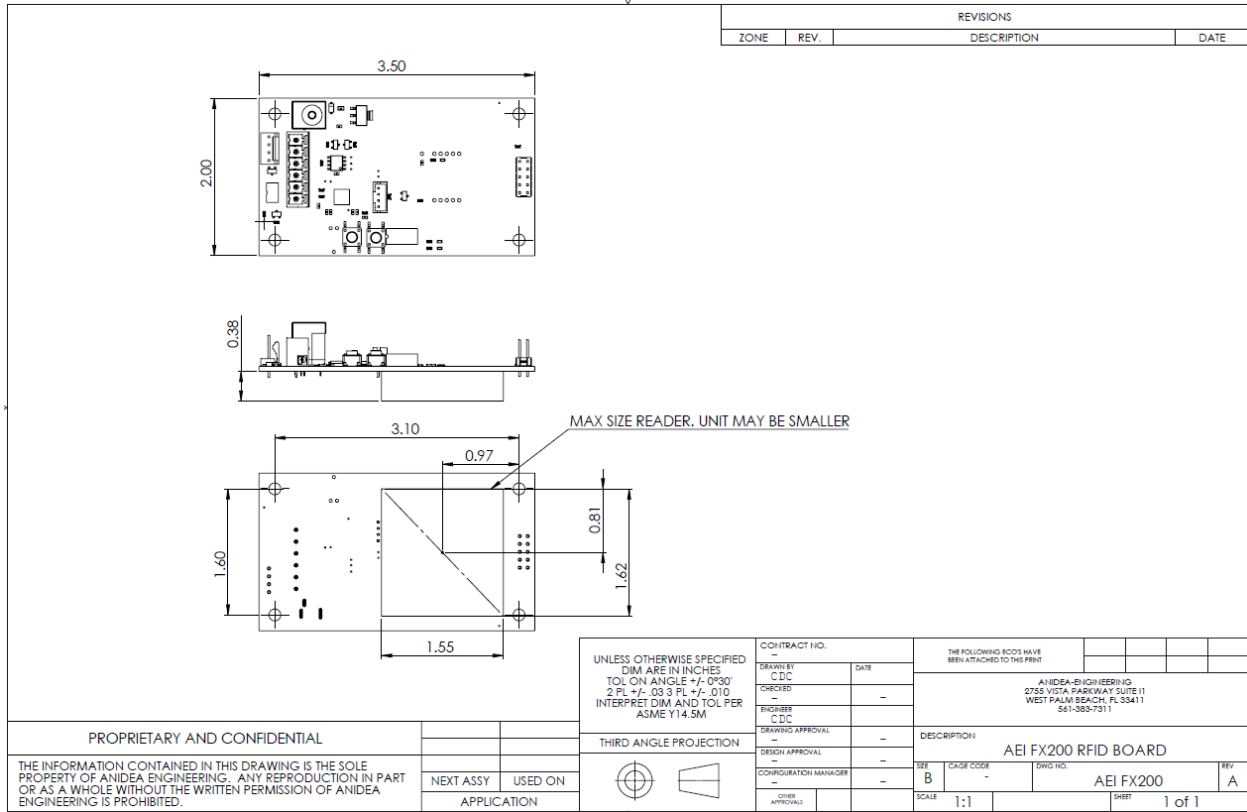
[address] – This is the address of the responding FX200

Destination – Address of the master, always 255.

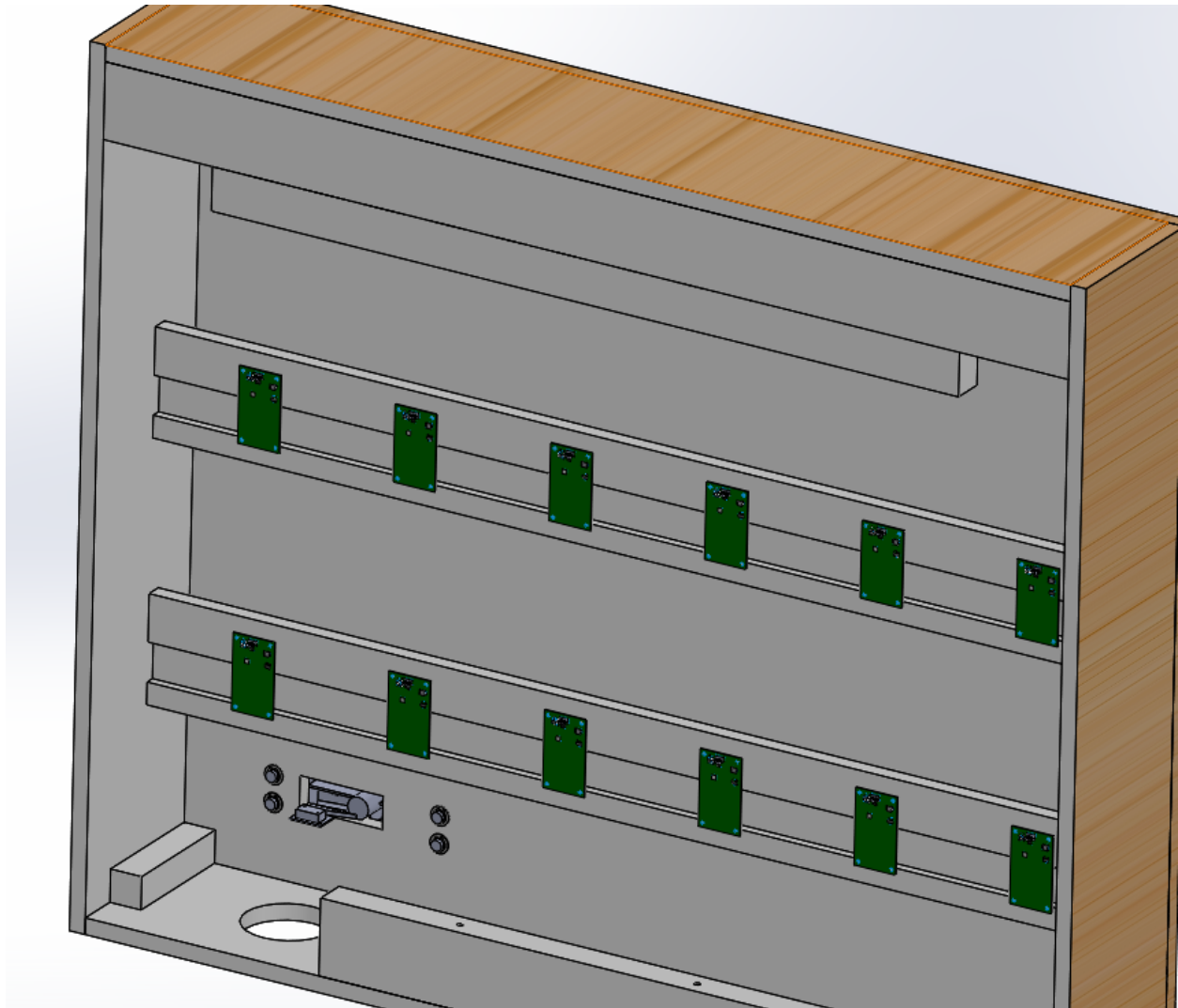
[reply] – The reply. Note that tag data is always prepended with : to make parsing easier. See table for example replies.

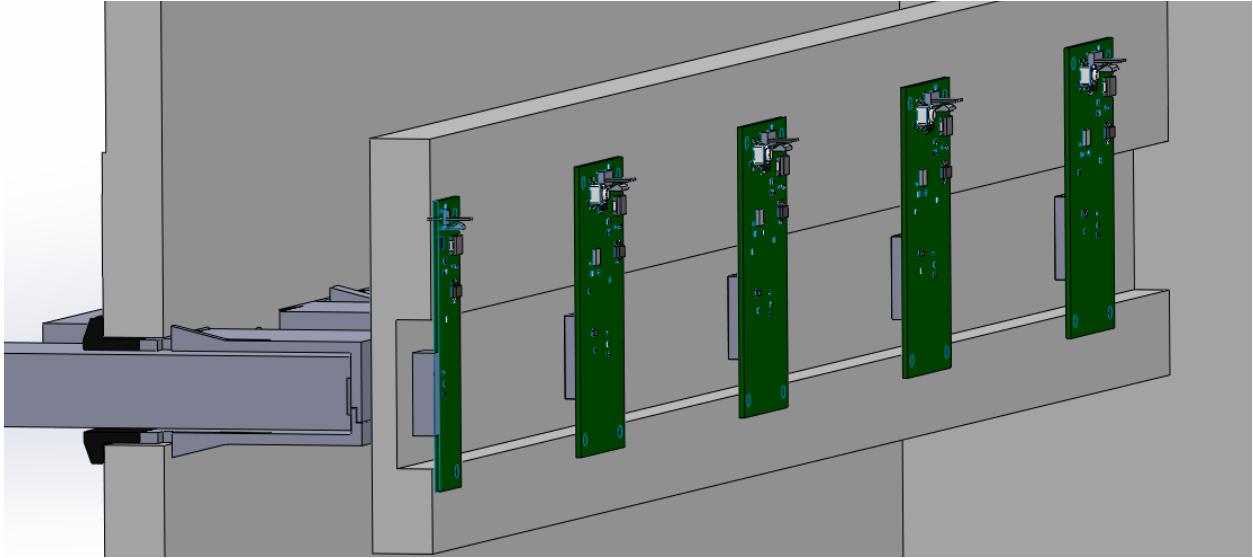
End token – This is the end of the command, always "!!!"

Mechanical Layout



Mounting Examples







Contact

Web: www.escaperoomtechs.com

Email: support@escaperoomtechs.com

Phone: 561-708-0007

References

Terminal Blocks: Onshore Tech

Header - OSTOQ063250 <https://www.digikey.com/products/en?keywords=OSTOQ063250>

Terminals - OSTTS06315B <https://www.digikey.com/products/en?keywords=OSTTS06315B>

2018/09/25