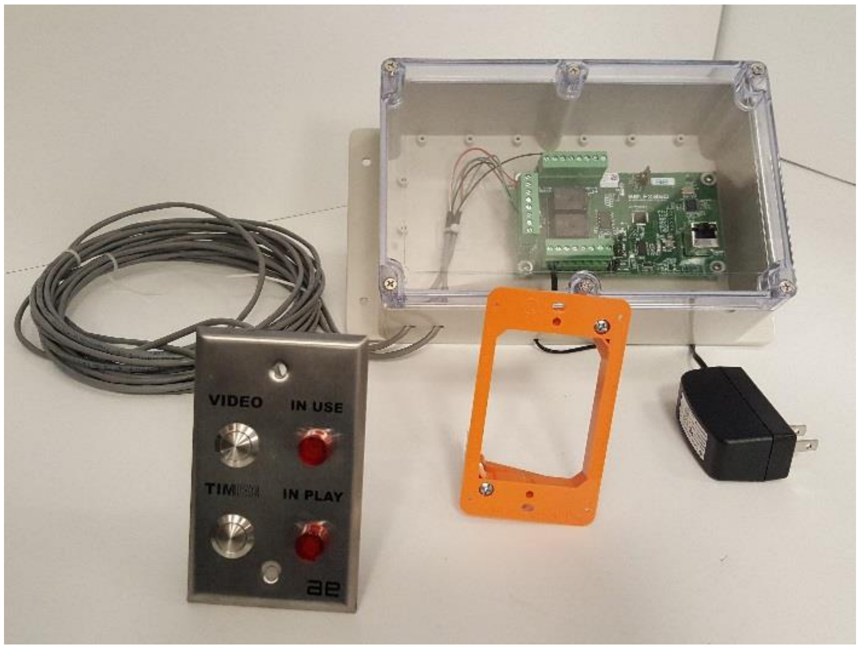
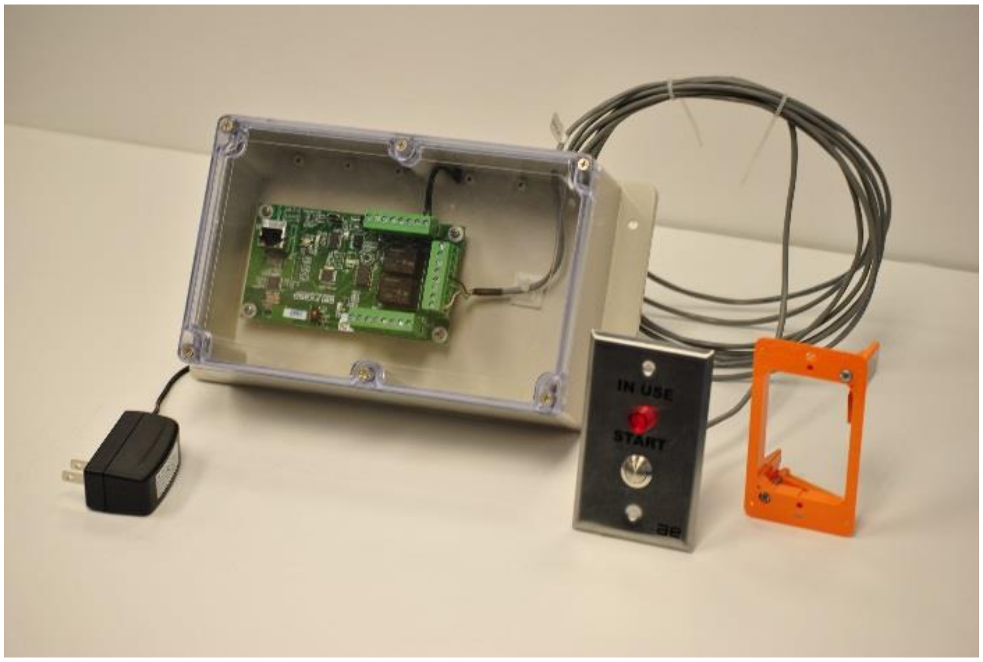
Ethernet Room Controller

Overview

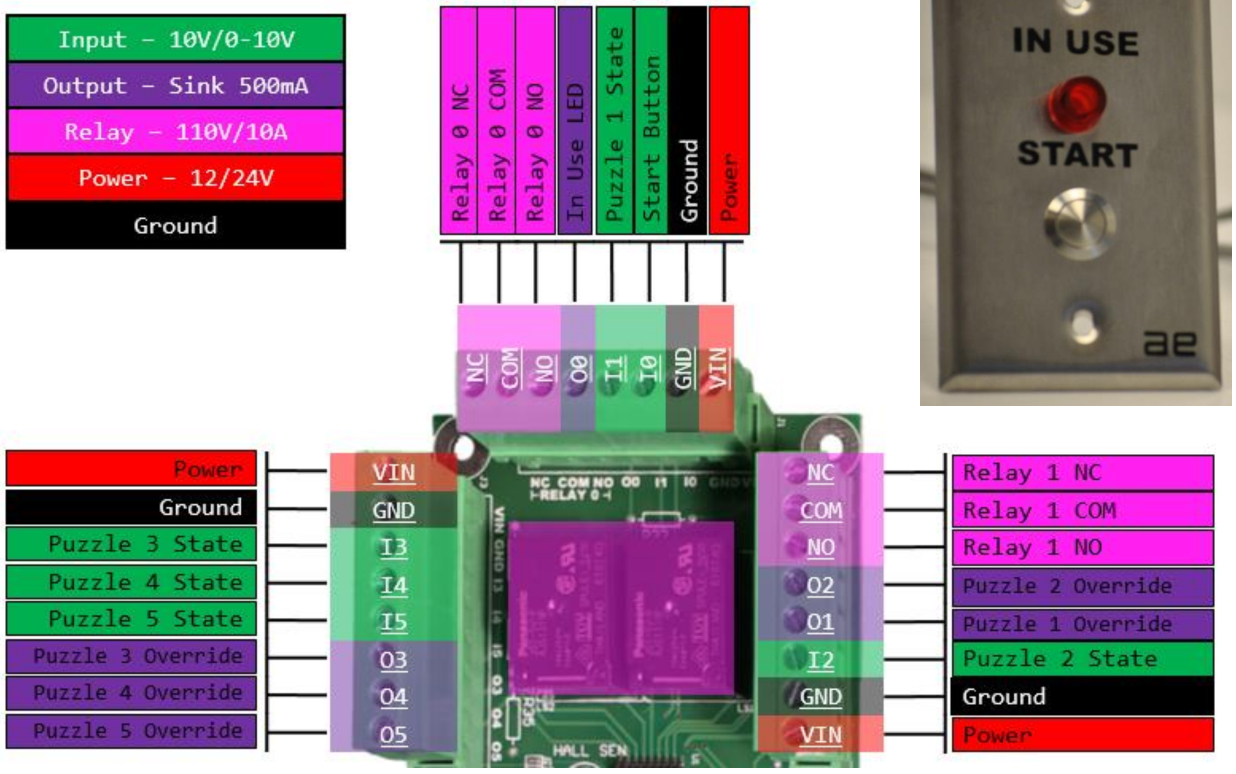
The FX350 ethernet-enabled room controller is designed to equip game masters with total room control. Monitor and override up to 6 props to ensure proper game flow, even when something goes awry.





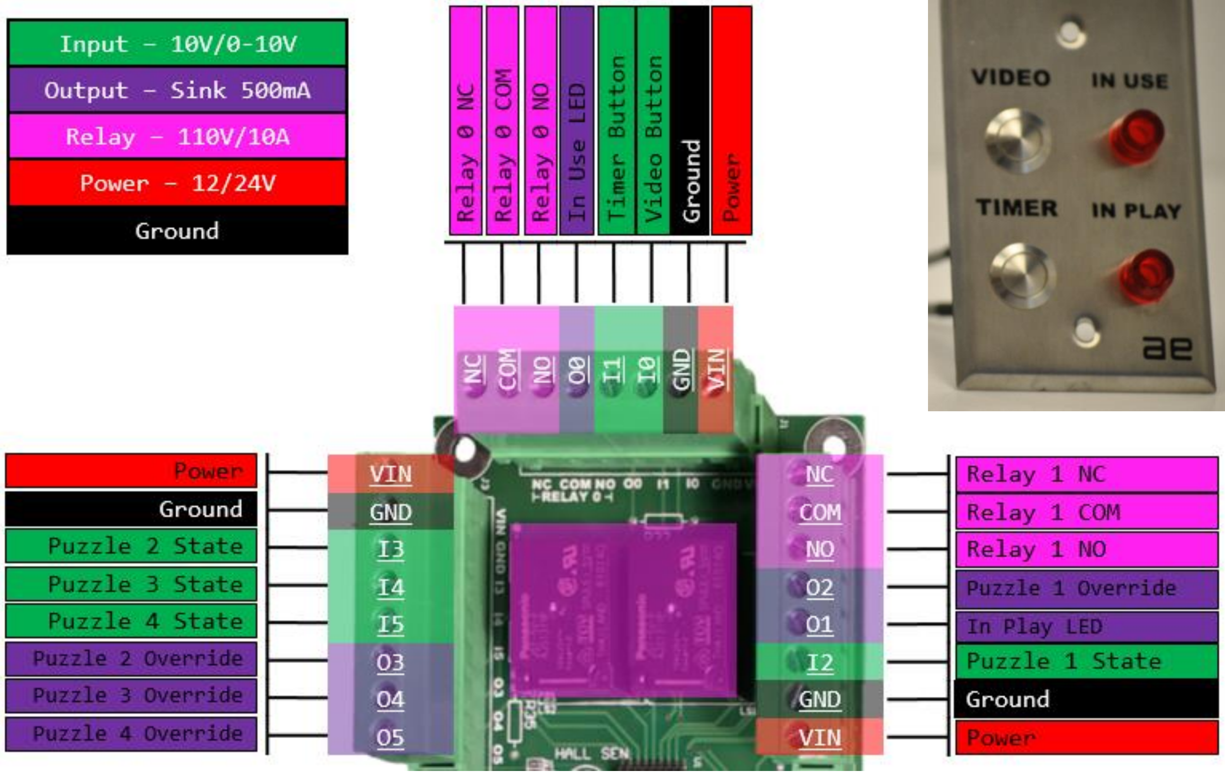
Kit Configurations

1 Button 1 LED Version



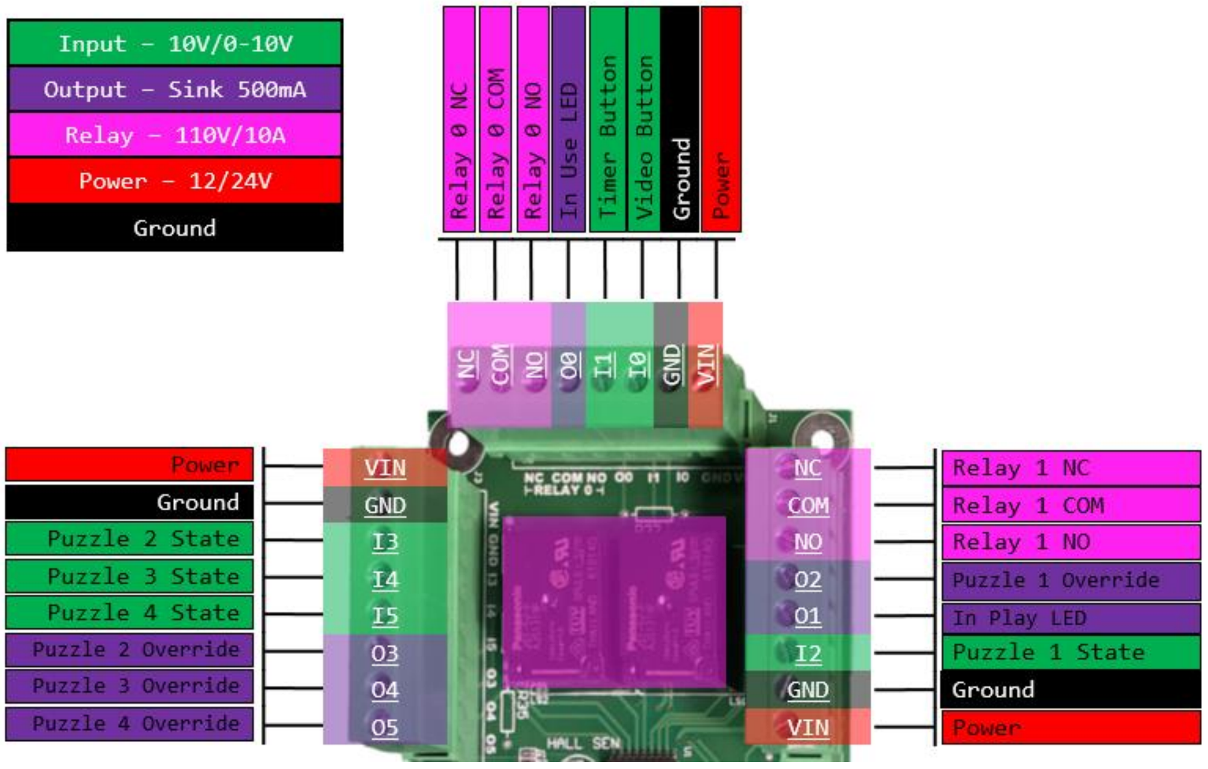
I/O	Function	Notes
INPUT0	Timer Button	Starts Room Countdown Timer
INPUT1	Puzzle State 1	Triggers **OUTPUT1 ON** on HIGH
INPUT2	Puzzle State 2	Triggers **OUTPUT2 ON** on HIGH
INPUT3	Puzzle State 3	Triggers **OUTPUT3 ON** on HIGH
INPUT4	Puzzle State 4	Triggers **OUTPUT4 ON**, RELAY0 OFF on HIGH
INPUT5	Puzzle State 5	Triggers **OUTPUT5 ON**, RELAY1 OFF on HIGH
OUTPUT0	In Use LED	Illuminates when room has started
OUTPUT1	Puzzle 1 Override	Outputs puzzle 1 solved signal
OUTPUT2	Puzzle 2 Override	Outputs puzzle 2 solved signal
OUTPUT3	Puzzle 3 Override	Outputs puzzle 3 solved signal
OUTPUT4	Puzzle 4 Override	Outputs puzzle 4 solved signal
OUTPUT5	Puzzle 5 Override	Outputs puzzle 5 solved signal
RELAY0	Puzzle 4 Relay	Initially HIGH, switches LOW once OUTPUT4 is HIGH
RELAY1	Puzzle 5 Relay	Initially HIGH, switches LOW once OUTPUT5 is HIGH

2 Button 2 LED Version



I/O	Function	Notes
INPUT0	Video Button	Starts Video / Room Intro Sequence
INPUT1	Timer Button	Starts Room Countdown Timer
INPUT2	Puzzle 1 State	Triggers **OUTPUT2 ON** on HIGH
INPUT3	Puzzle 2 State	Triggers **OUTPUT3 ON** on HIGH
INPUT4	Puzzle 3 State	Triggers **OUTPUT4 ON**, **RELAY0 OFF** on HIGH
INPUT5	Puzzle 4 State	Triggers **OUTPUT5 ON**, **RELAY1 OFF** on HIGH
OUTPUT0	In Use LED	Illuminates when room has started
OUTPUT1	In Play LED	Illuminates when countdown started
OUTPUT2	Puzzle 1 Override	Outputs puzzle 1 solved signal
OUTPUT3	Puzzle 2 Override	Outputs puzzle 2 solved signal
OUTPUT4	Puzzle 3 Override	Outputs puzzle 3 solved signal
OUTPUT5	Puzzle 4 Override	Outputs puzzle 4 solved signal
RELAY0	Puzzle 3 Relay	Initially HIGH, switches LOW once OUTPUT4 is HIGH
RELAY1	Puzzle 4 Relay	Initially HIGH, switches LOW once OUTPUT5 is HIGH

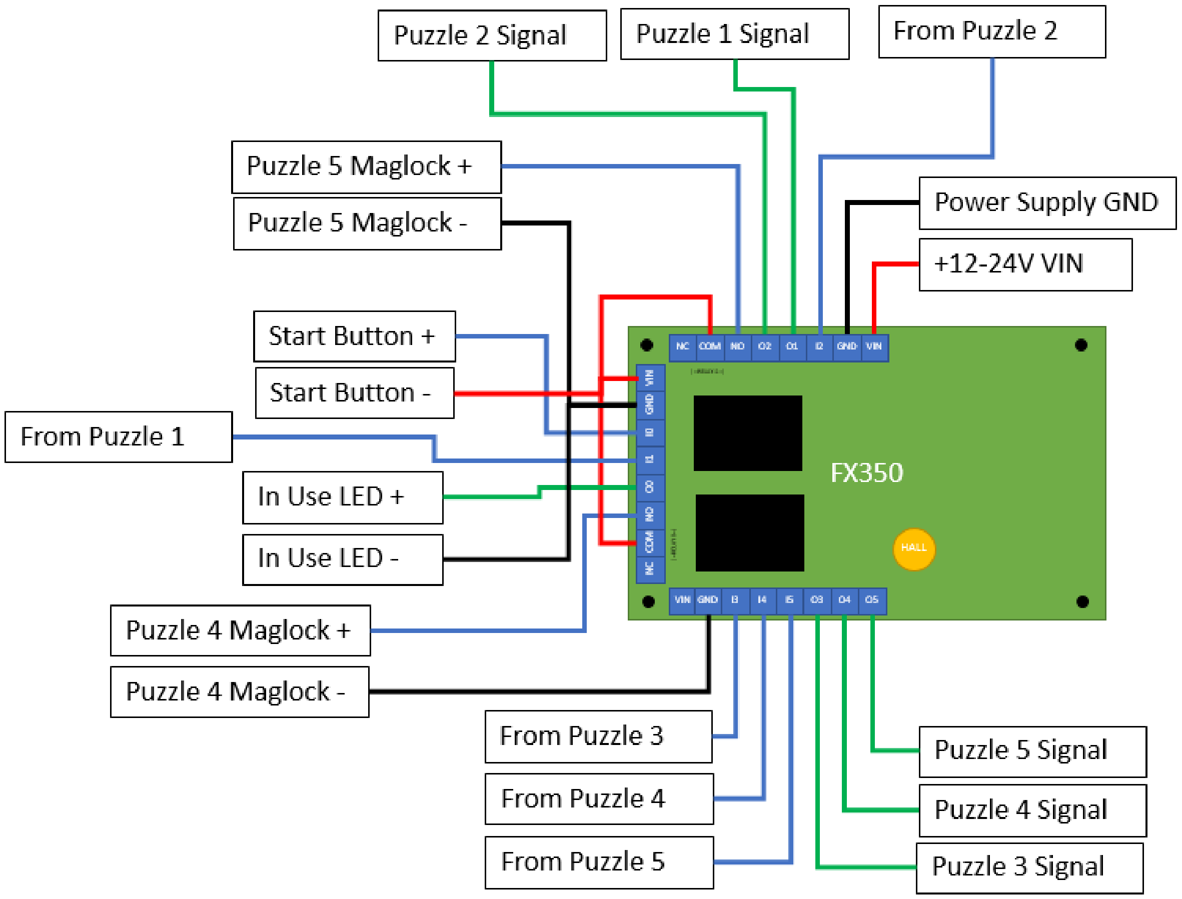
0 Button 0 LED Version



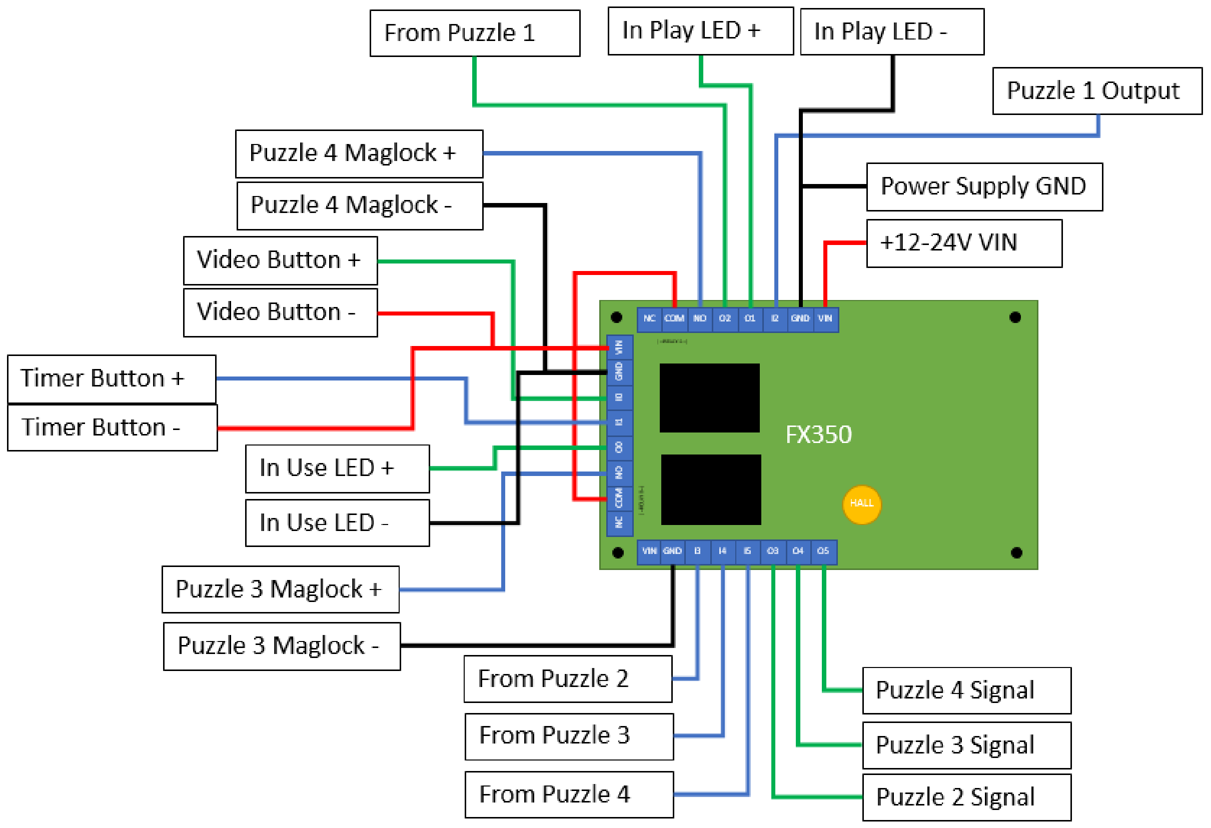
I/O	Function	Notes
INPUT0	Puzzle 1 State	Triggers **OUTPUT0 ON** on HIGH
INPUT1	Puzzle 2 State	Triggers **OUTPUT1 ON** on HIGH
INPUT2	Puzzle 3 State	Triggers **OUTPUT2 ON** on HIGH
INPUT3	Puzzle 4 State	Triggers **OUTPUT3 ON** on HIGH
INPUT4	Puzzle 5 State	Triggers **OUTPUT4 ON**, **RELAY0 OFF** on HIGH
INPUT5	Puzzle 6 State	Triggers **OUTPUT5 ON**, **RELAY1 OFF** on HIGH
OUTPUT0	Puzzle 1 Override	Illuminates when room has started
OUTPUT1	Puzzle 2 Override	Illuminates when countdown started
OUTPUT2	Puzzle 3 Override	Outputs puzzle 1 solved signal
OUTPUT3	Puzzle 4 Override	Outputs puzzle 2 solved signal
OUTPUT4	Puzzle 5 Override	Outputs puzzle 3 solved signal
OUTPUT5	Puzzle 6 Override	Outputs puzzle 4 solved signal
RELAY0	Puzzle 5 Relay	Initially HIGH, switches LOW once OUTPUT4 is HIGH
RELAY1	Puzzle 6 Relay	Initially HIGH, switches LOW once OUTPUT5 is HIGH

Wiring and Installation

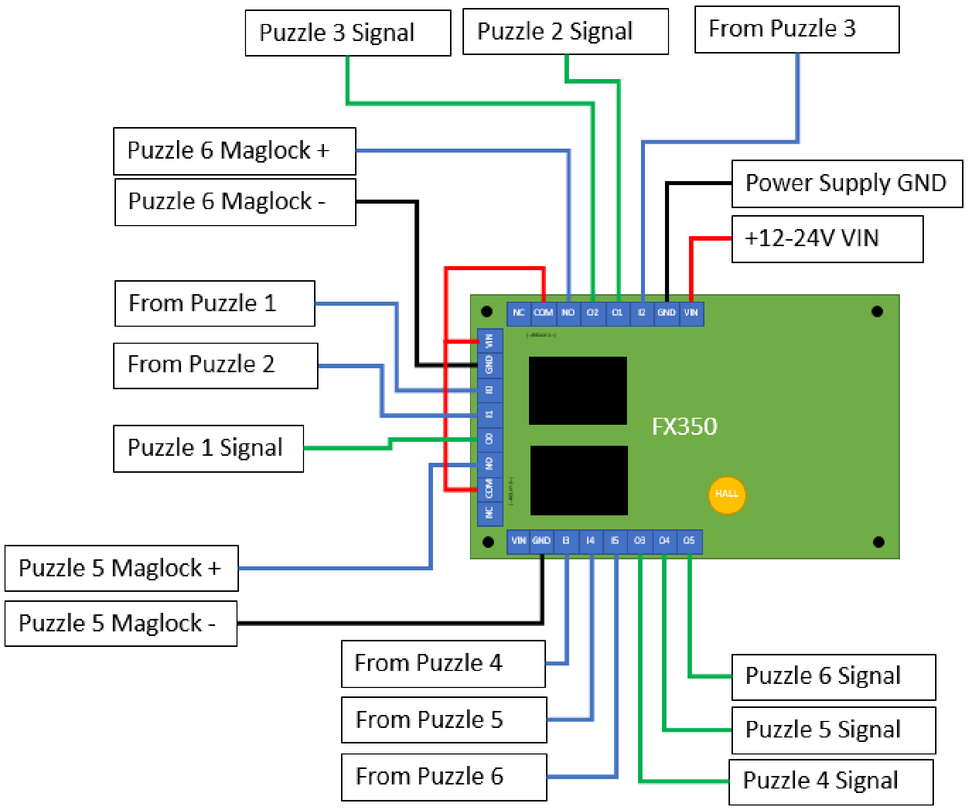
1 Button 1 LED Version



2 Button 2 LED Version

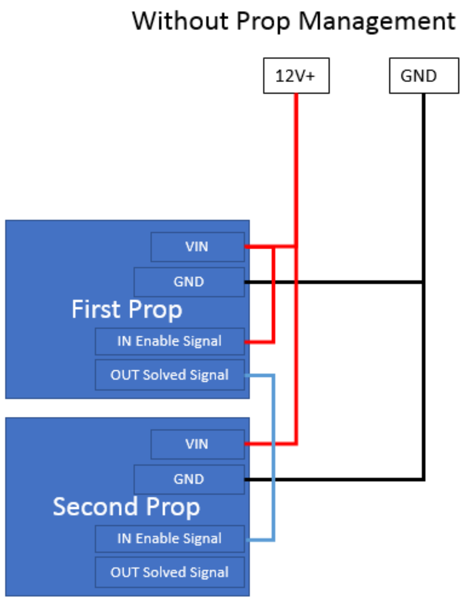


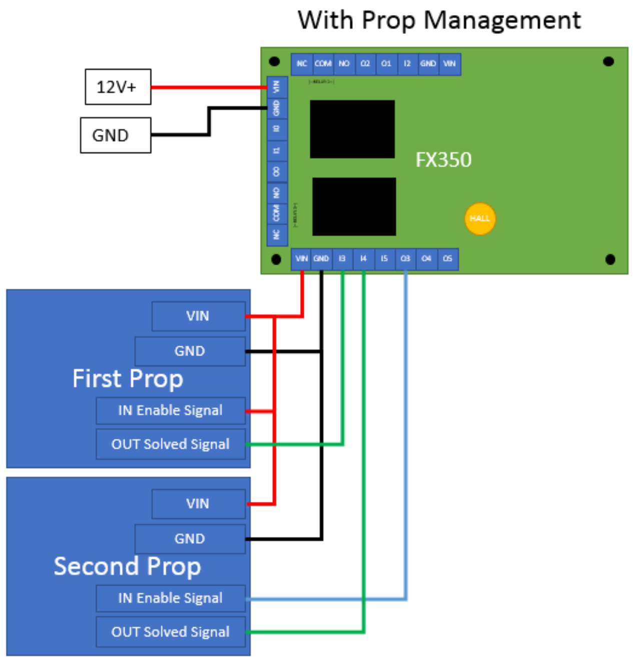
0 Button 0 LED Version



Example Two Prop Control and Monitoring

This diagram exemplifies two sequential props where the completion of one prop triggers the other. The left diagram shows how this could be accomplished without monitoring or overriding capability. The diagram on the right shows how an FX350 room controller may be used to detect solutions and override the second prop’s enable signal.



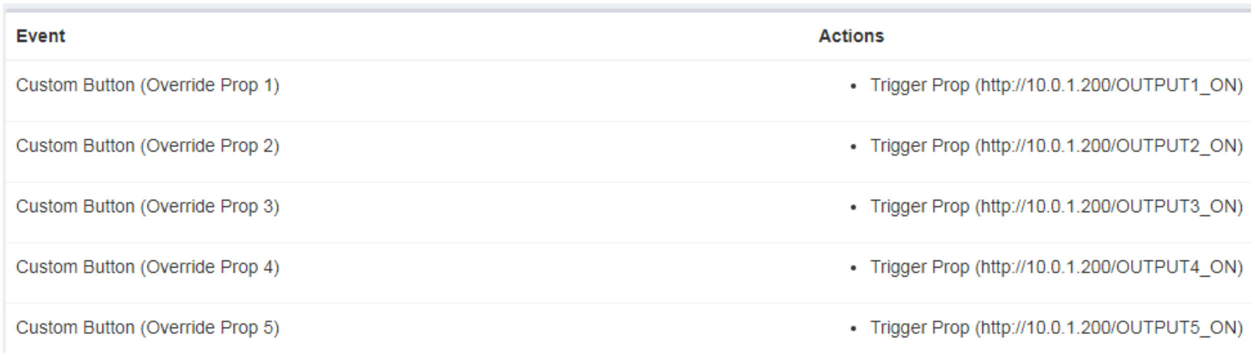


Integration with ERM

The following is a demonstration of the room controller integrated with ERM. Note that the controller must be preprogrammed for ERM networking. If you ordered the room controller kit and specified ERM as your control software, this will be done for you, otherwise reference the section below titled “User Programming”.

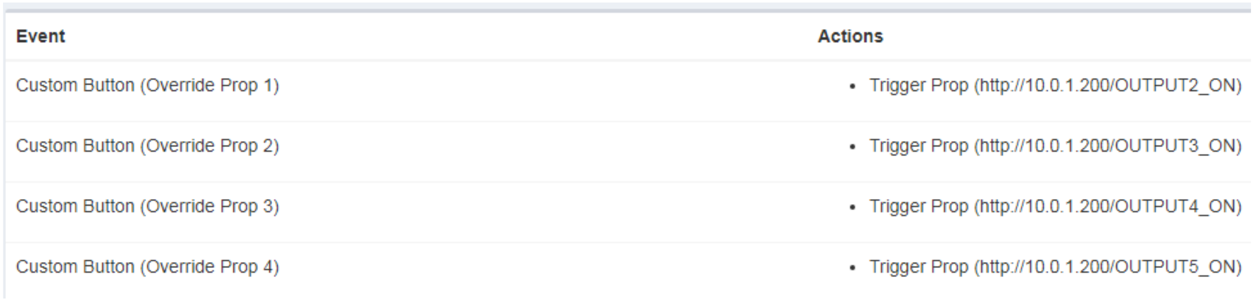
Events

1 Button 1 LED Version Events



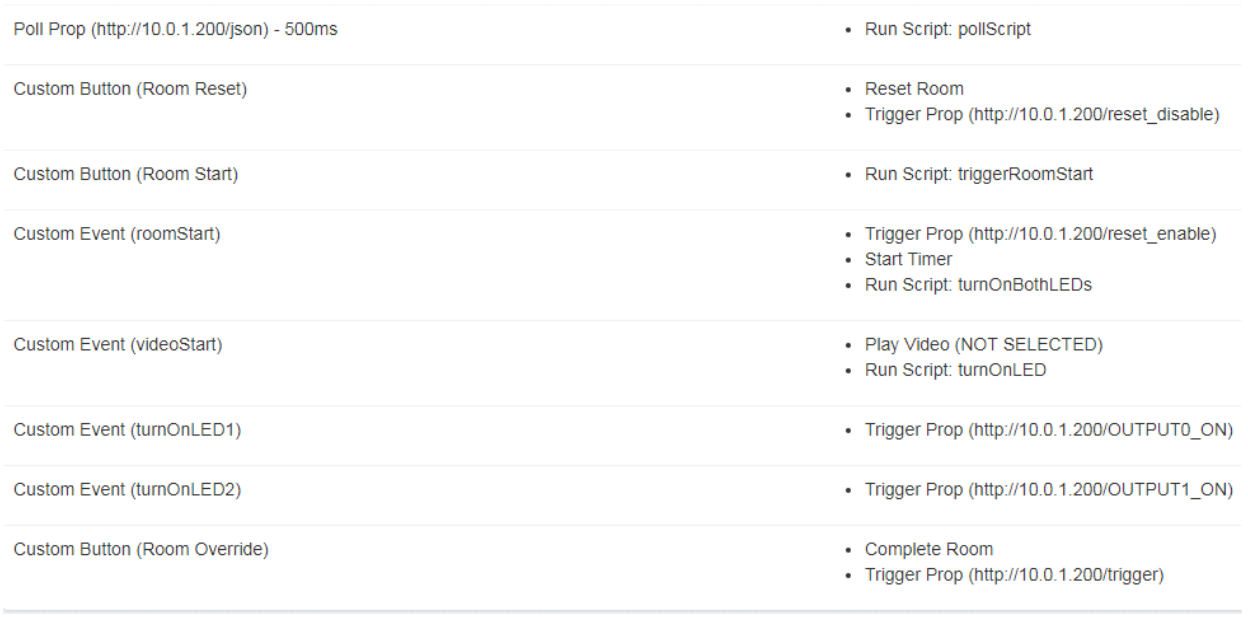
Event	Actions
Custom Button (Override Prop 1)	• Trigger Prop (http://10.0.1.200/OUTPUT1_ON)
Custom Button (Override Prop 2)	• Trigger Prop (http://10.0.1.200/OUTPUT2_ON)
Custom Button (Override Prop 3)	• Trigger Prop (http://10.0.1.200/OUTPUT3_ON)
Custom Button (Override Prop 4)	• Trigger Prop (http://10.0.1.200/OUTPUT4_ON)
Custom Button (Override Prop 5)	• Trigger Prop (http://10.0.1.200/OUTPUT5_ON)

2 Button 2 LED Version Events



Event	Actions
Custom Button (Override Prop 1)	• Trigger Prop (http://10.0.1.200/OUTPUT2_ON)
Custom Button (Override Prop 2)	• Trigger Prop (http://10.0.1.200/OUTPUT3_ON)
Custom Button (Override Prop 3)	• Trigger Prop (http://10.0.1.200/OUTPUT4_ON)
Custom Button (Override Prop 4)	• Trigger Prop (http://10.0.1.200/OUTPUT5_ON)

Common Events



Poll Prop (http://10.0.1.200/json) - 500ms	• Run Script: pollScript
Custom Button (Room Reset)	• Reset Room • Trigger Prop (http://10.0.1.200/reset_disable)
Custom Button (Room Start)	• Run Script: triggerRoomStart
Custom Event (roomStart)	• Trigger Prop (http://10.0.1.200/reset_enable) • Start Timer • Run Script: turnOnBothLEDs
Custom Event (videoStart)	• Play Video (NOT SELECTED) • Run Script: turnOnLED
Custom Event (turnOnLED1)	• Trigger Prop (http://10.0.1.200/OUTPUT0_ON)
Custom Event (turnOnLED2)	• Trigger Prop (http://10.0.1.200/OUTPUT1_ON)
Custom Button (Room Override)	• Complete Room • Trigger Prop (http://10.0.1.200/trigger)

The custom buttons may be given more useful names for the game master, for this example they are given generic names. Also note that all the Trigger Prop actions shown use the local IP 10.0.1.200 as an example. If you purchased the full kit and specified an IP when ordering, that IP will be preprogrammed into your board and is the one you should use. Otherwise see the section below titled "User Programming".

Custom Scripts

pollScript

This script is called repeatedly every 500ms to check the state of the room controller. It gets the state of the input button(s) and triggers the appropriate event when pressed. Uncomment the line which corresponds to your version of the room controller kit.

```
//Parse JSON string to object
var obj = JSON.parse(env.returnVal);

//Uncomment one of these
//state.version = 1; //Single Button Version
//state.version = 2; //Double Button Version
if (obj.INPUT0 == 1)
{
    if (state.version == 1) trigger("roomStart");
   else trigger("videoStart");
}

if (state.version == 2 && obj.INPUT1 == 1)
{
    trigger("roomStart");
}
```

triggerRoomStart

This is a simple script to call a custom event named **roomStart**.

```
trigger("roomStart");
```

turnOnBothLEDs

This script calls custom events named **turnOnLED1** and **turnOnLED2** to turn on LEDs, depending on which version of the controller has been selected in the **pollScript**.

```
trigger("turnOnLED1");
if (state.version == 2) trigger("turnOnLED2");
```

turnOnLED

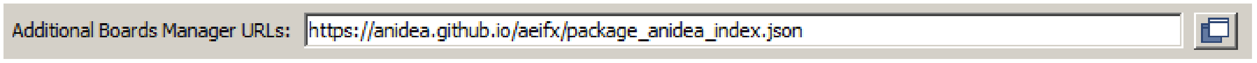
This is a simple script to call a custom event named **turnOnLED1**.

```
trigger("turnOnLED1");
```

User Programming

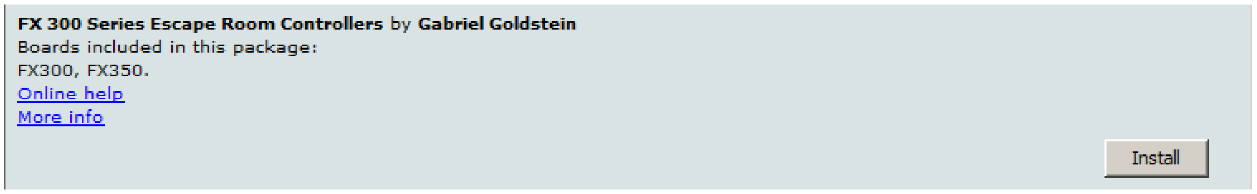
Follow this section if you wish to program your own room controller.

1. Download and install the Arduino IDE https://www.arduino.cc/en/main/software if you don't have it installed already.
2. Navigate to **File -> Preferences**. Add this URL to the board manager URLs textbox: https://anidea.github.io/aeifx/package_anidea_index.json



Additional Boards Manager URLs: https://anidea.github.io/aeifx/package_anidea_index.json

3. Load the board settings by selecting **Tools -> Board -> Boards Manager**. In the **Type** dropdown box choose **Contributed**. Click **Install** for the FX 300 package.



FX 300 Series Escape Room Controllers by **Gabriel Goldstein**
Boards included in this package:
FX300, FX350.
Online help
More info

4. Go to **Tools -> Board** and select FX350 towards the bottom.
5. Connect power to the board.
   a. The board will need external power (10V minimum) to be programmed. To power up, connect a power supply to VIN/Power and GND/Ground.
6. Plug in the micro USB connector to the FX350.
   a. Drivers are required and may take time to load. They are available with the Arduino tools install if they do not install automatically.
   b. NOTE: Plugging in only the USB cable may make the controller look like it is powered, but it is NOT! The board requires an external power supply as described above to function.
7. Go to **Tools -> Port -> COMx** where x is the proper COM port the controller is on. This is typically the last port listed.
8. In the **Anidea-FX-Framework.ino** file, scroll down until you see the game and network selection code.
9. Uncomment the room game to configure the controller for room management.

```
// Uncomment only one of these lines for the game you want
//myGame = new game_empty(); // Empty game to manually control inputs and
outputs only
myGame = new room(); // Used to control a whole room
//myGame = new simplegame(); //Simple game provided as an example
//myGame = new sequencedetect(); //Sequencedetect
//myGame = new sixwire(); //Sixwire
//myGame = new inputsequence(); //Detects a sequence of inputs
//myGame = new rfid();
```

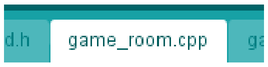
10. Next, set up your network information. Choose a unique MAC address and set an unused local IP (MyIP) address for your controller. If not using ERM, you must also set the IP of the server running the escape room software.

```
// This must be unique for each device
byte MyMac[] = {0x90, 0xA2, 0xDA, 0x0E, 0x94, 0xB6};

// This must be unique for each device on the network. Leave blank to
configure at run time.
IPAddress MyIP(10, 0, 1, 200);

// This should be the IP of the device running the management software. Not
needed for ERM
IPAddress HostIP(10, 0, 1, 105);
```

11. Here you may select your desired network interface. For this example, we will be using Escape Room Master.

```
// Uncomment only one of these lines for the network you want
//myNetwork = new network_empty(); //Empty network for use with FX300
myNetwork = new escaperoommaster(MyMac, MyIP, HostIP);
//myNetwork = new cluecontrol(MyMac, MyIP, HostIP);
//myNetwork = new mqtt(MyMac, MyIP, HostIP);
//myNetwork = new houdinimc(MyMac, MyIP, HostIP);
```

12. Next open the game_room.cpp file. If you don't see it in the header bar you can click the dropdown arrow on the right to access it.



Set the define to the value that corresponds with your room controller.

```
#include "game_room.h"

///////////////////////////////////////////////////////////
// Select which version of the room controller you have //
///////////////////////////////////////////////////////////

#define NUM_BUTTONS 0 // 0, 1, or 2

///////////////////////////////////////////////////////////
//                                                      //
///////////////////////////////////////////////////////////
```

13. Compile and upload to your room controller by selecting the upload button. Done!