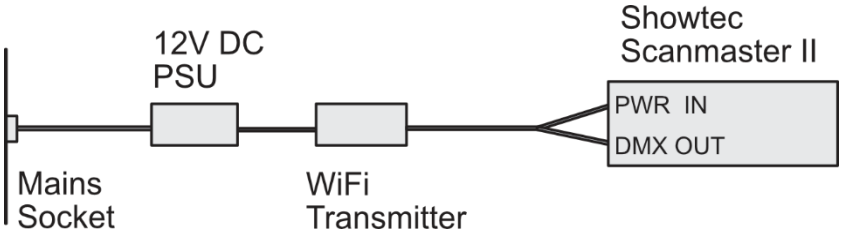
WiFi Showtec Scanmaster and DMX Transmitter with Magic Q

The Magic Q wing at Trinity has 10 faders. If a show needs to be busked, this can be limiting. In theory the solution is to switch to Magic Q's virtual page 2 where 10 more faders are available but switching backwards and forwards can become confusing and it is impossible to use a fader on page 1 and another on page 2 simultaneously.

The Showtec Scanmaster in combination with a DMX WiFi transmitter offers an additional 8 faders when Magic Q is configured to accept remote control.



Setting Magic Q for remote control

1) In Magic Q, go to Setup > General > Network to check the IP address. This should be 192.168.1.129 i.e. the PC's own address on the Trinity Tech network. This is where the WiFi transmitter will send UDP packets. (It is advisable to set the local network so that the MagicQ PC's IP address is locked to this value).

2) In the same configuration block, click *Ethernet Remote Protocol* and select ChamSys Rem (rx no header)

3) In Setup >Multi Console, set
Enable remote control to Yes,
Enable remote access to Yes,
Net Session Mode to None,
Net Session ID to 0,
Playback Sync Mode to PB Sync,
Program Sync Mode to No Sync

4) For each single scene or chase to be controlled from the Showtec, record the cue into one of cue stacks 101-108.
Raising fader 1 – 8 on the Showtec will write the intensity value into the cue.

Setting the Showtec Scanmaster Mk II

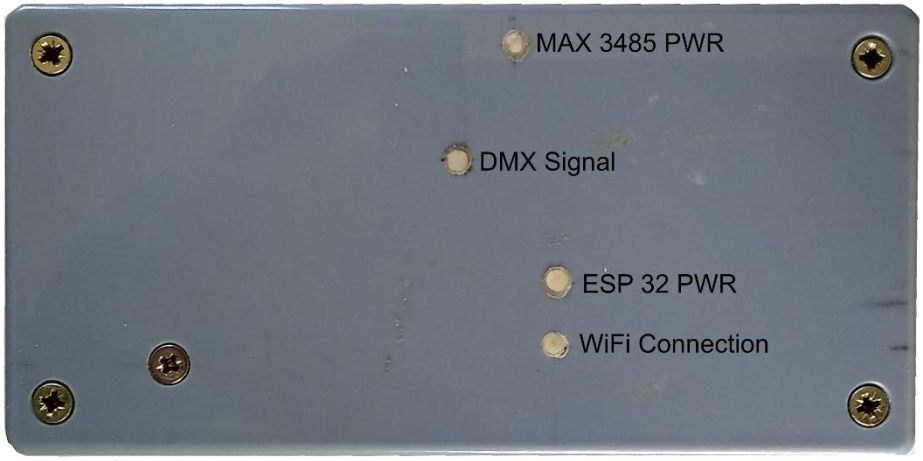
1) On power up, the Showtec goes into blackout mode, indicated by a tiny flashing LED in the display. Turn this off by pressing the blackout button.

2) Press and hold the program button for three seconds to put the unit into programming mode, indicated by another tiny flashing LED in the display. Finally, click the '1' button in the fixtures list. The faders will now send DMX signals on the first eight channels to the transmitter.

WiFi Transmitter

The Wifi transmitter is used to accept DMX data from the Showtec panel and transmit it to Magic Q.

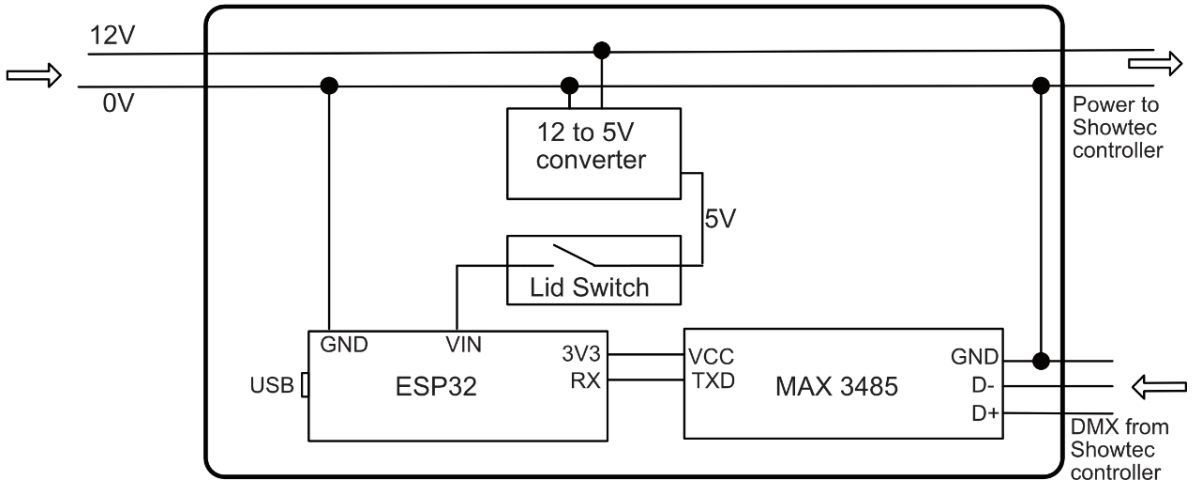
No action is required but there are 4 LEDs that indicate status:



Both power indicators should show on power up. The (faint) DMX signal should show when the Showtec is transmitting DMX to the transmitter. The WiFi connection to Magic Q can take up to 12 seconds to establish (shown by a fast even flash). Thereafter, the indicator should show an uneven slow flash to indicate a connection but no change of data. Very fast flashing indicates data being sent to Magic Q.

Note that the normal DMX network from Magic Q is entirely separate from the short DMX connection between the WiFi module and the Showtec.

Technical Notes



The Scanmaster requires a 12V supply and this is looped through the unit. The ESP32 accepts a 5V supply but its I/O are limited to 3.3V, which it provides to the MAX3485.

In programming mode, the USB connection provides the 5V but this is also seen on the Vin pin. Because we can't be certain that the PC's USB 5V is reverse protected and that it is likely that the two 5V sources are not identical, the lid switch ensures that the two supplies cannot be connected together.

Software notes

The ESP32 code is listed below. There are two sets of WiFi access information - Home and Theatre. If the ESP32 fails to connect to the theatre WiFi, it tries to connect to (Steve Kimpton's) home WiFi instead. This makes testing at both locations easy. Note however, this means that in order to avoid a failed connection at the theatre, Magic q must be running before the Scantec system is started.

The serial monitor is used to report on the WiFi connection at startup and then give live updates of the data sent to Magic Q Cuestacks 101 -108. This information is only available when the unit is connected to the PC with USB.

The ESP32 uses UDP to send wireless data to Magic Q on the PC. Although UDP includes some data checking, there is no confirmation that messages have actually arrived.

There are no delays in the loop function and, if there are no DMX packets ready to process, the loops will be very fast indeed although nothing will happen. The timer (timerDMX) is incremented with the received DMX packet duration whenever a packet is received and is used as a means of flashing the blue LED.

DMX data is sent to Magic Q if it has changed since the last time it was sent.

If editing, remember to set the board to ESP Dev Module - https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

```
1
2   // Steve Kimpton
3   // Showtec to Magic Q software updated March 9th 2023
4   // Change Trinity tech address - was 130, now 129
5   // Change home ssid and password from Plusnet to Sky
6   // Quick flash while coonnecting to wifi changed to 150ms
7   //NB - set board to ESP32 Dev Module other versions may misbehave!
8   #include <esp_dmx.h>         //V1.1.3 was used
9   #include <WiFi.h>            //
10  #include <WiFiUdp.h>
11  #define UDP_PORT 4210
12
13  int Blue = 2;  //Flasher
14  int blueOn = 0;
15
16  //For DMX stuff
17  int transmitPin = 17;
18  int receivePin = 16;
19  int enablePin = 21;
20
21  dmx_port_t dmxPort = 2;
22  byte data[DMX_MAX_PACKET_SIZE];
23  byte olddata[9];
24  int fader[9];
25  QueueHandle_t queue;
26  unsigned int timerDMX = 0;   // used for 2 second serial monitor
27  unsigned int timerMQ = 0;    // used for 75mS data send to Magic Q
28  bool dmxIsConnected = false;
29
30  int tries = 0;
31  unsigned long timenow;
32  unsigned long elapsed;
33  int packetcount = 0;
34  int lastpacketcount = 0;
35  int nrpackets = 0;
36  unsigned long oldtime;
37  int cuestack;
38  String cmdstr = "";
39  int udpmsgs=0;
40  int cuestackmsgs=0;
41
42  WiFiUDP Udp;
43
44  char buff[120];                      // Used to create varying string with mixed char
45  and int
46  int MQport = 6553;                   // Remote port on Magic Q
47  IPAddress MQIPh (192, 168, 0, 10);   // MagicQ home IP addr ('IPAddress' is a data
48  type!)
```

```
49  //IPAddress MQIP (192, 168, 1, 130);      // Magic Q theatre (original)
50  IPAddress MQIP (192, 168, 1, 129);      // Magic Q theatre
51
52  const char* ssidh = "SKYKFY2S";          //Home
53  const char* passwordh = "3Js1V9pYy1UX";  //Home
54
55  const char* ssid = "TRINITY-TECH";    //Theatre
56  const char* password = "B4CACF24";    //Theatre
57
58  ////////////////////////////
59
60  void delayflash() { //InitWiFi originally included 1 second delay, so use this time for
61  flashing.
62    for (int d = 0; d < 10; d++) {
63      digitalWrite(Blue, !(digitalRead(Blue)));
64      delay(150);
65    }
66  }
67

68  ////////////////////////////
69  void initWiFi() {
70    WiFi.mode(WIFI_STA);
71    WiFi.begin(ssid, password);
72    Serial.print("Connecting to Trinity Theatre Tech WiFi ..");
73    while ((WiFi.status() != WL_CONNECTED) && (tries < 6)) {
74      tries++;
75      Serial.print(" Try=");
76      Serial.print(tries);
77      Serial.print("..");
78      delayflash();
79      //delay(1000);
80    }
81    if (tries >= 6) {
82      WiFi.begin(ssidh, passwordh);
83      Serial.println("\nConnecting to WiFi home..");
84      while ((WiFi.status() != WL_CONNECTED)) {
85        tries++;
86        Serial.print(" Try=");
87        Serial.print(tries);
88        Serial.print("..");
89        MQIP = MQIPh;
90        delayflash();
91        //delay(1000);
92      }
93    }
94    Serial.print("\n");  Serial.println(WiFi.localIP());
95  }
96  ////////////////////////////////
97  void setup() {
98    pinMode(Blue, OUTPUT);
99    Serial.begin(115200);
100   initWiFi();
101   Serial.print("RRSI: ");
```

```
102     Serial.println(WiFi.RSSI());
103
104     dmx_config_t dmxConfig = DMX_DEFAULT_CONFIG;
105     dmx_param_config(dmxPort, &dmxConfig);
106     dmx_set_pin(dmxPort, transmitPin, receivePin, enablePin);
107
108     int queueSize = 1;
109     int interruptPriority = 1;
110     dmx_driver_install(dmxPort, DMX_MAX_PACKET_SIZE, queueSize, &queue, interruptPriority);
111   }
112   ////////////////////////////////
113
114   void loop() {
115     dmx_event_t packet;
116     if (xQueueReceive(queue, &packet, DMX_RX_PACKET_TOUT_TICK)) { /* If this code gets
117   called, it means we've recieved DMX data! */
118       packetcount++;
119       if (packet.status == DMX_OK) {
120         /* If this is the first DMX data we've received, lets log it! */
121         if (!dmxIsConnected) {
122           Serial.println("DMX connected!");
123           dmxIsConnected = true;
124         }
125
126         /* We can read the DMX data into our buffer and increment our timers. */
127         dmx_read_packet(dmxPort, data, packet.size);
128         timerDMX += packet.duration;
129
130         bool send = false;
131         int pcent;
132   // ---------------------------------------- We check the new data against the old
133   //                                            and create parts of the udp message
134         for (int i = 1; i < 9; i++) {
135           if (data[i] != olddata[i]) {
136             cuestackmsgs++;
137             pcent = data[i] / 2.55;
138             cuestack = 100 + i;
139             cmdstr = cmdstr + "\\82,";
140             cmdstr = cmdstr + cuestack;
141             cmdstr = cmdstr + ",";
142             cmdstr = cmdstr + pcent;
143             cmdstr = cmdstr + 'H';
144             send = true;
145           }
146
147         }
148
149         if (send) {
150           udpmsgs++;
151           Udp.beginPacket(MQIP, MQport);
152           Udp.print(cmdstr);
153           Udp.endPacket();
154           Serial.print(".");
155           digitalWrite(Blue, !(digitalRead(Blue)));
```

```
156           memcpy(olddata, data, 9);
157           Serial.println(cmdstr);
158           cmdstr="";
159         }
160   //---------------------------  This section is just to flash the blue light if nothing
161   much is happening
162      if ((timerDMX > 1500000) && (blueOn == 0)) {
163           timenow = micros();
164           digitalWrite(Blue, HIGH);
165           Serial.println("Blue on");
166           blueOn = 1;
167           elapsed = timenow - oldtime;
168           oldtime = timenow;
169           Serial.printf("timerDMX=%d packetcount in=%d elapsed=%d packets just
170   sent=%d  previously sent=%d ", timerDMX, packetcount, elapsed, udpmsgs, cuestackmsgs);
171           Serial.printf("%u %u %u %u %u %u %u %u \n", data[1], data[2], data[3], data[4],
172   data[5], data[6], data[7], data[8]);
173           packetcount = 0;
174           udpmsgs=0;
175           cuestackmsgs=0;
176         }
177         if ((timerDMX > 1600000) && (blueOn==1)) {
178           digitalWrite(Blue, LOW);
179           Serial.print(timerDMX);
180           Serial.println(" Blue off (after 100000 usecs");
181           blueOn = 0;
182           timerDMX = 0;
183         }
184
185   //------------------------------  We hope this bit never runs
186       }
187       else {
188         /* Uh-oh! Something went wrong receiving DMX. */
189         Serial.println("DMX error!");
190       }
191
192     } else if (dmxIsConnected) {
193       Serial.println("DMX timed out! ...");
194     }
195   }
196
```