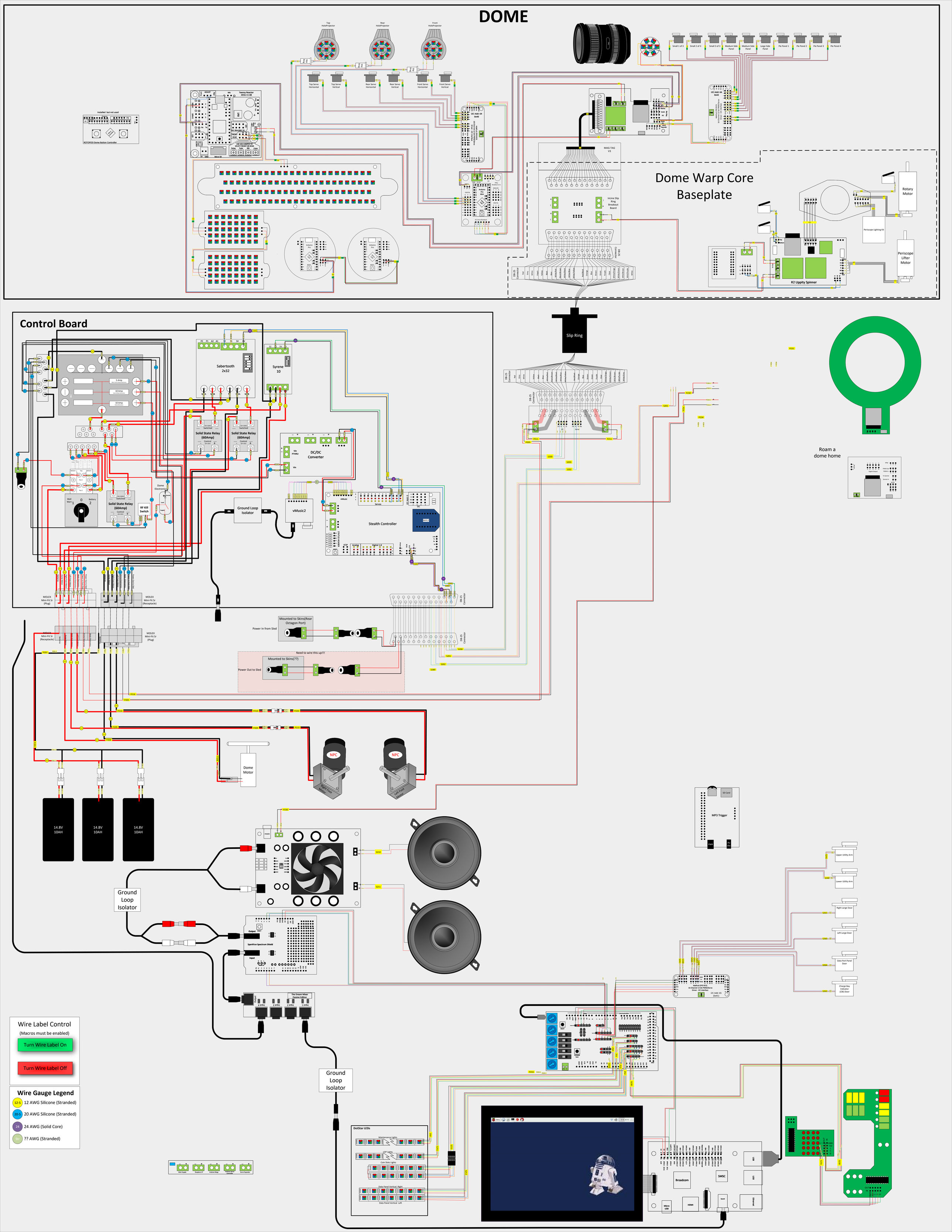
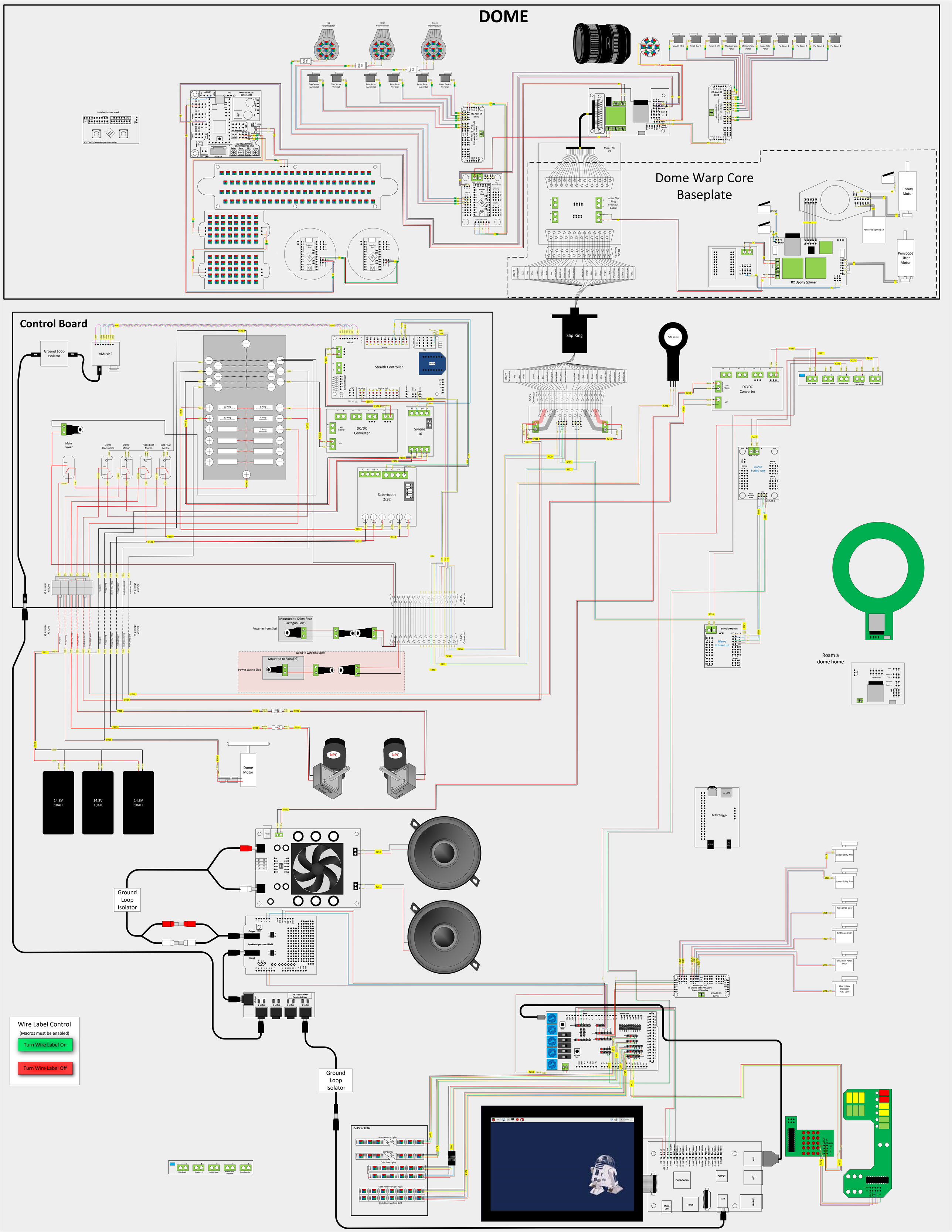
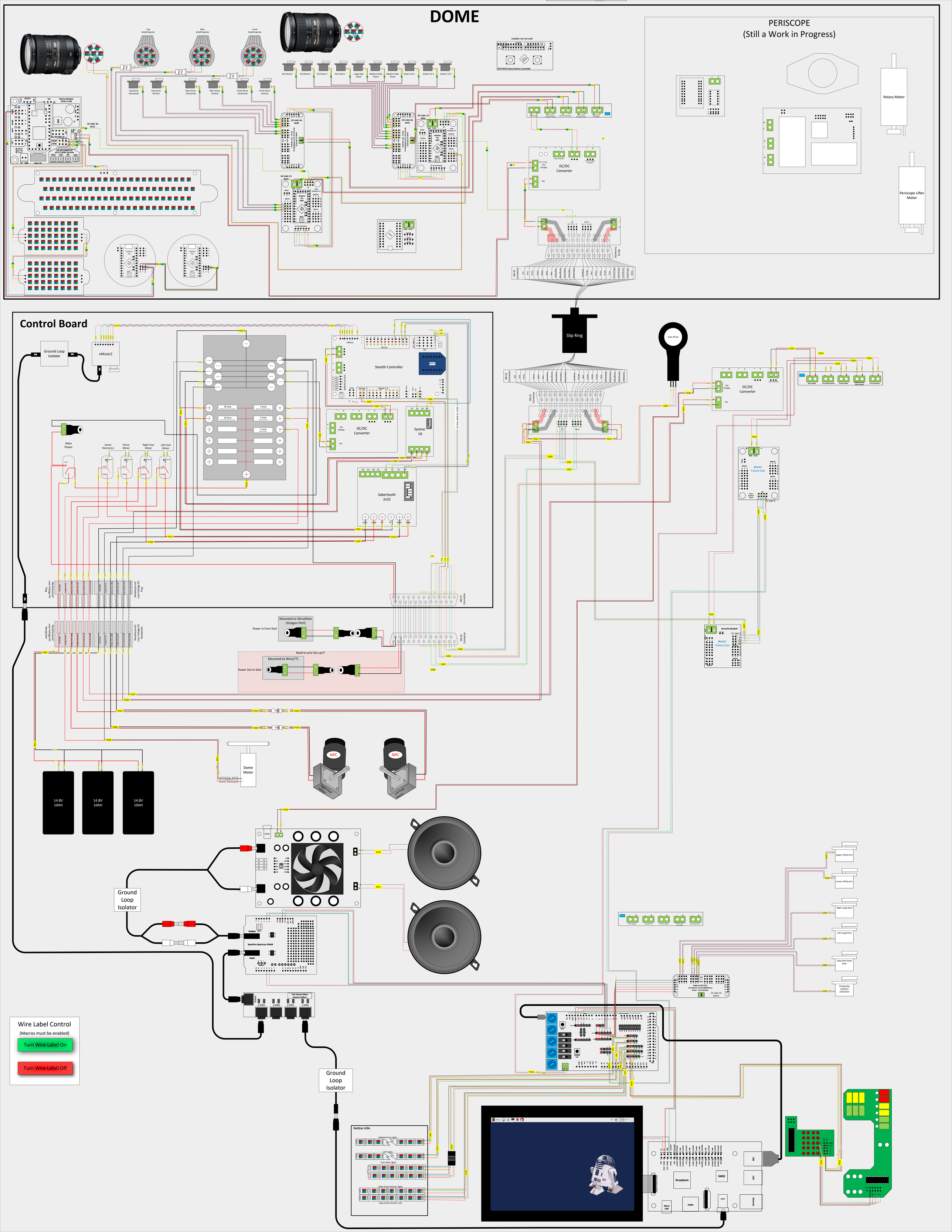
DOME

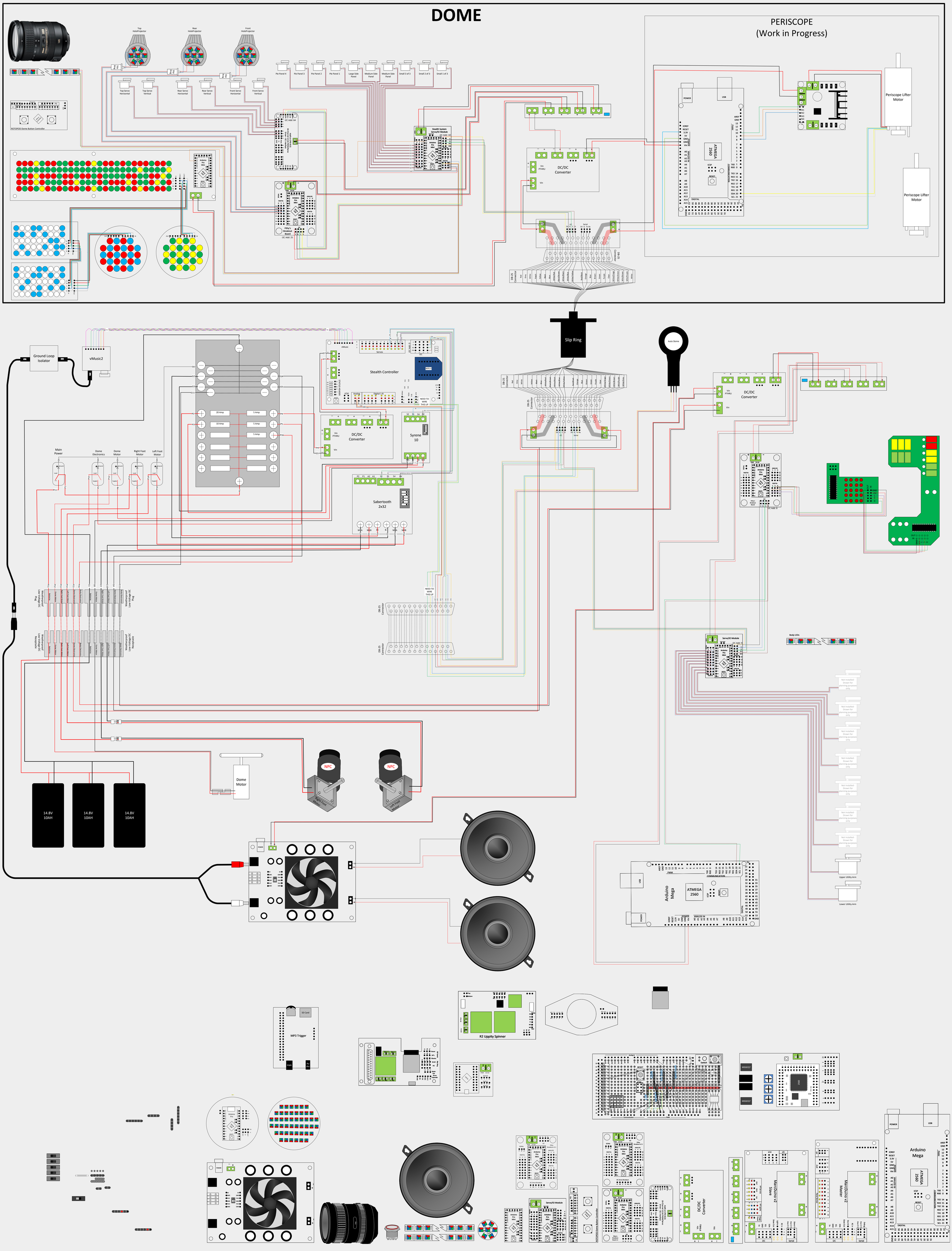

Top HoloProjector
Rear HoloProjector
Front HoloProjector
Top Servo Horizontal
Top Servo Vertical
Rear Servo Horizontal
Rear Servo Vertical
Front Servo Horizontal
Front Servo Vertical
Small 1 of 3
Small 2 of 3
Small 3 of 3
Medium Side Panel
Medium Side Panel
Large Side Panel
Pie Panel 1
Pie Panel 2
Pie Panel 3
Pie Panel 4
Installed but not used
ROTOPOD Dome Button Controller
Teensy Reactor 2016-11-08
I2C Add: 64 0x40
I2C Add: 66 0x42
MAG-TAG V3
Dome Warp Core Baseplate
InLine Slip Ring Breakout Board
DB-25 Connector
Rotary Motor
Periscope Lighting Kit
Periscope Lifter Motor
R2 Uppity Spinner
Control Board
Sabertooth 2x32
Syrene 10
Solid State Relay (60Amp)
DC/DC Converter
Dome Electronics
RF Kill Switch
Ground Loop Isolator
vMusic2
Stealth Controller
MOLEX Mini-Fit Sr (Plug)
MOLEX Mini-Fit Sr (Receptacle)
Slip Ring
Roam a dome home
Mounted to Skins(Rear Octagon Port)
Power In from Sled
Need to wire this up!!!
Mounted to Skins(??)
Power Out to Sled
Dome Motor
NPC
Right Foot Motor
Left Foot Motor
14.8V 10AH
14.8V 10AH
14.8V 10AH
MP3 Trigger
SD Card
Upper Utility Arm
Lower Utility Arm
Right Large Door
Left Large Door
Data Port Panel Door
Charge Bay Indicator (CBI) Door
Ground Loop Isolator
DotStar LEDs
Maintenance Lights
LDP Lights
Coin Slots Lights
Data Panel Vertical Right
Data Panel Vertical Left
Broadcom
SMSC
HDMI
Ethernet
USB
Wire Label Control
(Macros must be enabled)
Turn Wire Label On
Turn Wire Label Off
Wire Gauge Legend
12-S 12 AWG Silicone (Stranded)
20-S 20 AWG Silicone (Stranded)
24 24 AWG (Solid Core)
Str ?? AWG (Stranded)

DOME


Top HoloProjector
Rear HoloProjector
Front HoloProjector
Top Servo Horizontal
Top Servo Vertical
Rear Servo Horizontal
Rear Servo Vertical
Front Servo Horizontal
Front Servo Vertical
Small 1 of 3
Small 2 of 3
Small 3 of 3
Medium Side Panel
Medium Side Panel
Large Side Panel
Pie Panel 1
Pie Panel 2
Pie Panel 3
Pie Panel 4
Installed but not used
ROTOPOD Dome Button Controller
Teensy Reactor 2016-11-08
MAG-TAG V3
InLine Slip Ring Breakout Board
Dome Warp Core Baseplate
Rotary Motor
Periscope Lighting Kit
Periscope Lifter Motor
R2 Uppity Spinner
DB-25 Connector
Slip Ring
Control Board
Ground Loop Isolator
vMusic2
Main Power
Dome Electronics
Dome Motor
Right Foot Motor
Left Foot Motor
Stealth Controller
DC/DC Converter
Syrene 10
Sabertooth 2x32
Mounted to Skins(Rear Octagon Port)
Power In from Sled
Need to wire this up!!!
Mounted to Skins(??)
Power Out to Sled
14.8V 10AH
14.8V 10AH
14.8V 10AH
Dome Motor
NPC
NPC
Auto Dome
Blank/ Future Use
Roam a dome home
MP3 Trigger
Upper Utility Arm
Lower Utility Arm
Right Large Door
Left Large Door
Data Port Panel Door
Charge Bay Indicator (CBI) Door
Wire Label Control
(Macros must be enabled)
Turn Wire Label On
Turn Wire Label Off
DotStar LEDs
Maintenance Lights
LDP Lights
Coin Slots Lights
Data Panel Vertical Right
Data Panel Vertical Left
Broadcom
SMSC
HDMI
Micro USB
Ethernet
USB
SD Card


DOME
PERISCOPE
(Still a Work in Progress)
Rotary Motor
Periscope Lifter Motor
Top HoloProjector
Rear HoloProjector
Front HoloProjector
Pie Panel 4
Pie Panel 3
Pie Panel 2
Pie Panel 1
Large Side Panel
Medium Side Panel
Small 3 of 3
Small 2 of 3
Small 1 of 3
Top Servo Horizontal
Top Servo Vertical
Rear Servo Horizontal
Rear Servo Vertical
Front Servo Horizontal
Front Servo Vertical
Installed but not used
ROTOPOD Dome Button Controller
DC/DC Converter
Slip Ring
Auto Dome
Control Board
Ground Loop Isolator
vMusic2
Main Power
Dome Electronics
Dome Motor
Right Foot Motor
Left Foot Motor
Stealth Controller
Syrene 10
Sabertooth 2x32
Mounted to Skins(Rear Octagon Port)
Power In from Sled
Mounted to Skins(??)
Power Out to Sled
Need to wire this up!!!
14.8V 10AH
14.8V 10AH
14.8V 10AH
Dome Motor
NPC
Right Foot Motor
Left Foot Motor
Blank/ Future Use
Servo/IO Module
Upper Utility Arm
Lower Utility Arm
Right Large Door
Left Large Door
Data Port Panel Door
Charge Bay Indicator (CBI) Door
Wire Label Control
(Macros must be enabled)
Turn Wire Label On
Turn Wire Label Off
DotStar LEDs
Maintenance Lights
LDP Lights
Coin Slots Lights
Data Panel Vertical Right
Data Panel Vertical Left
Broadcom
SMSC
HDMI
Micro USB
USB
USB
Ethernet

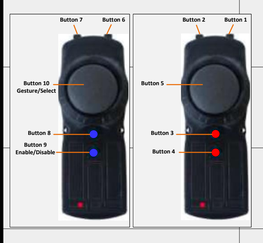
DOME


PERISCOPE
(Work in Progress)
Top HoloProjector
Rear HoloProjector
Front HoloProjector
Pie Panel 4
Pie Panel 3
Pie Panel 2
Pie Panel 1
Large Side Panel
Medium Side Panel
Small 3 of 3
Small 2 of 3
Small 1 of 3
Top Servo Horizontal
Top Servo Vertical
Rear Servo Horizontal
Rear Servo Vertical
Front Servo Horizontal
Front Servo Vertical
ROTOPOD Dome Button Controller
Stealth System Servo/IO Module
DC/DC Converter
Periscope Lifter Motor
Periscope Lifter Motor
Slip Ring
Ground
DB-25 Connector
Ground
Ground
Ground

Actions	Details
Open/Close Dome 1 – Opens the Dome 2 – Changes the Camaera Lens LED Colors to a random color each time it closes a few panels 3 – Sets the HP's to random pulse Right, Right	Double Click Double Click to get To the Details Page
Quick Wave 1 – Opens the Dome panels 2 – Changes the Camera Lens LED Colors to a random color each time it closes a few panels 3 – Changes the HPs to a random color each time it runs through a cycle of opening and closing Left, Left	Double Click Double Click to get To the Details Page
Random Jittery Wave 1 – Opens the Dome panels 2 – Changes the Camera Lens LED Colors to a random color each time it closes a few panels 3 – Changes the HPs to a random color each time it runs through a cycle of opening and closing Left, Right,	Double Click Double Click to get To the Details Page
Right, Left,	
Wave 1 1 - Waves the Dome Panels 2 - Flashes the Camera LEDs to Random Colors, 3 - Turns the HP LEDs to Rainbow Right, Right, Right	Double Click Double Click to get To the Details Page
Wave 2 1 - Waves the Dome Panels 2 - Flashes the Camera LEDs to Random Colors, 3 - Turns the HP LEDs to Rainbow Left, Left, Left	Double Click Double Click to get To the Details Page
Left, Right, Left	
Right, Left, Right	

Actions	Details
HP Rainbow 1 - Waves the Dome Panels 2 - Flashes the Camera LEDs to Random Colors, 3 - Turns the HP LEDs to Rainbow Up, Up	Double Click Double Click to get To the Details Page
HP Cyclone Blue 1 - Waves the Dome Panels 2 - Flashes the Camera LEDs to Random Colors, 3 - Turns the HP LEDs to Rainbow Down, Down	Double Click Double Click to get To the Details Page
HP Random Color Pulse 1 – Pulses the HP's a random Color at medium speed Down, Up	Double Click Double Click to get To the Details Page
HP Leia Message 1 – Plays the HP's at the Leia function Up. Down	Double Click Double Click to get To the Details Page

Actions	Details
Utility Arms Open/Close 1 – Opens and closes the Utility Arms Button 4	Double Click Double Click to get To the Details Page

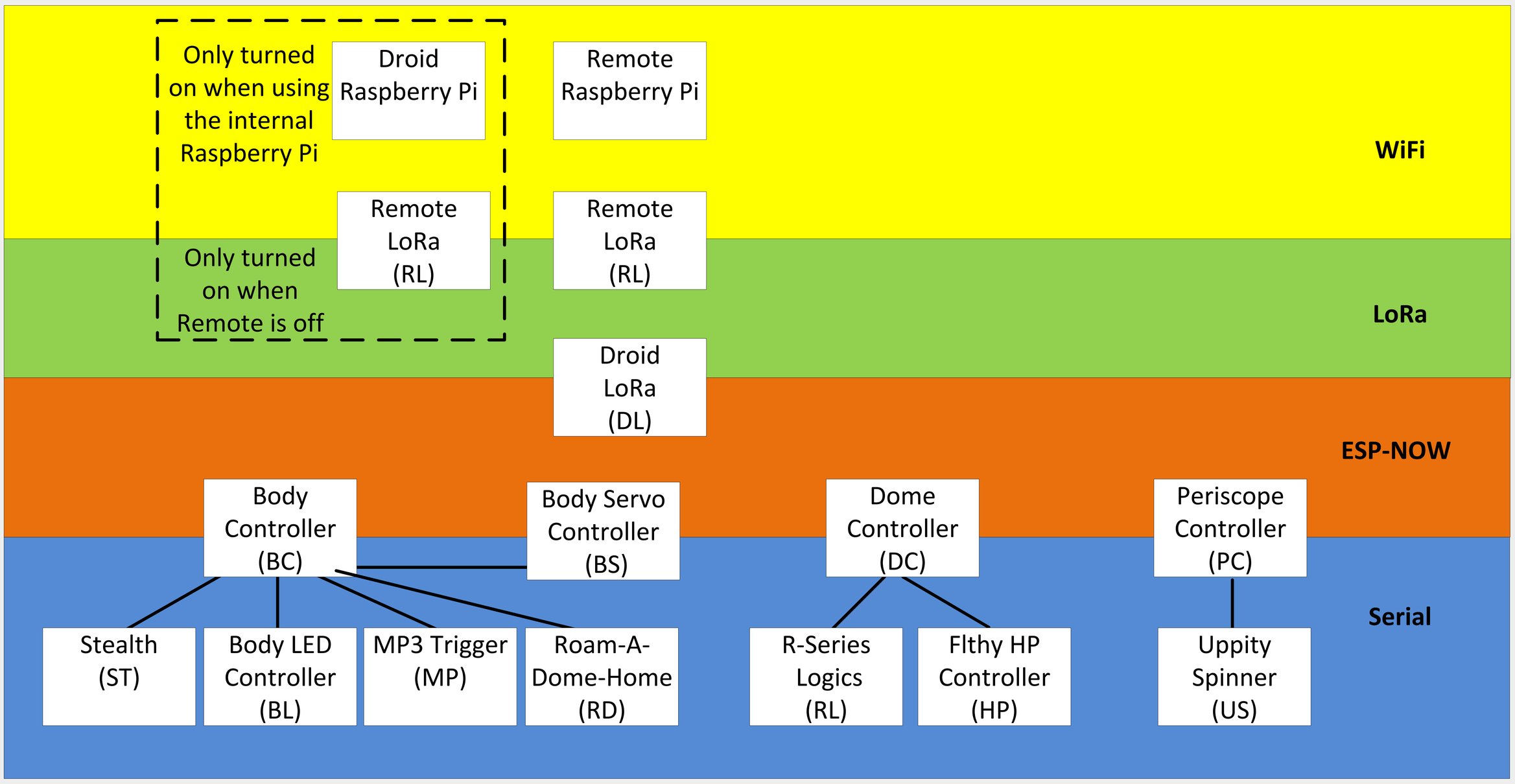


Actions	Details

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Teeces (Serial)
			Space Holer							

DOME
PERISCOPE
(Work in Progress)
Top Servo Horizontal
Top Servo Vertical
Rear Servo Horizontal
Rear Servo Vertical
Front Servo Horizontal
Front Servo Vertical
Pie Panel 4
Pie Panel 3
Pie Panel 2
Pie Panel 1
Large Side Panel
Medium Side Panel
Medium Side Panel
Small 3 of 3
Small 2 of 3
Small 1 of 3
MarcDuino v2 Slave
MarcDuino v2 Master
DC/DC Converter
Periscope Lifter Motor
Periscope Lifter Motor
ATMEGA 2560
DB-25 Connector
Slip Ring
Auto Dome
Ground Loop Isolator
vMusic2
Stealth Controller
Main Power
Dome Electronics
Dome Motor
Right Foot Motor
Left Foot Motor
Syrene 10
Sabertooth 2x32
Weatherproof Low Voltage DC Plug
Weatherproof Low Voltage DC Receptacle
14.8V 10AH
14.8V 10AH
14.8V 10AH
Dome Motor
LED Brightness
10K Potentiometer
For Development Use Only (will not be part of final build)
Sequence Change
Button 1
Button 2
Broadcom
SMSC
HDMI
Micro USB
Ethernet
Arduino Mega
ATMEGA 2560
MarcDuino v2 Slave
MarcDuino v2 Master
DC/DC Converter
Servo/IO Module
Teensy Reactor 2016-11-08


Only turned on when using the internal Raspberry Pi
Droid Raspberry Pi
Remote Raspberry Pi
WiFi
Remote LoRa (RL)
Remote LoRa (RL)
Only turned on when Remote is off
LoRa
Droid LoRa (DL)
ESP-NOW
Body Controller (BC)
Body Servo Controller (BS)
Dome Controller (DC)
Periscope Controller (PC)
Serial
Stealth (ST)
Body LED Controller (BL)
MP3 Trigger (MP)
Roam-A-Dome-Home (RD)
R-Series Logics (RL)
Flthy HP Controller (HP)
Uppity Spinner (US)

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Dome Button Cntr I2c = 11
Wave 1 1 - Waves the Dome Panels 2 - Flashes the Camera LEDs to Random Colors, 3 - Turns the HP LEDs to Rainbow	 Right, Right, Right	g=65656,4,10,6				case 6: Wave1(); break;				

```
void Wave1() {
  sendFlthyI2Cmd("A007");
  Serial.println("Wave Panels 1");
// Wave 1
                    // Attach to all the panel so we can move them
                    Servos[LTL1].attach(LITTLE_PANEL_1_OF_3);
                    Servos[LTL2].attach(LITTLE_PANEL_2_OF_3);
                    Servos[LTL3].attach(LITTLE_PANEL_3_OF_3);
                    Servos[MED1].attach(MED_PANEL_1_OF_2);
                    Servos[MED2].attach(MED_PANEL_2_OF_2);
                    Servos[LRG1].attach(LARGE_PANEL);
                    Servos[PIE1].attach(PIE1_SERVO_PIN);
                    Servos[PIE2].attach(PIE2_SERVO_PIN);
                    Servos[PIE3].attach(PIE3_SERVO_PIN);
                    Servos[PIE4].attach(PIE4_SERVO_PIN);

                    //Move the Servos to Nuetral position
                      Servos[LTL1].write(NEUTRAL,150);
                    Servos[LTL2].write(NEUTRAL,150);
                    Servos[LTL3].write(NEUTRAL,150);
                    Servos[MED1].write(NEUTRAL,150);
                    Servos[MED2].write(NEUTRAL,150);
                    Servos[LRG1].write(NEUTRAL,150);
                      Servos[PIE1].write(NEUTRAL,150);
                      Servos[PIE2].write(NEUTRAL,150);
                      Servos[PIE3].write(NEUTRAL,150);
                      Servos[PIE4].write(NEUTRAL,150);


                                                          //RUN THIS 4 TIMES
  for (int i=0; i<4; i++) {
    Serial.println("Wave");
                    colorWipe(basicColors[random(0,10)], 255);      // Turn Camera Lens LED Random Colors
                    Servos[LTL1].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[LTL2].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[LTL3].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[MED1].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[MED2].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[LRG1].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[PIE1].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[PIE2].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[PIE3].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[PIE4].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                    delay(WAVEDELAY * 2.5);
                    Servos[LTL1].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[LTL2].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[LTL3].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[MED1].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[MED2].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[LRG1].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[PIE4].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[PIE3].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[PIE2].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY);
                    Servos[PIE1].write(NEUTRAL,WAVESPEED);
                    delay(WAVEDELAY * 2);
  }

  // Disconnect from our servos
                    Servos[LTL1].detach();
                    Servos[LTL2].detach();
                    Servos[LTL3].detach();
                    Servos[MED1].detach();
                    Servos[MED2].detach();
                    Servos[LRG1].detach();
                    Servos[PIE1].detach();
                    Servos[PIE2].detach();
                    Servos[PIE3].detach();
                    Servos[PIE4].detach();

    colorWipe(C_BLUE, 255);                   // Turn Camera Lens LED Blue
    sendFlthyI2Cmd("A00354");
  Serial.println("Wave Done");


  i2cCommand=-1; // always reset i2cCommand to -1, so we don't repeatedly do the same command
  //digitalWrite(STATUS_LED, LOW);
}
```

Turn all HPs to Rainbow

Turn all HPs to pulse blue and speed of 4

Back to Gestures-Actions Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Dome Button Cntr I2c = 11

Wave 2
1 - Waves the Dome Panels
2 - Flashes the Camera LEDs to Random Colors,
3 - Turns the HP LEDs to Rainbow
	
	Left, Left, Left
		g=45454,4,10,7				case 7:
						Wave2();
						break;
						
						void Wave2() {
						
						digitalWrite(STATUS_LED, HIGH); // turn on STATUS LED so we can visually see we got the command on the board
						//Wave 2 / Scream
						
						Serial.println("Scream Sequence");
						
						sendMaster(2Ccmd("tmpvol=100,0#"); // Temp max volume for 8 seconds
						delay(150);
						sendMaster(2Ccmd("tmprnd=60"); // delay random sounds by 60 seconds so we dont stomp on sequence
						delay(1000);
						sendMaster(2Ccmd("S0403"); // Play Scream, SOUNDBANK 4, SOUND 3
						
						// Routine below should last as long as the sound we just started playing
						
						//Attach to all the pie panels
						Servos[LTL1].attach(LITTLE_PANEL_1_OF_3);
						Servos[LTL2].attach(LITTLE_PANEL_2_OF_3);
						Servos[LTL3].attach(LITTLE_PANEL_3_OF_3);
						Servos[MED1].attach(MED_PANEL_1_OF_2);
						Servos[MED2].attach(MED_PANEL_2_OF_2);
						Servos[LRG1].attach(LARGE_PANEL);
						Servos[PIE1].attach(PIE1_SERVO_PIN);
						Servos[PIE2].attach(PIE2_SERVO_PIN);
						Servos[PIE3].attach(PIE3_SERVO_PIN);
						Servos[PIE4].attach(PIE4_SERVO_PIN);
						delay(250);
						
						// Loop 7 times in panic mode
						for (int x=0; x<7;x++) {
						colorWipe(C_BLUE, 255); // Turn Camera Lens LED Blue
						sendFlthy(2Ccmd("A0061 "); // Turn All HP's Red
						// Open some the Servos, they could in theory do this at different speed to be more organic
						Servos[LTL1].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[LTL2].write(NEUTRAL+2,SPEED);
						Servos[LTL3].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[MED1].write(NEUTRAL+2,SPEED);
						Servos[MED2].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[LRG1].write(NEUTRAL+2,SPEED);
						Servos[PIE1].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[PIE2].write(NEUTRAL+2,SPEED);
						Servos[PIE3].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[PIE4].write(NEUTRAL+2,SPEED),true); // wait on servos
						
						// Now close/open the opposite Pie panels so they're waving at different intervals
						colorWipe(C_RED, 255); // Turn Camera Lens LED Red
						sendFlthy(2Ccmd("A0063"); //Turn all HP's Blue
						Servos[LTL1].write(NEUTRAL+2,SPEED);
						Servos[LTL2].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[LTL3].write(NEUTRAL+2,SPEED);
						Servos[MED1].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[MED2].write(NEUTRAL+2,SPEED);
						Servos[LRG1].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[PIE1].write(NEUTRAL+2,SPEED);
						Servos[PIE2].write(NEUTRAL-PIE_ADD_DEGREES,SPEED);
						Servos[PIE3].write(NEUTRAL+2,SPEED);
						Servos[PIE4].write(NEUTRAL-PIE_ADD_DEGREES,SPEED,true); // wait on servos
						
						}
						
						// end of panic sequence reset thing back to normal
						
						//reset HP LED to original value
						
						sendFlthy(2Ccmd("A0033 ");
						
						// Close all the pies and flaps
						Servos[LTL1].write(NEUTRAL,150);
						Servos[LTL2].write(NEUTRAL,150);
						Servos[LTL3].write(NEUTRAL,150,true);
						Servos[MED1].write(NEUTRAL,150);
						Servos[MED2].write(NEUTRAL,150);
						Servos[LRG1].write(NEUTRAL,150,true);
						Servos[PIE1].write(NEUTRAL,150);
						Servos[PIE2].write(NEUTRAL,150);
						Servos[PIE3].write(NEUTRAL,150);
						Servos[PIE4].write(NEUTRAL,150,true);
						//colorWipe(strip-Color(0, 0, 255)); // Blue
						
						// wait for everything to settle
						delay(500);
						
						// Detach all servos so we can go back to opening them manually
						Servos[LTL1].detach();
						Servos[LTL2].detach();
						Servos[LTL3].detach();
						Servos[MED1].detach();
						Servos[MED2].detach();
						Servos[LRG1].detach();
						Servos[PIE1].detach();
						Servos[PIE2].detach();
						Servos[PIE3].detach();
						Servos[PIE4].detach();
						colorWipe(C_BLUE, 255); // Turn Camera Lens LED Blue
						I2cCommand=-1; // always reset I2cCommand to -1, so we don't repeatedly do the same command
						//digitalWrite(STATUS_LED, LOW);
						
						}
							Turn all HPs to Red
							
							Turn all HPs to Blue
							
							Turn all HPs to pulse Blue slowly
									Turn on Scream function	

Try sending Scream

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Servo Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Dome Button Cntr I2c = 11
Open/Close Dome 1 – Opens the Dome 2 – Changes the Camaera Lens LED Colors to a random color each time it closes a few panels 3 – Sets the HP's to random pulse	Right, Right	g=656,4,10,5								

```
case 5:
    OpenClosePanels();
    break;
```

```
void OpenClosePanels() {

  //digitalWrite(STATUS_LED, HIGH); // turn on STATUS LED so we can visually see we got the command on the board
  //Open or close  All Pie Panels

  Serial.print("Pie Panels: ");
  colorWipe(C_RED, 255);                          // Red
  if (panelsOpen) { // Close the Pie Panels
   Serial.println("Closing");
   panelsOpen=false;
   // Attach to PANELS so we can move them
   Servos[LTL1].attach(LITTLE_PANEL_1_OF_3);
   Servos[LTL2].attach(LITTLE_PANEL_2_OF_3);
   Servos[LTL3].attach(LITTLE_PANEL_3_OF_3);
   Servos[MED1].attach(MED_PANEL_1_OF_2);
   Servos[MED2].attach(MED_PANEL_2_OF_2);
   Servos[LRG1].attach(LARGE_PANEL);
   Servos[PIE1].attach(PIE1_SERVO_PIN);
   Servos[PIE2].attach(PIE2_SERVO_PIN);
   Servos[PIE3].attach(PIE3_SERVO_PIN);
   Servos[PIE4].attach(PIE4_SERVO_PIN);


   // Close them in a non-sequencial order and not at the same time to make it more interesting
   // Basically 2, then 2
                Servos[PIE1].write(NEUTRAL,CLOSESPEED);
   colorWipe(basicColors[random(1,10)], 255);                    // Turn Camera Lens LED Random Colors
                Servos[PIE3].write(NEUTRAL,CLOSESPEED,true); // wait
                Servos[PIE4].write(NEUTRAL,CLOSESPEED);
   colorWipe(basicColors[random(1,10)], 255);                    // Turn Camera Lens LED Random Colors
                Servos[PIE2].write(NEUTRAL,CLOSESPEED,true); // wait
                Servos[LTL1].write(NEUTRAL,CLOSESPEED);
   colorWipe(basicColors[random(1,10)], 255);                    // Turn Camera Lens LED Random Colors
                Servos[MED1].write(NEUTRAL,CLOSESPEED,true); //wait
                Servos[LTL2].write(NEUTRAL,CLOSESPEED);
   colorWipe(basicColors[random(1,10)], 255);                    // Turn Camera Lens LED Random Colors
                Servos[LRG1].write(NEUTRAL,CLOSESPEED,true); //wait
                Servos[LTL3].write(NEUTRAL,CLOSESPEED);
   colorWipe(basicColors[random(1,10)], 255);                    // Turn Camera Lens LED Random Colors
                Servos[MED2].write(NEUTRAL,CLOSESPEED,true); //wait

   // Detach from the Pies
                Servos[LTL1].detach();
                Servos[LTL2].detach();
                Servos[LTL3].detach();
                Servos[MED1].detach();
                Servos[MED2].detach();
                Servos[LRG1].detach();
                Servos[PIE1].detach();
                Servos[PIE2].detach();
                Servos[PIE3].detach();
                Servos[PIE4].detach();

   Serial.println("Closed");
   colorWipe(C_BLUE, 255);                    // Turn Camera Lens LED Blue

  } else { // Open Pie Panels
   Serial.println("Opening");
   panelsOpen=true;
    colorWipe(C_GREEN, 255);                    // Turn Camera Lens LED Green
   // Attach to all the panel so we can move them
                Servos[LTL1].attach(LITTLE_PANEL_1_OF_3);
                Servos[LTL2].attach(LITTLE_PANEL_2_OF_3);
                Servos[LTL3].attach(LITTLE_PANEL_3_OF_3);
                Servos[MED1].attach(MED_PANEL_1_OF_2);
                 Servos[MED2].attach(MED_PANEL_2_OF_2);
                Servos[LRG1].attach(LARGE_PANEL);
                Servos[PIE1].attach(PIE1_SERVO_PIN);
                Servos[PIE2].attach(PIE2_SERVO_PIN);
                Servos[PIE3].attach(PIE3_SERVO_PIN);
                Servos[PIE4].attach(PIE4_SERVO_PIN);

   // Open them all at once.
                Servos[LTL1].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[LTL2].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[LTL3].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[MED1].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[MED2].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[LRG1].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[PIE1].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[PIE2].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[PIE3].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED);
                Servos[PIE4].write(NEUTRAL-PIE_ADD_DEGREES,OPENSPEED, true);

                // Detach from the Pies
                Servos[LTL1].detach();
                Servos[LTL2].detach();
                Servos[LTL3].detach();
                Servos[MED1].detach();
                Servos[MED2].detach();
                Servos[LRG1].detach();
                Servos[PIE1].detach();
                Servos[PIE2].detach();
                Servos[PIE3].detach();
                Servos[PIE4].detach();

   Serial.println("Opened");

  }
  I2cCommand=-1; // always reset I2cCommand to -1, so we don't repeatedly do the same command
   //digitalWrite(STATUS_LED, LOW);
}
```

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Dome Button Cntr I2c = 11

Utility Arms Open/Close
1 – Opens and closes the Utility Arms
	Button 4	b=4,4,9,3	case 3:
			UtilityArms();
			break;
			void UtilityArms() {
			
			digitalWrite(STATUS_LED, HIGH);
			
			// If the Arms are open the close them
			if (topUtilityArmOpen) {
			Serial.println("Close utility arms");
			topUtilityArmOpen=false;
			Servos[T_UTIL_ARM].attach(T_UTIL_ARM_SERVO_PIN);
			Servos[B_UTIL_ARM].attach(B_UTIL_ARM_SERVO_PIN);
			
			// pull arms slightly beyond neutral to make sure they're really closed. Will vary on your droid
			Servos[T_UTIL_ARM].write(NEUTRAL-30,UTILITYARMSSPEED);
			Servos[B_UTIL_ARM].write(NEUTRAL-30,UTILITYARMSSPEED,true); // wait on arm to reach position
			
			Servos[T_UTIL_ARM].write(NEUTRAL-1,UTILITYARMSSPEED);
			Servos[B_UTIL_ARM].write(NEUTRAL-1,UTILITYARMSSPEED,true);
			
			Servos[T_UTIL_ARM].detach(); // detach so we can move the arms freely
			Servos[B_UTIL_ARM].detach();
			// Open the arms if closed
			} else {
			Serial.println("Open utility arms");
			topUtilityArmOpen=true;
			Servos[T_UTIL_ARM].attach(T_UTIL_ARM_SERVO_PIN);
			Servos[B_UTIL_ARM].attach(B_UTIL_ARM_SERVO_PIN);
			
			// Set Servo position to Open. In our case NEUTRAL is 90, and open is close to zero,
			// but on one of my arms it needs to be just shy of that so it's set to 25
			// We're also moving them at a moderate speed
			Servos[T_UTIL_ARM].write(25,UTILITYARMSSPEED); // open at moderate speed
			Servos[B_UTIL_ARM].write(0,UTILITYARMSSPEED);
			}
			
			i2cCommand=-1; // always reset i2cCommand to -1, so we don't repeatedly do the same command
			digitalWrite(STATUS_LED, LOW);
			}							

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Dome Button Cntr I2c = 11

HP Rainbow
1 - Waves the Dome Panels
2 - Flashes the Camera LEDs to Random Colors.
3 - Turns the HP LEDs to Rainbow
	
	Up, Up
		g=252,6,25,1
		a=A007
		a=A001
		a=A00305
		a=A0045
		a=A0060
			Space Holer				Turn all HPs to Rainbow
							
							void rainbow(byte hp) {
							int inter = 10;
							long elapsed;
							byte frames;
							elapsed = millis() - tCounter[hp];
							frames = elapsed/inter;
							if(frames > 256*5) {
							tCounter[hp]=millis();
							}
							else {
							for(int i=0; i<NEO_JEWEL_LEDS; i++) {
							neoStrips[hp].setPixelColor(i, Wheel(((i * 256 / NEO_JEWEL_LEDS) + frames) & 255));
							}
							if(elapsed>=inter) {neoStrips[hp].show();}
							}
							}			

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Dome Button Cntr I2c = 11
HP Cyclone Blue 1 - Waves the Dome Panels 2 - Flashes the Camera LEDs to Random Colors, 3 - Turns the HP LEDs to Rainbow	Down, Down	g=858,6,25,4 a=A007 a=A001 a=A00305 **a=A0045** a=A0060	Space Holer				Turn all HPs to Cyclone Blue			

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Dome Button Cntr I2c = 11
HP Random Color Pulse	Down, Up	g=852,6,25,3 a=A007 a=A001 **a=A00305** a=A0045 a=A0060	Space Holer				Turn all HPs to Pulse Random Color Medium Speed			

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Dome Button Cntr I2c = 11
HP Leia Message	Up. Down	g=258,6,25,2 a=A007 **a=A001** a=A00305 a=A0045 a=A0060	Space Holer				Turn all HPs to Leia Message			

Back to Quick Reference Tab

Action	Gesture	Stealth Config.txt	Body Servo Expander I2C = 9	Body Arduino Pro Mini I2C = 8	Body Arduino Mega I2C = 7	Dome Expander I2C = 10	Flthy HP I2C = 25	HP Servo Board I2C = 65	Teeces (Serial)	Teeces (Serial)
Quick Wave 1 – Opens the Dome 2 – Changes the Camaera Lens LED Colors to a random color each time it closes a few panels 3 – Sets the HP's to random pulse	Left, Left	g=454,4,10,8				case 8: quickWave(); break;				

```
void Wave1() {
  sendFlthyI2Ccmd("A007");
  Serial.println("Wave Panels 1");
// Wave 1
                // Attach to all the panel so we can move them
                Servos[LTL1].attach(LITTLE_PANEL_1_OF_3);
                Servos[LTL2].attach(LITTLE_PANEL_2_OF_3);
                Servos[LTL3].attach(LITTLE_PANEL_3_OF_3);
                Servos[MED1].attach(MED_PANEL_1_OF_2);
                Servos[MED2].attach(MED_PANEL_2_OF_2);
                Servos[LRG1].attach(LARGE_PANEL);
                Servos[PIE1].attach(PIE1_SERVO_PIN);
                Servos[PIE2].attach(PIE2_SERVO_PIN);
                Servos[PIE3].attach(PIE3_SERVO_PIN);
                Servos[PIE4].attach(PIE4_SERVO_PIN);

                //Move the Servos to Nuetral position
                    Servos[LTL1].write(NEUTRAL,150);
                Servos[LTL2].write(NEUTRAL,150);
                Servos[LTL3].write(NEUTRAL,150);
                Servos[MED1].write(NEUTRAL,150);
                Servos[MED2].write(NEUTRAL,150);
                Servos[LRG1].write(NEUTRAL,150);
                    Servos[PIE1].write(NEUTRAL,150);
                    Servos[PIE2].write(NEUTRAL,150);
                    Servos[PIE3].write(NEUTRAL,150);
                    Servos[PIE4].write(NEUTRAL,150);


  for (int i=0; i<4; i++) {                                    //RUN THIS 4 TIMES
    Serial.println("Wave");
                colorWipe(basicColors[random(0,10)], 255);     // Turn Camera Lens LED Random Colors
                Servos[LTL1].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[LTL2].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[LTL3].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[MED1].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[MED2].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[LRG1].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[PIE1].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[PIE2].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[PIE3].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY);
                Servos[PIE4].write(NEUTRAL-PIE_ADD_DEGREES,WAVESPEED);
                delay(WAVEDELAY * 2.5);
                Servos[LTL1].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[LTL2].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[LTL3].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[MED1].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[MED2].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[LRG1].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[PIE4].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[PIE3].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[PIE2].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY);
                Servos[PIE1].write(NEUTRAL,WAVESPEED);
                delay(WAVEDELAY * 2);
}

// Disconnect from our servos
                Servos[LTL1].detach();
                Servos[LTL2].detach();
                Servos[LTL3].detach();
                Servos[MED1].detach();
                Servos[MED2].detach();
                Servos[LRG1].detach();
                Servos[PIE1].detach();
                Servos[PIE2].detach();
                Servos[PIE3].detach();
                Servos[PIE4].detach();

  colorWipe(C_BLUE, 255);                    // Turn Camera Lens LED Blue
  sendFlthyI2Ccmd("A0015F4");
  Serial.println("Wave Done");


  i2cCommand=-1; // always reset i2cCommand to -1, so we don't repeatedly do the same command
  //digitalWrite(STATUS_LED, LOW);
}
```

Turn all HPs to Rainbow

Turn all HPs to pulse blue and speed of 4