

Title: Task Board Specification document

Subject: Electrical component team

Contributors: Electrical team, Isai Mateo , Matthew Alaniz

Last update: 1/7/2020, need to recount inventory to ensure we have enough diodes leak sensor probes

General specifications:

The Board System acts as the peripheral module that handles the rest of tasks performed by the system of the sub aside from thrusters. This system handles tasks:

- Controlled 12V output sources for Pneumatics
 - for our purposes we prototype with solenoids, this task handles the drop weights and thrusters
 - The 12V is imputed from the buck converter and distributed in parallel as inputs to two p-channel mosfets.
 - The N-channel Mosfet's act as open circuits until they receive a 5v logic signal at their gete. At that instance they act as closed circuits allowing the 12V to be inputted into the solenoid

```
SolenoidDrop
int SSignal;
bool ReadytoDrop = false;

void setup() {
  // put your setup code here, to run once:
  pinMode(SSignal, OUTPUT);
  digitalWrite(SSignal, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(ReadytoDrop == true){
    digitalWrite(SSignal, HIGH); //Switch Solenoid ON
    delay(25);
    digitalWrite(SSignal, LOW); //Switch it solenoid off again
    delay(25);
    ReadytoDrop = false;
  }
}
```

- LCD screen battery/ statemachine machine back
 - This feature task is to provide feedback for the battery's so we can see the voltage levels of our batteries, as well as any state machine logic feedback deemed as necessary from the software team.
 - The battery feedback is important so we can track or its voltage and pull the sub out if the voltage drops too low, if the battery voltage drops too low they can become chemically damaged.

- LCD screen is 1 x 1.8 Inch TFT LCD Display Module. It connects to the board using a 10 pin connector, JST-XH 8S male. (128x160 pixels)

LCD

```
#include <TFT.h>
#include <SPI.h>

#define cs 10
#define dc 9
#define rst 8

TFT LCDScreen = TFT(cs, dc, rst);

float num = 0;
String number = "";
char printout[10];

void setup() {
  // put your setup code here, to run once:
  LCDScreen.begin();//initialize the LCD screen
}

void loop() {
  // put your main code here, to run repeatedly:
  if (num > 21){
    num = 0;//once num reaches 21 reset to 0
  }
  else {
    num = num + 1;//increment num by 1
  }

  LCDScreen.background(0, 0, 0);//Clear screen
  LCDScreen.stroke(0, 255, 0);//Text color set to Green
  LCDScreen.setTextSize(2);//change text font to small
  LCDScreen.text("Battery 1", 0, 0);//Print to tope left corner "Battery 1"

  number = String(num/9);//turn num/9 to a string
  number.toCharArray(printout, 10);//add the number string to a char array printout
  LCDScreen.setTextSize(5);//change text font to big
  LCDScreen.text(printout, 0, 20);//print the value in the printout array

  LCDScreen.setTextSize(2);
  LCDScreen.text("Battery 2", 0, 60);
  LCDScreen.setTextSize(5);
  LCDScreen.text(printout, 0, 80);
  delay(500);
}
```

- Bar 30 pressure sensor
 - Connects to board via headers.
 - Connects to pressure sensor via atx molex 4 pin connector. Logic IC2 signals to communicate

PressureSensor

```
#include <Wire.h>
#include "MS5837.h"
#include <String.h>

MS5837 sensor;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); //start the serial monitor
  Serial.println("Starting");
  Wire.begin(); //Initialize wire library
  sensor.init(); //must be called before operating sensor (returns true if successful)
  sensor.setFluidDensity(997); //kg/m^3 (997 freshwater, 1029 for seawater)
}

void loop() {
  // put your main code here, to run repeatedly:
  sensor.read();

  Serial.print("Pressure: ");
  Serial.print(sensor.pressure()); //print the pressure that the bar30 senses
  Serial.println(" mbar");

  Serial.print("Depth: ");
  Serial.print(sensor.depth()); //print the depth that the bar30 senses
  Serial.println(" m");
}
```

- Neopixel RGB light strip
 - Light strip has 3 inputs, VCC, Logic signal input, GND
 - The recommended VCC input range is 5V-3.7V & The manufacturer recommends a minimum signal voltage of 70% of the NeoPixel input voltage
 - We power it with 3.3V Vcc from the Teensy 3.2 MCU and the logic signals can come from a digital pin also with 3.3V

NEOPIXEL

```
#include <Adafruit_NeoPixel.h>

#define PIN 8 // LED PIN
#define NUMPIXELS 9 // # of LEDs

Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  // put your setup code here, to run once:
  strip.begin(); // Initializes NeoPixel library
  strip.show(); // Sets all LED's to off
}

void loop() {
  // put your main code here, to run repeatedly:
  for (int i = 0; i < NUMPIXELS; i++) { // Do for each led in the strip
    strip.setPixelColor(i, 0, 0, 255); //we pick the color Green
  }
  strip.show(); // This sends the updated pixel color to the hardware.
}
```

- Leak Sensor Module
 - Sensor is purchased from bluerobotics that senses water leaks

- The Sensor package has 2 pieces, a module to interface with the task board, and a sponge-like sensor that interfaces with the module.
- The module connects to the board via VCC, GND, digital Logic signal connections.
- When the module returns logic signal high, this means a leak has been detected, when the module logic signal returns low than no leak has been detected.

LeakSensor

```
int Leak1 = A1;
int value = 0;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); //begin the serial monitor
  Serial.println("Starting");

  pinMode(Leak1, INPUT); //set Leak sensor as input
}

void loop() {
  // put your main code here, to run repeatedly:
  value = analogRead(Leak1); //save Leak reading into value
  Serial.print("Leaking: ");
  Serial.println(value); //print the value leak sensor receives
}
```

Parts

- 2 N-channel mosfets
- 2 Schottky diodes
- LCD screen
- LCD screen connector
- Bar30 pressure sensor
- 4pin Atx connector
- Neo pixel RBG Lights strip
- Leak sensor Module (max 2)
- Probes(max 8)
- XT-60 connectors(x5) type of subject to connector
- 6 SMD resistors

Purchases necessary:

- 2 N-channel power mosfets
- 6 SMD resistors