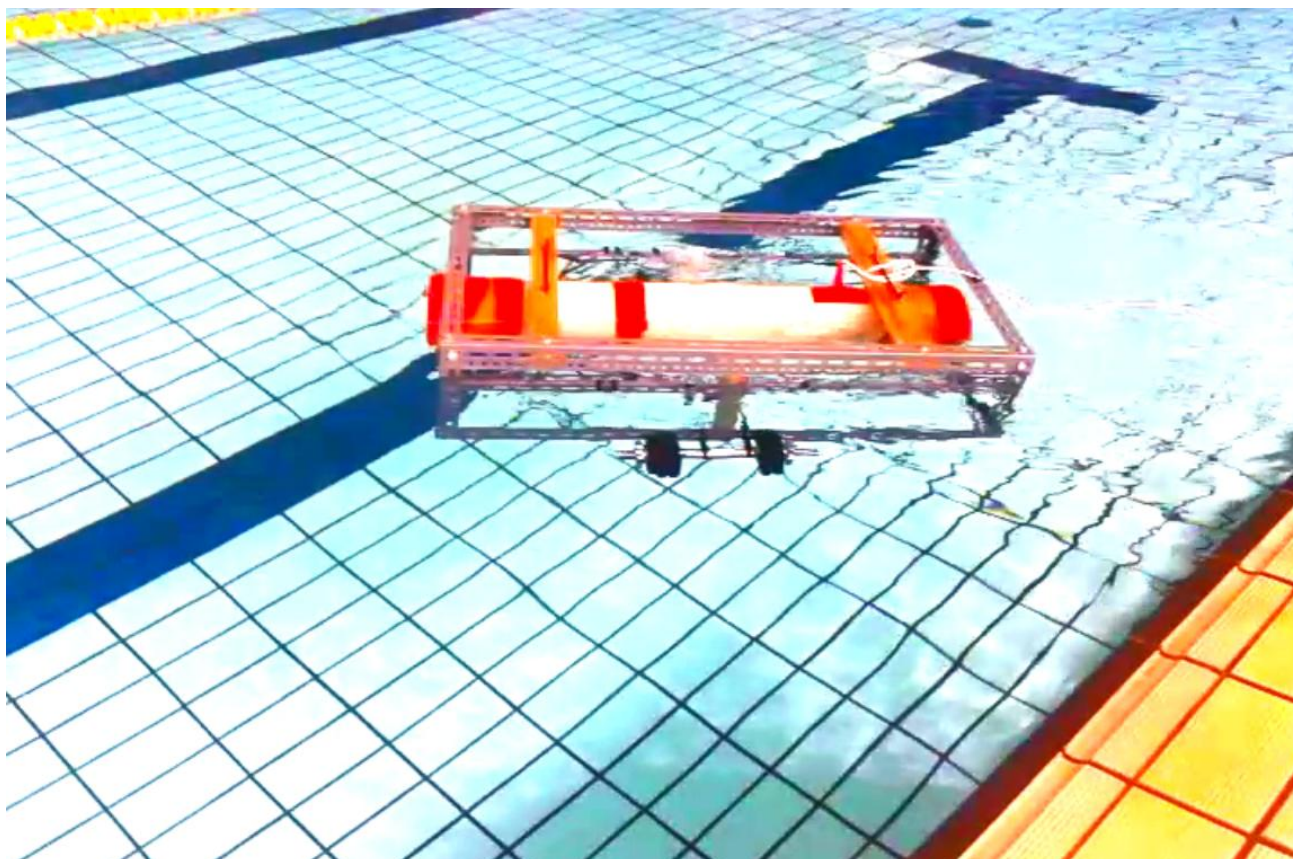


Sea Explorer



Un progetto di

*Balestrini Andrea;
Vallet David;
Ruggeri Giulio.*

SOMMARIO

I	Premessa	pag. 4
II	Introduzione al progetto	pag. 5
III	Schema a blocchi e Componenti	pag. 6
	III a. Schema a blocchi	pag. 6
	III b. Componenti	pag. 7
	Motori brushless	pag. 7
	ESC	pag. 7
	Eliche	pag. 8
	Batterie LiPo	pag. 8
	Microcontrollore Arduino MEGA	pag. 8
	Seeeduino ethernet shield V2_0	pag. 9
	DC/DC converter 12V/9V	pag. 9
	DC/DC converter 12V/5V	pag. 9
	IPCam	pag. 10
	Router	pag. 10
	Diodi di potenza IOSQ050	pag. 10
	Faretti Led	pag. 11
	Transistor BD179	pag. 11
	Trasduttore	pag. 11
	DHT22	pag. 11
	H2OhNo	pag. 12
	Regolatore lineare LM7805	pag. 12
	Tubi e tappi in PVC e supporti	pag. 12
	Lastra in plexiglass	pag. 13
	Viti, bulloni, rondelle, resistori, cavi, guaine e altri materiali	pag. 13
IV	Elettronica	pag. 14
	IV a. Schema del circuito	pag. 14
	IV b. Elettronica dei componenti	pag. 15
	Conversione AD Arduino	pag. 15
	Circuito transistor	pag. 16

V	Montaggio	pag. 17
VI	Funzionamento programma	pag. 21
VII	Flow-Chart e spiegazioni	pag. 22
	VII a. Programma principale	pag. 22
	VII b. Subroutines	pag. 27
	Pagina WEB	pag. 27
	Controllo motori	pag. 34
	Controllo luci	pag. 36
	Misurazione temperatura interna	pag. 37
	Misurazione temperatura esterna	pag. 38
	Monitoraggio carica batterie	pag. 39
	Rilevamento infiltrazioni d'acqua	pag. 40
VIII	Considerazioni finali	pag. 41
	VIII a. Test in acqua	pag. 41
	VIII b. Problemi riscontrati e possibili soluzioni	pag. 42

Sitografia: <http://www.arduino.cc/>
<http://forum.arduino.cc/>
<https://it.wikipedia.org/>
<http://www.hobbyking.com/>
siti e forum di elettronica vari

I. Premessa

Il Sea Explorer è un ROV (Remotely-Operated-Vehicle) ossia un veicolo pilotato da una postazione remota. Questi mezzi furono ideati dalla US Navy, a partire dal 1953, per trovare, distruggere o disinnescare ordigni esplosivi subacquei. Negli anni successivi le forme e gli utilizzi di questi mezzi si sono moltiplicati e sono ora utilizzati in molti campi quali: il recupero di oggetti, lavori di precisione subacquei (piattaforme petrolifere ecc.) o per la semplice esplorazione di luoghi inaccessibili all'uomo.

L'International Marine Contractors Association (IMCA) ha stabilito cinque classi di ROV.

Classe I → ROV da ispezione, completamente elettrici e normalmente dotati di telecamera. Usati principalmente per controllare piattaforme, porti o tubature.

Classe II → questi mezzi sono di dimensioni più grandi e usano sistemi di tipo elettrici e idraulici. Vengono usati per lavori leggeri come trasporto di oggetti o posizionamento di materiali sul fondale e sono quindi dotati di braccia meccaniche o altri sistemi.

Classe III → di dimensioni notevoli, sono ROV completamente idraulici. Data la loro facilità nel raggiungere grandi profondità sono spesso dotati di sensori per la scansione marina e utilizzati per la creazione di carte oceanografiche.

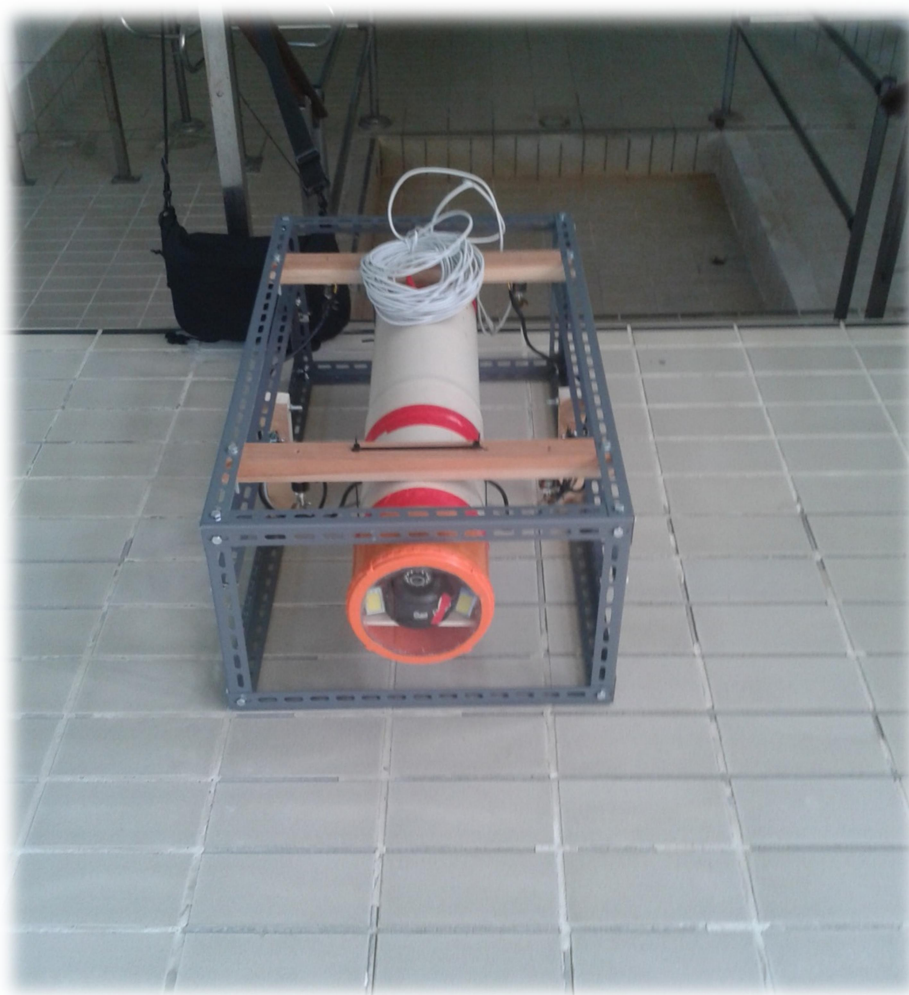
Classe IV → Sono ROV molto voluminosi e dotati di cingoli, di solito si poggiano sul fondale marino e poi vengono trainati via nave. Il principale uso a cui è destinato questo tipo di ROV è l'interramento di tubazioni marine.

Classe V → Ancora in fase di sviluppo. Sono dei ROV non più comandati da una persona a terra ma programmati prima dell'immersione per seguire una determinata rotta.

II. Introduzione al progetto

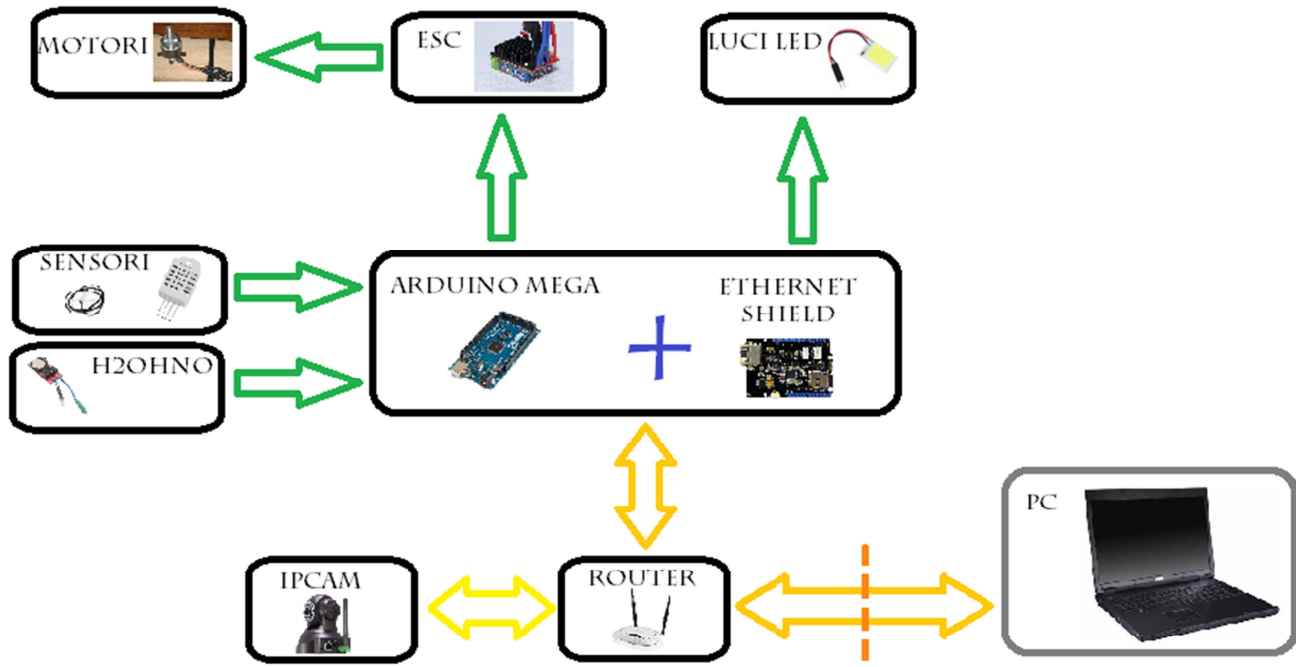
Il Sea Explorer può essere considerato come un ROV di classe I in quanto il suo scopo principale è dedicarsi all'esplorazione di ambienti acquatici quali piscine, condutture, vasche o ambienti simili. Non è progettato per eseguire operazioni meccaniche sul luogo di esplorazione. I suoi 6 motori, 4 per l'immersione e l'emersione e 2 per gli spostamenti orizzontali, gli permettono una mobilità non indifferente nonostante la stazza di circa 50Kg (25Kg di componenti più 25Kg di zavorra).

La zavorra è anche utile per bilanciare l'assetto del Sea Explorer in maniera simile a come fanno i sub, in modo da conferirgli un peso "zero" in acqua: questo piccolo accorgimento consente una serie di vantaggi quali una durata superiore delle batterie, un minore lavoro dei motori e una maggiore facilità di recupero del mezzo in caso di emergenza.



III. Schema a blocchi e componenti

III a. Schema a blocchi



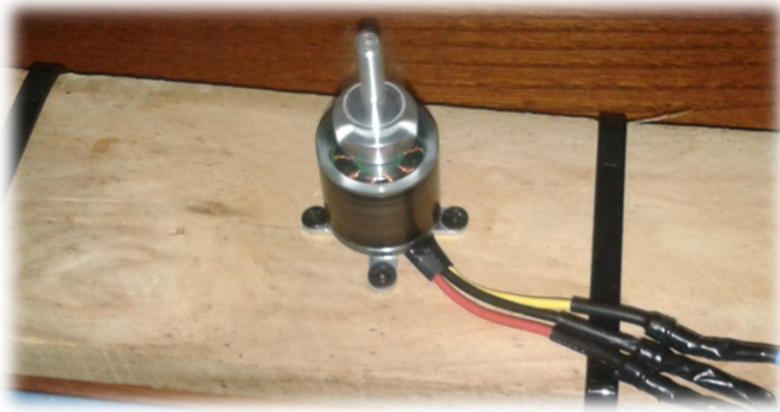
Lo schema a blocchi mostra sinteticamente il funzionamento ed i componenti presenti nel Sea Explorer. Le frecce verdi indicano segnali di tipo analogico/digitale mentre le frecce arancio e gialle segnali web (le prime wireless le seconde via cavo). La linea tratteggiata indica che il segnale è in uscita dal mezzo.

Come si può vedere il “centro di controllo” del Sea Explorer, composto da Arduino e Shield, riceve i dati dei sensori, che verranno inviati al computer, e comanda le luci e gli ESC (che a loro volta comandano i motori). Da computer si inviano i comandi e si ricevono i dati, opportunamente elaborati, inviati dai sensori. Il router permette la comunicazione tra IPCam, Arduino/Shield e computer.

III b. Componenti

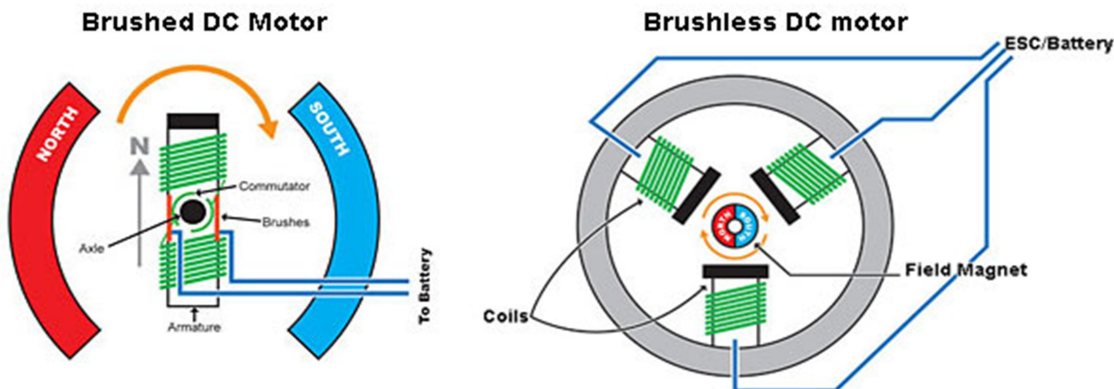
6x Motori brushless

Questi motori, alimentati da una tensione trifase, hanno una corrente massima di 20A e una velocità di 750 rpm/V. Non necessitano delle



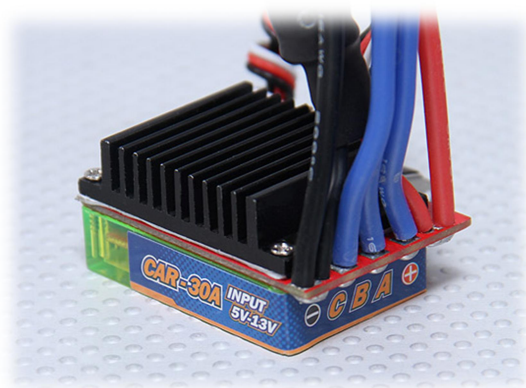
spazzole per funzionare e di conseguenza non hanno contatti elettrici striscianti sull'albero motore.

Questo ci permette di utilizzarli in un ambiente subacqueo.



Nell'immagine possiamo notare i differenti tipi di bobine dei due tipi di motori: Brushed (con spazzole) e Brushless (senza spazzole).

6x ESC (Electronic Speed Controller)



Questo apparecchio, alimentato tra i 5 e 13V, ci fornisce un'uscita trifase comandata tramite un segnale PWM trasmesso, in questo caso, da un microcontrollore. La corrente massima fornita da questo modello in particolare è di 30A.

6x Eliche

Rispettivamente 4 di diametro 40mm e 2 di diametro 60mm, sono utilizzate per fornire la propulsione del Sea Explorer.



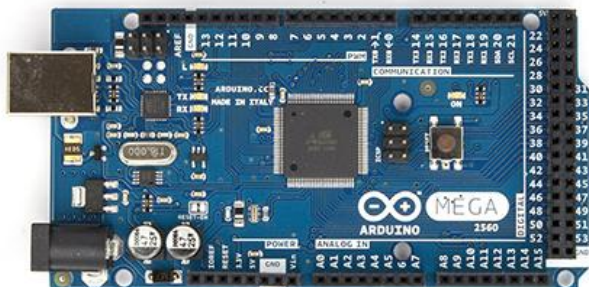
2x Batterie LiPo 11.1V 5Ah

Batterie al Litio Polimero altamente performanti, sono dotate di una grande corrente massima in uscita (150A di picco e 15A continuati) e di un'elevata capacità di carica: questo ne fa le batterie ideali per il nostro mezzo. L'unica pecca di questo tipo di batterie è che diventano inutilizzabili se la tensione delle singole celle che le compongono scende al di sotto dei 3V (quindi nel caso delle nostre batterie a 3 celle 9V in totale).



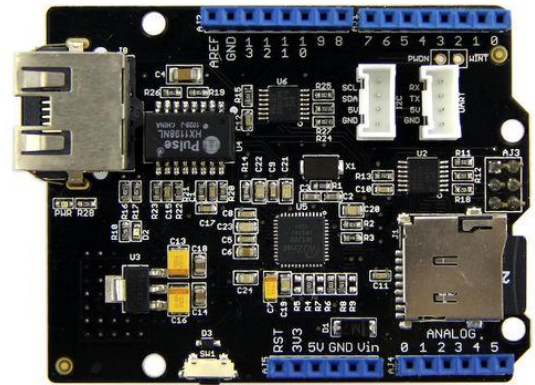
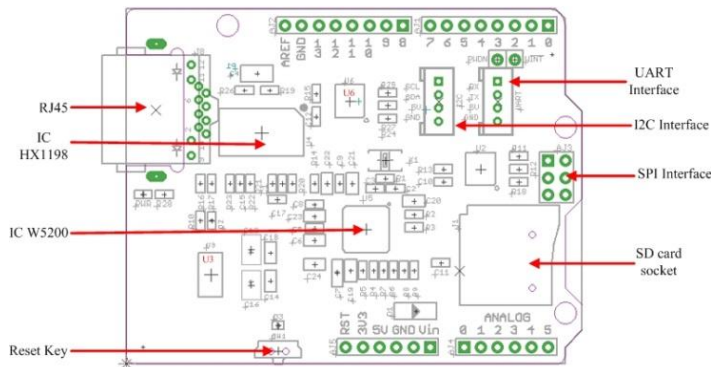
1x Microcontrollore Arduino MEGA

Arduino è un tipo di piattaforma basata sui microcontrollori della ATMEL, nel nostro caso l'ATmega2560. Questa board contiene 54 pin digitali di input/output (max 40mA) di cui 15 utilizzabili per inviare impulsi PWM, 4 UARTs (porte seriali hardware), un oscillatore al quarzo a 16 MHz, una connessione USB e un jack di alimentazione. Questo supporto integra all'interno tutto il necessario per poter funzionare semplicemente connettendo l'alimentazione alla porta USB o alimentando con il jack tra 6-20V (massimo) o 7-12V (raccomandato). È programmabile tramite linguaggio C o C++ in un ambiente apposito.



1x Seeeduino Ethernet shield V2.0

Questo shield, studiato apposta per i microcontrollori di tipo Arduino (o con piedinatura simile), consente di creare attraverso semplici righe di codice una pagina web utilizzabile senza problemi da qualsiasi dispositivo possa connettersi alla rete.



L'alimentazione viene presa dai pin di Arduino sottostanti così come la massa. I pin di Arduino che comandano la scheda sono il 4 (scheda SD) e il 10 (W5200) e l'interfaccia SPI.

È programmabile con un linguaggio analogo a quello di

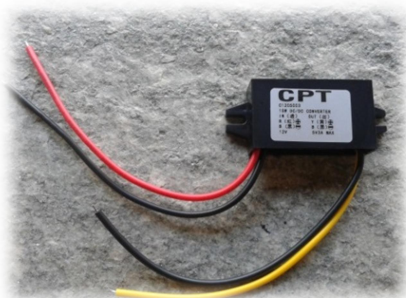
Arduino previa inclusione della libreria "EthernetV2_0.h"

1x DC/DC converter 12V/9V

Convertitore in corrente continua da 12 a 9V e corrente massima 2A. È utilizzato per alimentare il router.



1x DC/DC converter 12V/5V



Convertitore in corrente continua da 12 a 5V e corrente massima 3A.

È utilizzato per l'alimentazione della IP Cam.

1x IP Cam

Telecamera motorizzata in grado di ruotare su se stessa e alzare/abbassare l'obiettivo, è utilizzata per il controllo visivo. Dispone inoltre di visione a infrarossi per vedere nel buio.

L'antenna Wi-Fi viene utilizzata per l'invio dello streaming video al router a bordo del Sea Explorer.



1x Router

Utilizzato per la comunicazione computer-Arduino-IPCam. È alimentato a 9V per 0.6A. La sua capacità di trasmissione di 300MBps lo rendono adatto per trasmettere a una buona velocità lo streaming video.



2x Diodi di potenza IOSQ050

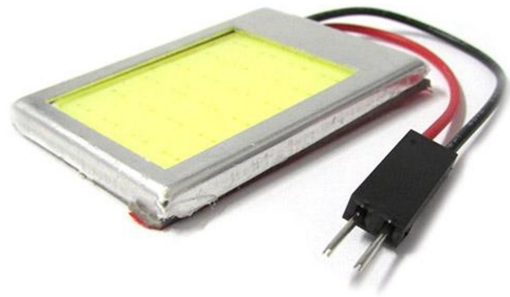
Diodi in grado di sopportare una corrente di decine di Ampère. Sono utilizzati per evitare l'autoscarica delle batterie una sull'altra.



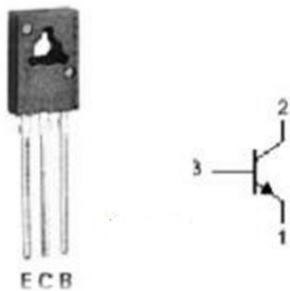
2x Faretti LED

Questi LED sono alimentati a 12V e hanno un consumo di circa 0.3A.

Sono necessari per garantire un'adeguata illuminazione alla telecamera.



1x Transistor BD179



È un normale transistor NPN con un h_{FE} di circa 100. Viene utilizzato per comandare l'accensione e lo spegnimento dei faretto LED.

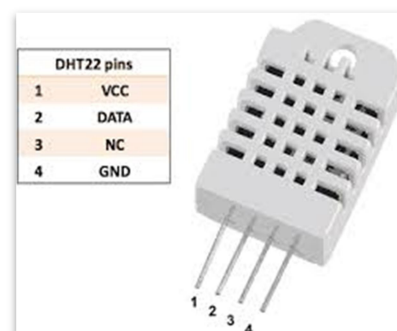
1x Trasduttore



Semplice resistore che varia in base alla temperatura, cui è possibile risalire grazie a un partitore di tensione costruito usando questo sensore e una resistenza.

1x DHT22

Questo sensore percepisce umidità e temperatura tramite due trasduttori ed è in grado di inviare i valori all'Arduino tramite comunicazione seriale uscente dal piedino numero 2 dell'integrato.

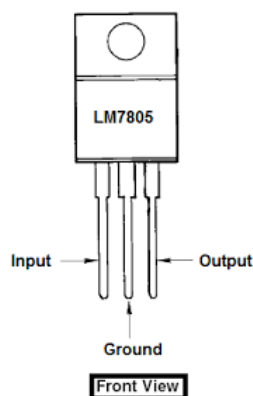


1x H2OhNo

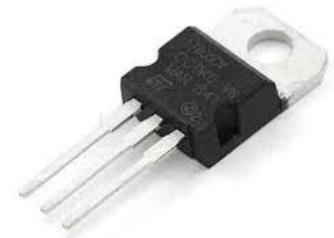
Questo tipo di sistema misura la presenza o meno di un corto-circuito tra due punte, nel caso si presenti questa eventualità vengono attivati un led e un segnale acustico. Avendo posizionato i due puntali nella parte inferiore del sottomarino è possibile utilizzare questo apparecchio come rivelatore d'acqua.



1x Regolatore lineare LM7805

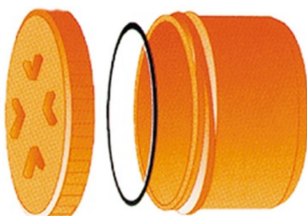


L'LM7805 è uno stabilizzatore di tensione utilizzato per ottenere una tensione di 5V. È una parte piuttosto importante della circuiteria interna poiché per i sensori è necessario avere una tensione di riferimento stabile in modo da non falsare le misure. Come componente è inadatto per forti correnti ma nel nostro caso funziona benissimo.



Tubi e tappi in PVC e supporti

Come impalcatura e struttura per tutto il Sea Explorer sono stati utilizzati tubi in PVC e un intelaiatura metallica/lignea per il sostegno dei motori. I tubi essendo impermeabili e resistenti sono un ottimo



contenitore per tutta la parte elettronica mentre le barre garantiscono per la solidità del tutto.

1x Lastra in plexiglass

Per avere una superficie trasparente dalla quale la telecamera potesse vedere si è usata una lastra in plexiglass (vista la buona resistenza del materiale) opportunamente lavorata a forma di oblò



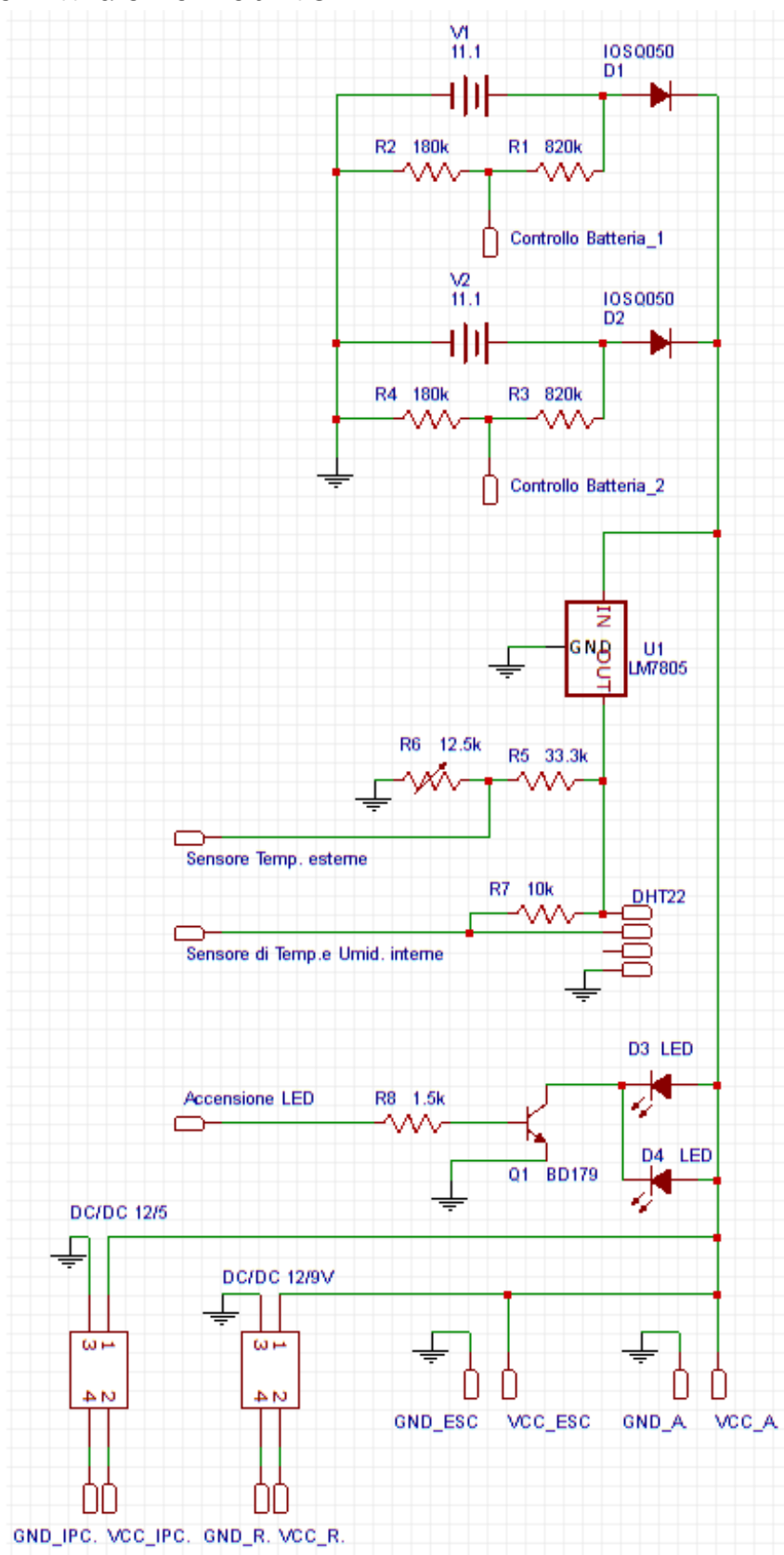
Viti, bulloni, rondelle, resistori, cavi, guaine e altri materiali

Sono stati utilizzati per fissare le varie parti e i componenti interni, per completare i circuiti, per isolarli dall' acqua, per effettuare collegamenti, ecc.



IV. Elettronica

IV a. Schema del circuito



IV b. Elettronica dei componenti

Conversione AD Arduino



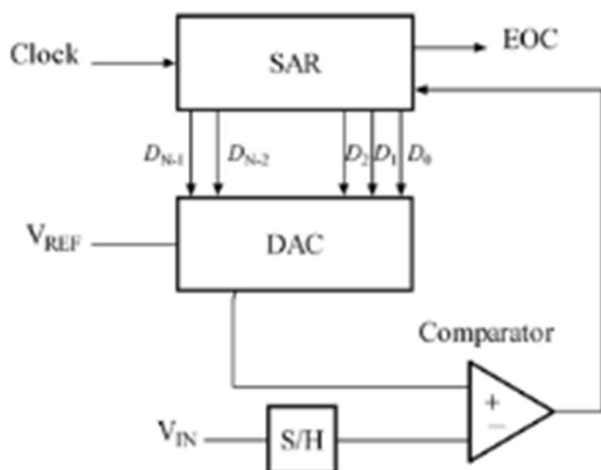
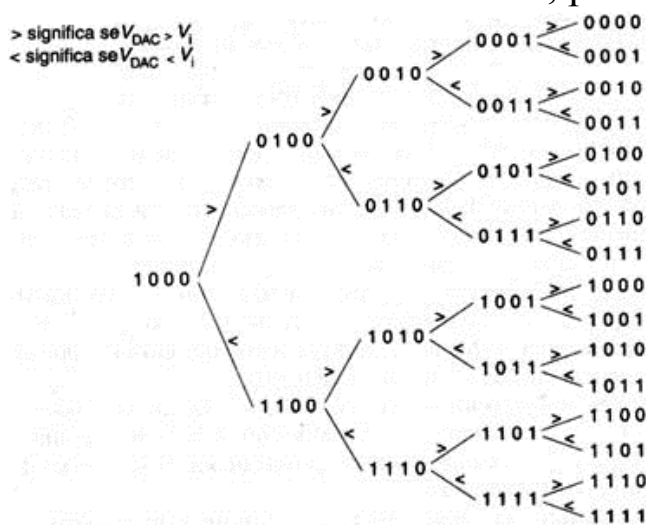
Per leggere i valori analogici immessi negli appositi piedini le schede di tipo Arduino fanno uso di un convertitore AD a 10 bit avendo così a disposizione 1024 valori (0-1023) per la conversione. Normalmente il convertitore è impostato per convertire valori da 0V (0) a 5V (1023), nel caso si volesse cambiare il range di misura è possibile fissare il valore massimo tramite il pin AREF presente sulla scheda al di sopra del pin 13.

Per effettuare la conversione la scheda utilizza un convertitore ad approssimazioni successive, questo tipo di ADC confronta dei valori digitali contenuti in un registro (convertiti tramite un DAC) con il valore in ingresso, andando a verificare se è maggiore o minore di quello dato. Una volta assegnato un valore al MSB effettuerà lo stesso passaggio con il bit successivo e così via. L'errore commesso

sarà pari a $Q/2$ con $Q = V_{ref}/2^{n_{bit}}$.

Il vantaggio di questo tipo di convertitore è la velocità di conversione, pari a $n_{bit} * T_{clk}$ che nel nostro caso è $10 * 16 * 10^{-6} = 160 \mu\text{sec}$

Nelle immagini si possono vedere lo schema a blocchi e un esempio di conversione con un ADC a 4 bit.



SAR → Successive Approximation Register

DAC → Digital Analog Converter

S/H → Sample & Hold

Circuito transistor

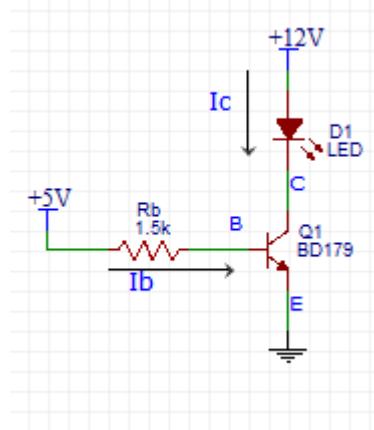
Per comandare l'accensione e lo spegnimento dei faretto LED si è deciso di utilizzare un transistor collegato a un'uscita digitale dell'Arduino MEGA. Conoscendo la corrente di collettore (misurata sperimentalmente come di circa 300mA) e l' h_{FE} si è potuta calcolare l'opportuna resistenza di base.

$$I_c = I_b * h_{FE}$$

$$I_b = (V_b - V_{be}) * R_b$$

$$0.3 = ((5 - 0.7) / R_b) * 100$$

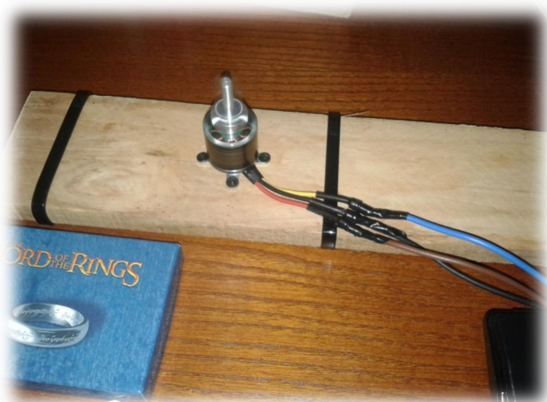
$$R_b = (4.3 / 3)k = 1.43k$$



Vista l'assenza dei resistori da 1.43K Ω si è optato per uno da 1.5 K Ω

V. Montaggio

Durante il montaggio abbiamo avuto notevoli difficoltà in merito all'impermeabilizzazione del ROV, problemi che abbiamo quasi del tutto risolto utilizzando colle apposite per PVC, colle ad alta tenuta (ad es. colle bicomponenti) e nastro adesivo specifico come ulteriore guaina di sicurezza. Di seguito sono riportate alcune immagini dello svolgimento dei lavori sul Sea Explorer comprese alcune fasi di test dei componenti



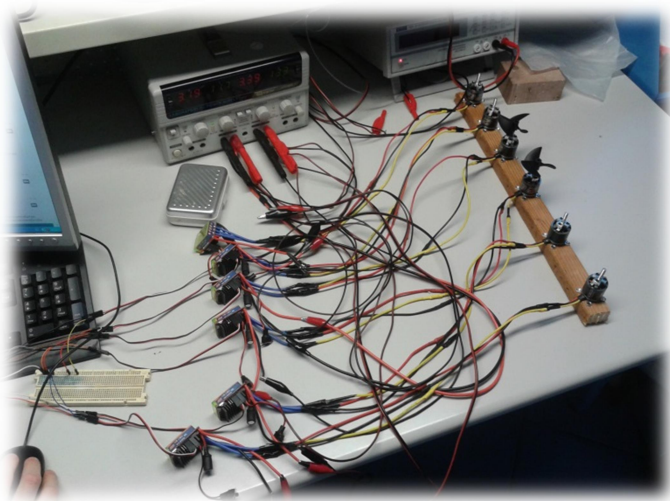
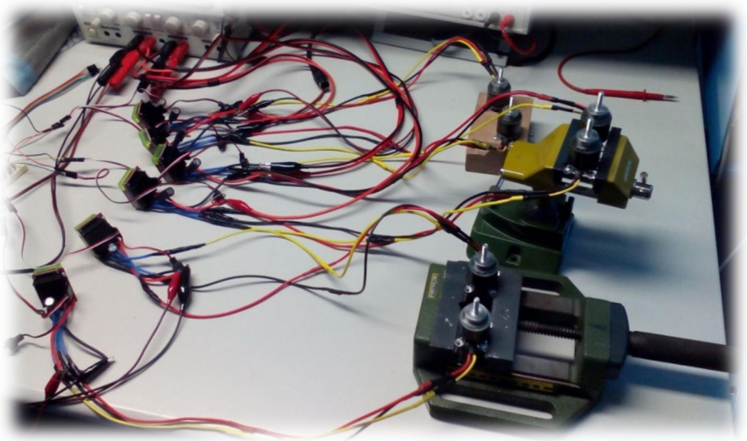
Test preliminare di un motore su tavola in legno



Test in acqua del motore con e senza elica per verificarne funzionamento e consumo



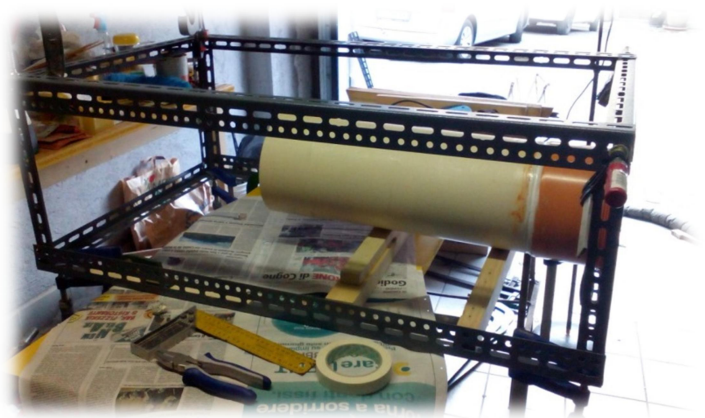
Test dei sei motori in contemporanea per verificarne funzionamento e consumi

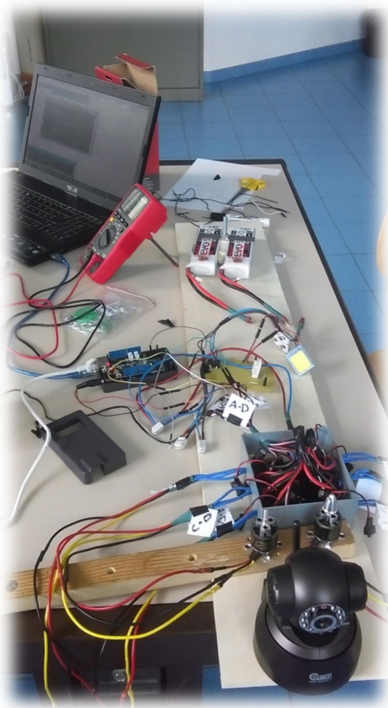


Prove motori con alcune eliche abbinate

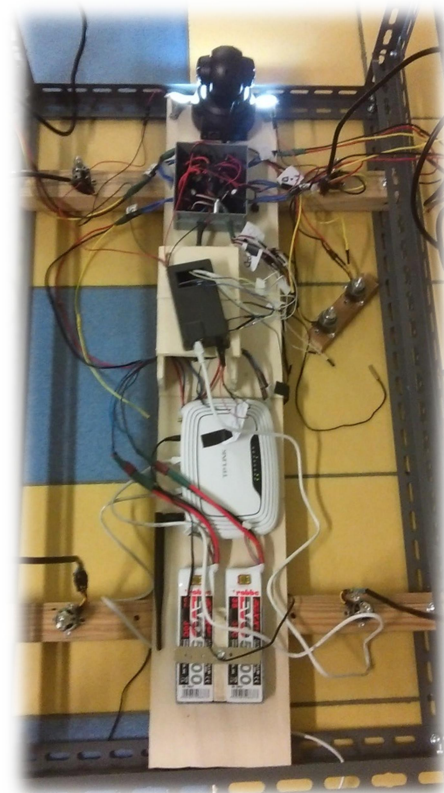
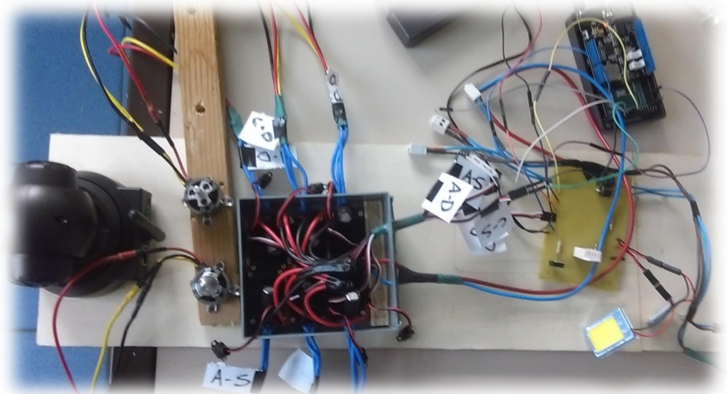


Montaggio del telaio e del corpo centrale del Sea Explorer

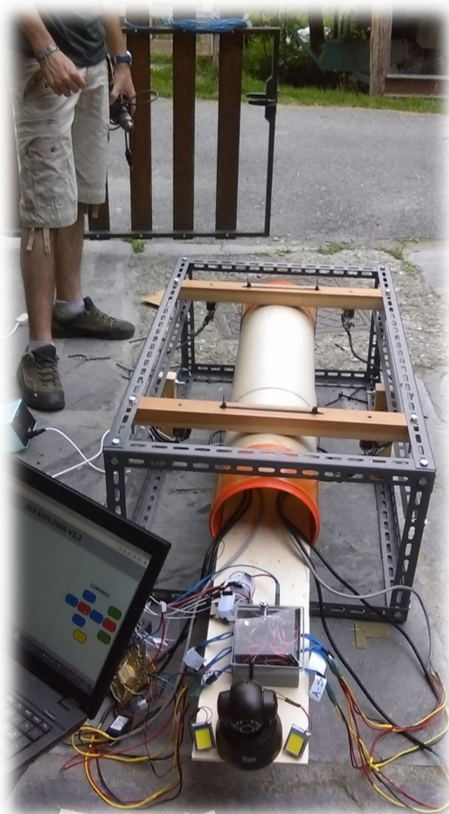




Alcune fasi di costruzione e test della circuiteria interna abbinate a controllo e debug del programma



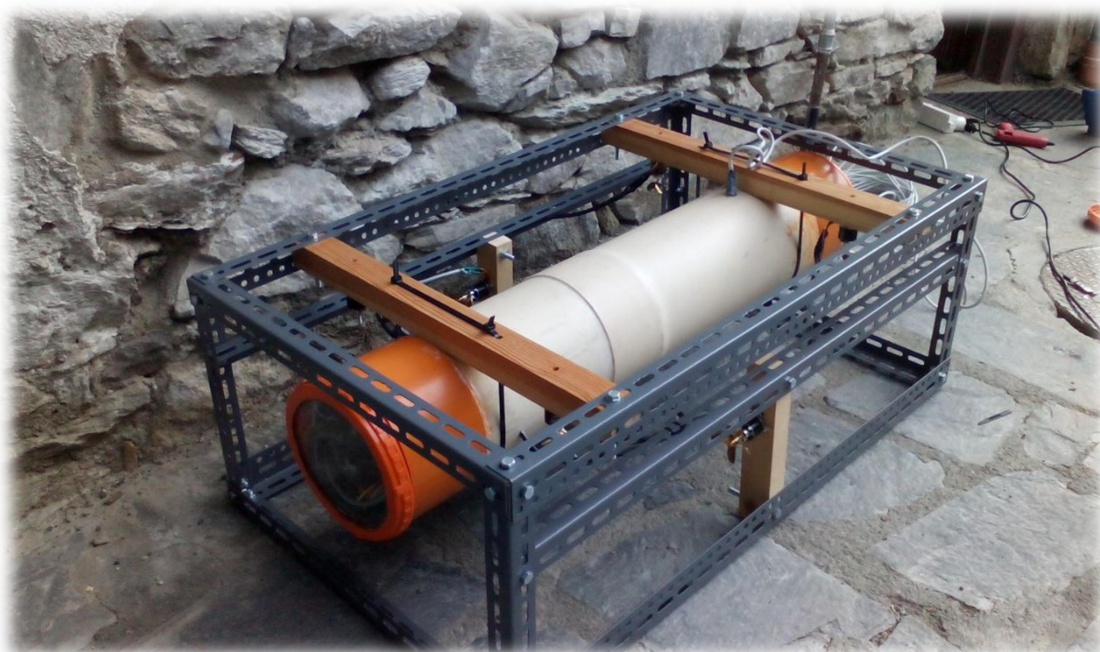
Controllo del posizionamento dei circuiti, del funzionamento delle singole parti e dei cablaggi



Verifiche e messe a punto finali

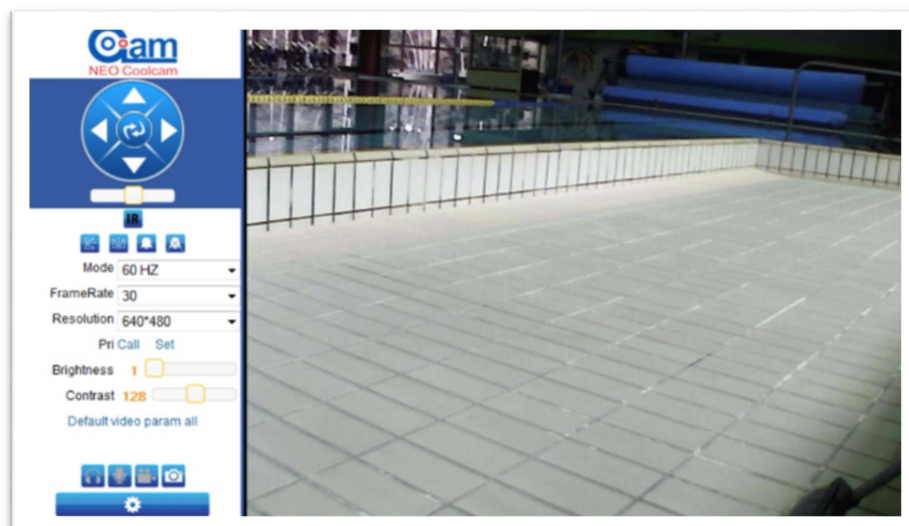


Sea Explorer pressoché completo e pronto al varo



VI. Funzionamento programma

Il funzionamento è piuttosto semplice e intuitivo: connettendo al computer l'apposito cavo LAN è possibile collegarsi alla pagina web di indirizzo "192.168.0.120" e utilizzarla per comandare il mezzo.



Come si può vedere sono presenti tutte le informazioni per governare il Sea Explorer.

Sulla sinistra dello schermo vengono mostrati i sensori e gli indicatori di bordo, al centro il video dell'IPCam mentre a destra i

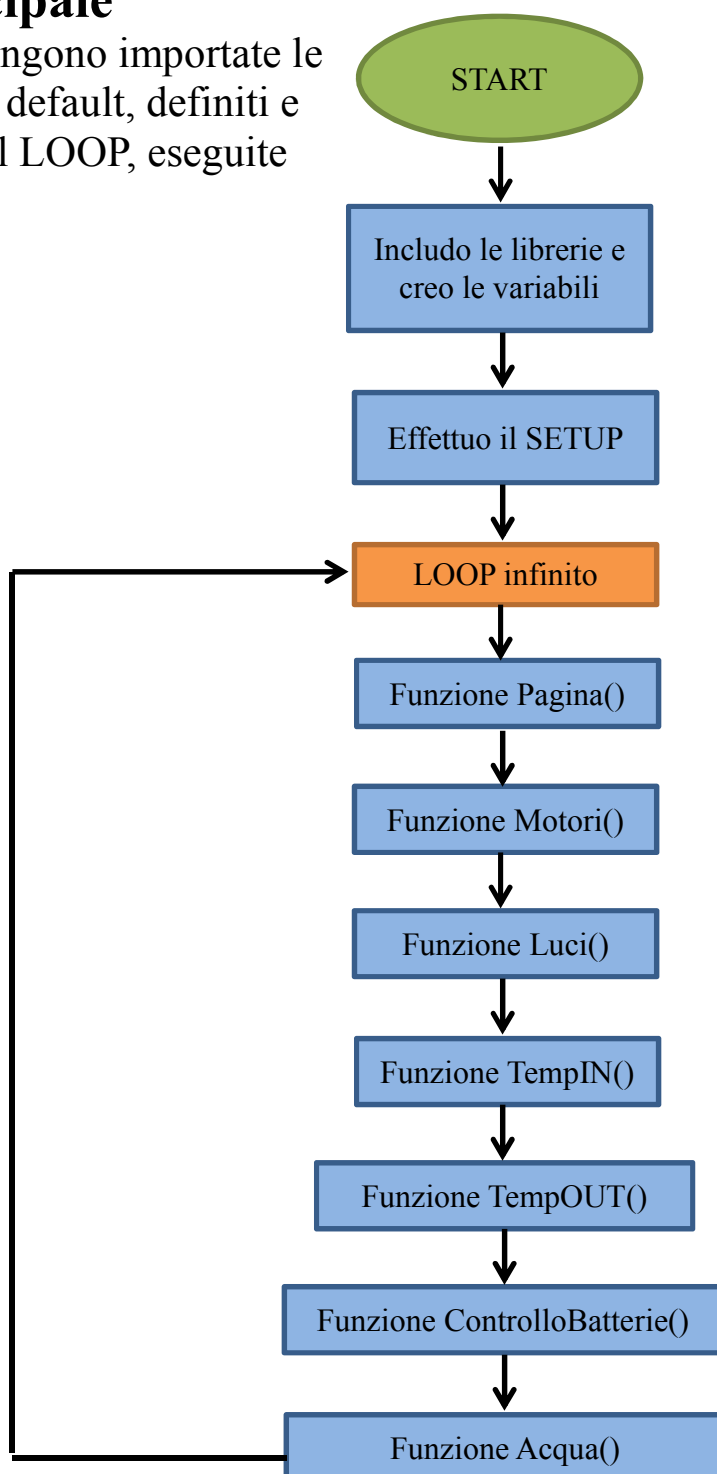
comandi per muoversi nelle tre direzioni (avanti/indietro, destra/sinistra, su/giù) assieme all'accensione e spegnimento luci. Per modificare i controlli e le impostazioni dell'IPCam è sufficiente collegarsi all'indirizzo "192.168.0.110:81".

VII. Flow-Chart e spiegazioni

Per facilitare la scrittura e la comprensione del programma si è deciso di dividerlo in più funzioni che vengono poi richiamate dal programma principale. Tutti i programmi sono scritti in linguaggio C/C++ apposito per Arduino

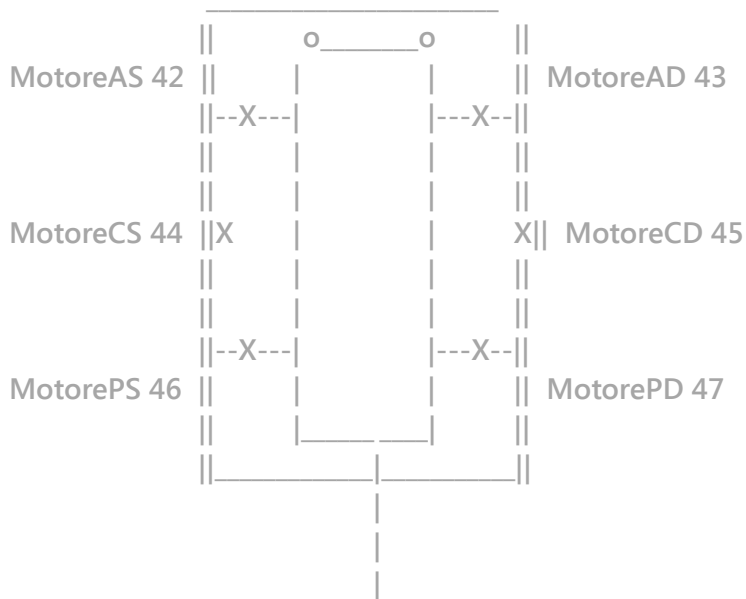
VII a. Programma principale

In questa parte di programma vengono importate le librerie, impostate le variabili di default, definiti e direzionati i pin e, all'interno del LOOP, eseguite le funzioni secondarie.



/*

BALESTRINI ANDREA, RUGGERI GIULIO, VALLET DAVID CLASSE V ELE
PROGETTO ESAME DI STATO 2015
SEA EXPLORER



*/

```
////////////////////////////////////  
/////INCLUDO LE LIBRERIE/////
```

```
////////////////////////////////////  
#include <SPI.h>  
#include <Servo.h>  
#include "DHT.h"  
#include <EthernetV2_0.h>
```

```
////////////////////////////////////  
////////DEFINISCO I PIN////////  
////////////////////////////////////
```

```
#define DHTPIN 19  
#define W5200_CS 10  
#define SDCARD_CS 4  
#define MotoreAS_PIN 42  
#define MotoreAD_PIN 43  
#define MotoreCS_PIN 44  
#define MotoreCD_PIN 45  
#define MotorePS_PIN 46  
#define MotorePD_PIN 47  
#define led_pin 40  
#define tempOUT_PIN A10  
#define panic_PIN A12  
#define Batteria_1_PIN A14  
#define Batteria_2_PIN A1
```

```
////////////////////////////////////  
/////INCLUDO LE VARIABILI/////
```

```
//ETHERNET SHIELD
```

```
byte mac[ ] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };  
IPAddress ip(192,168,0,120);  
byte subnet[ ] = { 255, 255, 255, 0 };  
EthernetServer server(80);  
EthernetClient client;  
String readString;
```

```
//MOTORI
```

```
Servo MotoreAD, MotoreAS, MotoreCD, MotoreCS, MotorePD, MotorePS;  
char direzione, movimento;  
int x,y, vel[5];  
#define x_max 4  
#define x_min 0  
#define y_max 4  
#define y_min 0
```

```
//TEMPERATURA INTERNA
```

```
#define DHTTYPE DHT22  
DHT dht(DHTPIN, DHTTYPE);  
// Connect pin 1 (on the left) of the sensor to +5V  
// Connect pin 2 of the sensor to whatever your DHTPIN is  
// Connect pin 4 (on the right) of the sensor to GROUND  
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor  
// Initialize DHT sensor for normal 16mhz Arduino  
float h, t;
```

```
//TEMPERATURA ESTERNA
```

```
int lettura;  
float Vx=1.0000, temp=25.0000;  
float num, den;
```

```
//LED
```

```
boolean acceso;
```

```
//BATTERIA
```

```
float E1, V1;  
float E2, V2;  
#define k1 5.638  
#define k2 5.6798
```

```
//H2OhNo!
```

```
int valore_acqua;  
boolean panic;
```



```

////////////////////////////////////
/////PROGRAMMA PRINCIPALE/////
////////////////////////////////////
void setup()
{
  Serial.begin(9600);
  Ethernet.begin(mac, ip, subnet);
  server.begin();
  dht.begin();
  delay(1000);

  pinMode(53,OUTPUT);
  pinMode(SDCARD_CS,OUTPUT);
  digitalWrite(SDCARD_CS,HIGH); //Deselezione la SD
  pinMode(led_pin, OUTPUT);

  //dichiaro i pin dei motori e li posiziono di default fermi
  MotoreAD.attach(MotoreAD_PIN);
  MotoreAD.writeMicroseconds(1500);
  MotoreAS.attach(MotoreAS_PIN);
  MotoreAS.writeMicroseconds(1500);
  MotoreCD.attach(MotoreCD_PIN);
  MotoreCD.writeMicroseconds(1500);
  MotoreCS.attach(MotoreCS_PIN);
  MotoreCS.writeMicroseconds(1500);
  MotorePD.attach(MotorePD_PIN);
  MotorePD.writeMicroseconds(1500);
  MotorePS.attach(MotorePS_PIN);
  MotorePS.writeMicroseconds(1500);
  delay(1500);

  //valori di default delle variabili
  vel[0]=1200;
  vel[1]=1400;
  vel[2]=1500;
  vel[3]=1600;
  vel[4]=1750;
  x=2;
  direzione='S';
  movimento='N';
  acceso=false;
  panic=false;

  Serial.write("endLoop");
  delay(200);
  digitalWrite(led_pin, LOW);
  delay(200);
  digitalWrite(led_pin, HIGH);

```

```
delay(200);  
digitalWrite(led_pin, LOW);
```

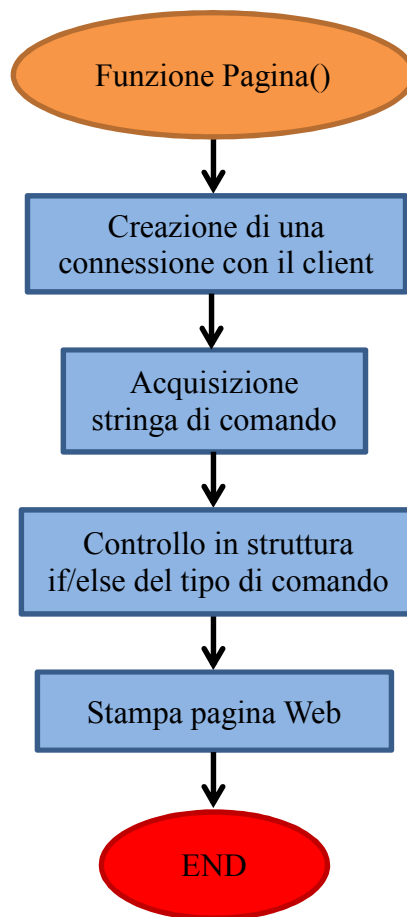
```
//inizio il loop
```

```
void loop()  
{  
  Pagina();  
  Motori();  
  Luci();  
  TempIN();  
  TempOUT();  
  ControllaBatterie();  
  Acqua();  
}
```

VII b. Subroutines

Pagina WEB

Questa funzione si occupa di stampare la pagina web contenente le letture dei sensori di bordo, le immagini dell'IPCam e i comandi del Sea Explorer. Essendo sostanzialmente una pagina di scrittura e di controlli if/else piuttosto intuitivi il Flow-Chart sarà semplificato per cause di dimensioni altrimenti troppo estese.



//FUNZIONE PAGINA WEB

```
void Pagina()
{
    EthernetClient client = server.available();
    if (client) {
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                if (readString.length() < 100)
                    readString = readString + c;
                if (c == '\n')
                {
                    //comandi in base al pulsante premuto
                    if (readString.indexOf("avanti")>0)
                    {
                        direzione='A';
                        if(x<x_max)
                            x++;
                        else if(x==x_max)
                            x=x_max;
                    }
                    else if(readString.indexOf("indietro")>0)
                    {
                        direzione='I';
                        if(x>x_min)
                            x--;
                        else if(x==x_min)
                            x=x_min;
                    }
                    else if(readString.indexOf("stop_1")>0)
                    {
                        direzione='S';
                        x=2;
                    }
                    else if(readString.indexOf("destra")>0)
                    {
                        direzione='R';
                    }
                    else if(readString.indexOf("sinistra")>0)
                    {
                        direzione='L';
                    }
                    else if(readString.indexOf("su")>0)
                    {
                        movimento='U';
                        if(y>y_min)
                            y--;
                        else if(y==y_min)
```

```

y=y_min;
}
else if(readString.indexOf("stop_2")>0)
{
    movimento='N';
    y=2;
}

else if(readString.indexOf("giu")>0)
{
    movimento='D';
    if(y<y_max)
        y++;
    else if(y==y_max)
        y=y_max;
}
else if(readString.indexOf("led")>0)
{
    if(acceso==true)
        acceso=false;
    else if(acceso==false)
        acceso=true;
}

//inizio pagina html
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head>");
client.println("<title>SEA EXPLORER</title>");
//definizione stile dei pulsanti
client.print("<style tipe=\"text/css\">");
client.print(".bottone1 {background: blue; font-weight: bold; font-family:Verdana; border-
radius: 20px; width: 90px; height: 50px; border-width: 2px; border: 4px solid #000000; }");
client.print(".bottone2 {background: red; font-weight: bold; font-family:Verdana; border-
radius: 20px; width: 90px; height: 50px; border-width: 2px; border: 4px solid #000000; }");
client.print(".bottone3 {background: green; font-weight: bold; font-family:Verdana;
border-radius: 20px; width: 90px; height: 50px; border-width: 2px; border: 4px solid #000000; }");
client.print(".bottone4 {background: yellow; font-weight: bold; font-family:Verdana;
border-radius: 20px; width: 90px; height: 50px; border-width: 2px; border: 4px solid #000000; }");
client.print(".bottone5 {background: yellow; font-weight: bold; font-family:Verdana;
border-radius: 8 px; width: 8px; height: 8px; border-width: 1px; border: 2px solid #000000; }");
client.print(".bottone6 {background: black; font-weight: bold; font-family:Verdana; border-
radius: 8 px; width: 8px; height: 8px; border-width: 1px; border: 2px solid #000000; }");
client.print(".bottone7 {background: green; font-weight: bold; font-family:Verdana;

```



```

border-radius: 20 px; width: 90px; height: 30px; border-width: 2px; border: 4px solid
#000000; }");
    client.print(".bottone8 {background: red; font-weight: bold; font-family:Verdana; border-
radius: 20 px; width: 90px; height: 30px; border-width: 2px; border: 4px solid #000000; }");
    client.print("</style>");
    client.print("<script language=\"JavaScript\">");
    client.print("</script>");
    client.print("<body bgcolor=\"grey\">");
    client.print("</head>");
    client.print("<body>");
    client.print("<table cellpadding=\"0\" border=\"0\">");
    client.print("<tr>");
    client.print("<p style=\"text-align: center;\">");
    client.print("<span style=\"font-family:verdana; font-size:40px;\"><strong>SEA EXPLORER
V4.1</strong></span></p>");
    client.print("</tr>");
    client.print("<tr>");
    //inizio tabella layout
    client.print("<table width=\"1352\" height=\"560\" bgcolor= \"lightblue\"
cellpadding=\"5\" border=\"2\">");
    client.print("<tr>");
    client.print("<td align=\"left\" width=\"25%\">");
    //inizio tabella dei dati
    client.print("<table style=\"text-align: center;\">");
    client.print("<tr>");
    client.print("<p style=\"text-align: center;\">");
    client.print("<span style=\"font-family:verdana; font-size:28px;\">DATI</span></p>");
    //sensori
    client.print("<span style=\"font-family:verdana; font-size:22px;\">Sensori</span>");
    client.print("</br>");
    client.print("<span style=\"font-family:verdana; font-size:15px;\">");
    client.print("Temperatura interna: ");
    client.print(t);
    client.print("C");
    client.print("</br>");
    client.print("</br>");
    client.print("Umidita' interna: ");
    client.print(h);
    client.print("%");
    client.print("</br>");
    client.print("</br>");
    client.print("Temperatura esterna: ");
    client.print(temp);
    client.print("C");
    client.print("</br>");
    client.print("</br>");
    client.print("</span>");
    client.print("</tr>");

```

```

client.print("<tr>");
client.print("</br>");
client.print("</tr>");
client.print("<span style=\"font-family:verdana; font-size:22px;\">Indicatori</span>");
client.print("</br>");
//direzioni
client.print("<tr>");
client.print("<span style=\"font-family:verdana; font-size:15px;\">");
client.print("Direzione: ");
client.print(direzione);
client.print("</br>");
client.print("Movimento: ");
client.print(movimento);
client.print("</br>");
//stato
client.print("Luci ");
if(acceso==true)
client.print("<input class=\"bottone5\"/> </p>");
else if(acceso==false)
client.print("<input class=\"bottone6\"/> </p>");
client.print("</br>");
client.print("<span style=\"font-family:verdana; font-size:22px;\">Stato batterie</span>");
client.print("</br>");
client.print("Batteria1 ");
client.print(E1);
client.print("V");
client.print("</br>");
client.print("Batteria2 ");
client.print(E2);
client.print("V");
client.print("</br>");
if(E1<=9.80||E2<=9.80) //Funzione per il controllo della carica delle batterie
client.print("<input class=\"bottone8\" name=\"WARNING\" type=\"button\"
value=\"WARNING\"/> </p>");
else
{
client.print("<input class=\"bottone7\" name=\"OK\" type=\"button\"
value=\"OK\"/> </p>");
}
client.print("</br>");
client.print("Allarme Titanic: "); //Funzione per il controllo di acqua a bordo
if(panic==true)
{
client.print("ICEBERG!!");
}
else if(panic==false)
{
client.print("Tutto ok");
}

```

```

}
client.print("</span>");
client.print("</tr>");
client.print("</table>");
//chiudo tabella dei dati
client.print("</td>");
//inizio parte video
client.print("<td style='text-align: center;' width='45%'>");
client.print("<iframe title='IpCam' align='middle' bgcolor='gray' "
frameborder='0' height='480' width='640' id='ipcam' scrolling='no' "
src='http://192.168.0.110:81/videostream.cgi'></iframe></td>");
//fine parte video
client.print("<td width='30%'>");
//inizio tabella comandi
client.print("<p style='text-align: center;'>");
client.print("<span style='font-family:verdana; font-
size:25px;'><strong>COMANDI</strong></span></p>");
client.print("<center>");
client.print("<table border='0'>");
client.print("<tr>");
client.print("<td>");
client.print("</td>");
client.print("<td>");
client.print("<input class='bottone1' onclick='location.href='avanti'' "
name='AVANTI' type='button' value='AVANTI' /></p>");
client.print("</td>");
client.print("<td>");
client.print("</td>");
client.print("<td>");
client.print("<input class='bottone3' onclick='location.href='su'' name='SU' "
type='button' value='SU' /></p>");
client.print("</td>");
client.print("</tr>");
client.print("<tr>");
client.print("<td>");
client.print("<input class='bottone1' onclick='location.href='sinistra'' "
name='SINISTRA' type='button' value='SINISTRA' /></p>");
client.print("</td>");
client.print("<td>");
client.print("<input class='bottone2' onclick='location.href='stop_1'' "
name='STOP_1' type='button' value='STOP_1' /></p>");
client.print("</td>");
client.print("<td>");
client.print("<input class='bottone1' onclick='location.href='destra'' "
name='DESTRA' type='button' value='DESTRA' /></p>");
client.print("</td>");
client.print("<td>");
client.print("<input class='bottone2' onclick='location.href='stop_2''

```

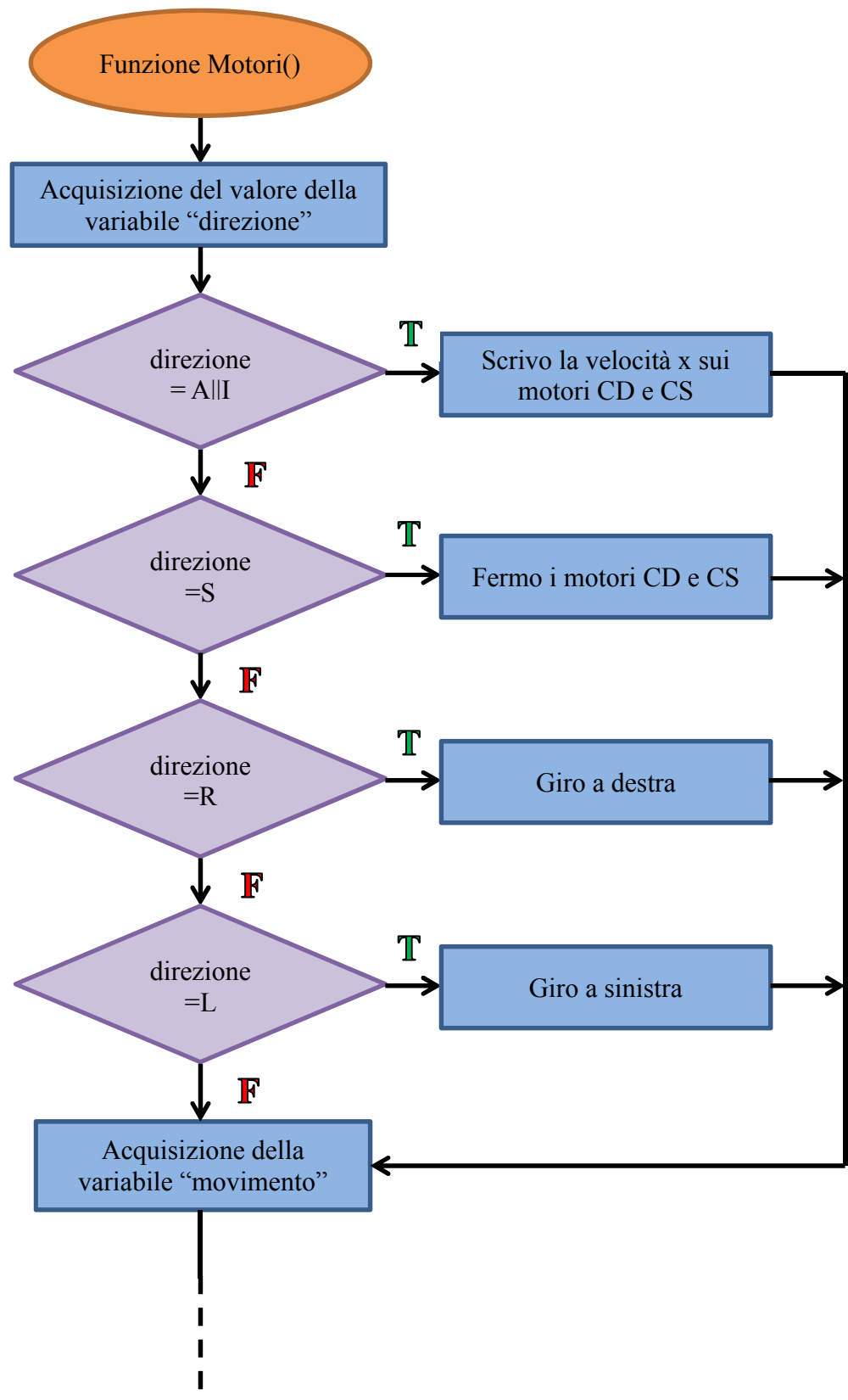
```

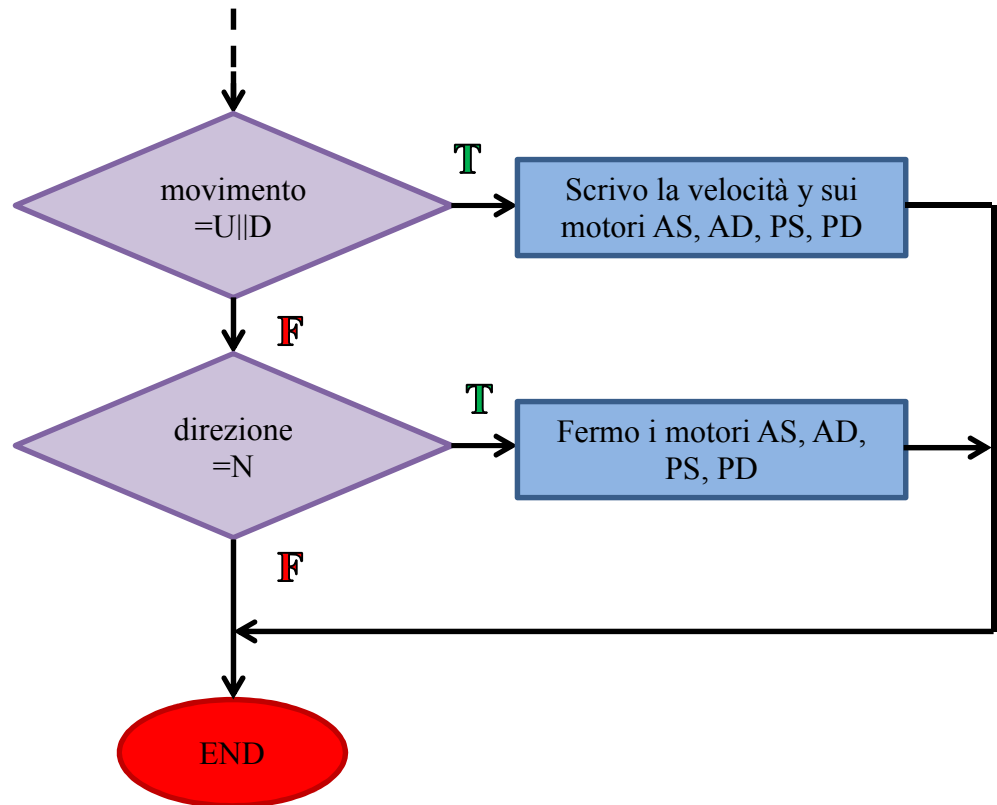
name="\STOP_2\" type="\button\" value="\STOP_2\" /></p>");
    client.print("</td>");
    client.print("</tr>");
    client.print("<tr>");
    client.print("<td>");
    client.print("</td>");
    client.print("<td>");
    client.print("<input class=\"bottone1\" onclick=\"location.href='indietro\"
name=\"INDIETRO\" type=\"button\" value=\"INDIETRO\" /></p>");
    client.print("</td>");
    client.print("<td>");
    client.print("</td>");
    client.print("<td>");
    client.print("<input class=\"bottone3\" onclick=\"location.href='giu\" name=\"GIU\"
type=\"button\" value=\"GIU\" /></p>");
    client.print("</td>");
    client.print("</tr>");
    client.print("</table>");
    //chiudo tabella comandi
    client.print("<input class=\"bottone4\" onclick=\"location.href='led\" name=\"LUCI\"
type=\"button\" value=\"LUCI\" /></p>");
    client.print("</td>");
    client.print("</tr>");
    client.print("</table>");
    //fine tabella layout
    client.print("</tr>");
    client.print("</table>");
    client.print("</center>");
    client.print("</body>");
    client.println("</html>");
    client.stop();
    readString=""; // pulizia stringa per prossima lettura
}
}
}
}
}
}

```

Controllo motori

Questa funzione si occupa di impostare la velocità e il senso di rotazione dei motori per permettere al Sea Explorer di muoversi nelle varie direzioni.





//DIREZIONE MOTORI

```

void Motori()
{
  /*controlla la direzione e la velocità impostate
  premento i tasti sulla pagina web*/
  switch(direzione)
  {
    case 'A' || 'T':
      MotoreCD.writeMicroseconds(vel[x]);
      MotoreCS.writeMicroseconds(vel[x]);
      break;

    case 'S':
      MotoreCD.writeMicroseconds(1500);
      MotoreCS.writeMicroseconds(1500);
      x=2;
      break;

    case 'R':
      MotoreCD.writeMicroseconds(vel[1]);
      MotoreCS.writeMicroseconds(vel[3]);
      break;

    case 'L':
      MotoreCD.writeMicroseconds(vel[3]);
      MotoreCS.writeMicroseconds(vel[1]);
      break;
  }
}
  
```

```

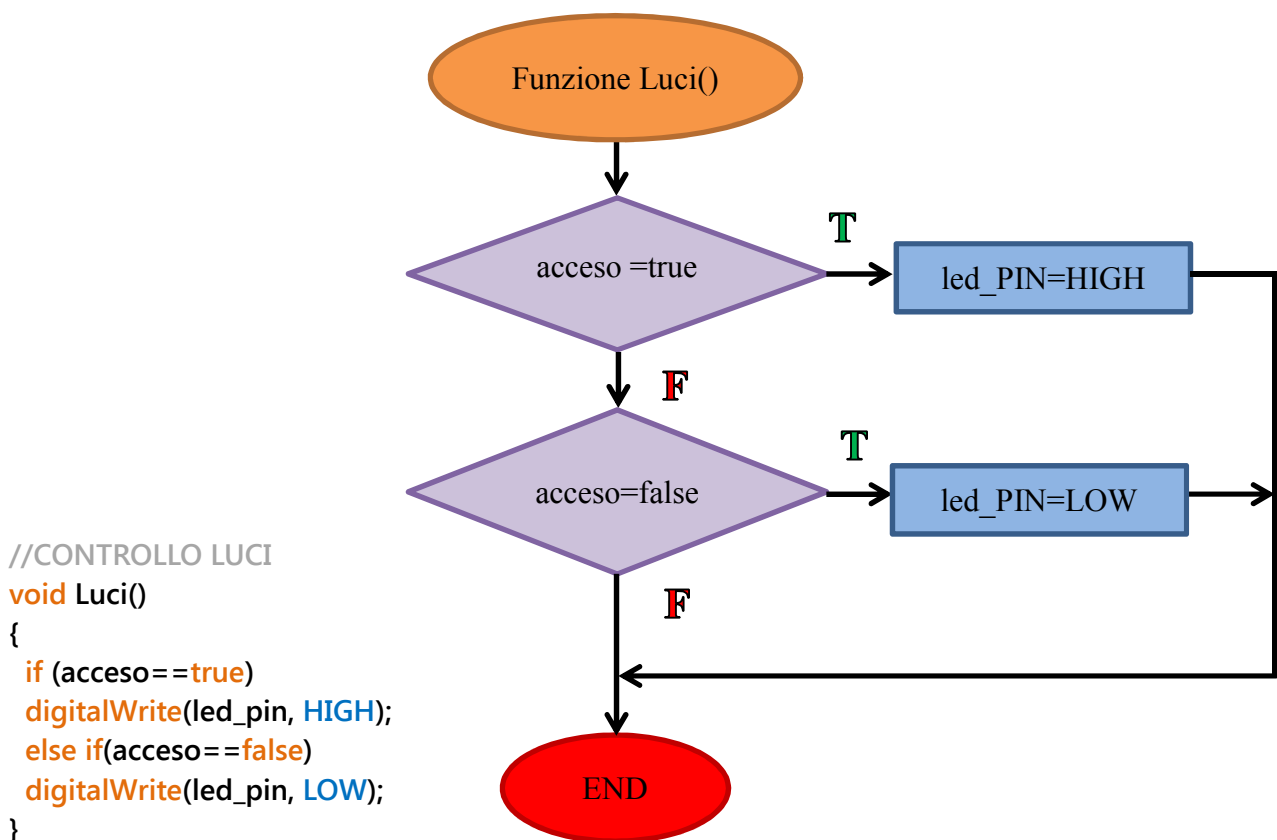
switch(movimento)
{
  case 'U' || 'D':
    MotoreAS.writeMicroseconds(vel[y]);
    MotoreAD.writeMicroseconds(vel[y]);
    MotorePS.writeMicroseconds(vel[y]);
    MotorePD.writeMicroseconds(vel[y]);
    break;

  case 'N':
    MotoreAS.writeMicroseconds(1500);
    MotoreAD.writeMicroseconds(1500);
    MotorePS.writeMicroseconds(1500);
    MotorePD.writeMicroseconds(1500);
    break;
}
}

```

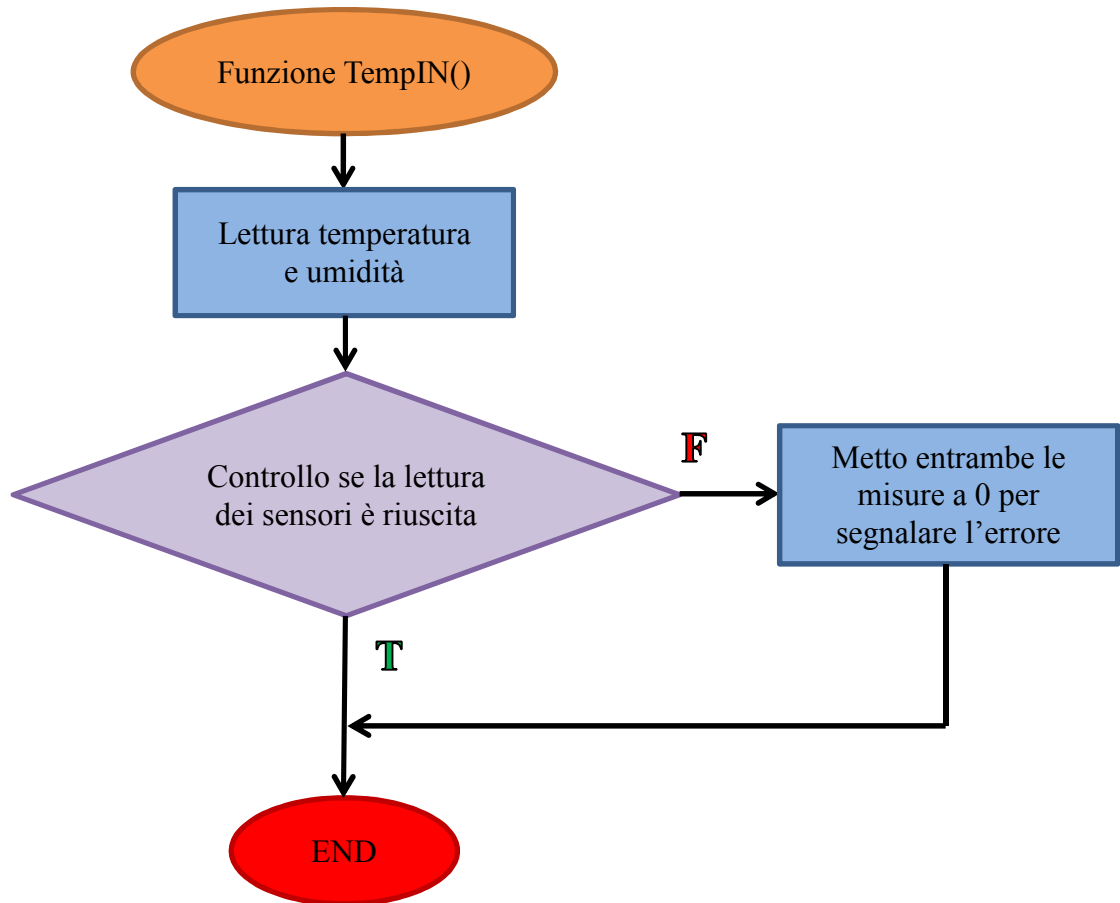
Controllo luci

Questa funzione si occupa di accendere e spegnere i LED in base al comando ricevuto.



Misurazione temperatura interna

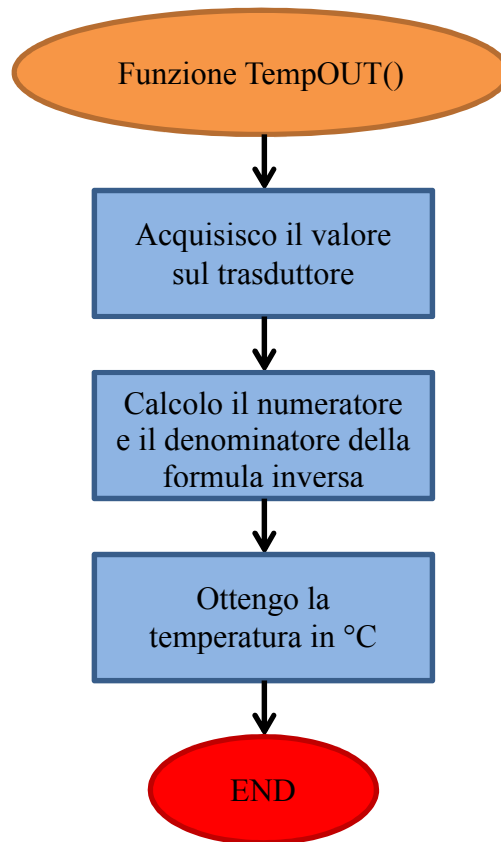
Questa funzione si occupa di rilevare tramite il DHT la temperatura e l'umidità interne al Sea Explorer.



```
//TEMPERATURA E UMIDITA' INTERNE
void TempIN()
{
    //La lettura occupa circa 250 millisecondi
    h = dht.readHumidity();
    t = dht.readTemperature(); //Lettura temperatura in gradi Celsius
    //se le letture falliscono segnalo l'errore
    if (isnan(h) || isnan(t))
    {
        h=0000;
        t=0000;
        return;
    }
}
```

Misurazione temperatura esterna

Questa funzione calcola la temperatura esterna a partire dalla tensione presente sul trasduttore.



//TEMPERATURA ESTERNA

void TempOUT()

{

lettura=**analogRead**(tempOUT_PIN); //lettura dal trasduttore

Vx=(lettura*5.0000)/1024.0000; //trasformo la lettura in V

num=Vx*62.4366-29.4366*5.0000;

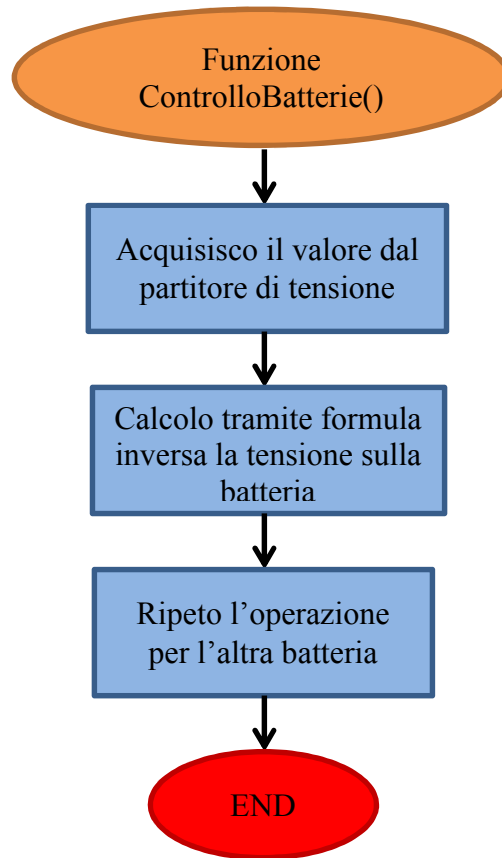
den=5.0000*-0.8005-Vx*-0.8005;

temp=num/den; //formula inversa per la temperatura

}

Controllo carica batterie

Questa funzione monitora la carica delle batterie LiPo, operazione molto importante poiché una scarica eccessiva può portare alla rottura delle batterie.



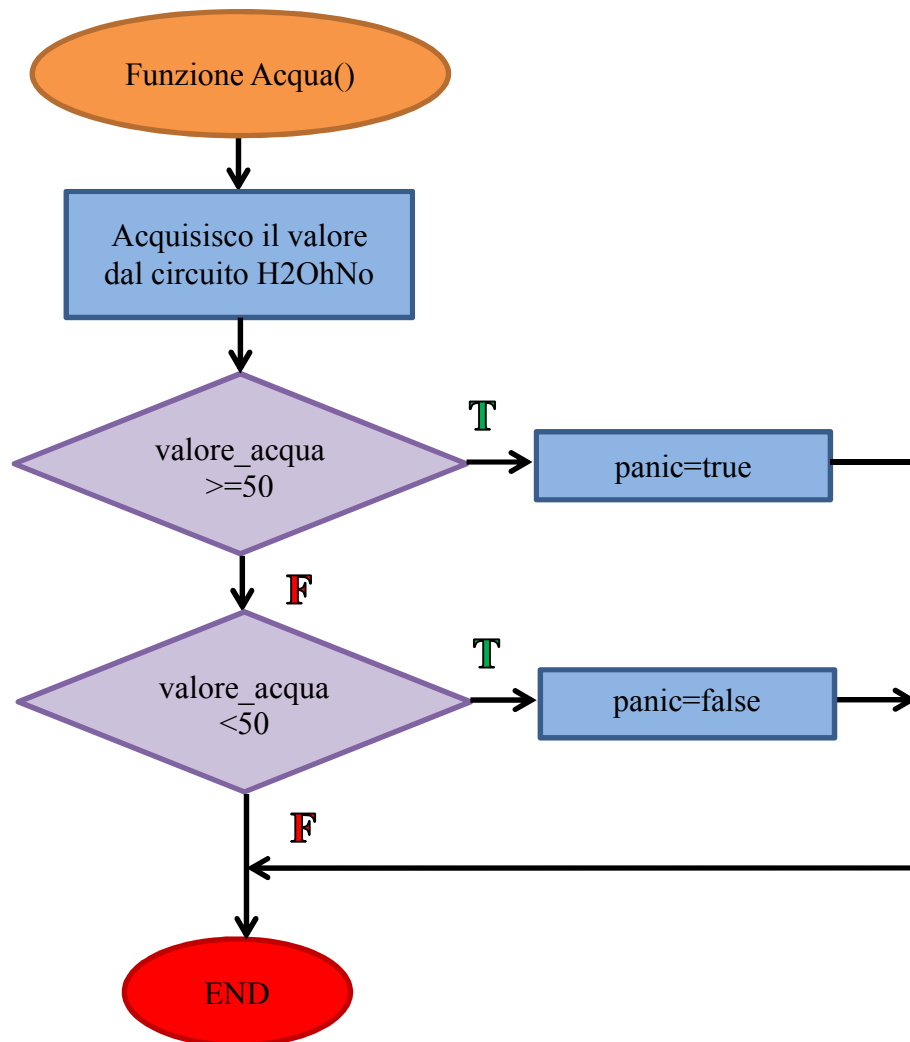
//CARICA BATTERIA

void ControlloBatterie()

```
{  
  V1=(analogRead(Batteria_1_PIN)*5.0)/1024.0; // lettura della tensione sulla resistenza  
  E1=(V1*k1); //moltiplico per (R1+R2)/R2  
  V2=(analogRead(Batteria_2_PIN)*5.0)/1024.0;  
  E2=(V2*k2);  
}
```


Rilevamento infiltrazioni d'acqua

Questa funzione rileva il passaggio di un pin del circuito H2OhNo da 0 a 1.2V circa nel caso venga segnalato un corto-circuito tra i piedini del sensore che servono a monitorare il livello dell'acqua



```
//H2OhNo
void Acqua()
{
  valore_acqua=analogRead(panic_PIN);
  if(valore_acqua>=50)
    panic=true;
  else if(valore_acqua<50)
    panic=false;
}
```

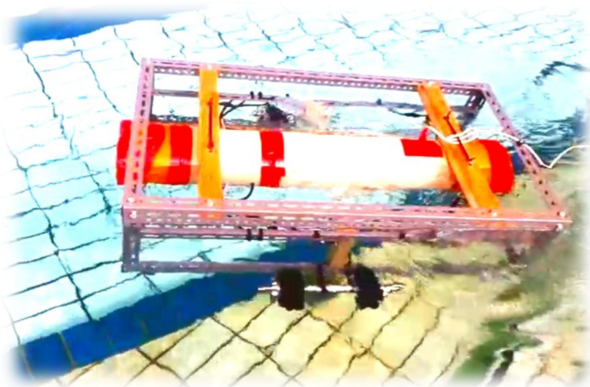
VIII. Considerazioni finali

VIII a. Test in acqua

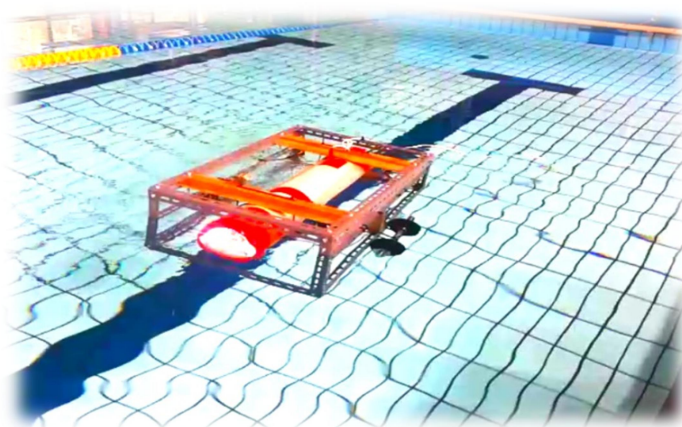
Concluso il lavoro sul Sea Explorer abbiamo provveduto a testarlo in acqua, abbiamo riscontrato come in effetti il ROV funzioni all'incirca come da progetto tranne alcuni problemi che esamineremo in seguito.



Primo test in acqua del progetto con “assistenza”



Test in autonomia



VIII b. Problemi riscontrati e possibili soluzioni

- 1) Si è notato che nonostante il peso dei componenti e dei sostegni metallici il ROV tende a galleggiare, si è pensato di ricorrere a dei pesi per risolvere il problema (momentaneamente manubri da palestra).
- 2) Nonostante i materiali appositi è ancora presente una piccola infiltrazione di acqua a causa della poca tenuta della guarnizione e del filetto presenti sul tappo anteriore. Una possibile soluzione è l'impiego di grasso al silicone per aiutare la tenuta stagna.
- 3) A causa dell'elevata potenza da dissipare (circa 1W) sui regolatori dell'Arduino MEGA e del poco spazio di areazione si crea un surriscaldamento del componente che crea problemi nel normale svolgimento del programma. Per ovviare al problema è stato ordinato un regolatore lineare DB7808 in modo da dimezzare la potenza dissipata da Arduino.
- 4) È presente un riflesso causato dai LED IR della telecamera, per ovviare al problema è sufficiente disabilitarli da software
- 5) Si è riscontrata una poca comodità nell'accendere/spegnere il Sea Explorer poiché al momento è possibile farlo solo collegando e scollegando manualmente le batterie (interne) .Si è quindi deciso di ordinare due interruttori impermeabili "T7111A5" da posizionare all'esterno del Sea Explorer.

WARNING: visto il procedere dei lavori al momento della scrittura della tesina è possibile che alcune parti di progettazione si rivelino incorrette e/o inesatte al momento dell'esposizione orale.