



Game INC.

Relazione Elaborato 2020/2021

Sezione C

Partecipante Marco Di Veroli.

Il progetto illustrato in seguito tratta di un pappagallo giocattolo, le cui funzioni sono di ricercare una persona adiacente e instaurarci una semplice conversazione.

La conversazione si basa sull'ascoltare l'oratore e in seguito ripetere per tre volte ciò che ha detto.

La ricerca adiacente si svolge in un campo d'area di 140 gradi fronte al giocattolo a una distanza massima di 3 metri.

Rilevata la persona, il pappagallo svolge una funzione dipendente dalla distanza pappagallo-uomo.

La realizzazione si suddivide in tre progettazioni: Logica, hardware e software

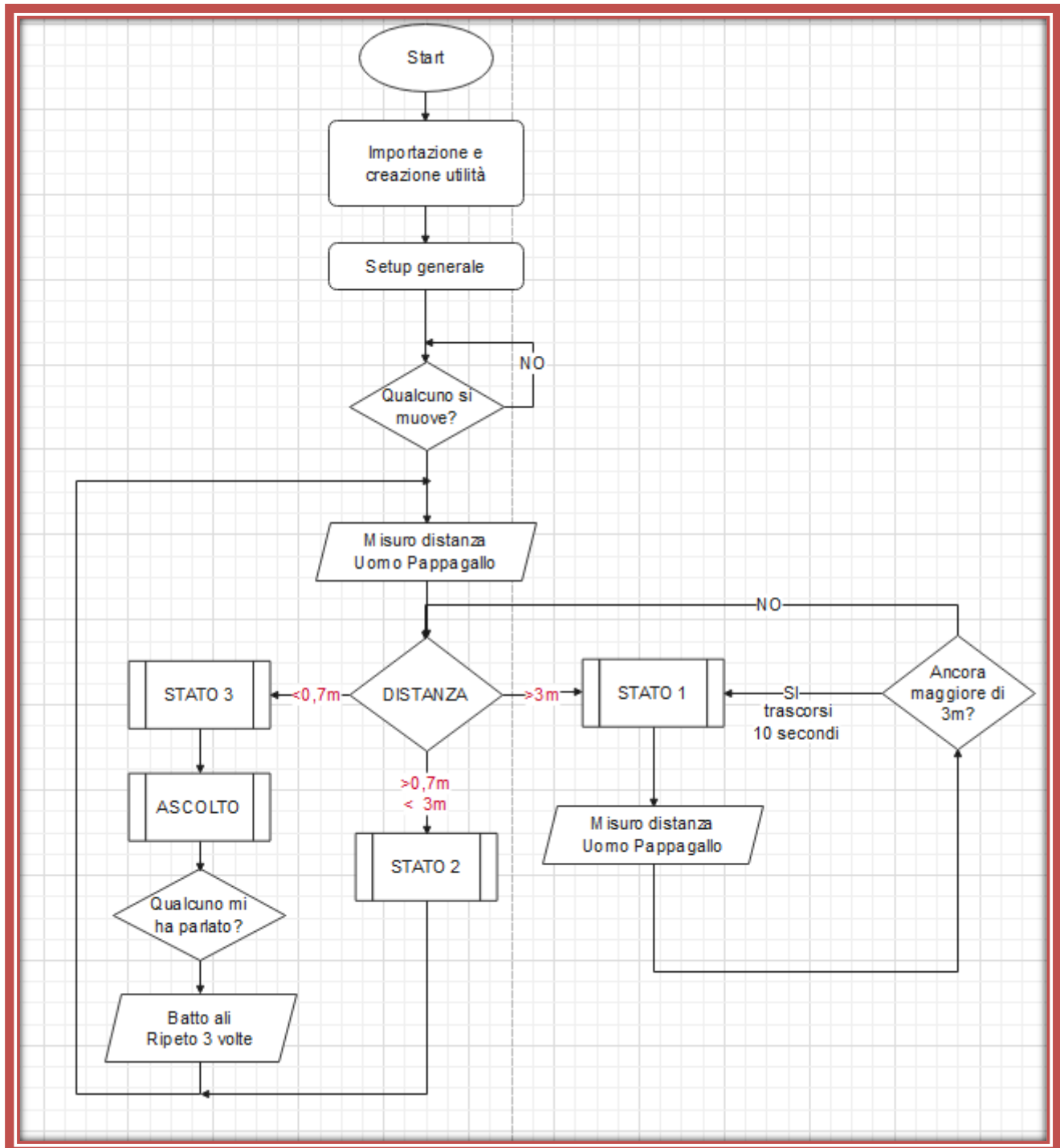
Nella prima parte sono spiegati i compiti da eseguire del pappagallo, ovvero la funzione dettata prima, la spiegazione avviene tramite dei flow chart (personalizzati) che restituiscano il funzionamento.

Nella seconda parte vi ritroviamo la progettazione hardware quindi l'elettronica dell'elaborato, suddivisa in due schemi principali di alimentazione e di segnali.

Nell'ultima fase dell'elaborato si mostrerà il codice sorgente, in linea di massima (con commenti annessi), il restante codice starà in un link delle referenze, contenente le librerie per far funzionare il pappagallo.

Gli schemi concettuali sono esposti in due prospettive diverse, flow chart e diagramma a stati.

Schematizzazione flow chart:



Blocchi di funzione:

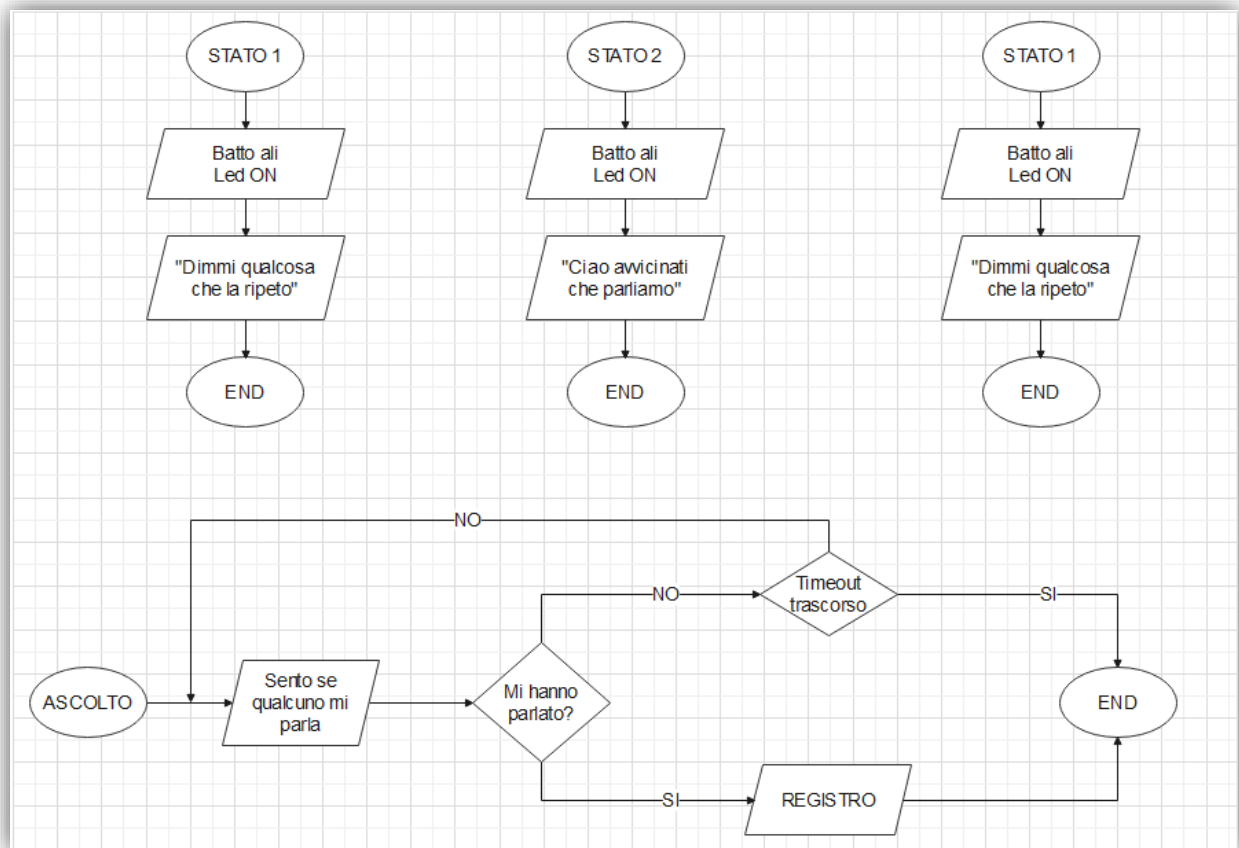
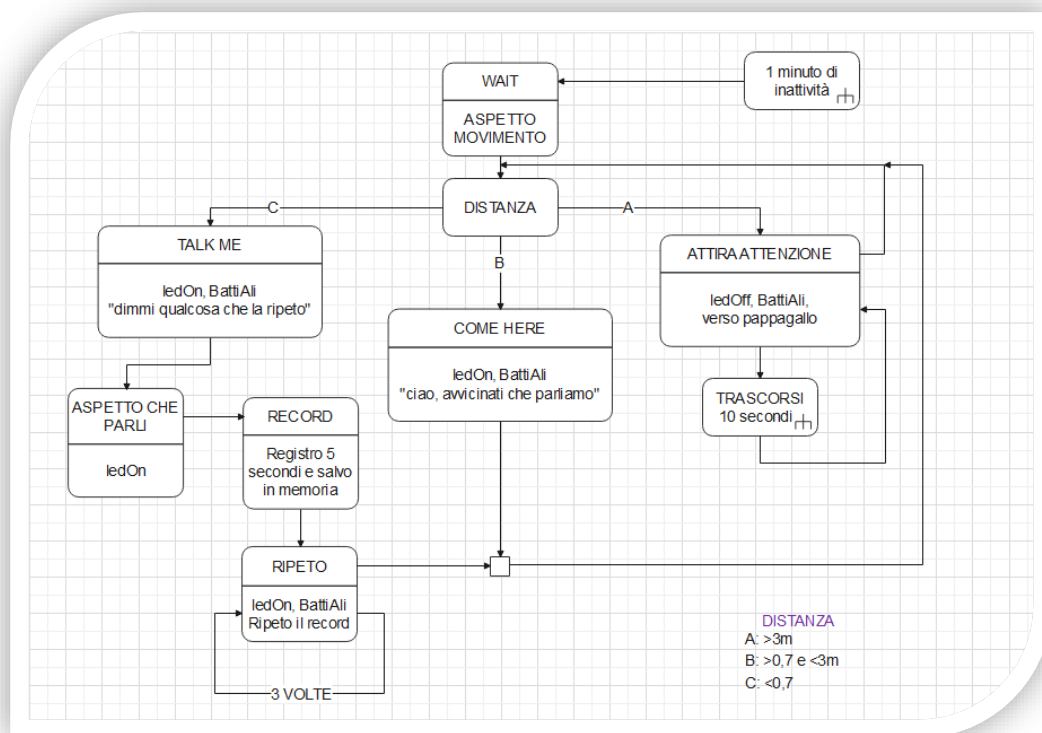


Diagramma a stati finiti:



La progettazione hardware:

Come detto prima essa prevede una parte di segnali e una di alimentazione.

La parte di alimentazione è collegata alla presa di casa e vi si generano due tensioni, utilizzate in seguito per il funzionamento dei sensori e degli attuatori.

Nel progetto si propongono due soluzioni diverse per il movimento delle ali del pappagallo, esso si può svolgere usando un motorino servo o un motorino DC, il motorino DC prevede un sistema meccanico aggiuntivo.

La testa del pappagallo dovrà essere mobile e controllata tramite un servo per ricercare le persone nella stanza, trovata una persona, ovvero aver misurato una distanza pappagallo-uomo il servo vi si sofferma.

Premettendo che l'attenzione della persona venga catturata dal pappagallo, si presuppone che essa si muove in linea retta verso il giocattolo.

Allorché il pappagallo funziona correttamente, avviando la sua routine di conversazione.

Se l'attenzione non dovesse essere stata intercettata e la persona esce dal campo visivo del pappagallo, il pappagallo si rimette a cercare intorno a sé.

La lista dei componenti è la seguente:

Elenco prima i dispositivi alimentati con tensione 5V:

Un microcontrollore, ESP32, ci permette di controllare i 32 pins in diverse modalità e ci restituisce una tensione di 3.3V in uscita, utilizzata in seguito.

Un sensore di movimento PIR dal range di 140° con una profondità massima di 3mt.

Un sensore di distanza ultrasuoni con portata 4mt.

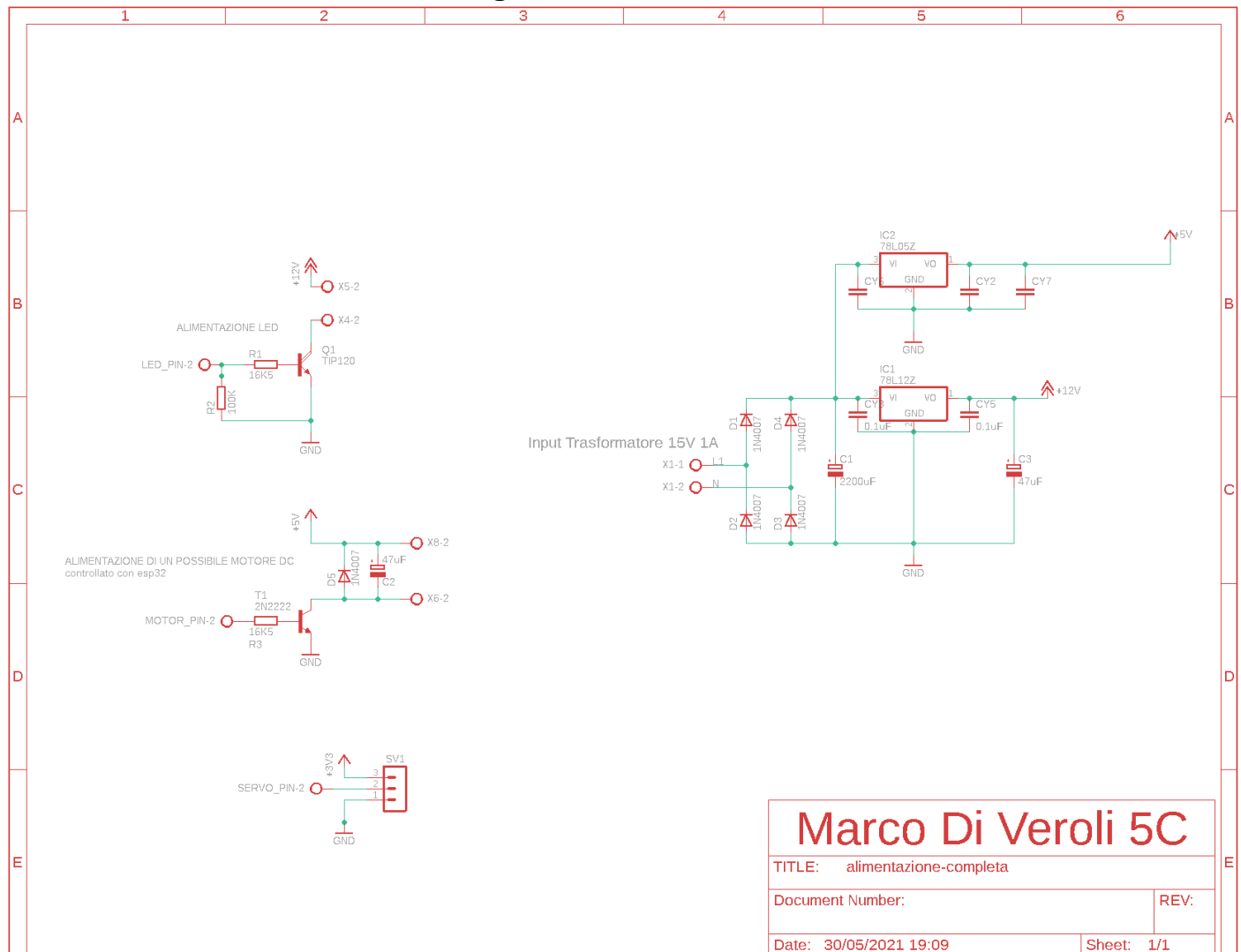
Un sistema di amplificazione e gestione segnali I2S con cui il MCU riprodurrà l'audio del pappagallo, con una potenza di 3W deve essere connesso a uno speaker da 4-8ohm.

Le alimentazioni di 3.3V e 12V sono applicate ai seguenti circuiti.

3.3V (ripulito da eventuali disturbi di linea usando un LDO "**lm2937**") per l'alimentazione di un microfono annesso a un circuito di condizionamento, denominato MAX9841.

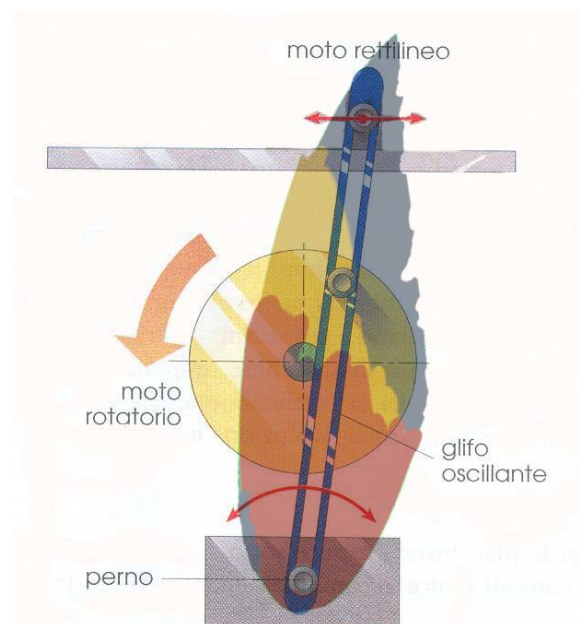
3.3V alimentazione motori Servo.

Schema di alimentazione e collegamento motori



Come già detto per le ali vi si può usare un motore DC con un meccanismo oppure un servo, nello schema di alimentazione e motori vi sono i circuiti di gestione, dei diversi motori e per la striscia LED, usando l'MCU.

Di seguito trovate la figura di come dovrebbe essere questo meccanismo, per semplificare il codice piuttosto di usare un segnale pwm per regolare la velocità del motorino DC ho preferito usare un trimmer



L'ultima parte la più completa sotto ogni aspetto, piena di funzioni è il programma vero e proprio.

Pappagallo 2020/2021:

```
#include <ESP32_Servo.h>

#include "Pappagallo.h"

#define FILESYSTEM SPIFFS

#define ALI 4

#define HEAD 2


#define LED 15

#define TRIG 5

#define PIR 14

#define ECHO 13


#define I2S_DOUT 25

#define I2S_BCLK 27 // I2S

#define I2S_LRC 26


Pappagallo polido(HEAD, ALI, "Poldo");


volatile int motion= 10;


int timer, timer_vieni;


void getMotion(){
    motion= 10;
}

/* SETUP CODE */

void setup() {
    FILESYSTEM.begin(true);
```

```

Serial.begin(115200);

pinMode(TRIG, OUTPUT);
pinMode(ECHO, INPUT);
pinMode(PIR, INPUT);
pinMode(RESET, INPUT);

poldo.setDistanceSensor(TRIG, ECHO);
poldo.setReproducer(I2S_BCLK, I2S_LRC, I2S_DOUT, 2);
poldo.setMic(ADC1_CHANNEL_5);
poldo.setLight(2);
poldo.Dormi();

attachInterrupt(digitalPinToInterrupt(PIR), getMotion, RISING);

timer_vieni= -5;
}

```

/* LOOP CODE */

```

void loop() {
    unsigned long timer_normalize= millis();
    int secondi= (int)millis()/1000;
    int human_distance; //Distanza pappagallo - uomo

    if (motion) { //Controllo se ci sono stati movimenti
        human_distance= poldo.check_humans_near(); //Mi cerco intorno umani vicino a me
        if (human_distance==0){ //Se non ho trovato nessuno mi disattivo

```



```

poldo.Dormi();

if (secondi-timer>=10){
    timer= secondi;
    poldo.Batti_ali();
    poldo.Reproduce("Verso.wav");

    //poldo.Batti_ali(); with DC motor uncomment
    motion-= 2; //Diminuisco attività nella stanza
}
}

else{ //Altrimenti controllo a che distanza sta l'umano

    motion= 10;
    timer_normalize= millis();
    poldo.Sveglia();

    if (human_distance<70){ //Se è vicino almeno 70cm gli chiedo di parlarmi e io ascolto
        poldo.Batti_ali();
        poldo.Reproduce("Talk_me.wav");//Parlami che ripeto audio
        //poldo.Batti_ali(); with DC motor uncomment
        detachInterrupt(digitalPinToInterrupt(PIR));
        //disabilito interrupt per evitare conflitti nella funzione record
        poldo.wait_noise(9, 3); //Aspetto che qualcuno mi parli, impostando un timeout

//La variabile anyone_talk restituisce true se wait noise ha sentito qualcosa nel campionamento
    if (poldo.anyone_talk()){
        poldo.Record(5);

        //Ascolto per 5 secondi chi mi sta parlando, con campione precedente
        for (int i= 0; i<3; i++){
            poldo.Batti_ali();
            poldo.Ripeti();//Ripeto ciò che mi è stato detto
            //poldo.Batti_ali(); with DC motor uncomment
        }
    }

    attachInterrupt(digitalPinToInterrupt(PIR), getMotion, RISING);
}

else{

```

```
    if (secondi-timer_vieni>5){  
        poldo.Batti_ali();  
        poldo.Reproduce("Come_here.wav");  
        timer_vieni= secondi;  
    }  
}  
timer= secondi;  
}  
}  
int delta= 1000-((millis()-timer_normalize)%1000);//milliseconds to normalize time  
delay(delta);//Normalize time, loop recycle every second  
}
```