**Progetto Greenhouse:**

Arduino svolge la funzione di controllore (e attuatore) della variazione dell’umidità all’interno di una serra. Il sistema possiederà due stati ,Manule(ManualState) e Automatico(Automatic), ma in entrambi arduino svolgerà solo il compito di fornire i dati al server java e svolger le operazioni che il server li comunicherà.

Per svolgere tale compito abbiamo sviluppato due classi:

class Automatic:public Task{

public:

Led\* ledAuto;

Led\* ledPump;

Sonar\* proxy;

CanaleCom\* canale;

PumpServo\* Pump;

Automatic(Led\* ledAuto, Led\* ledPump, Sonar\* proxy, CanaleCom\* canale, PumpServo\* Pump);

void init(int period);

void tick();

};

class ManualState:public Task{

public:

Led\* ledManual;

Led\* ledPump;

Sonar\* proxy;

CanaleCom\* canale;

PumpServo\* Pump;

ManualState(Led\* ledManual, Led\* ledPump, Sonar\* proxy, CanaleCom\* canale, PumpServo\* Pump);

void init(int period);

void tick();

};

Automatic esegue un controllo sulla distanza (che viene calcolata dal sonar qui chiamato proxy) e sull’arrivo di un messaggio inviato tramite bluetooth, se entrambe le condizioni sono soddisfatte verrà inviato “b” come segnale di richiesta al server per passare in modalità manuale. Successivamente controlla il valore relativo alla apertura della valvola e della sua portata:

* 4 = chiusa
* 3 = portata minima
* 2 = portata media
* 1 = portata massima

ManualState dopo l’accensione del ledManual riceve e invia al server il messaggio del dispositivo bluetooth poi esegue un controllo sulla distanza e sul messaggio in arrivo, nel caso in cui la distanza sia maggiore di quella limite o vengo ricevuto il valore “e” il sistema segnalerà al server di passare in modalità automatica inviando “a”. In seguito, controllerà la presenza di un messaggio proveniente dal server il quale, se presente, indicherà in modo preciso l’apertura della valvola tramite un apposito valore che verrà salvato sulla variabile “apertura”.

Ad occuparsi della comunicazione tra più dispositivi è la classe CanaleCom (non ver finale).

class CanaleCom {

public:

CanaleCom();

void sendUmidityBT(int val);

void sendState(int val);

void sendMessaggio(String val);

int getAndSendMsgBT();

String getMsg();

String getMsgBT();

int getValPump();

int getStatePump();

};

CanaleCom si occupa di instradare i messaggi dal bluetooth al server nello stato manuale (con il metodo: int getAndSendMsgBT(); ) dell’invio del valore attuale e dello stato dell’apertura della pompa ( void sendState(int val); void sendMessaggio(String val); ) e della ricezione dei messaggi dal server.