**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 i compiti di simulare l'apertura di una pompa, simulata da un servo motore, e invire al server eventuali messaggi provenienti da bluethoot.

Per svolgere tale compito abbiamo sviluppato due Task:

Automatic La classe è il task che descrive il comportamento del sistema in modalità automatica (che è lo stato iniziale).  
All'interno del task effettuiamo per prima cosa un controllo sul canale di comunicazione bluethoot ove se il messaggio è in arrivo e non vuoto, controllandone il contenuto (che deve essere uguale a "B") e la distanza percepita dal sonar(che dovrà essere minore di DIST), se il controllo dovesse andare a buon fine verrà inviata al Server java la richiesta di passare in modalità automatica.  
Successivamente l'applicazione risponderà ai messaggi in arrivo ogni secondo da Esp tramite seriale (inviati dal server java), i quali verranno sottoposti alla manipolazione delle stringhe al loro interno (subString), per separare l'intestazione dal valore simulato dell'umidità, questultima verrà reindirizzata sul canale bluethoot nel caso qualcuno fosse connesso, mentre l'intestazione verrà usata da una funzione switch per attuare l'appropriato movimento del servo o passare allo stato manuale, a seguire la legenda:

* m : il messaggio segnala ad arduino di passare in modalità manuale, definito da HMANUAL.
* f : segnala ad arduino di chiudere la pompa (settare l'angolo del servo a 0), definito da HPCLOSE.
* c, d, e : segnalano i vari gradi di apertura della pompa rispettivamente portata minima media e massima, definiti da HPmin, HPmed e HPmax.

ManualState In questo task ogni messaggio proveniente da bluethoot viene riderizionato al server, ed effettuiamo un controllo sulla distanza percepità dal sonar che in caso sia maggiore di DIST notificherà al sistema la necessità di passare in modalità manuale. Successivamente come nel task Automatic procede alla manipolazione della stringa del messaggio in arrivo da seriale separando le intestazioni dai dati veri e proprio, seguendo le specifiche delle intestazioni i dati verranno utilizzati come convenuto dalla seguente legenda:

* p : i dati si riferiscono all'apertura della pompa (posizione del servo), definito da HPUMPSERVO.
* r : i dati verranno reindirizzati al server, definito da HTRACE.
* a : segnala ad arduino di passare in automatico, definito da HAUTO.

Una classe importante è SharedState grazie alla quale possiamo gestire il passaggio da uno stato ad un'altro senza incontrare errori, utilizzando due variabli booleane automatic e manual per cambiare stato del sistema.

ServoPump la classe presenta il costruttore dove viene richiamato solo il metodo attach per collegare il pin analogico (PUMP), e due metodi uno per la chiusura della pompa simulata (closePump) e uno per il settaggio dell'angolo del Servo. NOTA IMPORTANTE: Per il progetto è stata utilizzata la libreria ServoTimer2 onde evitare conflitti con il timer, il che porta alla ridefinizione del movimento del servo passando da 0 a 180 gradi a 750(0) e 2250(180).

Led la classe ci permette di notificare visivamente sulla breadboard lo stato del sistema dove il led collegato al pin specificato dalla variabile "LEDa" segnala che il sistema si trova in modalità automatica, il led "LEDm" segnala la modalità manuale, e il led "LEDp" simulando la quantità di litri al minuto modificando la brillantezza.

Sonar il sonar (ECHO, TRIG) viene utilizzato per prendere la distanza dell'utilizzatore del dispositivo bluethoot dalla breadboard, cosi da simulare un'allontamento eccessivo o per delimitare una zona di utilizzo del sistema.

define.h il documento che contiene tutte costanti, valori attesi dei messaggi e definizioni dei pin utilizzati da arduino.