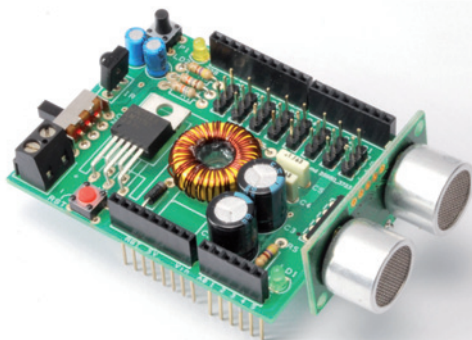


SHIELD DI CONTROLLO PER ROBOT ARDUSPIDERIN, ARDUBIPE E ARDUFILIPPO

(cod. ROBOT_SHIELD)

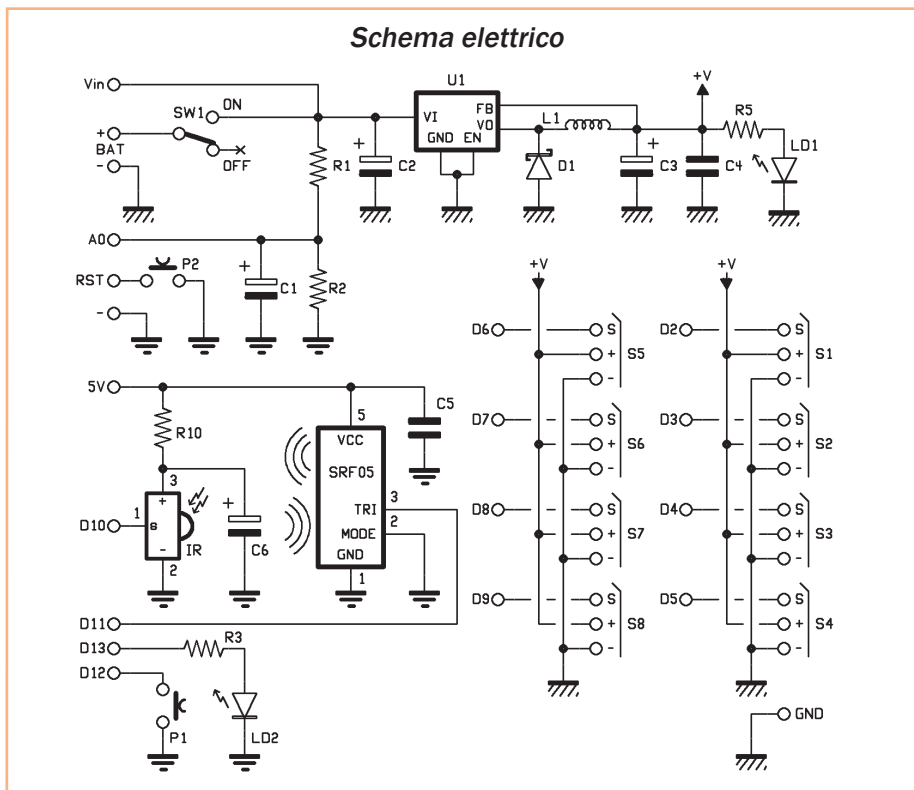
Shield per Arduino in grado di controllare fino a 8 servo, dotata di un proprio alimentatore interno (alimenta anche Arduino), con funzione di monitoraggio della tensione della batteria, controllo a distanza tramite un telecomando IR (un normalissimo telecomando di quelli usati per i televisori o video-registratori) e un sonar ad ultrasuoni che permette di riconoscere gli ostacoli. Sulla shield è disponibile il pulsante per resettare Arduino, un pulsante per uso generico ed un LED connesso al D13 di Arduino. Il comando per i servo avviene mediante i PWM che Arduino fornisce tramite le proprie linee D2+D9. La shield è compatibile con tutte le versioni originali di Arduino (duemilanove o UNO) ed anche con i cloni (Seeeduino). Questa shield è stata realizzata per gestire tre differenti robot come ARDUFILIPPO, ARDUSPIDERIN e ARDUBIPE (acquistabili separatamente). Arduino è in grado, tramite questa shield che comanda i servo ed un firmware ad hoc, di gestire ognuno dei robot, in quanto tutti si muovono grazie a servocomandi e rilevano gli ostacoli mediante un sonar. La scheda può essere alimentata con un unico pacco batteria da 6÷12 volt (due celle LiPo oppure 6÷8 celle NiCd o NiMH). Dimensioni 68 x 55 mm.



Realizzazione pratica

Lo shield è di facile realizzazione. Facendo riferimento al piano di montaggio e utilizzando un saldatore da non più di 30W, iniziate a saldare i componenti a basso profilo (resistenze, diodi, pulsanti, interruttore, LED...), quindi procedete con il montaggio dei condensatori (prestare attenzione alla polarità degli elettrolitici), della bobina e del regolatore, notando che questi ultimi due componenti devono avere il proprio corpo posizionato orizzontalmente sul circuito stampato (nota: per l'integrato LM2576-5, lo stampato prevede un piccolo dissipatore ma, se avrete l'accortezza di saldarlo in modo che rimangano alcuni millimetri tra esso e lo stampato, potete ometterlo). Sistemate ora i pin strip maschio a 3 poli in corrispondenza delle piazzole siglate S1, S2,.....S8, i pin strip maschio/femmina (a

6, 8 e 10 poli) tenendo rivolto verso l'alto il lato femmina. Noterete che introducendo i connettori femmina coi terminali lunghi nei rispettivi fori dello stampato dello shield, i terminali stessi faranno da pin-strip e permetteranno allo shield stesso di introdursi nei connettori del modulo Arduino. Terminate il circuito montando sulla scheda il sensore IR, il morsetto a 2 poli e lo strip femmina a 5 vie (mantenendo il suo corpo aderente) per il collegamento del sonar SRF05 sul quale deve essere necessariamente saldato un pin-strip maschio a passo 2,54 mm. Completate le saldature, verificate che non vi siano falsi contatti o cortocircuiti. Fatto ciò, potete fare un primo collaudo alimentando la scheda tramite il pacco batterie e verificando che si accendano il LED LD1 nonché il LED di alimentazione sulla scheda Arduino.



A questo punto la vostra piattaforma hardware è pronta per essere inserita nella meccanica del robot; per aiutarvi nel compito, riportiamo di seguito delle tabelle relative ai collegamenti dei servo con i vari robot. **Lo sketch per l'utilizzo con i vari robot è scaricabile gratuitamente dalla scheda del prodotto ROBOT_SHIELD disponibile su www.futurashop.it.**

Scegliere l'alimentazione

lo shield dispone di un proprio alimentatore interno in grado di fornire ai servocomandi una tensione stabilizzata ed

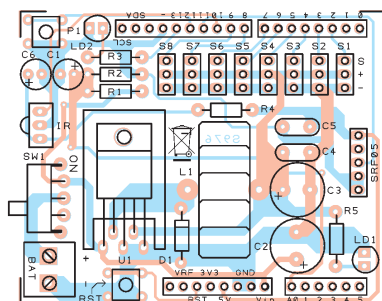
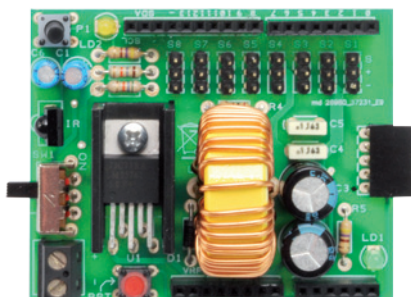
anche l'alimentazione alla scheda Arduino. Il dispositivo è alimentabile con un unico pacco batteria da 6+12 volt ma di tipo ricaricabile, in quanto (considerato l'elevato assorbimento di corrente) le normali pile a secco non durerebbero molto; vanno bene quattro elementi NiMH in serie, i quali a piena carica forniscono 1,5x4=6 V. Man mano che si scaricano, gli elementi scendono a 4 V: non siamo nelle condizioni ottimali di funzionamento del servo e, considerata la notevole variazione di tensione possiamo ipotizzare anche notevoli dif-

ferenze nelle prestazioni, ma se non scarichiamo le batterie del tutto possiamo farcela. Come alternativa restano gli elementi Li-Ion o ai polimeri di litio, che sono più performanti, più piccoli e leggeri (a parità di capacità) rispetto alle Ni-Cd e Ni-Mh (le LiPo possono erogare correnti molto elevate); insomma, ottimali per il controllo di motori. Purtroppo ogni singola cella fornisce una tensione nominale di 3,7 volt e quindi, mettendone due in serie, si ottengono 7,4V, che diventano 8,4 a piena carica: decisamente troppi per un servo normale.

Piano di montaggio

Elenco Componenti:

- R1: 56 kohm
- R2: 18 kohm
- R3, R5: 470 ohm
- R4: 100 ohm
- C1: 10 μ F 63 VL elettrolitico
- C2: 470 μ F 25 VL elettrolitico
- C3: 1000 μ F 16 VL elettrolitico
- C4, C5: 100 nF multistrato
- C6: 10 μ F 63 VL elettrolitico
- LD1: LED 3 mm rosso
- LD2: LED 3 mm verde
- U1: LM2576-5
- SW1: Deviatore a slitta
- P1: Microswitch
- RST: Microswitch
- IR: IR38DM
- L1: 100 μ H 5A
- D1: 1N5819
- SRF05: Modulo SRF05



Varie:

- Morsetto 2 poli
- Strip maschio 10 poli
- Vite 12 mm 3 MA
- Strip maschio 3 poli (8 pz.)
- Strip maschio 5 poli 90°
- Dado 3 MA
- Strip maschio 6 poli
- Strip femmina 5 poli
- Circuito stampato
- Strip maschio 8 poli (2 pz.)
- Dissipatore TO-220

Utilizzo con Robot Filippo

I servo vanno cablati seguendo le indicazioni della Tabella 1. Per l'alimentazione è previsto vengano usate 6 o 8 batterie ricaricabili NiMH o NiCd formato stilo, che vanno inserite in due distinti portabatterie e collegate in serie. Se optate per l'utilizzo di 6 celle dovrete usare due portabatterie a tre vani, mentre se usate otto celle dovrete usare due portabatterie da 4 vani. I por-

tabatterie vanno posizionati uno a destra ed uno a sinistra dei due servo, collegandoli in serie con due clip per batterie da 9 volt. Consigliamo di programmare Arduino prima di collegare i servo, onde evitare che un eventuale programma precedentemente in memoria fornisca segnali che possono portare a movimenti scomposti. Ricordatevi che se alimentate Arduino solo tramite la USB, per programmarla,

parte della tensione finirà sullo shield, quindi consigliamo di collegare il cavo USB solo quando è presente anche l'alimentazione principale sui punti BAT; diversamente, potrebbero esserci problemi con i servo. Programmate Arduino con lo sketch **filippo.pde** (scaricabile gratuitamente dalla scheda del prodotto ROBOT_SHIELD disponibile su www.futurashop.it.) ed avviate Serial Monitor, allorché

vi ritroverete una schermata con una serie di dati; quelli che vi interessano sono la tensione della batteria e la posizione di neutro dei servo. Inviare il comando “obs” per avere la lettura del sensore SRF05 ed il comando “lev” per verificare il livello della

batteria. Assicuratevi che la tensione letta corrisponda al reale valore delle batterie; per accertare ciò è sufficiente confrontare i dati con una misura effettuata con un buon tester. Concentratevi ora sull'impostazione del neutro dei servo, indispensabile in

quanto il robot, per funzionare correttamente, deve essere perfettamente centrato, cosa non sempre possibile a livello meccanico, in quanto i braccetti dei servo possono avere solo posizioni predefinite. Per centrare il servo che svolge la funzione di inclinare il ro-

Tabella 1 - Collegamento dei servo con robot Filippo

Servo	Pin Arduino	Connettore shield	Funzione
Servo 0	2	S1	Inclinazione (servo anteriore)
Servo 1	3	S2	Passo (servo in basso)

Tabella 2 - Collegamento dei servo con robot Bipe

Servo	Pin Arduino	Connettore shield	Funzione
Servo 0	2	S1	anca sinistra
Servo 1	3	S2	ginocchio sinistro
Servo 2	4	S3	piede sinistro
Servo 3	5	S4	anca destra
Servo 4	6	S5	ginocchio destro
Servo 5	7	S6	piede destro

Tabella 3 - Collegamento dei servo con robot Spiderin.

Servo	Pin Arduino	Connettore shield	Funzione
Servo 0	2	S1	Angolazione zampa anteriore destra
Servo 1	3	S2	Elevazione zampa anteriore destra
Servo 2	4	S3	Angolazione zampa posteriore destra
Servo 3	5	S4	Elevazione zampa posteriore destra
Servo 4	6	S5	Angolazione zampa posteriore sinistra
Servo 5	7	S6	Elevazione zampa posteriore sinistra
Servo 6	8	S7	Angolazione zampa anteriore sinistra
Servo 7	9	S8	Elevazione zampa anteriore sinistra

bot dovete inviare il comando "axx" dove xx è la posizione da assegnare. Questo dato può assumere i valori compresi tra 80 e 100, con il punto centrale posto a 90; se non riuscite a centrare il servo rimanendo all'interno di questi valori, dovete spostare fisicamente il braccetto del servo. Agite, se necessario, anche sulle due astine che portano il comando dalla squadretta del servo ai piedi. Potete agire su questo valore qualora riscontrate una tendenza a deviare dalla linea retta durante la camminata. Passate ora al servo del passo ed impostate il neutro con il comando "bxx", dove xx è la posizione da assegnare (valore compreso tra 80 e 100). Adesso siete pronti per il collaudo sul campo: potete usare il telecomando per attivare le funzioni oppure, in mancanza di quest'ultimo, usare il pulsante della shield che però avvia solo la camminata.

Utilizzo con Robot Bipe

I servo vanno cablati seguendo le indicazioni della Tabella 2. L'elettronica Arduino-based va posizionata sul dorso del robot, mentre, il pacco batterie deve stare sulla parte anteriore. Diciamo subito che per le nostre prove abbiamo utilizzato una batteria composta da due celle LiPo da 850 mah, che, essendo piccole e leggere, trovano facilmente posto e non appesantiscono la struttura. Per posizionare correttamente il sensore SRF05 dovete utilizzare un piccolo adattatore realizzato con uno strip femmina ad angolo retto e pin lunghi.

Lo sketch da caricare in Arduino si chiama **bipe.pde**. (scaricabile gratuitamente dalla scheda del prodotto ROBOT_SHIELD disponibile su www.futurashop.it). Per questo robot è importante la taratura del neutro dei servo. Tramite Serial Monitor si deve inviare il comando composto dalla lettera del servo da settare, seguito dal valore che si vuole impostare (compreso tra 80 e 100). Al Servo0 corrisponde la lettera "a" ed al Servo5 la "f"; gli altri vanno di conseguenza. Se il comando è andato a buon fine, su Serial Monitor ne avrete conferma; ad esempio, "set servo: a to 90". Dopo che avrete messo in funzione il robot, probabilmente dovrete ritoccare la posizione di neutro: ad esempio, se il robot tende a cadere in avanti o indietro, oppure se durante la camminata tende a deviare dalla linea retta. Dalle foto riportate nell'articolo noterete che il dorso è leggermente piegato in avanti nell'intento di bilanciare i pesi. A lavoro ultimato usate del nastro adesivo e delle fascette per mettere in ordine e raggruppare i cavi dei servocomandi, così il robot avrà un aspetto più ordinato. Nel Bipe sono previste ulteriori funzioni come l'inchino ed il calcio. Al pulsante 1 del telecomando corrisponde la funzione calcio con la quale il robot può colpire, ad esempio una pallina; al pulsante 2 corrisponde l'inchino. I parametri settabili da programma sono simili a quelli del robot Filippo:

```
int TimeOneStep = 4000;
int AmpPasso = 10;
```

```
int AmpRuota = 12;
int IncPasso = 8;
```

La modifica di questi parametri è associata alla costruzione meccanica (dipende ad esempio dalla ripartizione del peso delle batterie). Essendo il baricentro quasi tutto spostato verso l'alto, la struttura è comunemente instabile e tende a cadere; i grandi piedi ed un movimento lento favoriscono la stabilità ma, se aumentate la velocità di esecuzione i movimenti diventano più bruschi ed il pericolo di cadute aumenta.

Utilizzo con Robot Spiderin

I servo vanno cablati seguendo le indicazioni della Tabella 3. Per questo robot non abbiamo previsto la regolazione del neutro dei servo, perché alcune imprecisioni non vanno ad influenzarne il funzionamento.

L'unica accortezza è fissare il pacco batterie in modo che il robot presenti il baricentro esattamente al centro. Suggeriamo, per prima cosa, di programmare la scheda Arduino con il programma **spiderin.pde** (scaricabile gratuitamente dalla scheda del prodotto ROBOT_SHIELD disponibile su www.futurashop.it.) e successivamente collegare elettricamente i servo, che così facendo saranno tutti posizionati esattamente al centro. Successivamente collegherete meccanicamente i servo nella loro giusta posizione. Assicuratevi che essi siano allineati e che le zampe non siano né troppo aperte, né troppo chiuse. All'accensione, dopo un se-

condo (trascorso il warm-Up dell'alimentatore switching) tutti i servo saranno posizionati nel punto di neutro; potrebbe capitare che qualche servo vibri leggermente perché il meccanismo interno non ha ancora trovato il punto centrale. Ebbene, alle volte è sufficiente un piccolo colpo al servo per sistemare le cose.

Per questo robot non abbiamo previsto parametri da modificare, in quanto i movimenti sono molti e molto complessi e ci sarebbero troppe variabili da settare; abbiamo preferito trovare un setup che potesse offrire il miglior funzionamento in ogni situazione. Per il nostro prototipo la batteria LiPo che lo alimenta è stata fissata sotto la struttura, proprio in mezzo ai servo, assicurando, oltre al centraggio dei pesi, un baricentro molto basso. Giusto per verificare se le ipotesi iniziali erano corrette, abbiamo eseguito alcune misure sulle correnti in gioco, visto che si usano otto servo: durante la camminata la corren-

te media assorbita dai servo è stata di 1,6 A, mentre quella erogata dalla batteria si è attestata su 1 A.

Con una tensione sulla batteria di 8,4 volt, la potenza erogata è di 8,4 watt, mentre quella dissipata dai servo è di 8 W.

Il telecomando

La libreria, IRremote, ci serve per gestire i comandi da telecomando e deve essere scaricata dal sito di riferimento <http://github.com/shirriff/Arduino-IRremote>; per il suo funzionamento è sufficiente specificare a quale pin è connesso il sensore di ricezione dei segnali IR.

Per sapere se sono arrivati dati dal sensore si deve testare lo stato della variabile `irrecv.decode(&results)`; il codice arrivato è contenuto nella variabile `results.value`. Per sapere a quale tasto corrisponde il codice arrivato bisogna confrontare quest'ultimo con il codice di riferimento; allo scopo specifichiamo

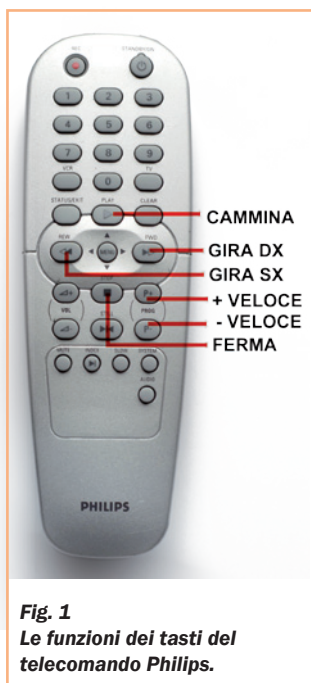


Fig. 1
Le funzioni dei tasti del telecomando Philips.

all'inizio dello sketch i codici relativi a ciascun pulsante. Ad ogni tasto sono associati due codici (CODE1 e CODE2) per

```

COM6
FILIPPO Aduino V1.0
Vbat= 7.8800001144
Livello batteria OK.
Recupero dati posizione neutro servi...
Neutro dei Servi: 87 90
Inizializzo Servi...
Servii inizializzati.
Invia 'obs' per testare il sensore di ostacoli
Invia 'lev' per testare il livello della batteria
Sono in Standby...
IR code= 1
TimeOneStep= 2000
IR code= 801
TimeOneStep= 2000
Autoscroll No line ending 9600 baud

```

Fig. 2 - Ricezione codici da telecomando Philips.

```

COM6
FILIPPO Aduino V1.0
Vbat= 7.8600001335
Livello batteria OK.
Recupero dati posizione neutro servi...
Neutro dei Servi: 87 90
Inizializzo Servi...
Servii inizializzati.
Invia 'obs' per testare il sensore di ostacoli
Invia 'lev' per testare il livello della batteria
Sono in Standby...
IR code= 20DF02FD
TimeOneStep= 2000
Autoscroll No line ending 9600 baud

```

Fig. 3 - Ricezione codici da telecomando LG.

distinguere la pressione momentanea da quella prolungata: nel primo caso i codici vengono inviati uno all'attivazione ed uno al rilascio, mentre nel secondo il trasmettitore ripete il codice corrispondente alla pressione. Quindi se si preme il pulsante Play viene inviato ciclicamente il codice 0x81C, mentre al rilascio viene inviato il codice 0x1C. Lo sketch è previsto per funzionare con telecomandi che usano la codifica Philips (fig. 1), che è la più usata nei televisori; se usate un trasmettitore di marca differente i codici dei pulsanti non corrispondono e dovrete modificare lo sketch, inserendo i codici che Serial Monitor mostra sullo schermo del PC quando trasmettete col vostro telecomando (Fig. 2 e Fig. 3).

Ad esempio, se utilizzate un telecomando LG noterete che il codice inviato è unico indipendentemente da come premete i pulsanti. Volendo, ad esempio, assegnare il pulsante "freccia in su" della navigazione a menù alla funzione di camminata, non dovrete fare altro che premerlo, leggere il codice IR ricevuto e trascriverlo nella riga dello sketch che assegna il codice per la camminata:

```
#define WALK_CODE1 0x20DF02FD
#define WALK_CODE2 0x20DF02FD
```

I parametri che possono essere modificati da programma per adattare i movimenti sono:

```
int TimeOneStep = 2000;
```

```
int AmpPasso = 30;
int IncPasso = 15;
int AmpRuota = 30;
int IncRuota = 18;
```

TimeOneStep è il valore iniziale del tempo impiegato per eseguire un passo, può essere modificato da telecomando da un minimo di 1 ad un massimo di 4 secondi.

AmpPasso è la massima ampiezza (in gradi) dell'angolo formato dalle gambe durante la camminata; il suo valore può variare da un minimo di 5 ad un massimo di 40.

IncPasso è la massima inclinazione del robot durante la camminata; il suo valore può variare da un minimo di 5 ad un massimo di 20 e deve garantire al robot di rimanere in equilibrio su una gamba mentre compie un passo.

AmpRuota è la massima ampiezza in gradi dell'angolo formato dalle gambe durante la rotazione; il suo valore può variare da un minimo di 5 ad un massimo di 40.

IncRuota è la massima inclinazione del robot durante la rotazione; il suo valore può variare da un minimo di 5 ad un massimo di 20.

Scelta della soglia di tensione della batteria

Il sistema effettua la lettura della tensione sulla batteria (attraverso il canale analogico A0 di Arduino), effettua il salvataggio del dato nella variabile `VbatRAW`, quindi divide il valore per 50; in questo modo ottiene la tensione in volt della batteria. Per evitare l'eccessiva scarica abbiamo previsto una protezione che, al di sotto di una certa tensio-

ne, toglie tensione a tutti i ser-vo; il consumo in corrente è quindi ridotto al minimo ma in tempi brevi è opportuno scollegare e ricaricare le batterie. La soglia è 6 volt, ideale per un pacco batterie composto da due celle LiPo o sei NiMH/NiCd; per pacchi da 8 di queste ultime potete portare la soglia ad 8 volt.

A tutti i residenti nell'Unione Europea. Importanti informazioni ambientali relative a questo prodotto

Questo simbolo riportato sul prodotto o sull'imballaggio, indica che è vietato smaltire il prodotto nell'ambiente al termine del suo ciclo



vitale in quanto può essere nocivo per l'ambiente stesso. Non smaltire il prodotto (o le pile, se utilizzate) come rifiuto urbano indifferenziato; dovrebbe essere smaltito da un'impresa specializzata nel riciclaggio. Per informazioni più dettagliate circa il riciclaggio di questo prodotto, contattare l'ufficio comunale, il servizio locale di smaltimento rifiuti oppure il negozio presso il quale è stato effettuato l'acquisto.

Prodotto e distribuito da:
FUTURA ELETTRONICA SRL
 Via Adige, 11 - 21013
 Gallarate (VA)
 Tel. 0331-799775
 Fax. 0331-778112
 Web site: www.futurashop.it
 Info tecniche: supporto@future.com

L'articolo completo del progetto è stato pubblicato su: *Electronica In* n. 166