

IL QUARTO CD-ROM DI I-DROID01

Con questo fascicolo hai trovato in allegato anche il quarto CD-ROM di I-Droid01, contenente le ultime versioni dei software di controllo e dei supporti per la programmazione. Grazie a questi upgrade potrai sfruttare al massimo tutte le potenzialità del robot, comprese quelle della pinza e del telecomando universale.

Come hai già sperimentato nel corso delle diverse fasi di montaggio, I-Droid01 ha visto accrescere nel tempo la propria dotazione elettronica e meccanica. L'ultimo passo di questa evoluzione riguarda i dispositivi della fase di montaggio corrente, in particolare la pinza, il sistema di rilevamento ostacoli a infrarossi e il telecomando universale. Parallelamente, anche l'equipaggiamento software è andato cambiando, in modo da adattarsi il meglio possibile alle funzioni e agli apparati disponibili. Tale evoluzione software è stata possibile grazie ai CD-ROM che hai ricevuto nel corso dell'opera, il quarto dei quali è allegato a questo fascicolo. Il quarto CD consente un ulteriore 'passo evolutivo' da parte del software: tra i suoi contenuti, infatti, si trovano le nuove versioni dei software di controllo, delle librerie per la programmazione (Java e C-like) e del Visual C-like Editor, tutte



Il quarto CD-ROM di I-Droid01 (sotto a sinistra), allegato a questo fascicolo, presenta un menu iniziale (sopra) con tre sezioni: SOFTWARE (con le nuove versioni del firmware per il B&V, dei programmi di controllo e dei supporti per la programmazione), MONTAGGIO (con i video di alcuni passi di assemblaggio) e MANUALI (in cui si trova tra l'altro la documentazione Java).

realizzate per consentire l'utilizzo dei nuovi sistemi oggetto di questa fase di montaggio. In più, viene incluso il firmware aggiornato per il B&V. Come per gli altri CD, anche in questo caso la sezione SOFTWARE risulta quindi molto importante.

Nel CD, però, sono presenti anche altre due sezioni: in MONTAGGIO sono disponibili i video relativi ad alcuni passi di assemblaggio, mentre in MANUALI viene fornita la documentazione relativa alle nuove versioni del Visual C-like Editor e dei supporti per la programmazione, come anche la descrizione del menu del display, modificato a causa dell'aggiornamento del firmware.





[1] SOFTWARE

La prima sezione del CD è SOFTWARE. Come di consueto, tramite le voci elencate in essa è possibile installare i vari programmi. Va ricordato come sia opportuno, prima di installare le nuove versioni dei software, disinstallare le versioni correnti, operazione che, come già detto in passato, porterà alla cancellazione di qualsiasi file contenuto nelle cartelle dei programmi. Prima di procedere con la disinstallazione, quindi, qualsiasi file di salvataggio va spostato in una cartella 'sicura'. In SOFTWARE si trovano la nuova versione del Visual C-like Editor, del PC Control e del Mobile Control, come anche la 'release' aggiornata del firmware per il modulo B&V e delle librerie Java. Queste ultime vengono accompagnate da diversi programmi di esempio, tra cui quelli mostrati nel corso dei fascicoli. Per quanto riguarda il primo dei contenuti software elencati, il Visual C-like Editor, esso mostra, come novità principale, la possibilità di utilizzare i comandi per il controllo della pinza e del telecomando universale, in C-like e in Visual C-like. Ovviamente queste nuove funzioni saranno utilizzabili quando la pinza e il telecomando saranno stati completati e connessi al robot. Possono essere impiegati da subito, invece, i comandi in C-like relativi ai sensori a infrarosso sulla breadboard. A tale proposito può essere utile eseguire il programma di esempio dal nome `ir_sensors_sample`, che mostra i blocchetti Visual C-like da impiegare nella gestione dei sensori, ottenuta sfruttando gli input a interrupt. Passando al PC Control, anch'esso presenta varie novità che riguardano la pinza e il telecomando universale, ma non solo. Una delle novità principali, infatti, riguarda piuttosto le immagini ottenute dalla CMOS camera che, grazie alla nuova versione del PC Control, possono ora essere memorizzate

sotto forma di filmato AVI in una cartella a scelta del proprio PC. Inoltre, passando con il cursore sull'immagine restituita nell'interfaccia di controllo dalla CMOS camera, viene ora mostrata una terna numerica, corrispondente ai valori in RGB del pixel indicato col cursore stesso. Per quanto riguarda gli hand tool 'avanzati', invece, il software di controllo permette di aprire e chiudere la pinza tramite la pressione prolungata, nell'interfaccia di controllo della base, rispettivamente dei tasti 9 e 3, mentre con la pressione prolungata del tasto 6 si interrompe il movimento della pinza stessa. Inoltre, sono presenti due nuove interfacce, dedicate al telecomando universale, identificate dalla presenza (in basso a destra della finestra del PC Control) di un piccolo telecomando iconizzato. Le due interfacce consentono di addestrare il telecomando e di utilizzarlo; è possibile spostarsi da una all'altra situazione premendo il tasto /. Anche con il Mobile Control è possibile addestrare e utilizzare il telecomando universale. Proprio in questo consiste la novità principale introdotta dalla versione inclusa nel CD 4. Ancora una volta, la gestione del telecomando viene realizzata tramite due interfacce, una utilizzabile per addestrare il dispositivo e l'altra per il suo utilizzo. Per spostarsi da un'interfaccia all'altra bisogna premere il tasto asterisco (*). In modo opposto rispetto al PC Control, poi, è possibile far aprire e chiudere la pinza del robot tramite l'interfaccia di controllo della base premendo in modo prolungato i tasti 3 (apertura), 9 (chiusura) e 6 (stop dei movimenti). Per i dettagli sulle nuove funzioni del PC e del

La prima sezione del CD-ROM è SOFTWARE; essa contiene gli aggiornamenti delle versioni dei vari strumenti software per I-Droid01. Sotto a sinistra, l'elenco principale della sezione; qui sotto, la pagina relativa alla libreria Java e agli esempi.

[1] SOFTWARE

In questa sezione trovi le versioni aggiornate del software per controllare e programmare I-Droid01.

Mobile Control e PC Control offrono le nuove interfacce per il controllo del telecomando universale, mentre il linguaggio Visual C-like consente di gestire la pinza, i sensori laterali e infrarossi e i codici del telecomando universale.

- Visual C-like Editor versione aggiornata
- PC Control e Mobile Control versioni aggiornate
- Installazione Brain & Vision versione aggiornata
- Libreria Java e codice di esempio
- Installa Adobe Acrobat Reader 7.0.5
- Installa Adobe Acrobat Reader 5.0.5 per sistemi Windows 95 e ME

Avvertenza

Disinstallare sempre le vecchie versioni del software prima di installare le nuove, anche sul telefono. **La disinstallazione del software dal PC rimuove tutto il contenuto delle cartelle di programma.** Si consiglia pertanto di **NON salvare nessun file in queste cartelle** ("C:\Programmi\I-Droid01\PC Control" e "C:\Programmi\I-Droid01\Visual C-like Editor").

In particolare, nel caso di Visual C-like Editor si potrebbero perdere eventuali programmi C-like salvati nella cartella dell'applicazione. In tale evento preleva altri file diversi da quelli creati dall'installazione, si raccomanda di spostarli o copiarli su un'altra cartella, fuori da "C:\Programmi\I-Droid01" (per esempio in "Documents"), **prima** della disinstallazione.

[1] SOFTWARE

In questa sezione trovi le versioni aggiornate del software per controllare e programmare I-Droid01.

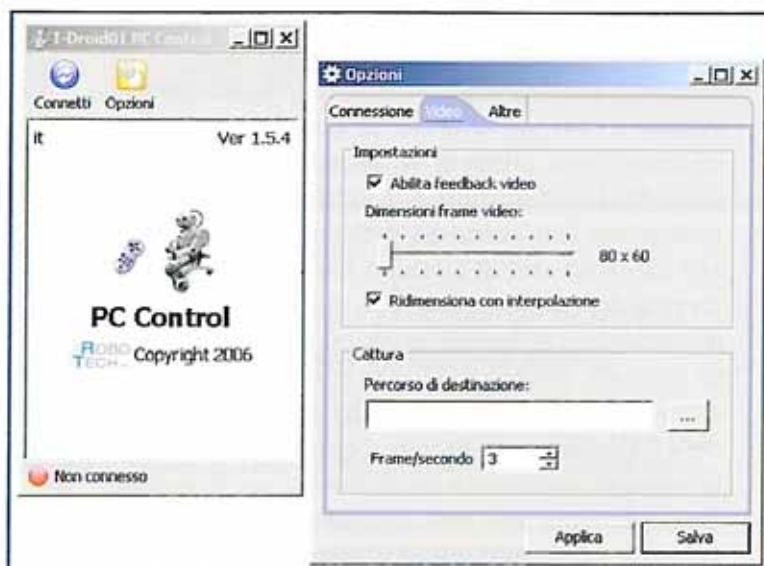
Mobile Control e PC Control offrono le nuove interfacce per il controllo del telecomando universale, mentre il linguaggio Visual C-like consente di gestire la pinza, i sensori laterali e infrarossi e i codici del telecomando universale.

La libreria Java contiene l'implementazione del protocollo di comunicazione utilizzato nel collegamento Bluetooth con il PC e il telefono cellulare e ti consente di scrivere programmi per controllare il robot sfruttando le capacità di calcolo del tuo PC. Nel codice di esempio, disponibile anche in formato binario, troverai alcuni esempi di utilizzo della libreria stessa (accesso ai moduli interni, accesso al bus, trasferimento file, streaming video, ecc.).

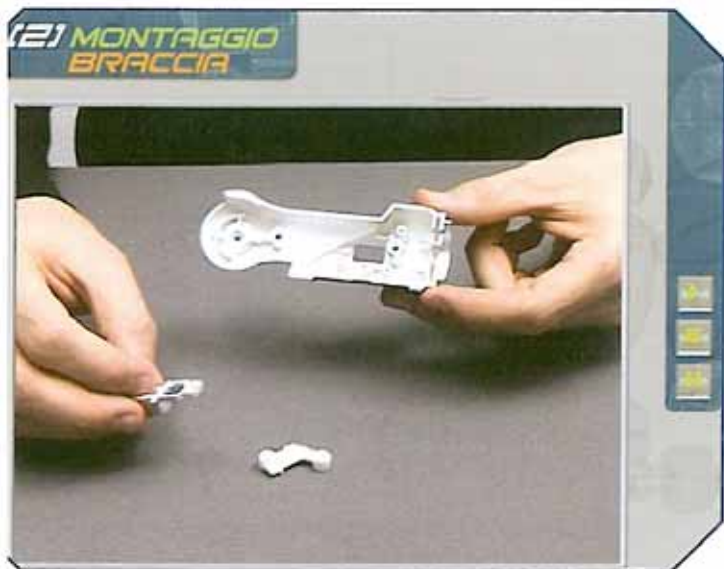
- Libreria Java
- Codice di esempio
- Sorgenti e progetti Eclipse
- Codice binario
- Esempi fascicolo

- Visual C-like Editor versione aggiornata
- PC Control e Mobile Control versioni aggiornate
- Installazione Brain & Vision versione aggiornata
- Libreria Java e codice di esempio
- Installa Adobe Acrobat Reader 7.0.5
- Installa Adobe Acrobat Reader 5.0.5 per sistemi Windows 95 e ME

Mobile Control è opportuno consultare le guide relative, accessibili dalla stessa sezione SOFTWARE del CD. Il quarto contenuto software che viene incluso nel CD-ROM è il firmware del modulo Brain & Vision. Oltre a fornire il supporto necessario all'utilizzo delle versioni aggiornate del PC e Mobile Control, come anche alla programmazione in Visual C-like, C-like e Java, il firmware consente alcune nuove funzioni nel menu del display di I-D01. In particolare, è possibile gestire il telecomando (Remote Control nel sottomenu Diagnostics), permettendo o meno la trasmissione e lanciando la procedura di apprendimento vocale. In più, è possibile settare il livello di forza della 'stretta' della pinza (sottomenu Diagnostics, voce Hand), scegliendo un valore compreso tra 1 e 6. Per una descrizione del nuovo menu del display e una breve introduzione all'addestramento del telecomando, è possibile fare riferimento all'apposita guida



Uno dei software contenuti nel CD è il PC Control. Sopra, una schermata che mostra, tra l'altro, la nuova opzione per la memorizzazione del filmato registrato dalla CMOS.



(2) MONTAGGIO

La sezione MONTAGGIO presenta diversi filmati, relativi ai passi di assemblaggio più importanti fra quelli illustrati negli ultimi fascicoli. Così, sono mostrati alcuni passaggi nel montaggio della breadboard e dei kit sensoriali, del modulo Arms, delle braccia e di alcuni degli hand tool.

(3) MANUALI

Nell'ultima sezione, MANUALI, vengono presentate le documentazioni. È possibile accedere alla guida per il Visual C-like Editor, che comprende tra l'altro la lista delle funzioni utilizzabili nella programmazione in C-like, come anche allo schema riassuntivo delle funzioni del nuovo menu del display. Inoltre, tramite la sezione MANUALI si può accedere alla documentazione che descrive le librerie Java incluse nella sezione SOFTWARE.

inserita nella sezione MANUALI del CD. In ogni caso, va ricordato ancora una volta che tutte le funzioni relative al telecomando universale e alla pinza saranno utilizzabili solo quando tali dispositivi saranno completati. La sezione SOFTWARE, infine, contiene le versioni aggiornate delle librerie per la programmazione in Java, comprensive di nuovi esempi. Le principali novità riguardano la possibilità di settare alcuni parametri di funzionamento della CMOS camera, di elaborare le immagini a bordo del robot, di far comunicare programmi Java e C-like, di usare la pinza e il telecomando. Anche in C-like sono disponibili nuove funzioni, in particolare per il settaggio della CMOS, l'elaborazione delle immagini a bordo del robot e l'uso di pinza e telecomando.

La sezione MONTAGGIO comprende i video relativi ad alcuni recenti passi di assemblaggio (sopra), mentre tramite MANUALI si può accedere anche alla documentazione Java (a destra).

Overview Package Class Use Tree Deprecated Index Help

PREV NEXT EDITED 10/18/06

I-Droid01 Communication Library

Communication Library

See: Description

Packages	
communication.handler	Connection management and protocol capabilities handlers
communication.handler.bus	Implementation of the Bus capability
communication.handler.internal	Implementation of the Internal capability
communication.handler.transfer	Implementation of the Transfer capability
communication.handler.video	Implementation of the Video capability
communication.protocol	Low level protocol encoding and decoding
communication.transport	Abstractions for the transport layer

Communication Library

License

Copyright (c) 2006, RoboTech srl

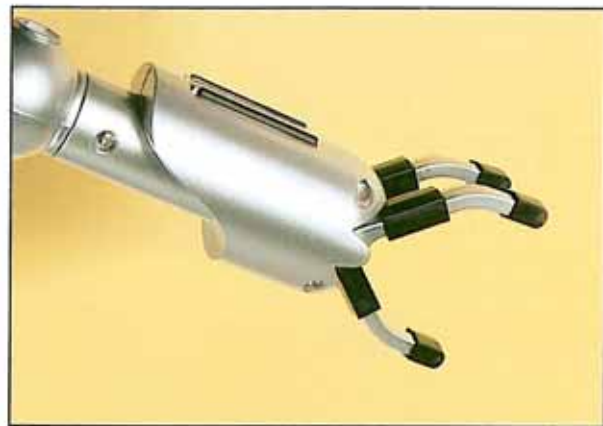
CONTROLLO DELLA MANO IN JAVA

In queste pagine viene mostrata una libreria in linguaggio Java per il controllo della mano di I-Droid01, che potrai utilizzare a breve nella realizzazione dei tuoi programmi quando l'assemblaggio della mano stessa sarà completato.

Nell'esempio proposto in questo fascicolo è presentata una libreria in linguaggio Java per la gestione della mano di I-Droid01. Prima di poterla utilizzare, però, dovrai attendere l'uscita numero 89, alla quale troverai allegata la scheda elettronica per il controllo della mano. A differenza degli esempi trattati nei fascicoli precedenti, in questo non viene implementata nessuna interfaccia grafica: potrai essere tu a crearne una, personalizzata in base alle tue esigenze, oppure potrai limitarti a utilizzare i metodi della libreria 'riciclandoli' quando necessario nei tuoi programmi.

LA CLASSE HANDCONTROL

La classe `HandControl` mette a disposizione una serie di metodi per la gestione della pinza di I-D01. A differenza del C-like e del Visual C-like, in Java sono previste due modalità di controllo. La prima, più semplice, permette di controllare la chiusura della mano impostando una precisa soglia di 'forza' per determinare quando arrestare il motore. Tale soglia, il cui valore può variare da 1 a 10, è inversamente proporzionale alla 'sensibilità' di I-Droid01 nel percepire la presa di un oggetto. Con un valore pari a 1, il motore che controlla la mano interromperà la presa appena sarà percepita una lieve resistenza. Con una soglia più elevata, al contrario, eserciterà una stretta maggiore prima di bloccare la presa. Sebbene sia possibile impostare qualsiasi valore tra 1 e 10, per evitare di sottoporre a stress eccessivo gli ingranaggi e la scheda della pinza, è consigliabile usare valori inferiori a 7. Per lo stesso motivo, del resto, in Visual C-like non sono disponibili i livelli superiori a 6. La seconda modalità di controllo, invece, è basata sul monitoraggio della corrente elettrica assorbita dal motore per stabilire se la mano sta afferrando un oggetto. Analizziamo ora in dettaglio i diversi metodi della libreria. Oltre al costruttore della classe, i primi metodi che incontriamo sono `hand_open` e `hand_close` che permettono, rispettivamente, di



Per usare la libreria Java presentata nella pagina seguente dovrai attendere l'uscita 89, dove troverai la scheda per la gestione della mano (sopra, come apparirà una volta completa).

aprire e chiudere la mano del robot utilizzando il controllo della forza. Il parametro `force` del metodo `hand_close` stabilisce la soglia di forza, secondo quanto detto in precedenza. Il metodo per l'apertura della mano non richiede alcun parametro: in questo caso, infatti, il livello di forza viene settato automaticamente dal software di controllo del robot. Il metodo successivo, `hand_stop`, forza l'interruzione dei movimenti della mano. Per ottenere le informazioni sullo stato della mano, sono presenti `get_hand_pos` e `get_error_type`. Il primo di questi metodi restituisce il valore 0 qualora la mano sia in una posizione indefinita, 1 se completamente aperta, 2 se completamente chiusa, 32 se in movimento. Infine è previsto anche il valore -1, per segnalare anomalie durante l'accesso ai registri di I-Droid01. Il secondo metodo, invece, è utile per ottenere informazioni circa eventuali problemi occorsi durante l'uso della mano del robot. Gli ultimi due metodi, `hand_open_current` e `hand_close_current`, servono per il controllo della mano basato sul calcolo della corrente assorbita dal motore. Per ulteriori dettagli, consulta nel CD la documentazione del protocollo I2C.

ESEMPI DI PROGRAMMAZIONE

Esempio di codice Java per l'implementazione di una libreria di controllo della mano di I-Droid01. Per la sua gestione sono previste due diverse tipologie di controllo, una 'di forza', per mezzo di una soglia numerica compresa tra 1 e 10, l'altra in base alla corrente elettrica assorbita dal motore. Per la prima modalità sono presenti i metodi `hand_open` e `hand_close`; per la seconda, invece, `hand_open_current` e `hand_close_current`. Gli altri metodi presenti nella libreria permettono di ottenere alcune informazioni specifiche sullo stato della mano di I-Droid01.

CLASSE HANDCONTROL

```

package communication.examples;

import communication.handler.internal.InternalHandler;
import communication.handler.internal.InternalModule;
import communication.handler.internal.HandData;
import communication.handler.internal.HandRegister;
import java.io.IOException;

public class HandControl {

    private InternalHandler internal;

    public HandControl(InternalHandler internal) {
        this.internal = internal;
    } // class constructor

    void hand_open() {
        int[] buf = new int[1];
        // Apertura per livello di forza
        buf[0] = HandData.OPEN;
        try {
            internal.writeRegister(InternalModule.HAND,
                HandRegister.COMMAND, buf);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } // hand_close

    void hand_close(int force) {
        int[] buf = new int[1];
        if ((force < 1) || (force > 10)) return;
        buf[0] = force;
        try {
            internal.writeRegister(InternalModule.HAND,
                HandRegister.FORCE_LEVEL, buf);
        } catch (Exception e) {
            e.printStackTrace();
        }
        buf[0] = HandData.CLOSE; // Chiusura per livello di forza
        try {
            internal.writeRegister(InternalModule.HAND,
                HandRegister.COMMAND, buf);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } // hand_close

    void hand_stop() {
        int[] buf = new int[1];
        buf[0] = HandData.STOP; // stop
        try {
            internal.writeRegister(InternalModule.HAND,
                HandRegister.COMMAND, buf);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } // hand_stop

    int get_hand_pos() {
        int[] buf = new int[1];
        try {
            internal.readRegister(InternalModule.HAND,
                HandRegister.COMMAND, buf);
            return buf[0];
        } catch (Exception e) {
            e.printStackTrace();
            return -1;
        }
    } // get_hand_pos

    int get_error_type() {
        int[] buf = new int[1];
        try {
            internal.readRegister(InternalModule.HAND, 6, buf);
            return buf[0];
        } catch (Exception e) {
            e.printStackTrace();
            return -1;
        }
    } // get_error_type

    void open_hand_current(int level) {
        int[] buf = new int[1];
        if ((level < 0) || (level > 80)) return;
        buf[0] = level;
        try {
            internal.writeRegister(InternalModule.HAND,
                HandRegister.OPEN_CURRENT, buf);
        } catch (Exception e) {
            e.printStackTrace();
        }
        // Apertura con soglia in corrente
        buf[0] = HandData.OPEN_CUR;
        try {
            internal.writeRegister(InternalModule.HAND,
                HandRegister.COMMAND, buf);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } // open_hand_current

    void close_hand_current(int level) {
        int[] buf = new int[1];
        if ((level < 0) || (level > 80)) return;
        buf[0] = level;
        try {
            internal.writeRegister(InternalModule.HAND,
                HandRegister.CLOSE_CURRENT, buf);
        } catch (Exception e) {
            e.printStackTrace();
        }
        // Chiusura con soglia in corrente
        buf[0] = HandData.CLOSE_CUR;
        try {
            internal.writeRegister(InternalModule.HAND,
                HandRegister.COMMAND, buf);
        } catch (Exception e) {
            e.printStackTrace();
        }
    } // close_hand_current

} // class HandControl

```