

UDOO CIRCUIT

Progetto a cura di Alessandro Paghi, Aldi Piroli e Mihai Dragusanu.

IN GENERALE...

UDOO-Circuit è un dispositivo finalizzato all'acquisizione di dati di crociera di veicoli a due ruote in movimento.

In particolare lo scopo del progetto è conseguire il tracciamento di un veicolo a due ruote in movimento, tramite le coordinate longitudine e latitudine, e l'acquisizione di alcuni parametri di moto fondamentali, come velocità di percorrenza e angolo di piega.

Il dispositivo è completamente sviluppato sulla board UDOO Neo, sia sul lato Linux che sul lato Arduino.

Il lato Arduino della board si interfaccia con un modulo GPS tramite la porta seriale UART_5, per l'acquisizione delle coordinate spaziali, temporali e relative alla velocità di percorrenza, con uno schermo LCD tramite il bus I2C (SDA, SCL), per la stampa a video dei parametri acquisiti dal modulo GPS, e con una serie di LED per un primo feedback luminoso dei parametri in gioco.

Il lato Linux della board si interfaccia con tre sensori già presenti sulla board stessa: un magnetometro, un accelerometro ed un giroscopio.

In definitiva il sistema designato, recupera i dati dal modulo GPS e li combina con i dati forniti dal Motion Sensor (accelerometro, giroscopio, magnetometro). Dopodiché un feedback dei dati viene in un primo momento fornito sul sistema hardware stesso grazie al display LCD ed ai Led, ed in un secondo momento su una mappa opportunamente creata utilizzando Google Maps.

LINUX PART (A9 PART I)

Funzionamento Generale

Questa parte prevede l'utilizzo del core A9 del processore IMX6Solo.

Step effettuati:

- Inizializzazione della parte Accelerometro del chip FXOS8700CQ in modalità +-2G e della Seriale Virtuale per la comunicazione A9-M4 situata al percorso /dev/ttyMCC;
- Acquisizione dei dati grezzi dell'Accelerometro;
- Conversione dei dati in grezzi in termini di angolo di inclinazione della board;
- Trasmissione del dato ricavato al core M4;
- Ricezione dei dati relativi a latitudine, longitudine e velocità dal core M4;
- Allocazione dei dati nel file /var/www/html/gpx_data/gpx_data.txt;

Materiale hardware e Software utilizzato

Hardware:

- UDOO Neo (A9 SIDE);
- FXOS8700CQ (Accelerometro, Magnetometro);

Software:

- Linux Shell;
- Python language

Motivazioni per le scelte di tali materiali

Abbiamo utilizzato il chip FXOS8700CQ in quanto esso è già integrato sulla board stessa ed è comunicante con il microprocessore IMX6Solo, lato A9, tramite bus I2C. In particolare il nome del bus è I2C4 e l'indirizzo del sensore è 0x1E.

Abbiamo utilizzato Python in quanto per questo linguaggio di programmazione è disponibile la libreria SMBUS che permette di utilizzare il bus I2C con board che integrano sistemi Linux.

Cenni sul codice

Librerie utilizzate

- Python-smbus: permette un accesso SMBUS attraverso le interfacce I2C presenti in /dev negli host Linux;
- Python-serial: permette di interfacciarsi con porte Seriali UART implementate sulla scheda.

Problemi Riscontrati

Difficoltà di comprensione del DataSheet del chip FXOS8700CQ

ARDUINO PART (M4 PART)

Funzionamento Generale

Questa parte prevede l'utilizzo del core M4 del processore IMX6Solo.

Step effettuati:

- Acquisizione, attraverso il modulo GPS NEO-6M, della posizione della scheda nello spazio in termini di latitudine e longitudine, e della velocità della stessa.

Il core M4 per comunicare con il GPS utilizza la porta seriale "Serial0", che nel microprocessore viene implementata attraverso la UART_5 sul pin 0/1.

- Stampa della posizione ricevuta dal GPS sul display LCD fornendo un primo feedback visivo all'utente.
- Invio di tutti i dati acquisiti dal core M4, attraverso la seriale virtuale "Serial", al core A9. Quest'ultima è stata implementata per far sì che il microprocessore possa far comunicare i due core, A9 e M4, fra di loro in media semplice ed intuitiva.

Viene chiamata seriale virtuale in quanto fisicamente è memoria dedicata proprio per la comunicazione fra i due core.

- Ricezione dell'angolo di inclinazione della scheda dal core A9 attraverso la seriale virtuale "Serial" ed accensione degli appositi led situati sulla Breadbord attraverso i pin GPIO, fornendo un secondo feedback visivo all'utente.

Materiale hardware e Software utilizzato

Hardware:

- UDOO Neo (M4 SIDE);
- Modulo GPS NEO-6M;
- Display LCD 16x02 e modulo convertitore Serial_Uart-I2C;
- LED di vario colore;

Software:

- Arduino IDE 1.5.4 ;

Motivazioni per le scelte di tali materiali

Abbiamo utilizzato il modulo GPS, in quanto il progetto prevede l'utilizzo di quest'ultimo per riuscire a ricavare la posizione della scheda nello spazio in termini di latitudine e longitudine;

Abbiamo scelto di utilizzare un display LCD in modo da fornire un feedback visivo all'utente.

Cenni sul codice

Librerie utilizzate

- Wire: libreria utilizzata per gestire i dispositivi elettronici che sono stati sviluppati con la tecnologia I2C;
- LiquidCrystal_I2C: libreria utilizzata per il controllo del Display LCD;
- TinyGPS++: libreria utilizzata per il controllo del modulo GPS;

Problemi Riscontrati

Conversione Float to String

La funzione `dtostrf()` della libreria `stdlib.h` non converte un dato float in un dato string.

Approssimazione

Il core M4 riesce a convertire con precisione solamente numeri float con 6 cifre dopo la virgola.

Libreria TinyGPS++

La libreria non è in grado, una volta acquisito il segnale GPS, di rilevare la perdita di tale segnale.

Display LCD

Per nessun motivo plausibile il Display LCD funziona esclusivamente al primo Upload del codice.

WEB INTERFACE PART (A9 PART II)

Funzionamento Generale

Questa parte prevede l'utilizzo del core A9 del processore IMX6Solo.

L'interfaccia web permette all'utente di avere un feedback storico riguardante le misure effettuate tramite l'utilizzo di una mappa.

Materiale hardware e Software utilizzato

Hardware:

- UDOO Neo (A9 SIDE);

Software:

- Web Server Apache;
- HTML language;
- CSS language;
- JavaScript language;

Motivazioni per le scelte di tali materiali

Cenni sul codice

Librerie utilizzate:

- Google Maps APIs (doc.);
- MapLabel for Google Maps V3 (doc.);

Info sui dati allocati:

I dati ricavati dai sensori sono salvati al percorso: `var/www/html/gpx_data/gpx_data.txt` nel seguente ordine: Latitudine, Longitudine, Velocità, Angolo Di Piega.

```
40.709604,-73.951192,3,5
40.709562,-73.951108,4,4
40.709539,-73.951085,7,7
40.709520,-73.950978,10,9
40.709498,-73.950940,12,10
40.709478,-73.950871,15,16
40.709478,-73.950810,17,17
40.709456,-73.950703,15,19
```

Problemi Riscontrati

- L'impossibilità di scrivere testo su una mappa, con solo le librerie di default di Google Maps JavaScript API. Per questo abbiamo usato una libreria esterna (MapLabel for Google Maps V3);
- Con solo la funzione 'Polyline' non si ha la sicurezza di tracciare linee che rappresentino la realtà del percorso fatto. Per questo si suggerisce di usare la classe 'Google Maps Directions API', che a causa del tempo limitato e del costo non è stato possibile implementare.

ANALISI DELL'ARCHITETTURA DELLA BOARD

In generale...

Tutte le versioni di UDOO Neo sono equipaggiate con un processore Freescale IMX6SoloX. Tale processore racchiude in un singolo chip un core ARM Cortex A9 ed un core ARM Cortex M4.



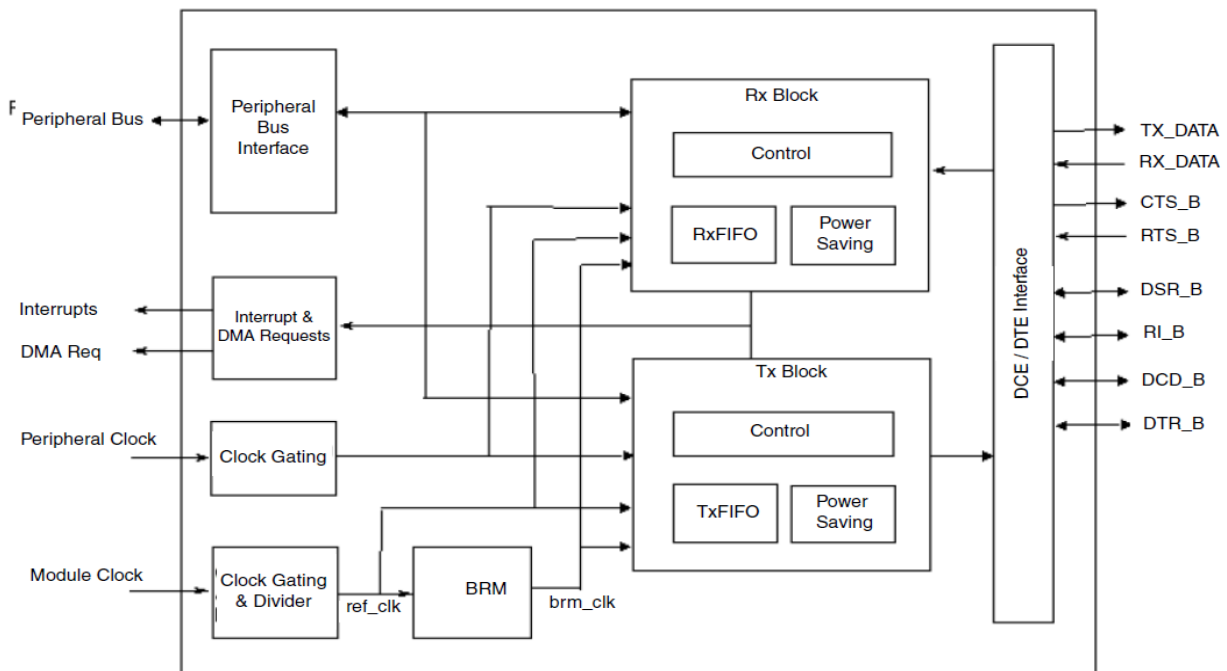
Quindi non ci sono due Chip separati come nella precedente versione di UDOO, ma un solo chip. Questo significa minore consumo di energia, minor costo ed una velocità di comunicazione più elevata, infatti non è presente una linea Seriale UART fra i due core come nella precedente board, ma una memoria condivisa ad alta velocità. Questo garantisce performance migliori, stabilità e maggiore robustezza del sistema stesso.

UART Serials

In Generale...

Universal Asynchronous Receiver/Transmitter (UART) fornisce una comunicazione seriale con i dispositivi esterni attraverso un convertitore di livelli ed un cavo RS-232

Diagramma a Blocchi



Info sul Diagramma a Blocchi

Interrupts

Fornisce una richiesta di Interrupt in Output;

DMA Request

I segnali utilizzati sono:

- Dma_req_rx: fornisce un output DMA request signal per l'interfaccia ricevente;
- Dma_req_tx: fornisce un output DMA request signal per l'interfaccia trasmittente;

Peripheral Clock

È usato come clock scrivente per TxFIFO, come clock leggente per RxFIFO e come sincronizzatore per gli input pin per il controllo del modem. Entra in funzione appena la UART viene alimentata ed abilitata.

Module Clock

È usato per tutti gli stati macchina, scrivere in RxFIFO, leggere in TxFIFO, etc. Entra in funzione quando la UART invia o riceve caratteri.

TX_DATA (UART Transmit)

È l'output del trasmettitore seriale. Quando si opera in modalità RS-232, i dati sono codificati secondo la logica NRZ e invertiti prima della trasmissione.

RX_DATA (UART Receive)

È l'input del ricevitore seriale. Quando si opera in modalità RS-232, i dati sono decodificati secondo la logica NRZ e invertiti prima di essere processati.

CTS_B (Clear To Send)

Normalmente, il ricevitore indica che è pronto per ricevere dati posizionando a livello low questo pin.

RTS_B (UART Request To Send)

È utilizzato per controllare il trasmettitore. Il modem o qualsiasi altro dispositivo terminale connesso con la UART pone a '0' il pin RTS_B per indicare che è pronto per ricevere dati.

DSR_B (Data Set Ready)

DCE usa questo segnale per informare DTE che è attivo, ha completato le operazioni di preparazione ed è pronto per comunicare.

È un segnale di output se la seriale è in DCE mode altrimenti è un segnale di input se la seriale è in DTE mode.

RI_B (Ring Indicator)

Se usato, DCE usa questo segnale per informare DTE che si è appena verificato un ring.

È un segnale di output se la seriale è in DCE mode altrimenti è un segnale di input se la seriale è in DTE mode.

DCD_B (Data Carried Detect)

Se usato, DCE usa questo segnale per informare DTE che ha rilevato il carrier signal a la connessione può essere stabilita.

È un segnale di output se la seriale è in DCE mode altrimenti è un segnale di input se la seriale è in DTE mode.

DTR_B (Data Terminal Ready)

È come se fosse il "Main Switch": è responsabile dello stabilimento della connessione fra trasmettitore e ricevitore.

È un segnale di input se la seriale è in DCE mode altrimenti è un segnale di output se la seriale è in DTE mode. Quando la connessione fra DTE e DCE è stata stabilita per la prima volta, DTR_B rimane attivo per tutta la durata della connessione.

UDOO Neo UARTs

Questa tabella fornisce una descrizione delle UART disponibili sulla board:

Nome	Funzione	Descrizione
UART_1	Cortex A9 serial debug	Tramite questa seriale è possibile avere accesso alla Console di Linux.
UART_2	Cortex M4 MQX debug	Le librerie del Freescale MQX real-time OS usano questa seriale per il Debug.
UART_3	WL1831 Bluetooth chip	È utilizzata per colloquiare con il chip Bluetooth montato sulla scheda.
UART_4	n.c.	n.a.
UART_5	Arduino pin 0 /1 serial	È utilizzata dal core M4 tramite l'oggetto Serial0 disponibile nell'Arduino IDE attraverso i pin della GPIO 0/1.
UART_6	Cortex A9 user serial	Non è utilizzata di default. Può essere connessa con il core A9.

I2C

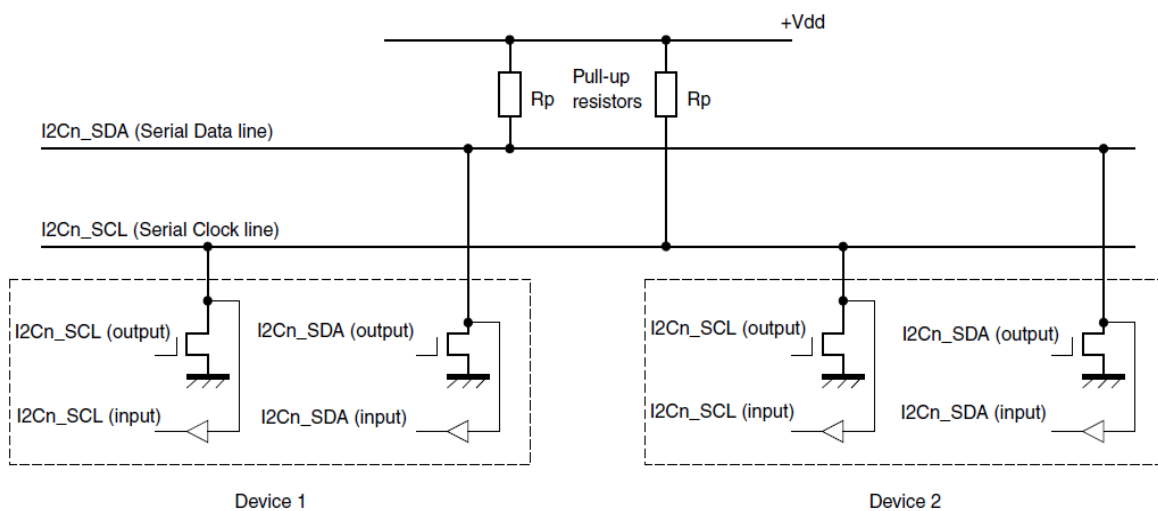
In generale...

I2C (Inter-IC) è un bus bidirezionale a due cavi seriali che fornisce un link di comunicazione fra circuiti integrati (ICs).

I2C è un metodo efficiente per scambiare dati fra dispositivi, minimizzando le interconnessioni fra gli stessi. Questo bus è perfetto per applicazioni che richiedono una connessione occasionale su breve distanza.

La flessibilità di I2C permette l'aggiunta di dispositivi al bus principale per l'espansione del sistema e lo sviluppo dello stesso.

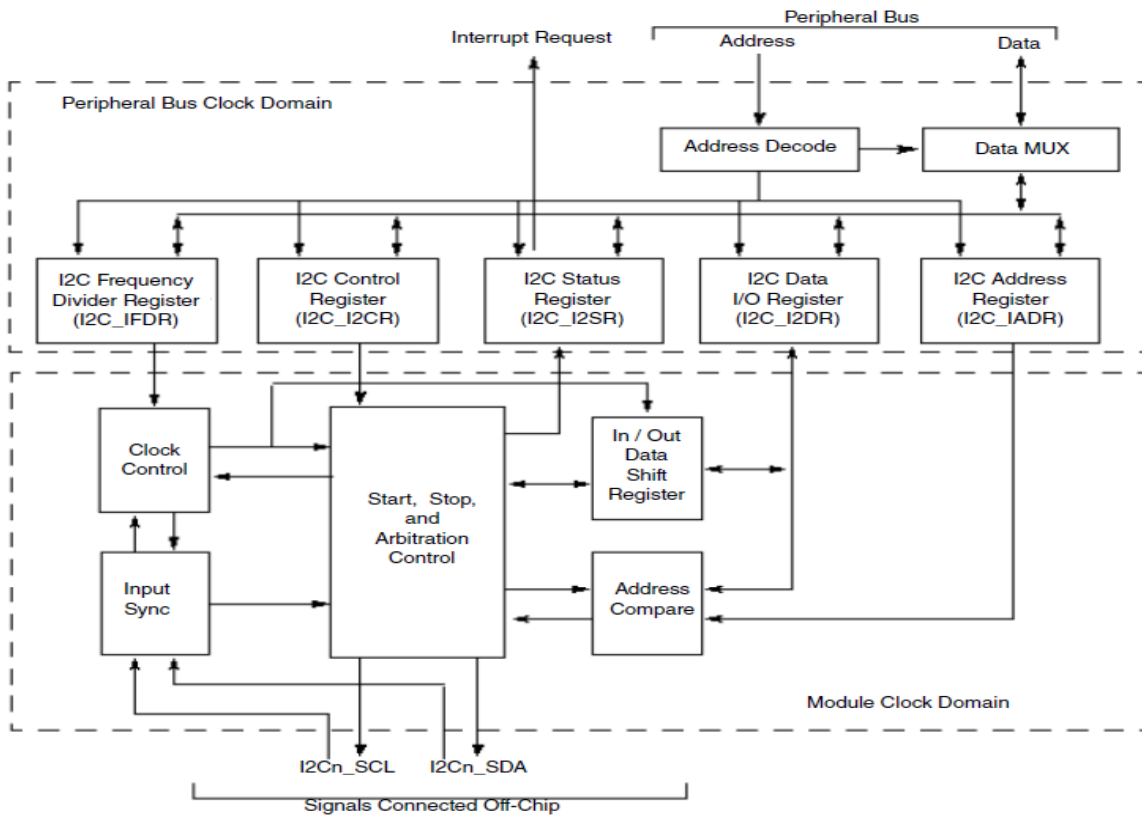
Connessione di dispositivi al bus I2C



La velocità dell'interfaccia I2C dipende dal carico presente nel bus I2C e dalle caratteristiche dello stesso.

Il sistema I2C è un vero e proprio multimaster bus che include la presenza di un arbitro e la presenza di un sistema di detenzione di collisione che previene da corruzione dei dati se più dispositivi provano ad accedere al bus simultaneamente.

Diagramma a Blocchi



Info sul Diagramma a Blocchi

I2C contiene 5 registri a 16 bit:

I2C Address Register

Contiene l'indirizzo dello slave usato dal protocollo I2C. La modalità slave, è la modalità di default per ogni indirizzo presente nel bus.

I2C Frequency Divider Register

Fornisce la possibilità di configurare il clock per la selezione del bit-rate.

I2C Control Register

È usato per abilitare I2C e gli interrupt I2C. Contiene inoltre bits che governano operazioni come slave e come master.

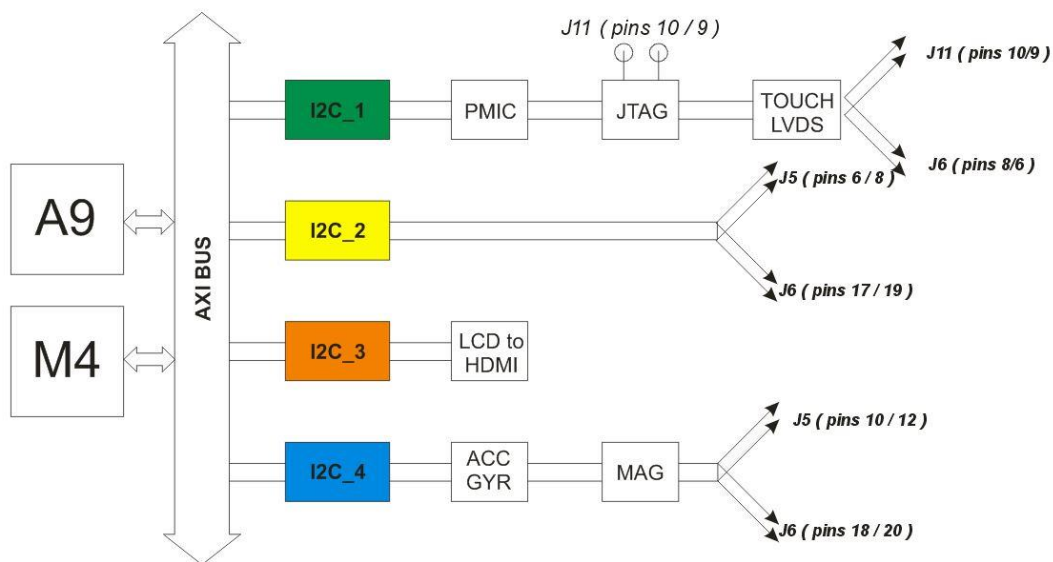
I2C Status Register

Contiene bits che indicano la direzione di transazione e lo stato della stessa.

I2C Data I/O Register

Contiene l'ultimo bit ricevuto o il prossimo bit da trasferire.

Udoo Neo I2Cs



Descrizione di I2C_4:

Questo canale è usato per la connessione del 9-axis motion sensor. Di default è connesso al core A9.

Dispositivo	Funzione	Indirizzo I2C
FXOS8700	Accelerometer / Magnetometer	0x1E
FXAS2100	Gyroscope	0x20