

Sommario

Il presente fascicolo vuole essere un riassunto delle fasi di costruzioni del primo modello di datalogger utilizzato per le imbarcazioni del Mètis Sailing Unipd e dei suoi successivi sviluppi. Vuole essere un riferimento per chiunque abbia intenzione di portare avanti il progetto e lo sviluppo, voglia costruire un nuovo modello o semplicemente voglia imparare ad utilizzare lo strumento utilizzando questo fascicolo come istruzioni per l'uso.

Indice

1	Introduzione	3
2	Obiettivi	5
3	Invio e ricezione dati	6
3.1	FPV Telemetry	7
3.2	Stringhe Disponibili	7
A	I caratteri ASCII	9

Capitolo 1

Introduzione

Questa guida si rivolge sia a chi desidera mettere in funzione DUILIO , sia a chi vuole intraprendere nuovi sviluppi del sistema, per questi ultimi, prima di intraprendere lo studio della presente guida, è fortemente consigliata la lettura di alcune guide di base sull'uso di Arduino, sui protocolli di comunicazione seriale e in particolare l'I²C, lo SPI e la comunicazione seriale UART (sulla quale si deve prestare molta attenzione alla questione byte-ASCII). Infine può tornare utile la conoscenza del protocollo OneWire.

Di fondamentale importanza è avere una basilare esperienza di programmazione e possibilmente un po' di conoscenza del linguaggio C.

Infine è desiderabile, per tutti i lettori, conoscere il modo in cui determinati sensori funzionano, benchè molte feature siano indicate nei capitoli dedicati è sicuramente fondamentale far riferimento ai datasheet per comprenderne meglio le caratteristiche. In particolare è utile conoscere il funzionamento di accelerometri, giroscopi e magnetometri MEMS, come interagire con essi e come si può ricavare l'assetto dai dati 'raw' ottenuti.

Per gli aspiranti alla sola utilizzazione del sistema si consiglia una lettura dei soli capitoli relativi alla messa in funzione dello strumento e all'interpretazione dei dati. Il capitolo su Amphitrite darà le istruzioni sull'interfaccia grafica appositamente studiata per interagire con DUILIO per la lettura dei dati in tempo reale e per l'invio di determinati comandi e settaggi. Questo software, scritto in Python2.7, è in grado anche di leggere i file *.txt salvati su SD dal datalogger.

Per una trattazione sommaria di alcuni argomenti utili si rimanda alle appendici, mentre per quanto concerne lo studio approfondito del filtro utilizzato per ricavare l'assetto si rimanda alle seguenti fonti:

- Sebastian O.H. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Rapp. tecn. Bristol University 2010.

- Sebastian O.H. Madgwick. <http://www.c-io.co.uk/open-source-imu-and-ahrs-algorithms>
- R. Vaidyanathan, Sebastian O.H. Madgwick, A.J.L. Harrison. Estimation of IMU and MARG orientation using a gradient descent algorithm. In: Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on (2011), pp 1-7.

Capitolo 2

Obiettivi

Si è voluto progettare e costruire un apparato elettronico utile per acquisire molteplici informazioni sulle imbarcazioni durante la navigazione. L'utilizzo di questi dati sarà poi da un lato finalizzato al miglioramento tecnico degli equipaggi, dall'altro servirà per affinare le tecniche di progettazione per valutare se effettivamente le barche rispondono secondo quanto voluto in fase di progettazione. Infine, in funzione di un progetto molto più ampio, le rilevazioni potranno essere utili come feedback per l'affinamento delle tecniche di simulazione CFD.

L'interesse nel produrre un sistema a basso costo e completamente customizzabile ci hanno indirizzato verso il mondo Arduino che, almeno in un primo momento, sarà il cervello del nostro apparato. Si prevede in futuro di ampliare il sistema e di conseguenza potrebbe essere utile l'inserimento di sistemi più potenti come ad esempio Raspberry abbinati a microcontrollori eventualmente standalone. La scarsa conoscenza ed esperienza in elettronica e in programmazione ci hanno fatto optare per l'utilizzo di schede preassemblate con una buona quantità di documentazione disponibile

Le funzionalità richieste sono:

- rilevamento dell'assetto;
- rilevamento della posizione;
- rilevamento del vettore velocità (modulo e verso);
- rilevamento delle condizioni ambientali (vettore velocità del vento, temperatura);
- predisposizione per la misura di carichi e deformazioni;
- salvataggio dati;
- invio dati in tempo reale (successiva implementazione);
- schermo per la visualizzazione id informazioni utili;

Capitolo 3

Invio e ricezione dati

La sintassi generale per l'invio e la ricezione di dati, siano essi comandi o dati rilevati, è la seguente:

Codice 3.1: sintassi per lo scambio seriale di informazioni

```
1 $TYPE, val_1, val_2, . . . , val_n*checksum\n
```

dove \$ è il carattere di inizio stringa, TYPE è un byte che identifica il tipo di stringa che stiamo inviando, dopo la virgola invece vengono inseriti i bytes che formano la stringa. La comunicazione seriale spedisce e riceve un byte per volta, i byte possono contenere informazioni numeriche o caratteri, dipende dal modo in cui vengono poi interpretati. Nelle appendici è presente una piccola guida sulla codifica ASCII al cui interno è presente la tabella [A.1] che mostra le conversioni nei vari tipi di codifica utilizzata.

È evidente che il formato utilizzato per inviare i dati sia del tutto simile alle stringhe NMEA, in questo caso però inviamo i byte in formato 'numerico', cioè il dato che ci arriva è pronto all'uso o al massimo va ricomposto con delle operazioni di Bit Shift per aumentarne la precisione. Questo ci permette di aumentare moltissimo le prestazioni, infatti si prenda ad esempio il valore 255, il massimo inviabile con un solo byte, se lo inviassimo sotto forma di caratteri dovremmo inviare tre bytes (uno per ogni carattere che compone il numero), facendolo sotto forma di byte riduciamo il peso dell'invio e aumentiamo la velocità di comunicazione.

Si deve però prestare attenzione al tipo di device che abbiamo di fronte, questo è possibile farlo solo qualora si abbia il controllo su entrambi i lati dello scambio di informazioni. Qualora si abbia a che fare con un GPS, per fare un esempio, dobbiamo uniformarci al suo protocollo che invece invia i caratteri singolarmente e quindi per interpretarli in modo corretto è necessario conoscere la codifica ASCII, un approfondimento è presente nelle appendici della guida.

type byte	command byte	significato
0	0	errore:
0	1	errore:
0	2	errore:
0	3	errore:
0	4	check connessione
0	5	inizia a salvare e inviare dati
0	6	ferma il salvataggio e l'invio di dati

Tabella 3.1: Lista dei comandi attualmente disponibili

3.1 FPV Telemetry

Lo scambio di informazioni in telemetria è reso possibile attraverso il modulo FPV che lavora a $533MHz$. Lato DUILIO è collegato ad una porta seriale (di default alla numero 3), lato computer invece si collega alla porta USB in quanto il modulo di terra è dotato di un convertitore USB-UART. La seriale lavora a $57600baud$.

Proprio per questo modulo abbiamo previsto il sistema di comunicazione indicato nel listato {3.1}.

Al momento il sistema è predisposto per la ricezione di comandi composti da un byte di type e uno di comando, per modificare la lunghezza è sufficiente andare a cambiare il valore della label LENGTH presente nel file Const.h allegato al codice sorgente, il valore lì presente tiene conto solo dei byte di comando e non quelli di tipo.

Per evitare sovraccarichi della seriale in accensione, nel caso di temporanee perdite di segnale o interruzione del software è presente un check nel codice di DUILIO che in caso di silenzio prolungato (60 secondi attualmente, ma si può modificare attraverso la label *CONNECTION_TIMEOUT* In Amphitrite è invece presente una funzione automatica che a intervalli predefiniti (attualmente 60 cicli della funzione che controlla l'orologio, quindi 60s) invia un byte \$0,4 * 4.

3.2 Stringhe Disponibili

Le tabelle seguenti indicano quali sono e cosa contengono le stringhe disponibili. È possibile inviarle sia come byte ascii (generalmente è il metodo usato per salvare su SD), sia come byte numerici (usato per migliorare le prestazioni per l'invio wifi). La differenza tra i due è che nel secondo caso, per migliorare ulteriormente la prestazione, non sono indicati i 'reference' cioè le parti di stringa che indicano l'unità di misura o l'ordine dei dati.

type string	acronimo
\$MVUP	Mètis Vela UniPd
\$MVIC	Mètis Vela Infusion Check

Tabella 3.2: Stringhe ascii disponibili

type byte	stringa di riferimento
1	\$MVUP
10	\$MVIC

Tabella 3.3: Relativi type byte in telemetria

Non è possibile eliminarli dalle stringhe su file di testo per una questione di comprensibilità.

Appendice A

I caratteri ASCII

Decimale	Binario	Esadecimale	Codice Ascii	Decimale	Binario	Esadecimale	Codice Ascii	Decimale	Binario	Esadecimale	Codice Ascii
0	00000000	00	nul	43	00101011	2B	+	86	01010110	56	V
1	00000001	01	soh	44	00101100	2C	,	87	01010111	57	W
2	00000010	02	stx	45	00101101	2D	-	88	01011000	58	X
3	00000011	03	etx	46	00101110	2E	.	89	01011001	59	Y
4	00000100	04	eof	47	00101111	2F	/	90	01011010	5A	Z
5	00000101	05	enq	48	00110000	30	0	91	01011011	5B	[
6	00000110	06	ack	49	00110001	31	1	92	01011100	5C	\
7	00000111	07	bel	50	00110010	32	2	93	01011101	5D]
8	00001000	08	bs	51	00110011	33	3	94	01011110	5E	^
9	00001001	09	ht	52	00110100	34	4	95	01011111	5F	_
10	00001010	0A	lf	53	00110101	35	5	96	01100000	60	`
11	00001011	0B	vt	54	00110110	36	6	97	01100001	61	a
12	00001100	0C	ff	55	00110111	37	7	98	01100010	62	b
13	00001101	0D	cr	56	00111000	38	8	99	01100011	63	c
14	00001110	0E	so	57	00111001	39	9	100	01100100	64	d
15	00001111	0F	si	58	00111010	3A	:	101	01100101	65	e
16	00010000	10	dle	59	00111011	3B	;	102	01100110	66	f
17	00010001	11	dc1	60	00111100	3C	<	103	01100111	67	g
18	00010010	12	dc2	61	00111101	3D	=	104	01101000	68	h
19	00010011	13	dc3	62	00111110	3E	>	105	01101001	69	i
20	00010100	14	dc4	63	00111111	3F	?	106	01101010	6A	j
21	00010101	15	nak	64	01000000	40	@	107	01101011	6B	k
22	00010110	16	syn	65	01000001	41	A	108	01101100	6C	l
23	00010111	17	etb	66	01000010	42	B	109	01101101	6D	m
24	00011000	18	can	67	01000011	43	C	110	01101110	6E	n
25	00011001	19	em	68	01000100	44	D	111	01101111	6F	o
26	00011010	1A	sub	69	01000101	45	E	112	01110000	70	p
27	00011011	1B	esc	70	01000110	46	F	113	01110001	71	q
28	00011100	1C	fs	71	01000111	47	G	114	01110010	72	r
29	00011101	1D	gs	72	01001000	48	H	115	01110011	73	s
30	00011110	1E	rs	73	01001001	49	I	116	01110100	74	t
31	00011111	1F	us	74	01001010	4A	J	117	01110101	75	u
32	00100000	20	spazio	75	01001011	4B	K	118	01110110	76	v
33	00100001	21	!	76	01001100	4C	L	119	01110111	77	w
34	00100010	22	"	77	01001101	4D	M	120	01111000	78	x
35	00100011	23	#	78	01001110	4E	N	121	01111001	79	y
36	00100100	24	\$	79	01001111	4F	O	122	01111010	7A	z
37	00100101	25	%	80	01010000	50	P	123	01111011	7B	{
38	00100110	26	&	81	01010001	51	Q	124	01111100	7C	
39	00100111	27	'	82	01010010	52	R	125	01111101	7D	}
40	00101000	28	(83	01010011	53	S	126	01111110	7E	~
41	00101001	29)	84	01010100	54	T	127	01111111	7F	
42	00101010	2A	*	85	01010101	55	U				

Figura A.1: Elenco dei caratteri Ascii e dei corrispettivi valori Hex, Bin, Dec

Bibliografia

- [1] “A sounding rocket as a test bench for cost effective measurements: Development of a sounding rocket demonstrator test bench for aerospace technologies and atmospheric measurements”. In: *Metrology For Aerospace (MetroAeroSpace)*, *IEEE* (giu. 2015), pp. 574–579.
- [2] Ross Garrett. *Fisica della Vela*. A cura di Zanichelli. 1990.
- [3] Guillerm Lionel Huertz Pierre Emmanuel. “Database building and statistical methods to predict sailing yacht hydrodynamics”. In: *Ocean Engineering* (2014).
- [4] Sebastian O.H. Madgwick. *An efficient orientation filter for inertial and inertial/magnetic sensor arrays*. Rapp. tecn. Bristol University, 2010.
- [5] Sebastian O.H. Madgwick. <http://www.x-io.co.uk/open-source-imu-and-ahrs-algorithms/>.
- [6] A. Mastrangelo. *Disegno e verifica di un datalogger basato su Arduino per piccole imbarcazioni da regata (Skiff)*. 2014.
- [7] R. Vaidyanathan Sebastian O.H. Madgwick A.J.L. Harrison. “Estimation of IMU and MARG orientation using a gradient descent algorithm”. In: *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on* (2011), pp. 1–7.
- [8] Andrea Vallicelli. *TAC 11 Design Nautico, riflessioni tematiche ed esercitazioni progettuali*. A cura di Sala Editori. 2002.
- [9] velarossa. “La barca in "Numeri"”. 2007.

Elenco delle figure

A.1	Elenco dei caratteri Ascii e dei corrispettivi valori Hex, Bin, Dec	10
-----	--	----

Elenco delle tabelle

3.1	Lista dei comandi attualmente disponibili	7
3.2	Stringhe ascii disponibili	8
3.3	Relativi type byte in telemetria	8

Elenco dei codici

3.1	sintassi per lo scambio seriale di informazioni	6
-----	---	---