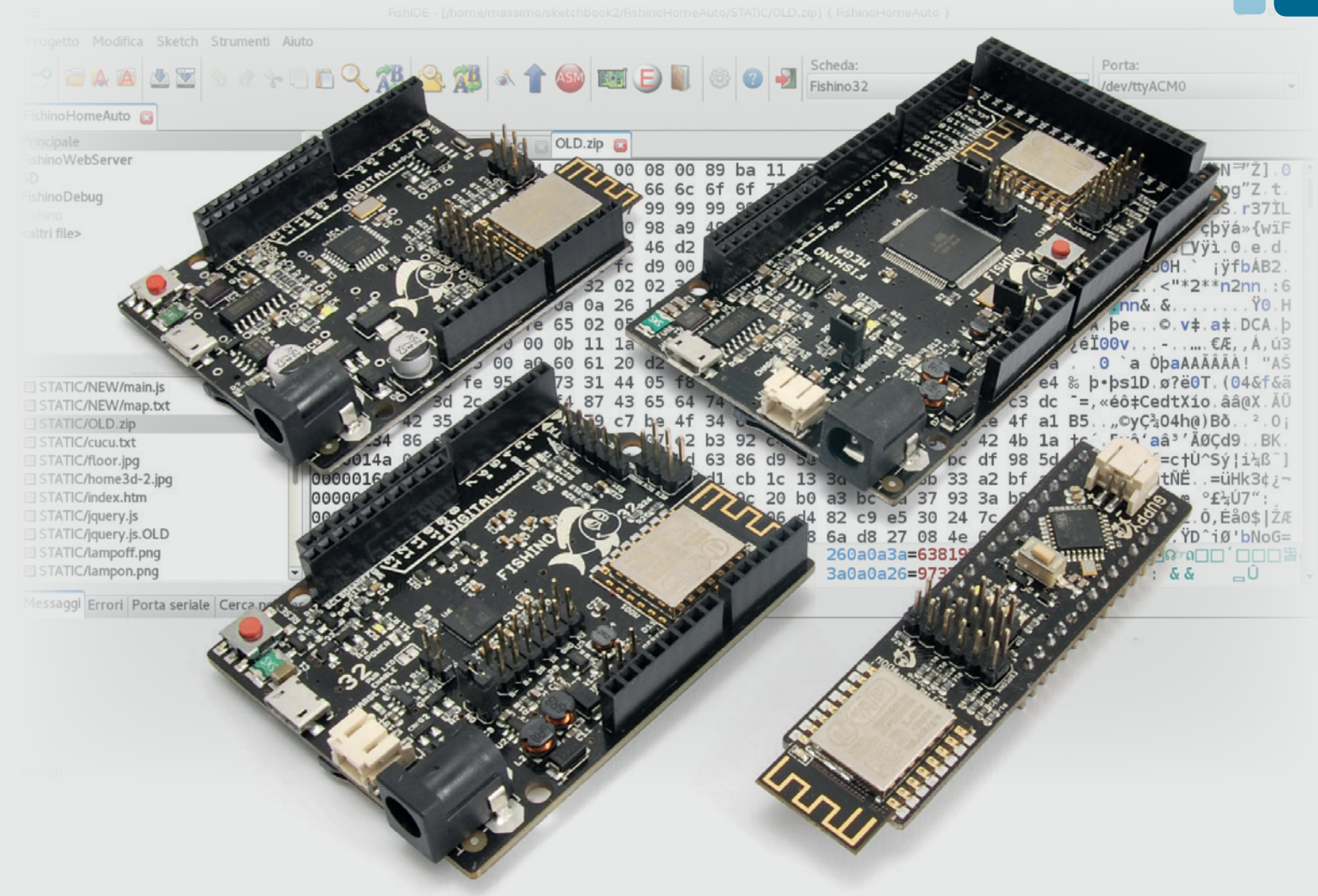

FishIDE

Un nuovo ambiente di sviluppo nato per le nostre schede Fishino, ma adatto anche alle Arduino ufficiali. Prima puntata.

di MASSIMO DEL FEDELE

Durante lo sviluppo delle schede Fishino, soprattutto quelle a 32 bit, ci siamo resi conto di dover scrivere un gran numero di librerie software e di sketch di esempio e di test dedicati. Se per gli sketch semplici l'IDE di Arduino si è rivelato adeguato, per quelli più complessi e soprattutto per lo sviluppo delle librerie ha iniziato a mostrare i propri limiti. L'IDE di Arduino, infatti, pur essendo in grado di lavorare su più file e di aprire più finestre, non è in grado di mostrare la relazione tra lo sketch e le librerie utilizzate; aprendo più finestre, inoltre, durante gli errori di compilazione non è in grado di mostrare gli stessi nel codice sorgente selezionandoli. Senza contare che l'IDE Arduino è un po' "scarno" per chi, come noi, è abituato alle "comodità" dei moderni sistemi di sviluppo, soprattutto in fase di editing del codice, come per esempio la possibilità di commentare blocchi di codice con un solo comando, di spostare l'indentatura, di eseguire ricerche e sostituzioni in file multipli, eccetera. Abbiamo quindi deciso di sviluppare un nostro IDE prendendo come modello il sistema di svilup-

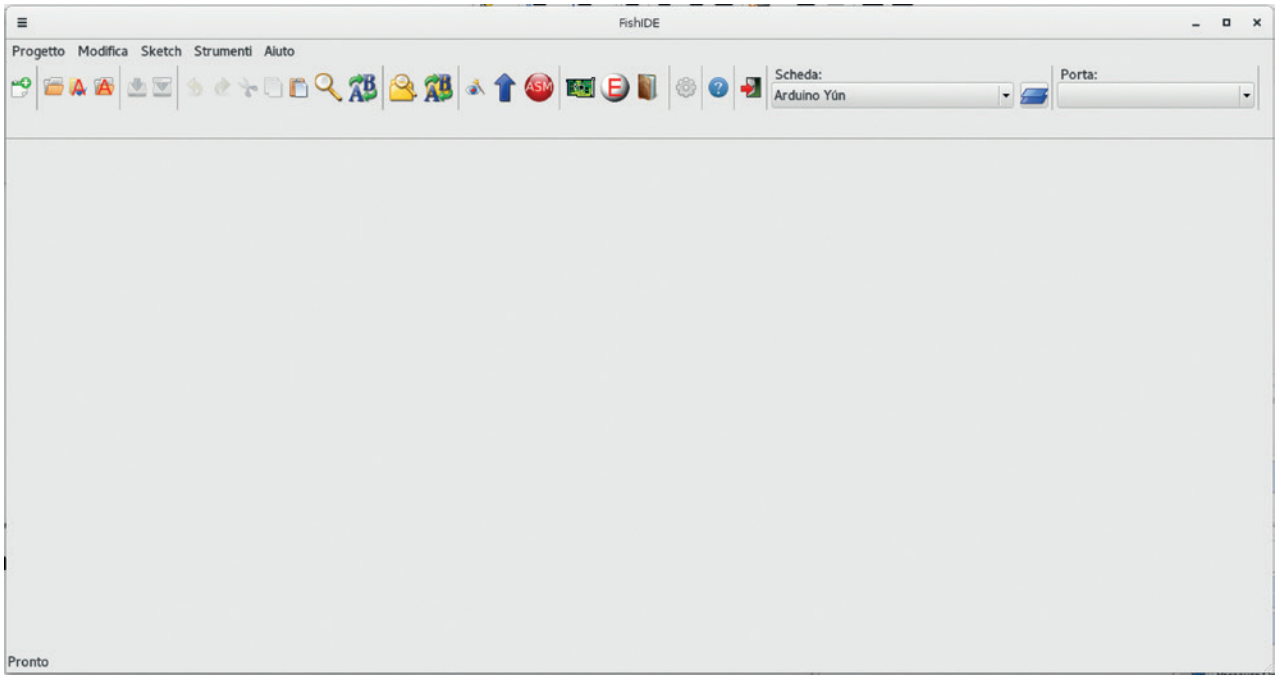

Fig. 1 *- Schermata principale.*

po utilizzato per scrivere programmi in C/C++, un sistema open source disponibile sul sito web *www.ultimatepp.org*. Questo sistema comprende un IDE, una sterminata serie di librerie software per sviluppare efficientemente applicazioni in C/C++ portabili tra Windows e Linux, e molto altro. Tra quello che offre è compreso un ottimo editor per programmi, comprendente tante "comodità" molto apprezzabili durante lo sviluppo. Lo stesso FishIDE è stato sviluppato utilizzando questo sistema. Chiaramente, l'editor è solo il punto di partenza, perché il sistema Arduino è piuttosto differente da un sistema di sviluppo C/C++, soprattutto per le seguenti motivazioni:

- nel C/C++ occorre specificare in qualche modo quali librerie utilizzare; questo si fa solitamente tramite un makefile oppure selezionandole tramite un'interfaccia grafica;
- Arduino, a differenza del C standard, provvede automaticamente ad "agganciarsi" ad alcuni file necessari per la compilazione degli sketch; uno sketch Arduino non ha bisogno, se non utilizza librerie esterne, di direttive #include;
- gli sketch di Arduino hanno un'estensione non standard (.ino) e permettono alcune libertà nella scrittura del codice, che non esistono nel C++ standard.

Abbiamo quindi preso l'editor di cui sopra, l'abbiamo "circondato" da un'interfaccia grafica, aggiungendogli alcuni strumenti, indispensabili e/o soltanto comodi nello sviluppo del software.

INSTALLAZIONE DI FISHIDE

Prima di addentrarci nelle caratteristiche del nostro sistema di sviluppo spendiamo due parole sull'installazione, anche se questo è un termine improprio, visto che l'IDE consiste in un solo file eseguibile compresso (.zip), da posizionare in qualsiasi parte del nostro hard disk.
Non è richiesta alcuna cartella particolare, tutti i file e i programmi necessari verranno scaricati e installati direttamente e automaticamente a seconda delle schede selezionate.
L'IDE è disponibile attualmente per i seguenti sistemi operativi :

- Linux a 32 bit (sviluppata sull'ultimo Ubuntu Linux);
- Linux a 64 bit (sviluppata sull'ultimo Ubuntu Linux);
- Windows a 32 bit (funzionante anche sulle versioni a 64 bit);
- Mac OS/X (versione beta, non nativa, sviluppata su Sierra).

L'ultima versione, quella per Mac, è ancora in beta (al momento della stesura dell'articolo) e presenta alcune problematiche che vedremo in seguito. FishIDE è scaricabile da *www.fishino.it*.
Per utilizzare FishIDE è sufficiente fare doppio clic sul file .zip, decomprimerlo in una cartella a scelta ed eseguirlo.
L'IDE si presenterà all'inizio con una schermata di AutoSetup, che permette di installare in modo automatico i file relativi alle schede di Arduino di base (quelle ad 8 bit) , alcune librerie tra quelle più utilizzate e gli esempi di base.
Non abbiamo volutamente inserito l'installazione automatica

dei package relativi alle schede Fishino, essendo i package piuttosto corposi e, soprattutto, volendo evitare di imporre l'installazione di schede che poi non verranno utilizzate.
Pur essendo possibile saltare questa fase e procedere manualmente in seguito (se si vuole utilizzare l'IDE solo per schede differenti, per esempio, è conveniente), consigliamo di eseguire questa semplicissima operazione in modo da avere il nostro sistema di sviluppo pronto all'uso da subito.
La finestra di dialogo dell' AutoSetup propone la richiesta: "Eseguire il setup automatico?"; cliccando su *Si* inizia il setup.
La procedura durerà alcuni minuti, nei quali appariranno sullo schermo i file in fase di installazione. Al termine apparirà la schermata principale del FishIDE (**Fig. 1**).
Prima di addentrarci nell'utilizzo vediamo subito di completare l'installazione aggiungendo le schede Fishino e le relative librerie; selezioniamo innanzitutto il menu delle impostazioni, che potremo aprire sia selezionandolo dal menu a tendina, alla voce *Strumenti>Impostazioni* oppure, più semplicemente, cliccando sull' "ingranaggio" presente nelle icone degli strumenti; si aprirà la finestra di dialogo proposta dalla **Fig. 2**.
Lasciamo perdere al momento le impostazioni varie che vedremo in dettaglio in seguito e selezioniamo il tab "Percorsi dei file", allorché appare la finestra di dialogo di **Fig. 3**.
Vediamo qui di seguito gli elementi della schermata in dettaglio nei suoi vari elementi.

- Cartella degli sketch: questo è il percorso della cartella principale che conterrà i nostri sketch; possiamo tranquillamente utilizzare la stessa dell'IDE di Arduino, oppure sceglierne una nuova. L'AutoSetup dovrebbe essere in grado di trovare la cartella automaticamente; nel caso in cui il campo fosse vuoto, è possibile selezionarlo manualmente utilizzando il pulsante con la cartellina sulla destra.
- URL per i package: analogamente all'IDE di Arduino è possibile specificare più URL (percorsi internet) contenenti le descrizioni ed i file di schede. Il nostro AutoSetup ha precompilato questo campo con le URL relative alle schede originali Arduino, le schede Fishino ed i moduli WiFi ESP8266 e derivati. È possibile aggiungere un numero illimitato di URL per schede aggiuntive; i percorsi si trovano solitamente sul sito del produttore.
- URL per le librerie : questa è una novità rispetto all'IDE di Arduino, visto che in quest'ultima non è possibile inserire dei percorsi addizionali per le librerie; l'unico percorso,

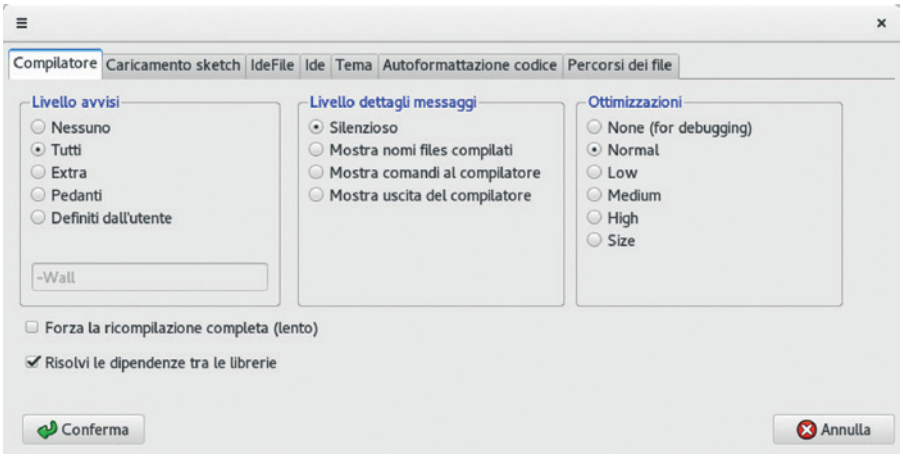

Fig. 2 *- Scheda Compilatore.*

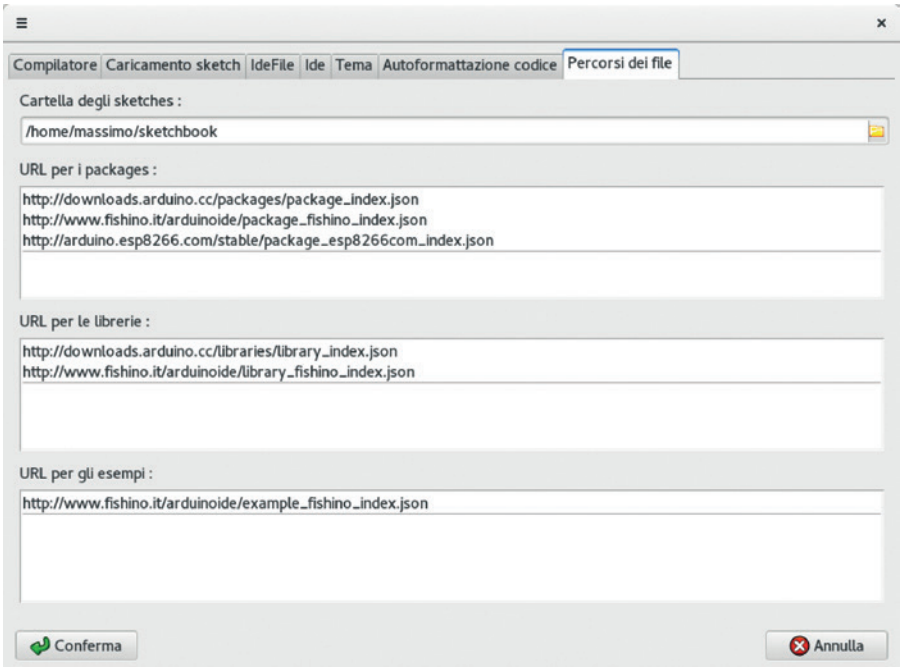

Fig. 3 *- Tab Percorsi dei file.*

che punta al sito di Arduino, è prefissato nel programma. Nella nostra IDE, per contro, è possibile inserire un numero illimitato di percorsi anche se, attualmente, gli unici disponibili sono quello relativo alle librerie "ufficiali" di Arduino e quelle da noi sviluppate per le schede "Fishino". Torneremo comunque in seguito sull'argomento librerie.

- URL per gli esempi: anche qui ci troviamo di fronte ad una novità; nell'IDE di Arduino gli esempi "base" sono fissi e sono contenuti nel pacchetto di installazione. Al contrario, in FishIDE possono essere contenuti in vari server remoti e da questi installati, ottenendo quindi una notevole flessibilità. Attualmente sul sito di Fishino sono contenuti tutti gli esempi base di Arduino e qualche demo per le schede Fishino.

Completiamo la fase di impostazione dei percorsi cliccando sul pulsante "Conferma" e torniamo all'IDE.

INSTALLAZIONE DELLE SCHEDE FISHINO

Come abbiamo accennato, l'AutoSetup provvede ad un'installazione minima contenente il supporto alle sole schede di Arduino, per evitarvi di perdere tempo nel download di librerie e driver di schede che poi potrebbero non essere utilizzate.
Per installare schede aggiuntive utilizziamo il "Gestore di schede", accessibile sia dal consueto menu *Strumenti>Gestore schede* che dall'apposito pulsante della toolbar. Cliccandovi visualizziamo la relativa finestra di dialogo nella quale risulta già installato il primo package: quello corrispondente alle schede Arduino di base, grazie all'AutoSetup visto prima. Utilizzando la scrollbar sulla destra andiamo alla fine della lista (**Fig. 4**).
Qui sono presenti due package: quello per le schede Fishino a 32 bit (fishino pic32) e quello per le Fishino ad 8 bit (fishino avr); sulla destra potete vedere i due pulsanti relativi con la dicitura "installa", mentre sulla sinistra ci sono le caselle di selezione con la versione dei pacchetti (in questo caso, 7.0.0 per entrambi).
Com'è intuibile, è possibile installare una qualsiasi delle versioni presenti; per esempio, se il pacchetto più recente contiene un bug, è possibile retrocedere a quello precedente. Siccome vogliamo installare i pacchetti più recenti, controlliamo nella casella che l'ultima versione sia effettivamente la 7.0.0 e facciamo clic su "installa" per ogni pacchetto.
La prima installazione di un pacchetto può durare parecchi minuti, dovendo scaricare tutti gli strumenti di sviluppo (compilatori, eccetera) che spesso sono piuttosto corposi; solitamente negli aggiornamenti questo non avviene (a meno che non vengano aggiornati anche gli strumenti, cosa che avviene raramente) ed il procedimento risulta velocissimo.
Completata l'installazione la schermata presenterà la scritta "rimuovi" al posto di "installa" sui pulsanti a destra, ad indicare che il pacchetto è installato ed è possibile rimuoverlo. Selezionando invece una versione precedente, la scritta passerà a "retrocedi", mentre per un pacchetto più nuovo sarà "aggiorna", in modo da permetterci un controllo completo di quanto installato.

INSTALLAZIONE DELLE LIBRERIE DI FISHINO

Una volta installato il package con le definizioni delle schede, per le nostre Fishino è indispensabile scaricare anche una serie di librerie software che permettono di gestirne le caratteristiche aggiuntive rispetto alle schede Arduino; per questo utilizzeremo il gestore delle librerie, raggiungibile tramite il menu *Strumenti>Gestore librerie* oppure dall'apposito pulsante della toolbar.
Lo strumento, una volta avviato, si presenta come nella **Fig. 5**; il suo utilizzo è semplicissimo,

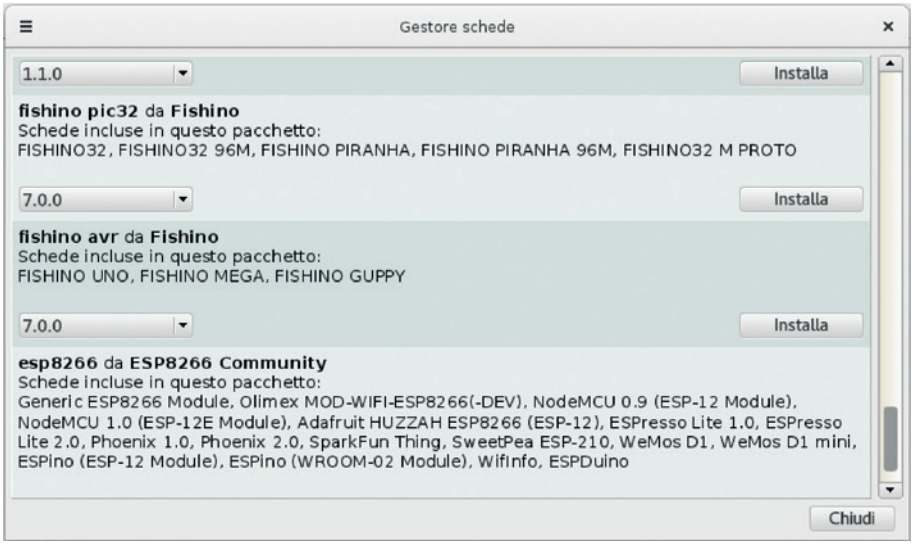

***Fig. 4** - Gestore schede.*

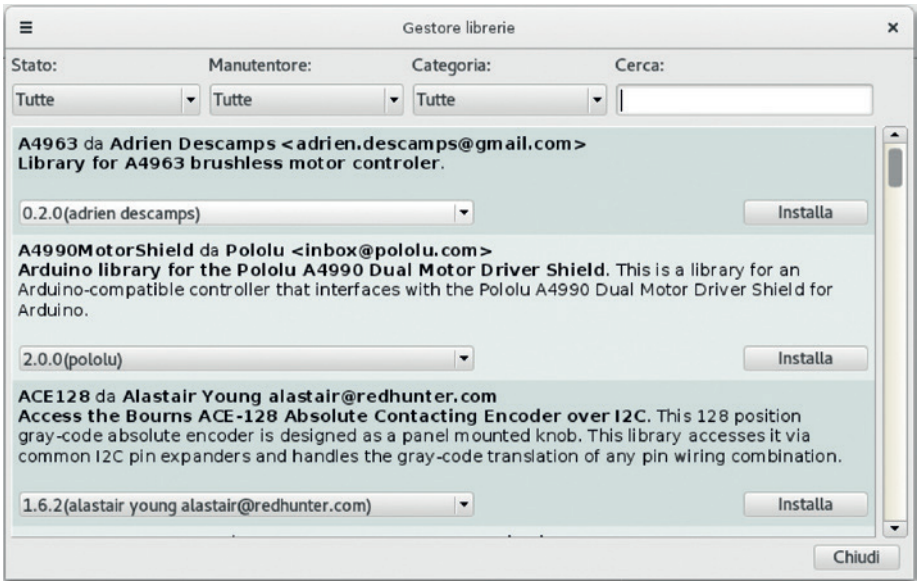

Fig. 5 *- Gestore librerie.*

ma permette di eseguire ricerche e selezioni delle librerie secondo vari criteri. Nella **Fig. 5** potete distinguere i seguenti elementi:

- in alto, tre caselle di selezione (Stato, Manutentore e Categoria) ed una casella di testo dove inserire una parte del nome o della descrizione per restringere la ricerca;
- sotto alla prima riga (analogamente al gestore di pacchetti) l'elenco delle librerie disponibili con le caselle di selezione versione sulla sinistra ed i pulsanti di installazione sulla destra.

Potete tranquillamente provare le varie opzioni e, tramite la barra di scorrimento sulla destra, vedere le librerie che sono state installate automaticamente dall'AutoSetup, che presenteranno il pulsante di destra con la scritta "rimuovi" al posto di "installa". Per installare le librerie di Fishino rimettiamo tutto a posto e, dalla casella Manutentore, selezioniamo *Fishino*, in modo da restringere la visualizzazione alle sole librerie da noi gestite; la schermata apparirà come in **Fig. 6**.
È sufficiente fare clic sul pulsante "installa" per ognuna delle librerie a cui si è interessati.
La libreria indispensabile per il funzionamento del modulo WiFi è la Fishino; le altre sono opzionali ed offrono funzionalità aggiuntive.

COLLOCAZIONE DELLE LIBRERIE

Chi utilizza abitualmente l'IDE di Arduino sa che le librerie vengono inserite in una cartella 'libraries' che si trova all'interno degli sketch. Aprendo però la cartella degli sketch gestita da FishIDE non troverete le librerie installate dal gestore, che vengono caricate in un'altro posto; nella 'libraries' restano tutte le librerie installate "a mano" o gestite dall'IDE di Arduino, che vengono comunque correttamente rilevate ed utilizzate da FishIDE.
Come mai un posizionamento "fuori standard" ? Principalmente per evitare che un utente inesperto, ma anche uno più esperto ma distratto, possa metterci mano per sbaglio. Tramite FishIDE è infatti possibile caricare ed aggiornare le librerie situate su un server remoto solo se queste non vengono "pasticciate" a mano; inoltre, se uno le modificasse a mano (magari per migliorarne qualche caratteristica) le modifiche verrebbero sovrascritte da un eventuale aggiornamento, o addirittura cancellate se disinstallassimo la libreria.
Abbiamo quindi optato per gestire le librerie su tre livelli, ben separati:

- le librerie delle schede, cioè quelle vincolate all'hardware,

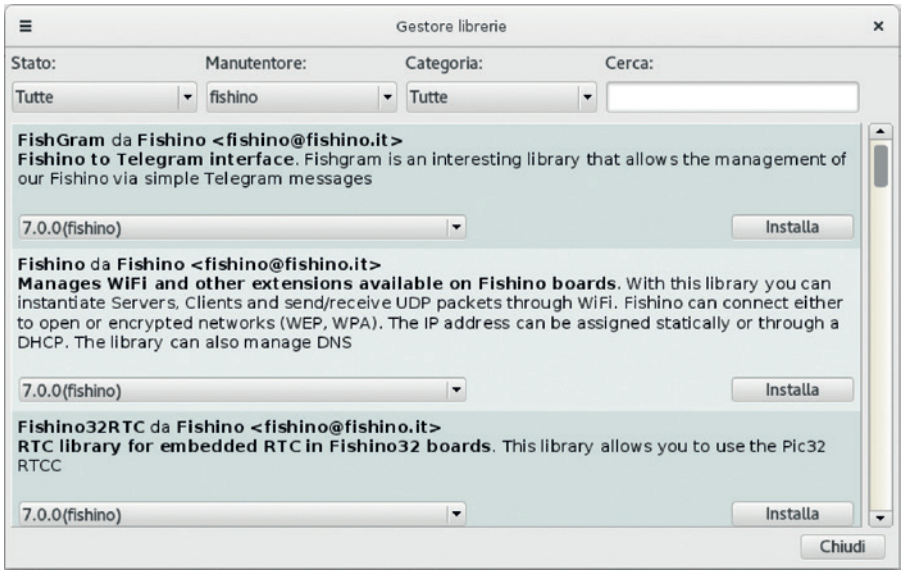

Fig. 6 *- Visualizzazione, in Gestore Librerie, delle sole librerie Fishino.*

che sono contenute nei pacchetti del gestore di schede;
- le librerie "di sistema", cioè quelle gestite dal Gestore Librerie;
- le librerie "utente", che possono essere installate a mano o caricando un file .zip nella solita cartella 'libraries'.

Le prime sono gestite esattamente come fa l'IDE di Arduino; si trovano in una cartella del package per la scheda selezionata e sono "intoccabili", o quasi, come vedremo in seguito.
Le seconde, gestite dal Gestore Librerie, si trovano in un'altra cartella di sistema, ed anch'esse sono quasi "intoccabili".
Le terze, nella cartella "libraries" possono essere "pasticciate" liberamente senza il pericolo di vederle cancellate o sovrascritte dal Gestore Librerie.
Vedremo in seguito, parlando delle impostazioni in dettaglio, che un file delle librerie nei primi due livelli, anche se aperto, **non è modificabile**, quindi apparirà nell'editor del codice in modalità sola lettura, in modo da evitare modifiche non volute, a meno di non abilitare un'apposita casella nelle impostazioni che attiva la scrittura dei file di sistema.
In questo modo potremo comunque vedere e studiare il codice delle librerie, vedere dove si verificano eventuali errori o avvisi di compilazione, ma non potremo fare danni a meno di ...non volerlo davvero!
Ma cosa succede se ci sono due librerie con lo stesso nome in due posti diversi? Come noto, a differenza di un sistema di sviluppo C/C++ standard, Arduino gestisce le librerie in modo quasi automatico; è sufficiente inserire una #include nello sketch ed l'IDE si occupa di trovare la libreria corretta. Ovviamente questo funziona finchè non ci sono 2 librerie simili, ovvero con #include identici. In questo caso, non potendo specificare manualmente quale delle due scegliere, il sistema utilizza una strategia particolare per trovare la libreria "corretta"... che non è detto che sia quella giusta!
In poche parole, ogni libreria contiene un file 'library.properties' che ne specifica alcune caratteristiche, due delle quali sono importanti per il meccanismo di selezione:
- la piattaforma supportata;
- gli include file che definiscono la libreria.

Vediamoli in dettaglio iniziando dal primo: la piattaforma.
La piattaforma definisce la tipologia delle schede; nel caso delle schede Fishino abbiamo due piattaforme definite, 'avr' per le schede ad 8 bit e 'pic32' per quelle a 32 bit. I package di Arduino forniscono la piattaforma 'avr' come base, ed altre, per esempio 'sam' per Arduino Due. Le schede con il modulo ESP8266 forniscono la piattaforma 'esp8266'.
Cosa c'entra con le librerie?
Presto detto! Alcune librerie non vanno a toccare direttamente l'hardware delle schede, ma utilizzano solo i comandi standard di Arduino, per esempio 'pinMode()' e 'digitalWrite()'; non accedono quindi direttamente ai registri del controller e non utilizzano caratteristiche che possono cambiare in base alla piattaforma scelta. In queste librerie, nel file 'library.properties' il parametro 'platform' è indicato con un asterisco, '*', ad indicarne l'universalità.
Altre librerie per contro sono specifiche per alcune piattaforme e non è detto, anzi, è quasi sicuro che non funzionino su altre; in queste il parametro 'platform' indica una o più piattaforme supportate; nel caso della libreria 'Fishino' ad esempio sono supportate le piattaforme 'avr' e 'pic32' ma non 'sam' ed 'esp8266' per le quali la libreria non è adatta.
Tornando al meccanismo di scelta della libreria "giusta", avviene come di seguito:
- viene eseguita una prima scansione per la sola piattaforma della scheda selezionata; ad esempio, se utilizziamo una Fishino32 la ricerca avviene per la piattaforma 'pic32'; vengono SCARTATE tutte le librerie che non dichiarano ESPLICITAMENTE quella piattaforma, anche quelle universali con '*';
- la scansione avviene nelle librerie utente (cartella 'libraries', che ha la precedenza), poi nella cartella delle librerie di sistema (quelle gestite dal Gestore Librerie) e, per ultimo, in quelle della scheda (che ovviamente appartenendo alla piattaforma della scheda la supportano sicuramente);
- alla prima libreria trovata (ovvero alla prima che dichiara un file #include come quello da noi inserito nello sketch), la ricerca si ferma e viene utilizzata quella libreria;
- se non viene trovata nessuna libreria, la ricerca riparte da capo considerando anche le librerie 'universali', cioè quelle che dichiarano '*' come piattaforma, sempre nell'ordine librerie utente-sistema-scheda;
- nel caso anche qui non venga trovato nulla, il sistema tenta l'utilizzo di una libreria specifica per una piattaforma 'sbagliata'; alcune librerie di Arduino, per esempio, dichiarano la piattaforma 'avr' ma in pratica funzionano su molte più schede;
- infine, se anche in quest'ultimo caso non si trova la libreria,

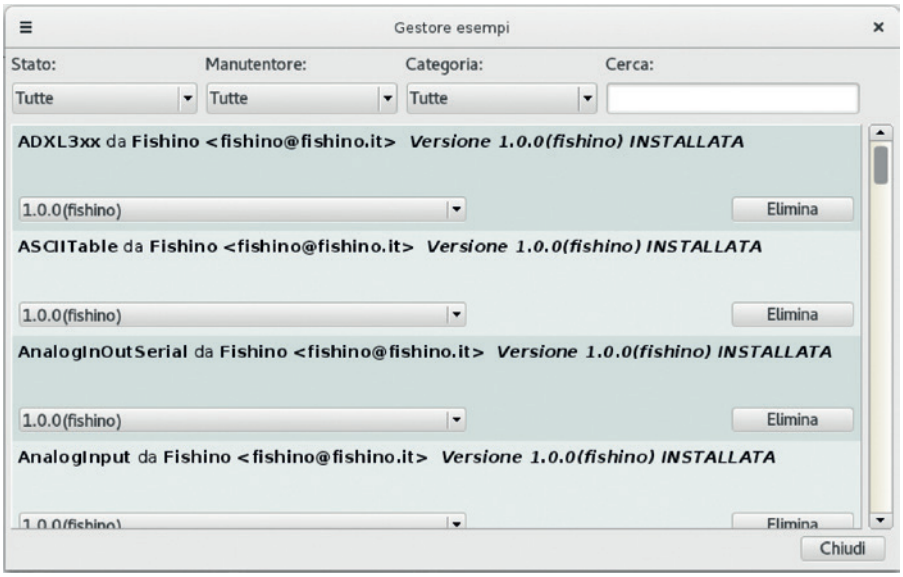

***Fig. 7** - Finestra Gestore esempi.*

viene mostrato un errore di compilazione.

Con questo metodo FishIDE utilizza quindi sempre la libreria che appare più adatta e, nel caso volessimo modificarne una di sistema, ci basterà copiarla nella cartella 'libraries' e lavorare su quella, certi che avrà la precedenza sull'originale.

INSTALLAZIONE DEGLI ESEMPI

Anche gli esempi sono installabili, aggiornabili e rimuovibili tramite un apposito modulo, il Gestore esempi, richiamabile tramite il menu *Strumenti>Gestore esempi*, oppure con l'apposito pulsante della toolbar. Per brevità mostriamo solo la schermata del modulo, il cui funzionamento è assolutamente intuitivo (**Fig. 7**). Come per il Gestore librerie, anche qui disponiamo di alcuni selettori per restringere la visualizzazione agli esempi richiesti, insieme ai consueti selettori di versione e pulsanti di installazione.
Anche gli esempi vengono installati in una cartella di sistema, nascosta all'utente, come le librerie; è comunque possibile sia aprirli e provarli direttamente (si aprono in modalità sola lettura) che importarli nella cartella degli sketch per poter essere modificati, come vedremo a breve parlando della schermata principale di FishIDE.

UTILIZZIAMO FISHIDE

Ed eccoci finalmente arrivati alla parte pratica e quindi all'utilizzo del nostro nuovo IDE: lanciamo FishIDE e, apertasi la finestra di lavoro, osserviamo i pulsanti a sinistra della toolbar (**Fig. 8**). Questi pulsanti corrispondono al menu 'Progetto' e permettono di aprire e salvare sketch, rinominarli ed importare gli esempi.
Che cos'è un 'progetto'? Ebbene, a differenza dell'IDE di Arduino, che si limita a gestire i file in una cartella, il nostro FishIDE utilizza un vero e proprio file di progetto, nella stessa cartella, avente il nome dello sketch e l'estensione '.fish' (abbiamo una fantasia illimitata!!!).
Questo file attualmente è poco utile, ma destinato ad estensioni future che appariranno a breve. In sintesi, raccoglie tutti i file presenti nella cartella, permetterà di "metterli in ordine" di visualizzazione nel FishIDE, di raggrupparli e di nasconderne alcuni che potrebbero essere dei documenti di contorno, quali PDF o altro.

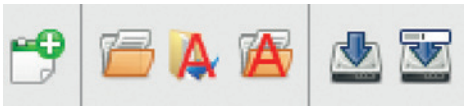

***Fig. 8** - Pulsanti per l'accesso ai comandi del menu Progetto.*

Il file .fish contiene anche l'elenco delle finestre aperte quando chiudiamo FishIDE e le posizioni del cursore in ogni editor; ci permetterà quindi di chiudere uno sketch e ritrovarcelo aperto e con il cursore posizionato allo stesso punto in cui l'avevamo lasciato. Non solo, nel caso avessimo aperto anche altri file, per esempio i sorgenti di una libreria, anche questi verranno ripristinati.
Torniamo ora ai pulsanti della toolbar, le cui funzioni sono le seguenti:

Nuovo progetto; permette di creare uno sketch da zero;

Apri progetto; permette di aprire uno sketch esistente;

Importa esempio; copia uno sketch di esempio nella cartella degli sketch e lo apre;

Apri esempio; apre un esempio in modalità sola lettura senza copiarlo;

Salva sketch; salva lo sketch corrente;

Salva sketch con nome; permette di salvare lo sketch corrente in un'altra cartella.

Per iniziare, utilizziamo il comando '**Importa esempio**', scegliendo un esempio abbastanza complesso che ci permetterà di esaminare tutte le principali caratteristiche di FishIDE; una volta cliccato sul pulsante corrispondente, apparirà un menu da cui impartire il comando '**Esempi delle librerie di sistema**', perché vogliamo usare un esempio della

Fig. 9 *- Finestra aperta dal comando Importa esempio.*

libreria **Fishino**, e quindi l'esempio '**FishinoHomeAuto**'. In **Fig. 9** vedete come apparirà la schermata una volta aperto l'esempio; in essa abbiamo indicato in rosso i vari elementi dell'interfaccia.

- l'**AREA EDITOR** è, chiaramente, l'area in cui verrà scritto il nostro sketch e/o comunque il codice della libreria, o qualunque file di testo.
- l'**AREA LIBRERIE** permette di selezionare le librerie utilizzate (o comunque librerie che si vogliono esaminare, come vedremo in seguito) o i file dello sketch, alla voce 'Principale', la prima della lista. L'ultima voce ('<altri file>') ci permetterà di aprire qualsiasi file esterno al nostro progetto, per esempio il file di un altro sketch da cui vogliamo copiare qualcosa. La cosa importante è che quest'area NON contiene nomi di file ma è una sorta di raggruppamento logico dei medesimi.
- l'**AREA FILE** contiene invece l'elenco dei file appartenenti alla voce selezionata nell' AREA LIBRERIE.
- l'**AREA FINESTRE AUSILIARIE** contiene una serie di tab che ci permetteranno di visualizzare i messaggi di compilazione, il monitor seriale, gli errori ed i file trovati col comando 'Cerca nei file' che vedremo in seguito.
- il **SELETTORE SCHEDA** ci permetterà di selezionare la nostra scheda tra quelle disponibili; nell'esempio abbiamo selezionato una Fishino32.
- il **SELETTORE PORTA** ci permetterà di selezionare la porta a cui è connessa la nostra scheda, tra quelle disponibili; potrà essere anche una porta di rete.
- il **TAB PROGETTO** ci permetterà di passare da un progetto all'altro; FishIDE permette infatti di aprire più progetti nella stessa schermata e di passare facilmente con un click da uno all'altro per potere, ad esempio, copiare spezzoni di codice.
- il **TAB FILE** ci permetterà di passare da un file all'altro tra quelli aperti nel progetto corrente.

All'importazione dell'esempio vediamo aprirsi il **TAB PROGETTO** contenente il nome **FishinoHomeAuto**, che è il nome del nostro sketch ed il **TAB FILE** con il file **FishinoHomeAuto.ino**; nel **AREA LIBRERIE** vediamo, oltre alla voce '**Principale**' (preselezionata) l'elenco delle librerie usate dal progetto, oppure non usate ma aperte manualmente (in grigetto).
Avendo selezionato la voce '**Principale**' nell'AREA LIBRERIE, nell'AREA FILE vedremo l'elenco dei file contenuti nella cartella del nostro sketch.
Per familiarizzare con il nostro IDE, proviamo a scegliere un file (nell' AREA FILE) che sia un po' "strano", ovvero non un file di testo ma un'immagine; per

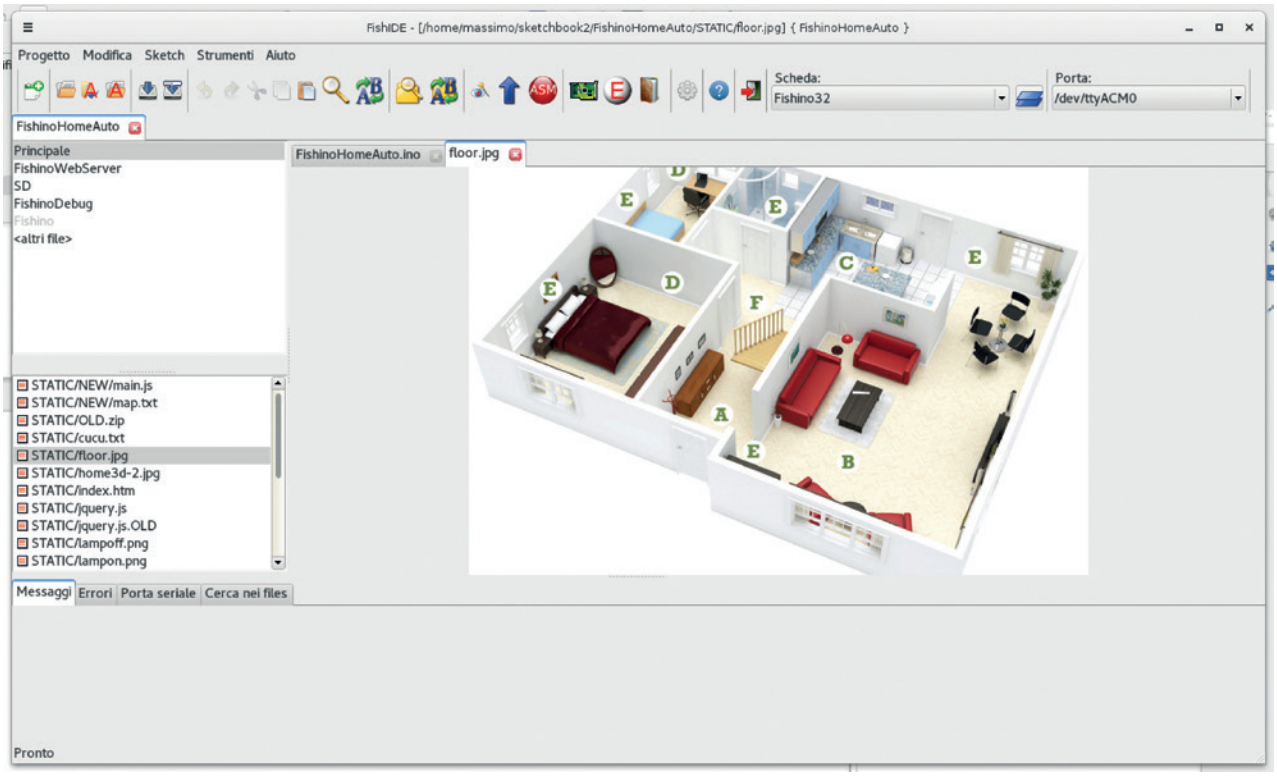

Fig. 10 *- Il file immagine importato in AREA FILE.*

esempio il file STATIC/floor.jpg (**Fig. 10**).
Sorpresa: FishIDE è in grado di visualizzare anche le immagini! Attualmente il supporto per le immagini è piuttosto limitato, non tanto per i formati ma per quello che ci si può fare : l'immagine viene visualizzata così com'è, senza scalarla (quindi potrebbe anche uscire dalla finestra, come parzialmente avviene nel nostro esempio) e senza poterla nè spostare nè modificare. Zoom e pan verranno aggiunti in una versione successiva; non prevediamo di inserire comandi per modificare le immagini, esistendo già molti programmi specializzati che lo farebbero comunque meglio. Si tratta nulla di più di una "comodità" che ci permette di visualizzare un file di immagine.
Tutti i file con formato non riconosciuto, per esempio un file .zip, verranno visualizzati come file esadecimali, anche qui senza possibilità di modifica (**Fig. 11**).
La visualizzazione di file in formato esadecimale risulta comodissima in casi particolari, per esempio per cercare le stringhe di testo in un file binario scaricato da una scheda.
Come avrete sicuramente notato, man mano che apriamo nuovi file cliccandoli nell'AREA FILE appaiono nuovi tab nel TAB FILE (**Fig. 12**). Questi tab permettono di passare da un file all'altro

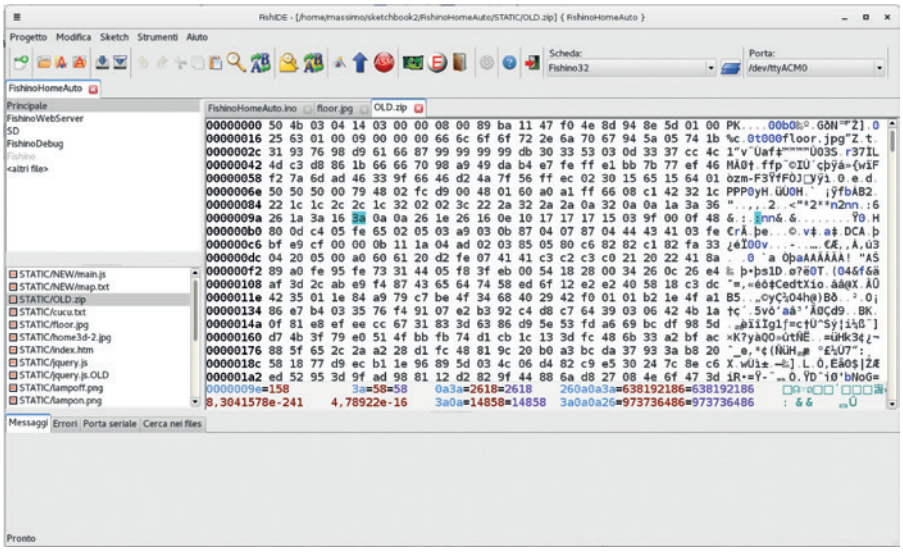

Fig. 11 *- Visualizzazione di file in formato non riconosciuto.*

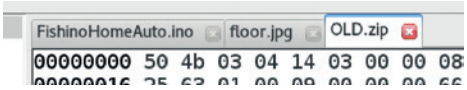

Fig. 12 *- Tab corrispondenti ai file aperti.*

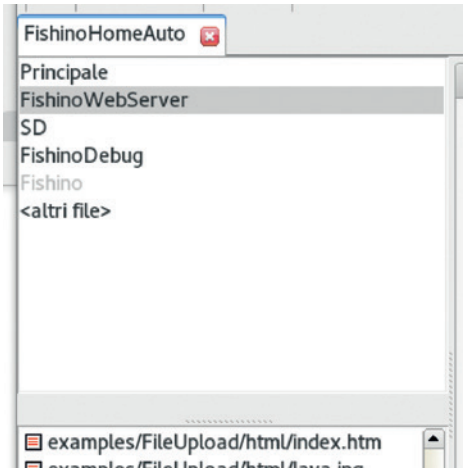

Fig. 13 *- Scelta di Fishino WebServer.*

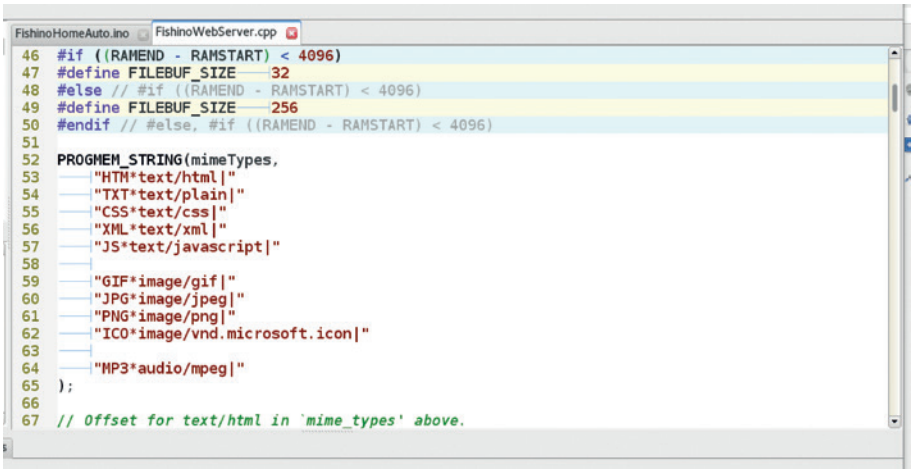

Fig. 14 *- File aperto nell'editor.*

tra quelli aperti, e di chiuderli cliccando sulla crocetta rossa. Chiudiamo quindi i file aperti salvo il primo, il *FishinoHomeAuto.ino*, che è il nostro sketch.
Ora possiamo dare un'occhiata alle librerie utilizzate: nel riquadro AREA LIBRERIE selezioniamo la voce FishinoWebServer (**Fig. 13**). Nel riquadro AREA FILE utilizziamo la scrollbar per scorrere l'elenco fino a visualizzare il file *FishinoWebServer.cpp* e selezioniamolo. Il corrispondente file verrà aperto e visualizzato nell'editor (**Fig. 14**).
Se nelle impostazioni non avete modificato alcunché, vedrete che il file *FishinoHomeAuto.ino*, che abbiamo preso dagli esempi importandolo (quindi creandone una copia nella nostra cartella degli sketch) è modificabile; il file FishinoWebServer, per contro, appartenendo ad una libreria di sistema non è modificabile, quindi non potrete scrivere nella corrispondente finestra dell'editor.
Questo comportamento dell'IDE può essere modificato tramite le impostazioni (**Fig. 15**) alla voce '**Imposta le librerie di sistema a sola lettura**'; deselezionando la casella potrete modificare i file.
All'inizio vi sconsigliamo di abilitare la scrittura, per non vanificare la protezione dei file importanti da parte dell'IDE; in seguito, quando sarete sufficientemente esperti nella gestione, questo vi permetterà di modificare direttamente le librerie senza doverne fare una copia.

COMPILAZIONE E CARICAMENTO DELLO SKETCH

Restando sempre sulla demo FishinoHomeAuto, proviamo ora a compilare lo sketch, ma senza caricarlo sulla scheda; quello che viene chiamato 'verifica sketch' nell'IDE di Arduino.
È sufficiente premere l'icona con l'occhio per avviare la compilazione; nell'AREA FINESTRE AUSILIARIE verrà attivata la finestra Messaggi in cui vedremo scorrere in sequenza tutti i file compilati, fino ad ottenere, in caso di successo, la consueta occupazione di memoria dello sketch (**Fig. 16**).
Nel menu *Impostazioni* esistono varie opzioni per modificare sia i messaggi che appaiono nella schermata proposta nella succitata **Fig. 16**, sia il livello degli avvisi di compilazione, fino ai dettagli più piccoli; le impostazioni standard sono ottimali per non avere un output troppo prolisso e vanno bene nella maggior

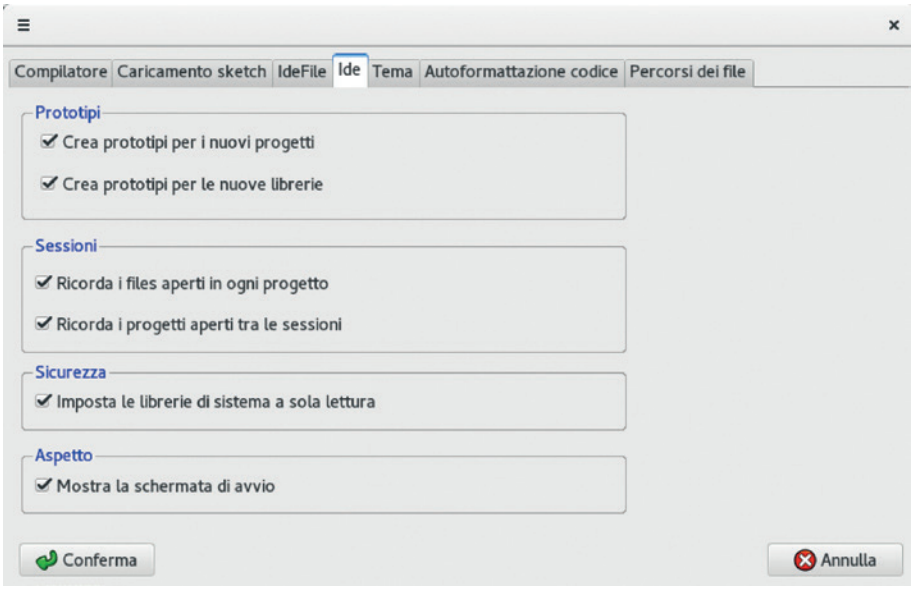

Fig. 15 *- Modifica del comportamento predefinito.*

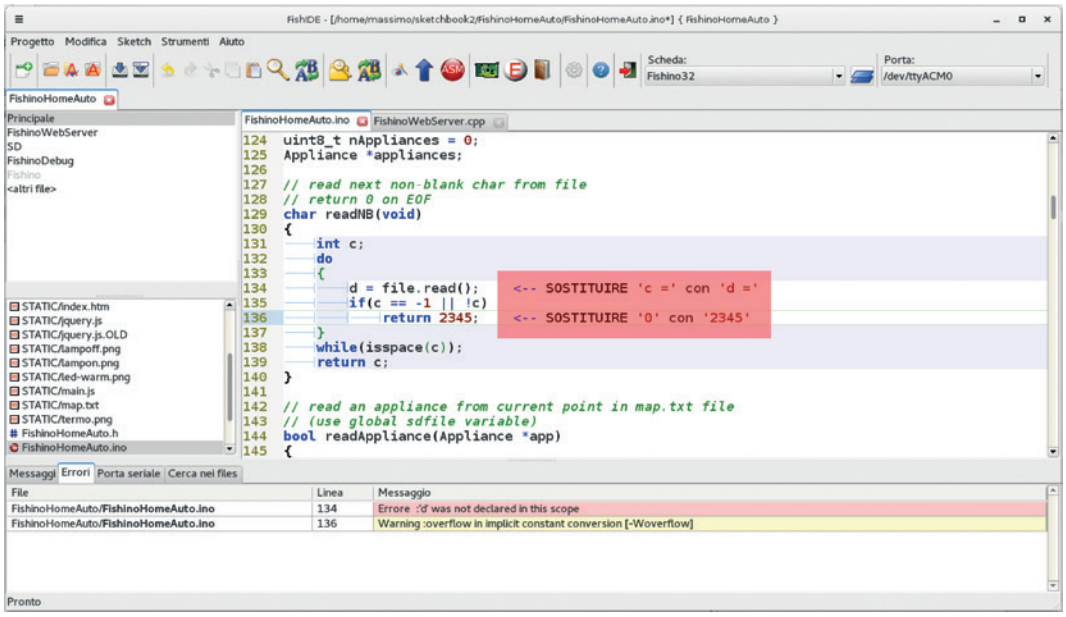

Fig. 17 *- Inserimento volontario di alcuni errori per verificare.*

parte dei casi. Potete comunque sperimentare con le varie opzioni; due di queste, il livello di avvisi e le ottimizzazioni, possono essere determinanti per scoprire errori subdoli nei programmi; anche qui abbiamo scelto per la prima un livello di avvisi molto elevato (decisamente più elevato di quello dell' IDE di Arduino, infatti in molte librerie scaricate vedrete apparire parecchi Warning), mentre le ottimizzazioni sono state impostate in modo da ridurre al minimo la dimensione dello sketch, come per l'IDE di Arduino. Solitamente, salvo per sketch molto piccoli, l'opzione senza ottimizzazione, che sarebbe perfetta per il debugging, non è applicabile poichè il file binario risulta troppo grosso e non risulta caricabile dal controller.

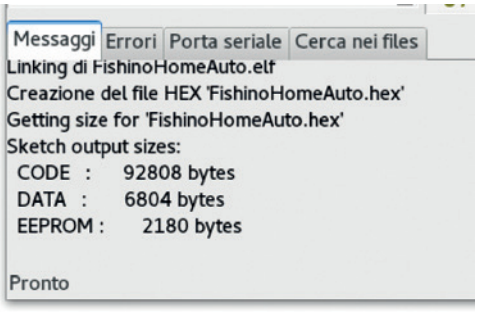

Fig. 16 *- Finestra Messaggi.*

Per caricare lo sketch è sufficiente ora cliccare sul pulsante con la freccia blu e, se avrete precedentemente selezionato scheda e porta corretti, verrà avviato il caricamento dello sketch.

IN CASO DI ERRORI

Una delle cose più interessanti del FishIDE è l'immediatezza nell'identificazione degli errori di compilazione; per vedere come funziona questa caratteristica, riposizioniamoci sullo sketch *FishinoHomeAuto.ino* facendo clic sul relativo tab in alto ed inseriamo volutamente un errore di sintassi ed uno che normalmente "scapperebbe" al controllo se operassimo con un livello di avvisi più basso. Con riferimento alla **Fig. 17**, eseguiamo le modifiche riportate nel riquadro in rosso e rilanciamo la compilazione facendo clic sull'icona con il simbolo dell'occhio; una volta terminata, il pannello inferiore si posizionerà sulla finestra degli errori/avvisi, mostrandone l'elenco come in **Fig. 17**: in rosso saranno evidenziati gli errori e in giallino gli avvisi. Cliccando sulla linea dell'errore/avviso, il cursore si posizionerà nell'editor in corrispondenza dello stesso, permettendovi di correggere facilmente il codice.

In caso abbiate a che fare con uno sketch con più file o un errore in una libreria, cliccando sull'errore verrà aperto in automatico il file corrispondente, se non è quello già visualizzato, ed il cursore si posizionerà sempre nella giusta posizione. Ricorreggiamo ora i due punti e rilanciamo la compilazione, che, se abbiamo fatto le cose per bene, avviene senza intoppi.

CONCLUSIONI

Bene, con questo abbiamo concluso anche il discorso sulla compilazione e con esso la prima parte della descrizione del FishIDE, l'ambiente di sviluppo da noi creato per le piattaforme Fishino, che contrariamente all'IDE Arduino nasce predisposto per supportare la famiglia Fishino. In questa prima puntata ci siamo soffermati sulle caratteristiche dell'IDE, evidenziando le differenze con quello di Arduino; inoltre abbiamo descritto installazione ed utilizzo dell'ambiente, che proseguiremo nella prossima puntata, descrivendo il preziosissimo editor di FishIDE.