Il progetto

Sketch

Sketch è il nome che l'ambiente di sviluppo Arduino utilizza per indicare un programma. Uno sketch è l'unità di codice che viene caricato ed eseguito su una scheda Arduino.

Commenti

Nel programma si mettono dei commenti per spiegare il significato ed il funzionamento del programma.

```
/*
 * Lampeggio
 *
 * Questo è l'esempio base di Arduino. Si accende un LED per un secondo
 * poi viene spento per un secondo, e così via ...
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */
```

Tutto il testo compreso tra /* e */ è ignorato dal compilatore quando viene compilato il programma e quindi non viene eseguito dal programma sulla scheda.
Gli altri asterischi all'inizio della riga non sono indispensabile e sono presenti solo per motivi estetici
Questo testo viene inserito per spiegare a chi legge il codice sorgente cosa fa il programma e come funziona.
E' una buona abitudine commentare i programmi e mantenere i commenti aggiornatic quando si modifica il codice.
Esiste anche un altro stile per inserire commenti brevi su una sola riga.
I commenti brevi iniziano con // e proseguono fino alla fine della riga.
Ad esempio:

```
int ledPin = 13; // LED connesso al piedino digitale 13
```

il testo “LED connesso al piedino digitale 13” è un commento.

Variabili

Una variabile è un'area di memoria per la memorizzazione di dati. Una variabile ha un nome, un tipo, e un valore.
Ad esempio, l'istruzione del paragrafo precedente dichiara una variabile con il nome ledPin, il tipo intero, e un valore iniziale di 13.
La variabile è utilizzata per indicare a quale pin digitale della scheda Arduino è collegato il LED. Ogni volta che il nome “ledPin” nome appare nel codice, verrà utilizzato il valore contenuto nella variabile. In questo caso, la di non creare la variabile ledPin e usare direttamente il numero 13 nelle parti del programma in cui si deve specificare il numero di pin. Il vantaggio di utilizzare una variabile è che è più facile spostare il programma su un pin diverso: si deve solo modificare la riga che assegna il valore iniziale alla variabile.
Spesso, tuttavia, il valore di una variabile cambia mentre il programma viene eseguito. Ad esempio, è possibile memorizzare il valore letto da un ingresso in una variabile.

Funzioni

Una funzione è una porzione di codice identificato da un nome che può essere utilizzato da un programma.
Ad esempio un progetto per Arduino deve contenere una funzione denominata setup().
Questa funzione viene eseguita dal bootloader al reset della scheda.
Nell'esempio seguente il pin digitale il cui numero è contenuto nella variabile ledPin viene configurato come uscita:

?

```
void setup(){
 pinMode (ledPin, OUTPUT); // imposta il pin digitale come uscita
}
```

La prima riga fornisce informazioni sulla funzione, come il suo nome, "setup". Il testo prima e dopo il nome specificano il tipo restituito e parametri. Il codice tra le {} è chiamato il corpo della funzione: sono le istruzioni eseguite dalla funzione.

È possibile chiamare una funzione già definita sia nel progetto stesso sia come parte del linguaggio di Arduino. Ad esempio, l'istruzione pinMode(ledPin, OUTPUT); chiam la funzione pinMode() predefinita nel linguaggio di Arduino passando i due parametri ledPin e OUTPUT. Questi parametri sono utilizzati dalla pinMode() per decidere quale pin configurare ed in che modo.

La funzione pinMode() può configurare un pin sia come ingresso sia come uscita. Quando configurato come ingresso, un pin è in grado di rilevare lo stato di un sensore come ad esempio un pulsante. Se configurato come uscita, il pin può pilotare un attuatore come ad esempio un LED.

Per modificare lo stato di un pin di uscita si usa la funzione digitalWrite().
Ad esempio, l'istruzione:

```
digitalWrite(ledPin, HIGH);
```

imposta il pin identificato dal valore di ledPin (pin 13) al valore alto (5 volt). Se si scrive LOW invece il pin assume il valore basso (0 volt).

La funzione delay() determina una attesa di una durata che dipende dal numero passato come parametro specificato in millesimi di secondo. Quindi:

```
delay(1000);
```

determina una attesa di un secondo.

setup () e loop ()

Ci sono due funzioni speciali che sono parte di ogni programma per Arduino: setup() e loop().
La funzione setup() viene chiamato una sola volta, quando il programma si avvia. In questa funzione vanno messe le operazioni di inizializzazione come le configurazioni dei pin di ingresso ed uscita e le inizializzazione delle librerie
La funzione loop () viene chiamata ripetutamente alla massima velocità consentita dal processore. Tale velocità dipende dal tempo impiegato ad eseguire un singolo loop e quindi in sostanza dalla lunghezza del codice della funzione loop().
E' sempre necessario inserire nel progetto le funzione setup() e loop() anche se sono vuote.

Ultime modifiche: lunedì, 9 gennaio 2017, 10:41