Programmare

I microcontrollori e i microprocessori sono elementi programmabili. Possono lavorare autonomamente se opportunamente programmati. I microcontrollori hanno architettura RISC mentre i microprocessori sono di tipo CISC.

* Architettura RISC: Reducted Instruction Set Computing; architettura ridotta. I microcontrollori eseguono piccoli programmi e uno alla volta
* Architettura CISC: Complex Instruction Set Computing; architettura complessa. I microprocessori eseguono programmi complessi e molti insieme.

Per poter eseguire comandi di acquisizione dati, controllo di variabili, comandi vari, c’è bisogno di piccoli programmi che un microcontrollore con architettura ridotta può sostenere.

Cosa significa programmare? Significa scrivere in un file di testo un elenco di comandi opportunamente scritti con una determinata sintassi a seconda del linguaggio

Una volta scritto, il programma va compilato per constatare se ci sono errori e poi trasformato in un file comprensibile al microcontrollore o microprocessore. Il file in formato testo è detto sorgente, quello compilato è l’eseguibile.

Compilatore

File eseguibile o in formato binario

File sorgente in formato testo

Un linguaggio può essere:

* Ad alto livello, se il linguaggio è molto comprensibile ad una persona
* A basso livello, se il linguaggio è vicino a quello del micro.

Una ulteriore classificazione dei linguaggi ad alto livello è il seguente:

* Linguaggi compilati: quelli che una volta processati, restituiscono un file eseguibile con estensione.exe (es. fortran, C/C++..)
* Linguaggi interpreti: sono interpretati. Vengono sempre processati ma hanno bisogno dell’ide per poterli eseguire (es. Python, basic..)

Ide, Tolchain e interpreti

Ide = Integrated Development Environment: ambiente di sviluppo integrato, suite software progettato per la realizzazione di applicazioni che aggrega strumenti di sviluppo per creare e testare il programma. Consiste quindi di strumenti editor di codice sorgente, strumenti di automazione e debugger. La maggior parte degli IDE ha compilatori e interpreti

Interprete = programma che permette di testare il file testuale creato per un determinato linguaggio; trasforma il file in eseguibile. Il compilatore è uno strumento molto più semplice di un IDE, non fornisce gli stessi strumenti dell’IDE

Tolchain = Insieme di programmi che concorrono nella realizzazione di un prodotto finale.

Sintassi C++

Il C++ è un linguaggio ad alto livello; è case sensitive nel senso che è sensibile alle lettere maiuscole e alle lettere minuscole. Ogni programma scritto in c++ ha estensione cpp ed è necessario di una libreria base iostream.h. Una libreria è un file testuale con estensione .h (header) che serve a rendere più semplice la sintassi. Ogni programma in C++ ha un template

***#include <iostream>***

***using namespace std;***

***int main(){***

***}***

Tra le parentesi { } va scritto il codice programma

Importante ricordare:

* I commenti non vanno processati e possono essere di due tipi:
	+ Di riga. Hanno inizio con // e non vanno chiusi. Devono essere scritti su una sola riga dove c’è //
	+ Su più righe. Hanno inizio e fine e possono essere scritti su più righe. Iniziano con /\*

Finiscono con \*/

* Ogni linea comando termina con ;

Sintassi per scrivere sul monitor

***#include <iostream>***

***using namespace std;***

***int main(){***

***cout<<”Ciao”;***

***}***

cin<< è il comando utile per scrivere sul monitor

* Per poter scrivere una frase, bisogna includere la scritta tra le virgolette “ “
* Andare daccapo \n ( cout<<”\nciao”;)
* Saltare più righe \n\n\n (cout<<”\n\n\nciao”;)
* Lasciare lo spazio di un tab \t (cou<<”\tciao”;)

***#include <iostream>***

***using namespace std;***

***int main(){***

***cout<<”\n\tCiao”; //scrive Ciao alla seconda riga del prompt dei programmi e uno spazio di tab***

***}***

Ripetere più volte un comando

Per poter ripetere più volte un comando, basta scriverlo più volte. La sintassi for, permette di ripetere più volte un comando senza doverlo scrivere.

La sintassi è la seguente: ***for(i=0;i<n; i++){ }***

La spiegazione è il seguente: per i che va da zero e, fino a che non arriva a n, deve essere incrementato di una unità.

Ogni volta che i incrementa di una unità, vengono eseguiti i comandi scritti nelle { }

***#include <iostream>***

***using namespace std;***

***int main(){***

***int i; //la variabile i deve essere sempre dichiarata ed è un intero***

***for(i=0;i<100;i++){ //I parte da 0 fino a 100 e la frase “ciao” viene ripetuto 100 volte***

***cout<<”Ciao”;}***

***}***

La variabile intera i può essere inizializzata da un numero diverso da zero

***#include <iostream>***

***using namespace std;***

***int main(){***

***int i; //la variabile i deve essere sempre dichiarata ed è un intero***

***for(i=10;i<100;i++){ //I parte da 10 fino a 100 e la frase “ciao” viene ripetuto 100-10 volte***

***cout<<”Ciao”;}***

***}***

Scrivere frasi e valori di variabili

Come già visto nelle righe precedenti, una frase viene scritta tra “ “ ; il valore di una variabile è scritta nel seguente modo cout<<nome\_variabile

***#include <iostream>***

***using namespace std;***

***int main(){***

***int i; //la variabile i deve essere sempre dichiarata ed è un intero***

***i=10;***

***cout<<”i=”; //compare i=***

***cout<<i; //compare il valore della variabile, quindi 10***

***}***

***#include <iostream>***

***using namespace std;***

***int main(){***

***int i; //la variabile i deve essere sempre dichiarata ed è un intero***

***for(i=0;i<100;i++){ //I parte da 0 fino a 100 e “ciao” viene ripetuto 100 volte***

***cin<<”Ciao”<<”\t”<<i<<”\n”; //la parola ciao viene scritta 100 volte insieme al valore di i che incrementa***

***}}***