MIKROELEKTRONIKA

Esistono in commercio tanti ambienti di sviluppo per microcontrollori, una delle ditte che secondo me offre prodotti con un buon rapporto qualità/prezzo è la Mikroelektronika, <u>www.mikroe.com</u>.

Nel suo sito vengono venduti una miriade di prodotti e schede elettroniche, come anche i propri ambienti di sviluppo per microcontrollori della Microchip, della AVR o per microcontrollori in tecnologia ARM.

Vengono offerti compilatori in linguaggio C, Basic e Pascal, con tutte le librerie di supporto dotate di documentazione ed esempi. I compilatori sono liberamente scaricabili dal loro sito e funzionano senza alcuna limitazione per programmi il cui codice hex sia inferiore a 2KB.

Come punto di partenza possiamo approfondire il compilatore MikroC per microcontrollori PIC ad 8 bit,.

All'avvio del programma troviamo la seguente schermata.

E mikroC P	RO for PIC v.7.2.0			
<u>File</u> <u>E</u> dit	View Project Build Run Tools	Help		
1 🖪 💽 - I	19 14 19 💷 🗅 🗗 🗗 🕄	👌 🗛 🖨 🖨 😓 🗄	🔧 🔈 📚 😓 🖳 🧎 🖽 🔐 🕌 🔐 🍕 Debug layout 💿 🕞 😱 🐻 🗐 🖉 🔟 📖 🤌 沙 🕢 (🖕 🖷	
💮 Start Page	2			F 🔜 🗉
Nev	v Project	Get Started Lat	test News	Project
<u>ре</u>	en Project	Welcome Links		Manage
	an Evamples Folder		Welcome to mikroC PRO for PIC	
	in examples rotacini		mkroc PRO for PIC Heb mkroc PRO for PIC web page	Library
Recent Pro	ojects:			Manag
		Circles Manage & Ho Rep Fight	Get started with mikroC PRO for PIC	<u></u>
		BUF	Quick start guide	
		Line and Line	Fear Link of the Hillion of	
		Cude to Key File License	Register your mikroC PRO for PIC Learn how to register your mikroC PRO for PIC Guide to key file licence	
V Show Pa	age On Startup			
Messages	Quick Converter			
Errors	Varnings V Hints			
Line	Message No.	Message Text	Unit	
				-

Il primo passo sarà quello di creare un nuovo programma con "**NEW PROJECT**" e successivamente selezionare **STANDALONE PROJECT**.

C mikroC PRO for PIC v.7.2.0				
<u>File Edit View Project Build</u>	<u>R</u> un <u>T</u> ools <u>H</u> elp			
	669 10 10 10		🕻 🖪 🗃 🕺 Debug layout	
💮 Start Page				
New Project	New Project Wizard			23
	Steps:	Project type:		
Open Project Open Examples Folde Recent Projects:	1. Project type 2. Project settings 3. Add files	Standard project Visual TFT project	Create plain project	
		Select project type that best sutis your ne	eds. Ωε	ancel
V Show Page On Startup				

Consiglio di iniziare a realizzare un primo progetto con il microcontrollore **PIC16F88**, che andrà selezionato nelle schermate successive.

Image: Construction of the second	Run Iools <u>H</u> elp	02	🗞 🗞 🗄 💻 🍞 🖪 🖻 者 🦂 🕴 Debug lay	out 🔻 🖬 💽
💮 Start Page				
New Project	New Project Wizard	Project Settings	5	× 1
Open Project	1. Project type 2. Project settings 3. Add files	Project Name: Project folder:	prova C:\Users\Daniele\Desktop\mikroc\	Browse
Recent Projects:		<u>D</u> evice name: Device clock	P16F88 8.000000 MHz	•
		Open Edit Project of Enter project name Checking 'Open Ed This enables you to Note: Project na	vindow to set Configuration bits , project folder, select device name and enter a device dock t Project' option will open 'Edit Project' window after closing to easily setup your device and project. me and project folder must not be left empty.	(for example: 80.000). nis wizard.
		Back	Next 🌳	Cancel

Terminiamo l'avvio del progetto con il pulsante FINISH

C mikroC PRO for PIC v.7.2.0			
<u>File Edit View Project Build</u>	<u>R</u> un <u>T</u> ools <u>H</u> elp		
1 🔁 😼 - 19 🖄 😼 🛛 🗋	6 6 🗐 🗄 🗠 🖯	🗋 🔎 😓 🗄 🎭 🎭 🗞 🎚 🖳 🦹 🖊 🕞 🔐 🗍 Debug layout	- 🖬 😡
🕜 Start Page			
New Project	New Project Wizard		X
-	Steps:	Select existing files you want to add to project	
Open Project	1. Project type 2. Project settings	Add File To Project:	<u>A</u> dd
Open Examples Folde	3. Add files	File Name	Remove
Recent Projects:			Re <u>m</u> ove All
			ancel
l			

Al termine della creazione del nuovo progetto verrà automaticamente creato un file con il nome del progetto e con estensione .C.

Si possono creare ed aggiungere altri file tramite il **PROJECT MANAGER** posizionato a destra.



Sempre a destra troviamo il **LIBRARY MANAGER** dove sono elencate tutte le librerie disponibili, che vanno selezionate a seconda delle necessità.

C mikroC PRC) for PIC v.7.2.0-C:\Users\Daniele\Desktop\mikro	roc\prova.mcppi	
<u>File Edit Vi</u>	ew <u>P</u> roject <u>B</u> uild <u>R</u> un <u>T</u> ools <u>H</u> elp		
18 8- 6	1 🔀 📴 🖴 🗋 🗗 🗗 🛃 🛛 🕭 - 🔒	🗣 🙆 😓 🐁 🔈 🎭 😓 🖳 🧎 🖊 🕞 🔐 🍂 Debug layout 💿 🔽 😡 🐻 🖉 🖉 🖉	
Start Page	🔀 📄 prova.c 🔀	III Library Manager	-ta 🔛 📘
1 🖓 🗸	oid main() {		Pro
		Search Library Manager	.* Q
		Search Library Manager	t: Manager - prova.mcppi Library Manager
		Out_wire Out_transfer	
III Messages	Quick Converter	⊕	
V Errors	✓ Warnings	B. D. PWM	
Line	Message No.	Message Text	
		B Software_I2C	
		B Sound	
		B SPI	-

Tramite il menù PROJECT in alto ed EDIT PROJECT presente nella finestra, possiamo accedere alla configurazione del nostro microcontrollore, parte questa molto importante ai fini della funzionalità del nostro programma.

C mikroC PRO for PIC v.7.2.0-C:\Users	;\Daniele\Desktop\mikroc\prova.mcppi								
<u>File Edit View Project Build E</u>	<u>un T</u> ools <u>H</u> elp								
B B - P K 📴 🖬 D G	i 🛃 🗄 📴 🕞 🖨 🖓 🍰	2		1	A 🖻 者 🕽	d Debu	g layout	- 🖬 🐼 🚥	a 1 📖 🤌
🚮 Start Page 🖾 📄 prova.c 🔀									
1 = void main() {									
	(FRID) .								
• • •	Edit Project			283					23
	Oscillator Selection			Â	-MCU and Oscilla	ator			
	INTRC oscillator; port I/O function on both RA6	OSC2/CLK	Opin a ▼		MCU Name		P16E88	-	
	Watchdog Timer				Pico Name		F 101 00	•	
	Disabled		•		MCU Clock Fr	requency [Mi	Hz]	8.000000	
	Power-up Timer								
	Enabled				Build Type			Heap	
	RA5/MCLR/VPP Pin Function			=	Release	ICD Deb	bug	Size 0	
	Disabled		•						Load Scheme
	Brown-out Reset				Configuration F	Registers			
	Enabled				CONFIG1 :	\$2007 :	0x3F50		Save Scheme
	Low-Voltage Programming				CONFIG2 :	\$2008 :	0x3FFF		
	Disabled		•						Default
	Data EE Memory Code Protection								Deraut
	Disabled		•						
	Flash Program Memory Write								
	Disabled		•						
< III	In-Circuit Debugger Mode					ſ	Genera	Output Settings	
Messages Duick Convertor	Disabled		•	Ŧ					
V Errors	-		<u>о</u> к		Canc	el			
Line Message No.	Message Text							Unit	
		REGIST	TER 15-1:	co	ONFIG1: CONF	IGURATIO	WORD	1 REGISTER (ADDRE	SS 2007h)
		R/P-1	R/P-1 R/P	-1	R/P-1 R/P-1	CPD LVP	R/P-1	R/P-1 R/P-1 R/P-1	R/P-1 R/P-1 R/P-1
Nel microcontrollore Pl	C16F88 della	bit 13	OOPMA DED	00	MALL MALV	OFD LVF	DURLIN	MOLAL TOJOZ PWATER	bit 0
Mirochin ci sono duo ro	gistri a 14 bit								
Millochip ci sono due re	gistil a 14 bit,	DIT 13	1 = Code pr	rote	ction off	be Protection	DITS		
che servono per impost	are	0 = 0000h to 0FFFh code-protected (all protected)							
		bit 12 CCPMX: CCP1 Pin Selection bit 1 = CCP1 function on RB0							
determinate caratteristic	che e funzioni,	0 = CCP1 function on RB3							
		bit 11 DEBUG: In-Circuit Debugger Mode bit							
tramite EDIT PROJECT	si vanno	1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/D pins 0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger							
ad impostare proprio qu	lesti due registri	bit 10-9	WRT<1:0>:	: Fla	ash Program Mem	nory Write En	able bits		
ad impostare proprio qu			11 = Write p 10 = 0000h	prote	ection off DOFFh write-prote	cted, 0100h	to OFFFh r	may be modified by EECO	N control
Si consiglia la lettura de	el datasheet del		01 = 0000h	to C	07FFh write-prote	cted, 0800h	to OFFFh r	may be modified by EECO	N control
		bit 8	00 = 0000h CPD: Data	FE	OFFFh write-prote Memory Code Pr	otection bit			
microcontrollore per col	mprendere bene	Die O	1 = Code pr	rote	ction off	orcoron pic			
il aignificata di agni hit			0 = Data EE	Eme	emory code-prote	cted			
li significato di ogni bit.		DIT /	1 = RB3/PG	Joita GM p	pin has PGM fund	tion, Low-Vo	Itage Prog	ramming enabled	
		1000	0 = RB3 is (digit	tal I/O, HV on MC	LR must be	used for pr	rogramming	
		bit 6	BOREN: Br	rowr	n-out Reset Enab	le bit			
			0 = BOR dia	sabi	led				
		bit 5	MCLRE: RA	A5/	MCLR/VPP Pin Fu	Inction Select	t bit		
			0 = RA5/MC	CLR	VPP pin function	is digital I/O,	MCLR int	ernally tied to Vop	
EGISTER 15-2: CONFIG2: CONFIGURATION	WORD 2 REGISTER (ADDRESS 2008h)	bit 3	PWRTEN:	Pow	ver-up Timer Enat	ble bit			
	IESO FOMEN		1 = PWRT (0 = PWRT (disa enal	bled				
bit 13	bit 0	bit 2	WDTEN: W	/atch	hdog Timer Enabl	le bit			
bit 13-2 Unimplemented: Read as '1'			1 = WDT er	nabi	led				
bit 1 IESO: Internal External Switchover bit		bit 4 1-	0 = WDT di	sabi	ied Iscillator Selection	bits			
1 = Internal External Switchover mode enabled 0 = Internal External Switchover mode disable	d	201.4, 14	111 = EXT	RC	oscillator; CLKD	function on F	RA6/DSC2	CLKO	
bit 0 FCMEN: Fail-Safe Clock Monitor Enable bit			110 = EXT	RC	oscillator; port I/C	D function on	RA6/OSC	2/CLKO	on on PA7/DSO4/DLKLsis
1 = Fail-Safe Clock Monitor enabled 0 = Fail-Safe Clock Monitor disabled			101 = INTE 100 = INTE	RCO	oscillator; OLKO fu	function on HA	both RA6/0	DSC2/CLKO pin and RA7/	OSC1/CLKI pin

- 110 = INTRC oscillator; DCKC function on RA6/DSC2/CLKO pin and port I/D function on RA7/DSC1/CLKI pin 100 = INTRC oscillator; port I/D function on both RA6/DSC2/CLKO pin and RA7/DSC1/CLKI pin 101 = ECIC; port I/D function on RA6/DSC2/CLKD

 - 010 = HS oscillator 001 = XT oscillator 000 = LP oscillator

Legend: R = Readable bit

-n = Value at POR

W = Writable bit

'1' = Bit is set

U = Unimplemented bit, read as '0'

x = Bit is unknown

'0' = Bit is cleared

Nel nostro caso troviamo su OSCILLATOR SELECTION l'opzione INTRC Oscillator, port function I/O on both RA6 e RA7.

Significa che abbiamo scelto di utilizzare il clock interno al microcontrollore lasciando liberi I due pin RA6 e RA7, normalmente utilizzati per il collegamento di un quarzo esterno.

Questa scelta ci consente di **non avere componenti esterni** e di avere **due linee libere** che possono essere utilizzate come ingresso (RA7) o uscita (RA6).

L'opzione **WATCHDOG TIMER**, risulta disabilitata. Questo bit se abilitato, attiva un circuito interno al microcontrollore che si chiama WATCHDOG (cane da guardia). Il watchdog (WDT) non è altro che un timer che ciclicamente, ogni periodo di tempo impostabile, resetta il microcontrollore.

Facciamo l'ipotesi che il watchdog sia impostato a 16 msec, ciò significa che ogni 16 msec il WDT resetta il microcontrollore, costringendolo a riavviare il suo programma dall'inizio. E' ovvio che per poter lavorare insieme al WDT dovremo evitare questo reset ripetitivo, e per questo all'interno del nostro programma, dovremo prevedere un reset del WDT prima che esso resetti il microcontrollore.

In questo modo avremo la certezza che qualora si crei una anomala situazione di loop infinito o di stallo del programma, qualcuno (cioè il WDT) resetterà il microcontrollore consentendo il riavvio ed al ripristino del programma.

Pensiamo ad esempio ad un microcontrollore che gestisce una centralina di un antifurto, un eventuale situazione di stallo del programma o di loop infinito causata anche da un baco del programma, potrebbe causare il mancato funzionamento del sistema, e potremmo non accorgercene per ore visto che non abbiamo un feedback immediato sul funzionameto del sistema. In questo caso il WDT consentirebbe il ripristino automatico del sistema.

Il WDT è comunque complesso da utilizzare e se non occorre se ne può anche fare a meno come nel nostro caso dove è disabilitato.

POWER UP TIMER. Questa funzione invece può essere molto utile e conviene in linea di massima attivarla sempre. In pratica se la funzione è abilitata, all'accensione del microcontrollore ci sarà un breve tempo di attesa prima dell'avvio del programma. Questo ritardo è molto utile perché in questo modo si evita di far partire il programma prima che la tensione di alimentazione sia perfettamente stabile e non soggetta a fenomeni transitori o ai disturbi che potrebbero crearsi all'accensione.

RA5/MCLR function. In ogni microcontrollore c'è un piedino adibito a reset hardware, chiamato **MCLR (Master CleaR)**. Disabilitando il bit si ha la possibilità di utilzzare lo stesso piedino come normale ingresso/uscita.

BROWN OUT RESET. Anche questa funzione può essere molto utile. In pratica se attivata viene monitorata la linea di alimentazione del microcontrollore, e se la tensione scende al di sotto di un valore per una certa durata, viene automaticamente resettato il microcontrollore. Serve ad inibire il funzionamento del programma nei casi in cui la tensione di alimentazione non sia perfettamente stabile, come ad esempio allo spegnimento del circuito.

LOW VOLTAGE PROGRAMMING. Nei microcontrollori della Microchip, sono possibili due possibili modi di programmazione, High Voltage (HV) e Low Voltage (LV). Normalmente mediante I comuni programmatori si utilizza il primo, dove verrà utilizzata una tensione di circa 13,5 Volt sul piedino MCLR.

E' anche possibile però programmare mediante una tensione bassa con il metodo LOW VOLTAGE, in questo caso si potrà utilizzare la stessa tensione di alimentazione del circuito, ma la linea RB3 dovrà essere lasciata libera. Nel nostro caso prevediamo di utilizzare un normale programmatore, perciò possiamo lasciare questa funzione disabilitata.

DATA EE MEMORY CODE PROTECTION e FLASH PROGRAM MEMORY CODE PROTECTION.

Questi due bit servono per abilitare la protezione del codice scritto nella EEPROM (dati) o nella flash (programma). Nel nostro caso non volendo proteggere il codice la funzione è disabilitata.

FLASH PROGRAM MEMORY WRITE. Consente di abilitare la scrittura nella FLASH, cioè la memoria dove viene memorizzato il programma da eseguire. Nel nostro caso la funzione è disabilitata.

IN CIRCUIT DEBUGGER MODE. E' possibile attivare questa funzione se si vuol fare il debug del programma, occorre in questo caso compiare con la spunta su ICD Debug nella sezione BUILD TYPE. Nel nostro caso la funzione è disabilitata.

CCP1 FUNCTION ON RB0. CCP sta per Capture Compare Pwm. Le 3 modalità Capture, Compare e PWM fanno capo al bit RB0 o RB3 a seconda dell'impostazione scelta.

La modalità Capture consente di rilevare il tempo di un timer interno ogni qualvolta ci sia un segnale all'ingresso associato al modulo Capure (nel nostro caso RB0).

La modalità Compare invece confronta il valore di un registro con il valore di un timer interno, e in caso di corrispondenza, questa viene segnalata sul pin associato (nel nostro caso RB0).

La modalità PWM (Pulse Width Modulator) invece consente la generazione di un segnale ad onda quadra con duty-cicle variabile sul pin associato (nel nostro caso RB0).

Questa descrizione è molto sintetica, e il modulo CCP necessità dell'utilizzo di altri registri, si rimanda in caso di necessità, alla lettura del data sheet.

FAIL SAFE CLOCK MONITOR. Questa funzione consente al microcontrollore di utilizzare atuomaticamente il clock interno qualora ci siano dei malfunzionamenti sul clock esterno. Nel nostro caso utilizzando solo il clock interno la funzione non è significativa.

INTERNAL EXTERNAL SWITCHOVER. Anche questa funzione ha senso se si utilizza il clock esterno, in pratica se attiva questa scelta, il programma verrà avviato utilizzando il clock interno fino a quando il clock esterno non sarà stabile.

Consiglio comunque un'attenta lettura del datasheet del microcontrollore, per comprendere il funzionamento di tutte queste funzioni.

A questo punto siamo pronti a scrivere il codice nel file con estensione .C.

Il codice va scritto all'interno del Main e come in tutti I programmi in linguaggio C possiamo creare le nostre funzioni e variabili all'interno di un unico file o in più file.

Proviamo a scrivere un semplicissimo programma per far lampeggiare un led collegato alla linea RB0.

C mikroC PF	RO for PIC v.7.2.0-C:\U	ers\Daniele\Desktop\mikroc\prova.mcppi			[
File Edit	view <u>P</u> roject <u>B</u> uild		🍇 🚷 🗍 🖳 🎙 🗛 🖻 🔒 🙀	🚺 Debug layout 🔍 🕞 🎚	3 ••) [<u>4</u>] ••) •) • (•) • (•) • • • • • • • •	
	<pre>void main() { OSCCON=0xFC; CMCON=0x07; ANSEL=0x00; TRISE=0x00; while(1){ LED=1; delay_ms(5) LED=0; delay_ms(5) } "" "" Delay_ms(5) }</pre>	<pre>//abilito il clock primario ad &A //disabilito i comparatori //seleziono PORTA come digitale 00);</pre>	EH z			roject Manager 🕒 Library Manager 🔐
int room		72 Units				
; V Errors	vvarnings	V HINTS				
Line	Message No.	Message Text		Unit		^
0	1139	Available RAM: 352 [byte:	s], Available ROM: 4096 [bytes]			
0	122	Compilation Started		prova.c		
18	123	Compiled Successfully		prova.c		
0	127	All files Compiled in 140 ms	s			
0	1144	Used RAM (bytes): 2 (1%) Free RAM (bytes): 350 (99%)	Used RAM (oytes): 2 (1%) Free RAM (bytes): 350 (99%)	E
0	1144	Used ROM (program word	s): 41 (1%) Free ROM (program words): 4055 (99	%) Used ROM (program words): 41 (1%) Free ROM (progra	
0	125	Project Linked Successful	Y	prova.mcpp		
0	128	Linked in 47 ms				
0	129	Project 'prova.mcppi' com	pleted: 203 ms			
0	103	Finished successfully: 26 g	gen 2019, 16:29:04	prova.mcpp		-
1:1	Insert	Compiled	C:\Users\Daniele\Desktop\mikroc\prova.c			

Il pulsante indicato dalla freccia avvia la compilazione del programma, nella finestra in basso viene indicato il risultato della compilazione.

A questo punto è possibile simulare il programma (F9 e poi shit+F5) o programmare il microcontrollore con l'apposito circuito programmatore.

C mikroC PRO for PIC v.7.2.0-C:\Users\Daniele\Desktop\mikroc\prova.mcppi		
<u>File Edit View Project Build Run Tools H</u> elp		
E B- E K B = D & & A 1 b + F C 2 b + A 1 C 2	Watch Values	8
🚮 Start Page 🔯 📄 prova.c 🔀	🗄 🕃 🚰 🗛 🐢 🕫 I 🛛 🖷 📰 🖓 🖓	
	🗌 🕴 🕂 Add 💥 Remove 📎 Properties 🙀 Add All 候 Remove All	
• sbit LED at RB0_bit;	Select variable from list:	
· Pvoid main() {	Search for variable by assembly name:	
• CMCON=0x07; //disabilito i comparatori	Name Malue Address	
ANSEL=0x00; //seleziono PORTA come digitale	PORTE 0 0x0006	
<pre> 10</pre>		
<	PC= 0x000003 0.00 us	
III Messages III Ouick Converter		

Il punto di forza di questo compilatore è la presenza di una guida ben fatta e di tantissime librerie ed esempi. Tramite F1 si accede alla guida ed andando sotto alle librerie hardware troviamo davvero di tutto. Bisogna però ricordarsi di selezionare le librerie che si vogliono utilizzare dal Library Manager visto all'inizio.



Se invece si vuol avviare la programmazione mediante apposito programmatore della mikroelektronica, bisogna cliccare sul pulsante indicato dalla freccia, e si avvierà la suite di programmazione dove è possibile avviare la programmazione o anche la lettura del chip.

C mikroC PRO fo	or PIC v.7.2.0-C:\Users\Daniele\Desk	top\mikroc\prova.mcppi			
<u>File Edit View</u>	<u>Project Build Run Tools</u>	<u>H</u> elp	1		
B 🔁 🗗	3 📴 🖬 🗋 🗗 🗗 🛃 🗄 🕻	👌 🖯 🖨 🖉 😓 🛛 🗞	🔈 😒 🏷 🛛 🖳 🦷 🧎 🖉 🔐	Debug layout 👻 📊	☑ 🖬 🕴 🖉 🔝 🖬 🕗 😥 🕡 🛛 😓 🔶
🚮 Start Page 🗵	3 📄 prova.c 🖾				
 sbit voi o <lio< li=""> o o o o<th></th><th>Configuration Bits Oscillator Watchdog Timer Power Up Timer Master Clear Brown Out Detect Low Voltage Program Data EE Protect In-Circuit Debugger CCP1 pin selection Int./Ext. switch over Fail-safe Clock mon.</th><th>INTRC; RA6 as I/O Disabled TA5 as I/O pin Tabled Tabled</th><th>Code Protect None None None</th><th>Voltage Options Power target MCU from programmer Power target MCU from programmer Power target MCU from programmer Power target MCU from programmer Provide the target MCU from programmer Provide the target MCU from programmer Provide the target MCU from programmer Provide t</th></lio<>		Configuration Bits Oscillator Watchdog Timer Power Up Timer Master Clear Brown Out Detect Low Voltage Program Data EE Protect In-Circuit Debugger CCP1 pin selection Int./Ext. switch over Fail-safe Clock mon.	INTRC; RA6 as I/O Disabled TA5 as I/O pin Tabled Tabled	Code Protect None None None	Voltage Options Power target MCU from programmer Power target MCU from programmer Power target MCU from programmer Power target MCU from programmer Provide the target MCU from programmer Provide the target MCU from programmer Provide the target MCU from programmer Provide t
une f	0%				
0					
0	HEX File: Loaded 🔶	C:\USERS\DANIELE\DESKTOP	NMIKROC\PROVA.HEX		