LA SECONDA PARTE DEL FIRMWARE

In questo Workshop proseguiamo la presentazione del sorgente del firmware di esempio, iniziata nel fascicolo precedente, con il quale potremo costruire una scheda di controllo per servocomandi analogici.

Nel fascicolo precedente abbiamo introdotto in modo molto semplificato gli obiettivi del nostro semplice progetto: realizzare una scheda di controllo basata sul PIC 16F628 in grado di pilotare alcuni servocomandi attraverso una serie di comandi di testo impartiti via linea seriale. Passiamo ora a dare alcune indicazioni in più sulle caratteristiche di funzionamento della scheda su cui caricheremo il firmware di esempio. Anche in questo caso il firmware è volutamente molto semplice e pensato esclusivamente per fini didattici. Con firmware di maggiore complessità è possibile arrivare a risultati ben più performanti. Come già sappiamo, la scheda supporterà la ricezione di comandi via linea seriale e potrà controllare fino a quattro servocomandi analogici, generando impulsi con durata compresa all'incirca tra gli 0,5 µs e i 2,5 µs (con una risoluzione di 256 posizioni per motore). La scheda permetterà, inoltre, di accendere o spegnere ognuno dei quattro servo attraverso due apposite istruzioni (le istruzioni interpretate dal firmware sono mostrate nella tabella sottostante). Lasciamo, comunque, i commenti più tecnici necessari a comprendere il funzionamento dell'intero sistema per i prossimi fascicoli e dedichiamoci, nelle prossime pagine, alla seconda parte del codice (le indicazioni per la sua compilazione e la configurazione del PIC verranno fornite nel corso del prossimo Workshop).

COMANDO	EFFETTO
M*nxxx*	Muove il servocomando '*n*' (*n* compreso tra 0 e 3 inclusi) in posizione '*xxx*' (*xxx* numero di tre cifre compreso tra 000 e 255).
O*n*	Attiva il servocomando '*n*'.
o*n*	Disattiva il servocomando '*n*'.

La seconda parte del firmware

```
311                     NOP
312                     NOP
313                     NOP
314                     NOP
315                     NOP
316                     NOP
317                     NOP
318                     NOP
319                     NOP
320                     NOP
321                     NOP
322                     NOP
323                     NOP
324                     NOP
325                     NOP
326                     NOP
327                     NOP
328                     NOP
329                     SUBWF ASM_DELAY
330                     BTFSS STATUS,2
331                     GOTO INIZIO0
332                FINE0:
333                     MOVLW TEMP_REG /* ripristino del contenuto del registro W */
334           }
335           CONTROLLO_SERVO0 = 0;
336      }
337
338   /* se il servo corrente è il servo numero 1 */
339   else if(servo_corrente==1)
340   {
341        /* copiamo nella variabile ASM_DELAY la posizione desiderata per il
342        servo numero 1 */
343        ASM_DELAY = posizioni_servo[servo_corrente];
344
345        /* iniziamo la generazione dell'impulso */
346        CONTROLLO_SERVO1 = 1;
347
348        /* Generazione del ritardo di base */
349        Delay_us(IMPULSO_MINIMO_us);
350        /* genera il ritardo associato allo step */
351        asm{
352             MOVWF TEMP_REG /* salvataggio del contenuto del registro W */
353             INIZIO1:
354                  MOVLW 0X01
355                  NOP
356                  NOP
357                  NOP
358                  NOP
359                  NOP
360                  NOP
361                  NOP
362                  NOP
363                  NOP
```

```
364                     NOP
365                     NOP
366                     NOP
367                     NOP
368                     NOP
369                     NOP
370                     NOP
371                     NOP
372                     NOP
373                     NOP
374                     NOP
375                     NOP
376                     NOP
377                     NOP
378                     NOP
379                     NOP
380                     NOP
381                     NOP
382                     NOP
383                     NOP
384                     NOP
385                     NOP
386                     NOP
387                     NOP
388                     NOP
389                     SUBWF ASM_DELAY
390                     BTFSS STATUS,2
391                     GOTO INIZIO1
392                FINE1:
393                     MOVLW TEMP_REG /* ripristino del contenuto del registro W */
394            }
395
396            CONTROLLO_SERVO1 = 0;
397        }
398
399        /* se il servo corrente è il servo numero 2 */
400        else if(servo_corrente==2)
401        {
402         /* copiamo nella variabile ASM_DELAY la posizione desiderata per il
403         servo numero 2 */
404         ASM_DELAY = posizioni_servo[servo_corrente];
405
406         /* iniziamo la generazione dell'impulso */
407         CONTROLLO_SERVO2 = 1;
408
409         /* Generazione del ritardo di base */
410         Delay_us(IMPULSO_MINIMO_us);
411         /* genera il ritardo associato allo step */
412         asm{
413              MOVWF TEMP_REG /* salvataggio del contenuto del registro W */
414              INIZIO2:
415                   MOVLW 0X01
416                   NOP
417                   NOP
418                   NOP
419                   NOP
420                   NOP
```

```
421                 NOP
422                 NOP
423                 NOP
424                 NOP
425                 NOP
426                 NOP
427                 NOP
428                 NOP
429                 NOP
430                 NOP
431                 NOP
432                 NOP
433                 NOP
434                 NOP
435                 NOP
436                 NOP
437                 NOP
438                 NOP
439                 NOP
440                 NOP
441                 NOP
442                 NOP
443                 NOP
444                 NOP
445                 NOP
446                 NOP
447                 NOP
448                 NOP
449                 NOP
450                 SUBWF ASM_DELAY
451                 BTFSS STATUS,2
452                 GOTO INIZIO2
453           FINE2:
454                 MOVLW TEMP_REG /* ripristino del contenuto del registro W */
455     }
456
457     CONTROLLO_SERVO2 = 0;
458  }
459
460 /* Se il servo corrente è il numero 3 */
461  else
462  {
463     /* copiamo nella variabile ASM_DELAY la posizione desiderata per il
464     servo numero 3 */
465     ASM_DELAY = posizioni_servo[servo_corrente];
466
467     /* iniziamo la generazione dell'impulso */
468     CONTROLLO_SERVO3 = 1;
469
470     /* Generazione del ritardo di base */
471     Delay_us(IMPULSO_MINIMO_us);
472     /* genera il ritardo associato allo step  */
473     asm{
474           MOVWF TEMP_REG /* salvataggio del contenuto del registro W */
475           INIZIO3:
476               MOVLW 0X01
```

```
477                 NOP
478                 NOP
479                 NOP
480                 NOP
481                 NOP
482                 NOP
483                 NOP
484                 NOP
485                 NOP
486                 NOP
487                 NOP
488                 NOP
489                 NOP
490                 NOP
491                 NOP
492                 NOP
493                 NOP
494                 NOP
495                 NOP
496                 NOP
497                 NOP
498                 NOP
499                 NOP
500                 NOP
501                 NOP
502                 NOP
503                 NOP
504                 NOP
505                 NOP
506                 NOP
507                 NOP
508                 NOP
509                 NOP
510                 NOP
511                 SUBWF ASM_DELAY
512                 BTFSS STATUS,2
513                 GOTO INIZIO3
514           FINE3:
515                 MOVLW TEMP_REG /* ripristino del contenuto del registro W */
516       }
517
518       CONTROLLO_SERVO3 = 0;
519   }
520
521    servo_corrente++;  /* elaboriamo il prossimo servo */
522    /* se sono stati elaborati tutti i servocomandi, riavviamo dal numero 0 */
523    if(servo_corrente==NUMERO_SERVO) servo_corrente = 0;
524
525    Delay_ms(5);
526   }  /* FINE while(1) */
527  }
```