

# MORSE SU PIC16F877

## ENCODER E DECODER

Davide Vosti

27 settembre 2006

(Doc. SSL-260806\_AS-it)

## Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Codice Morse</b>	<b>3</b>
2.1	Tabella dei codici . . . . .	3
2.2	Estensione del codice Morse . . . . .	3
<b>3</b>	<b>Informazioni generali</b>	<b>3</b>
<b>4</b>	<b>Codificatore</b>	<b>3</b>
4.1	Flowchart . . . . .	5
4.1.1	main . . . . .	5
4.1.2	parsechar . . . . .	5
4.1.3	bufferize . . . . .	5
4.1.4	sendbuffer . . . . .	5
4.1.5	code_char . . . . .	6
4.1.6	echodit . . . . .	6
4.1.7	signaldown . . . . .	6
<b>5</b>	<b>Decodificatore</b>	<b>12</b>
5.1	Flowchart . . . . .	12
5.1.1	main . . . . .	12
5.1.2	capture . . . . .	12
5.1.3	calculateup . . . . .	12
5.1.4	calculatedown . . . . .	12
5.1.5	code_morse . . . . .	13
<b>6</b>	<b>Osservazioni</b>	<b>13</b>

## 1 Introduzione

Questo documento è atto a descrivere il funzionamento del software per la codifica e la decodifica Morse.

## 2 Codice Morse

Il codice Morse è un metodo per trasmettere informazioni. Originariamente il codice comprendeva le lettere del nostro alfabeto e le 10 cifre.

Lettere e cifre sono codificate con punti (dit) o righe (dah) (es: lettera  $a \Rightarrow .-$ ).

Con il passare degli anni la tecnologia ha fatto passi in avanti, rendendo questa codifica obsoleta. Tuttavia il codice Morse trova ancora uso fra i radioamatori, usi particolari in caso di emergenza o come beacon.

### 2.1 Tabella dei codici

In Figura 1 è rappresentata la codifica per le lettere e le 10 cifre del nostro alfabeto.

### 2.2 Estensione del codice Morse

In principio il codice Morse non prevedeva la codifica dei simboli di punteggiatura come punti di domanda ed esclamativi. In seguito fu introdotta e standardizzata questa possibilità. La versione attuale del programma non prevede questa estensione.

## 3 Informazioni generali

Il codice Morse è generato secondo la frequenza delle lettere usate nella lingua inglese. Lettere con frequenza maggiore hanno simboli più corti. Questo non permette di codificare le lettere secondo un'algoritmo, per questo motivo è necessario tabellarle. I simboli per il codificatore sono memorizzati in un'unica tabella. Ogni simbolo è salvato in un registro da 8 bit dove i 3 bit più significativi memorizzano la lunghezza del simbolo e i 5 bit meno significativi memorizzano la codifica vera e propria. Al contrario, nel decodificatore i simboli sono suddivisi su più tabelle a dipendenza della loro lunghezza.

I simboli sono codificati mediante 1 e 0. L'1 identifica una riga, mentre lo 0 identifica il punto.

I segnali sono generati sul pin 0 della porta D, mentre la ricezione avviene sul pin 1 della porta C (modulo CCP).

Il tempo di riferimento è la durata di un dit. Un dah ha durata 4 dit. Il tempo di attesa tra un simbolo e l'altro è un dit mentre tra lettere l'attesa è di 4 dit. Tra una parola e l'altra si attende per 7 dit.

## 4 Codificatore

Il codificatore resta in attesa che l'utente inserisca una linea di testo tramite il terminale (lunghezza massima della stringa 79 caratteri). Alla ricezione di un "enter" il programma codifica il buffer ritornando a terminale i caratteri codificati e generando un segnale sul pin 0 della porta D.

## INTERNATIONAL MORSE CODE

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to five dots.

A	• —	U	• • —
B	— • • •	V	• • • —
C	— • — •	W	• — —
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	— — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • —	1	• — — — —
L	• — • •	2	• • — — —
M	— —	3	• • • — —
N	— •	4	• • • • —
O	— — —	5	• • • • •
P	• — — •	6	— • • • •
Q	— — • —	7	— — • • •
R	• — •	8	— — — • •
S	• • •	9	— — — — •
T	—	0	— — — — —

Figura 1: Codice Morse originario.

## 4.1 Flowchart

### 4.1.1 main

Il main non fa altro che rimanere in attesa dell'arrivo di un carattere. Quando questo accade viene chiamata una funzione per elaborare il carattere appena ricevuto. In Figura 2 è mostrato il diagramma di flusso.

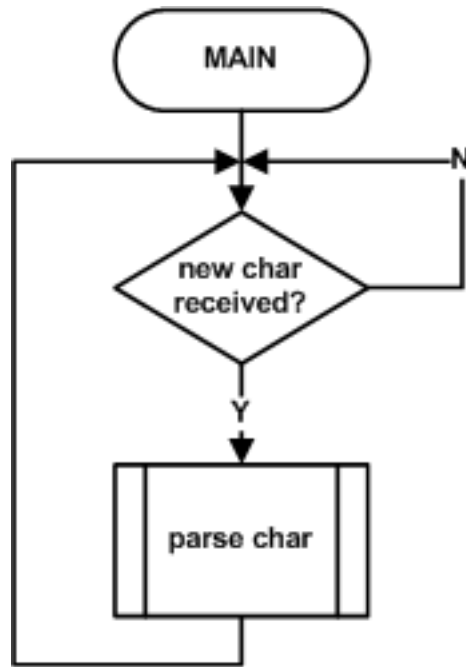


Figura 2: Diagramma di flusso del main.

### 4.1.2 parsechar

La funzione parsechar interpreta il carattere appena ricevuto. Se si tratta di un enter viene codificato e spedito il buffer, altrimenti il carattere viene bufferizzato. In Figura 3 è mostrato il diagramma di flusso.

### 4.1.3 bufferize

La funzione bufferize si occupa di salvare in un buffer il carattere appena ricevuto. Il buffer si trova in banco 3 ed ha una lunghezza di 79 caratteri. Questo è dovuto al limite standard della console (80 caratteri) meno il carattere del prompt. In Figura 4 è mostrato il diagramma di flusso.

### 4.1.4 sendbuffer

La funzione sendbuffer legge il buffer e chiama la funzione code\_char per codificare il carattere appena letto. Alla fine il buffer con i caratteri spediti viene ripulito. In Figura 5 è mostrato il diagramma di flusso.

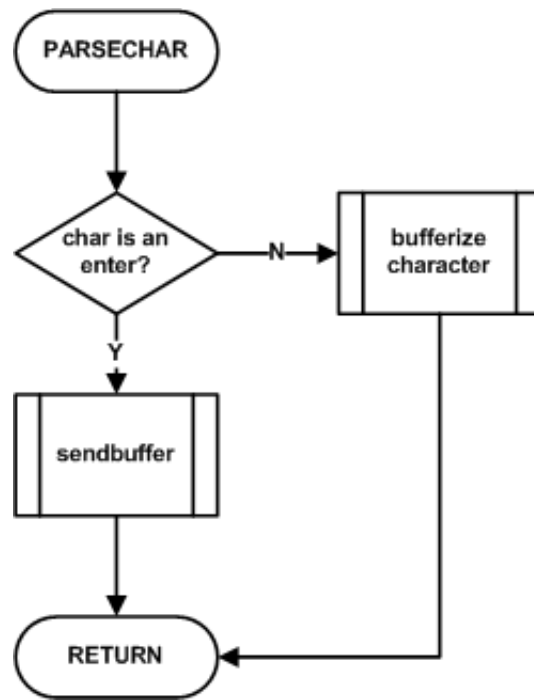


Figura 3: Diagramma di flusso di parsechar.

#### 4.1.5 code\_char

La funzione `code_char` controlla il carattere da trasmettere. Se il carattere è un'enter viene ignorato. Se si tratta di uno spazio viene generato un tempo di attesa, altrimenti viene controllato il tipo di carattere. Se si tratta di una cifra viene codificata direttamente, in caso contrario viene controllato se la lettera è maiuscola o minuscola. In caso di lettera minuscola questa viene trasformata in maiuscolo. In questo modo è possibile dimezzare la lunghezza della tabella contenente i simboli. In Figura 6 è mostrato il diagramma di flusso.

#### 4.1.6 echodit

La funzione `echodit` (ed `echodah`) si occupano di fare l'eco sul terminale dei punti o delle righe. Inoltre generano sul pin 0 della porta D un segnale di durata variabile a dipendenza del tipo di simbolo. Il segnale viene spento dalla funzione `signaldown`. In Figura 7 è mostrato il diagramma di flusso.

#### 4.1.7 signaldown

La funzione si occupa di mantenere il segnale per una durata determinata. `signaldown` fa uso del `timer0`. Ad ogni interrupt (ogni 10ms) viene decrementato il contatore `tmoverflow`. Quando questo è zero il segnale viene spento. In Figura 8 è mostrato il diagramma di flusso.

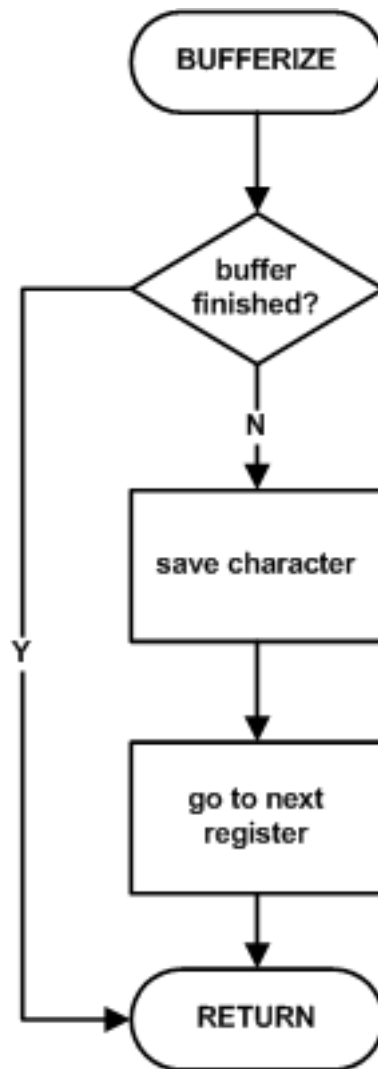
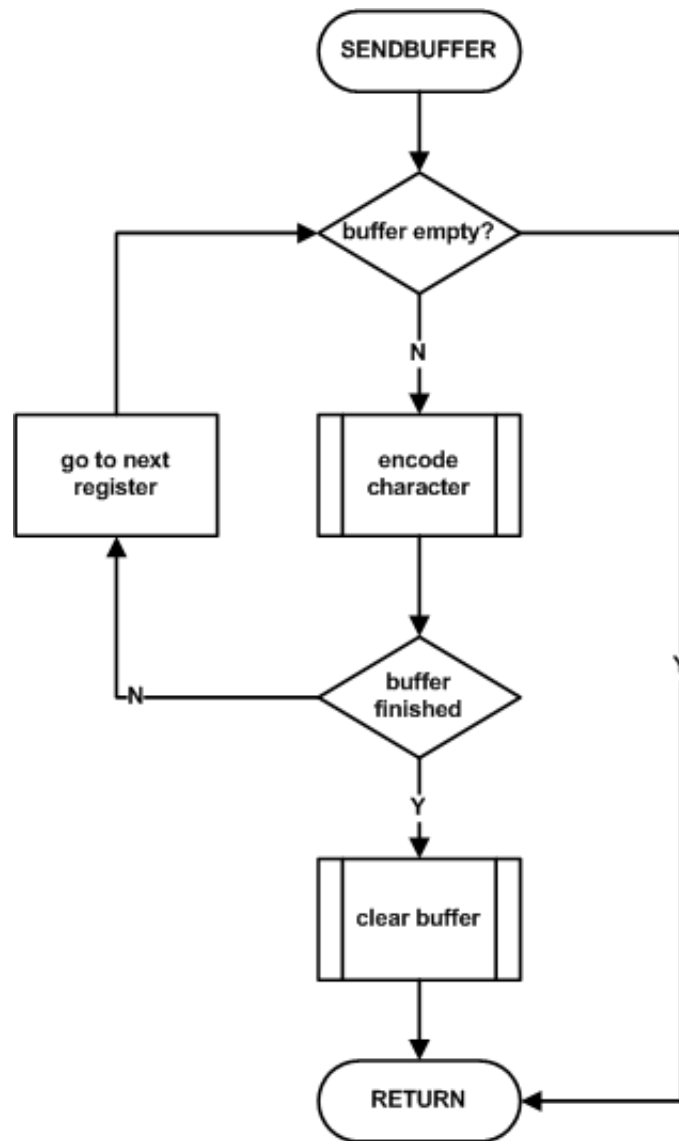


Figura 4: Diagramma di flusso di bufferize.

Figura 5: Diagramma di flusso di `sendbuffer`.



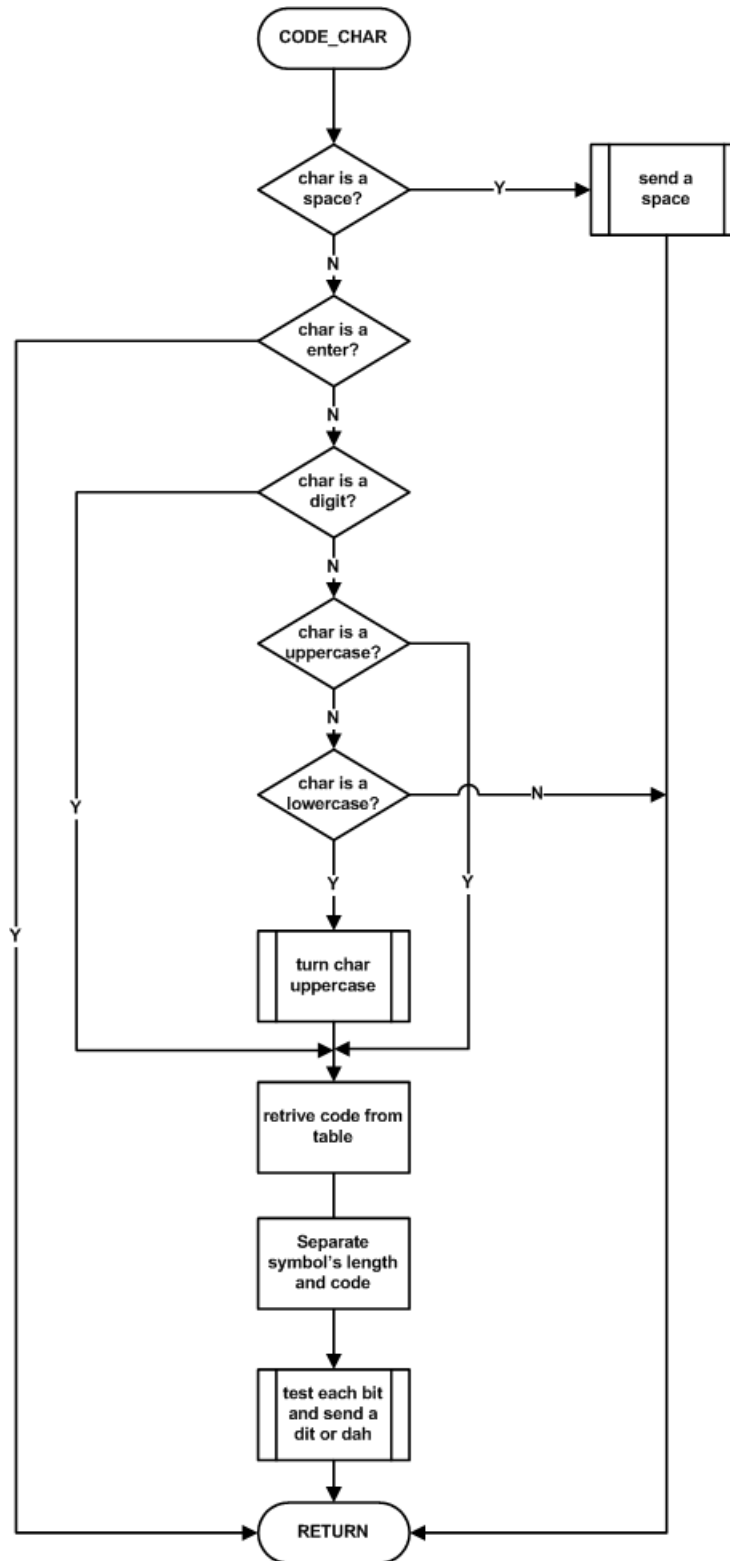


Figura 6: Diagramma di flusso di code\_char.

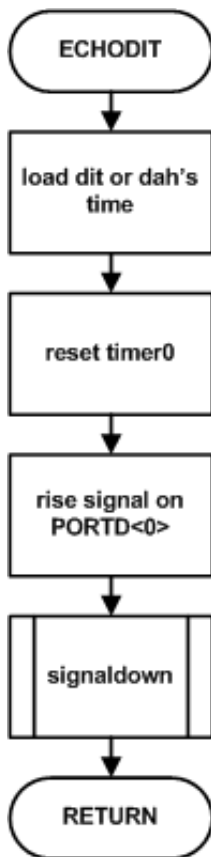


Figura 7: Diagramma di flusso di echodit.

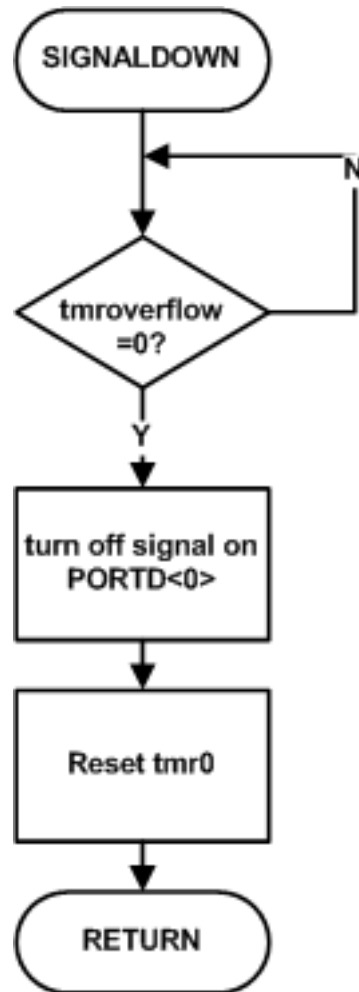


Figura 8: Diagramma di flusso di signaldown.

## 5 Decodificatore

### 5.1 Flowchart

#### 5.1.1 main

Il main si occupa unicamente di controllare l'arrivo di un nuovo fronte (tramite il modulo CCP), sia di salita che di discesa. A questo punto viene chiamata una funzione per smistare il segnale appena ricevuto. In Figura 9 è mostrato il diagramma di flusso.

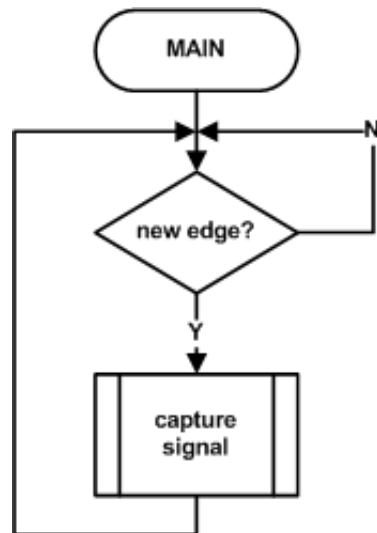


Figura 9: Diagramma di flusso del main.

#### 5.1.2 capture

La funzione capture si occupa di smistare il segnale ricevuto all'apposita funzione di gestione. Se il fronte ricevuto è di salita verrà chiamata `calculatedown`, mentre in caso di fronte di discesa sarà `calculateup` ad essere chiamata. In Figura 10 è mostrato il diagramma di flusso.

#### 5.1.3 calculateup

La funzione `calculateup` determina il tempo di durata di un segnale "alto" ed imposta un registro di conseguenza. Al termine viene cambiato il metodo di cattura del segnale, in questo caso viene impostato su fronte di salita. In Figura 11 è mostrato il diagramma di flusso.

#### 5.1.4 calculatedown

`calculatedown` ha lo stesso principio di funzionamento di `calculateup`, ma anziché codificare un simbolo interpreta la durata di un segnale "basso". A questo punto il simbolo viene decodificato. In Figura 12 è mostrato il diagramma di flusso.

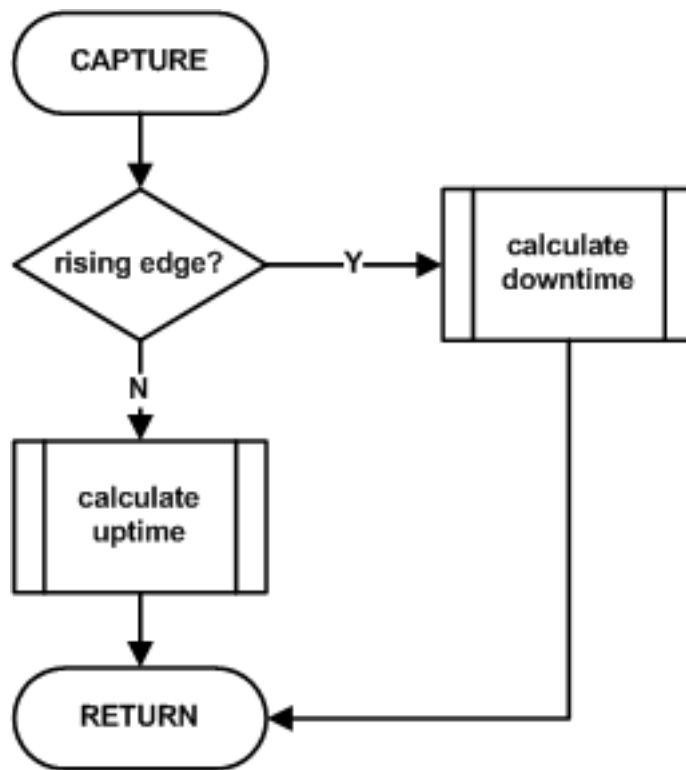


Figura 10: Diagramma di flusso di capture.

### 5.1.5 code\_morse

La funzione `code_morse` prende dalla rispettiva tabella il carattere decodificato e lo manda tramite la USART al terminale. In Figura 13 è mostrato il diagramma di flusso.

## 6 Osservazioni

Attualmente i programmi sono funzionanti ma con alcuni errori. Il codificatore funziona senza alcun errore, mentre nel decodificatore ho notato 2 anomalie:

- Il primo carattere ricevuto non viene decodificato.
- L'ultimo carattere non viene decodificato. La causa è che il decoder rimane in attesa di un nuovo fronte prima di decodificare. Una soluzione è quella di far partire la decodifica dopo un tempo di 4-5 dit.

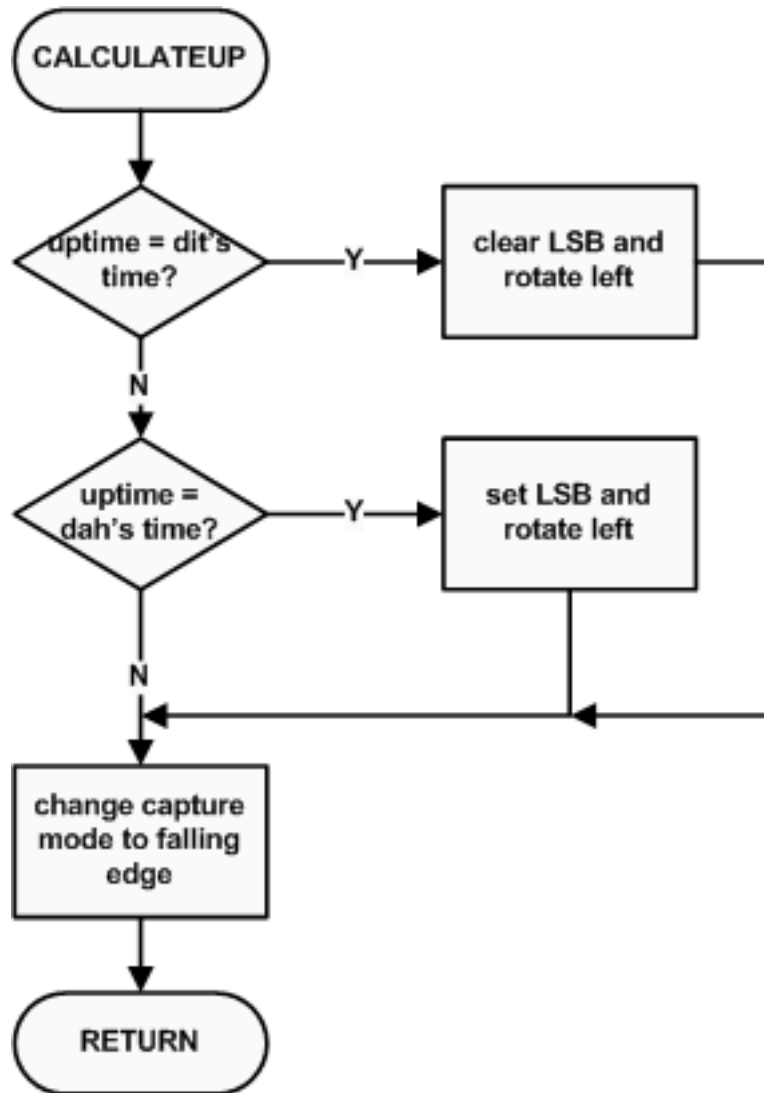


Figura 11: Diagramma di flusso di calculateup.

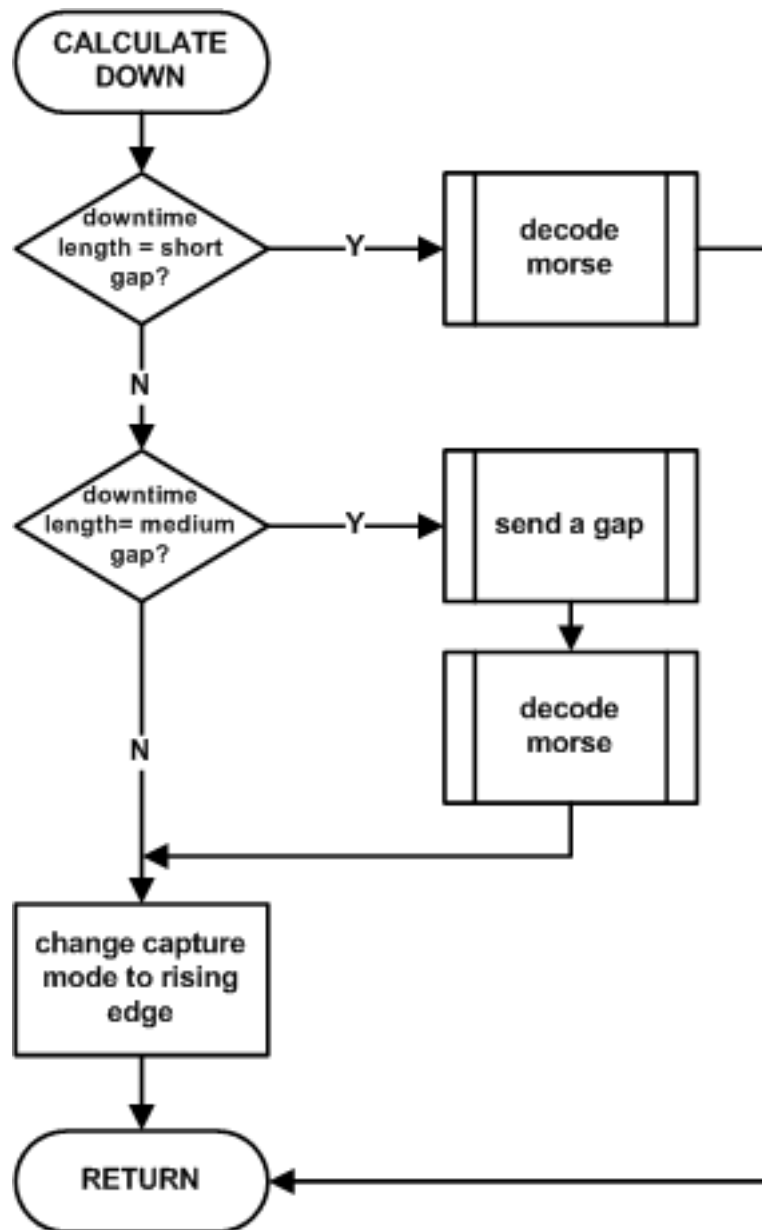
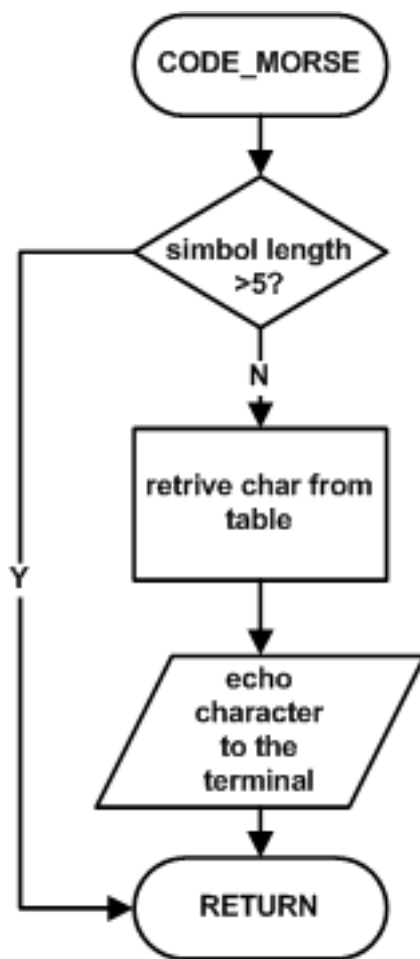


Figura 12: Diagramma di flusso di calculatedown.

Figura 13: Diagramma di flusso di `code_morse`.