Guida all’utilizzo della libreria wait.h

Sommario

[Includere la libreria 3](#_Toc527822)

[Configurazione dei sensori 3](#_Toc527823)

[Utilizzo 5](#_Toc527824)

[Metodi 6](#_Toc527825)

[WaitTouch(int port) 6](#_Toc527826)

[WaitLight(int port, int threshold, bool higher) 6](#_Toc527827)

[WaitDistance(int port, int distance) 7](#_Toc527828)

[WaitSound(int port, int threshold) 7](#_Toc527829)

[WaitTime(long millis) 7](#_Toc527830)

[WaitRotations(int port, int times, int speed) 7](#_Toc527831)

[WaitDegrees(int port, int degree, int speed) 8](#_Toc527832)

## Includere la libreria

Prima di tutto creare un nuovo file o aprirne uno già esistente. Successivamente bisogna la stringa di codice

#include "/path/wait.h" che permette di importare i metodi della libreria wait.

Tra i doppi apici deve essere specificato il percorso in cui si trova il file della libreria.

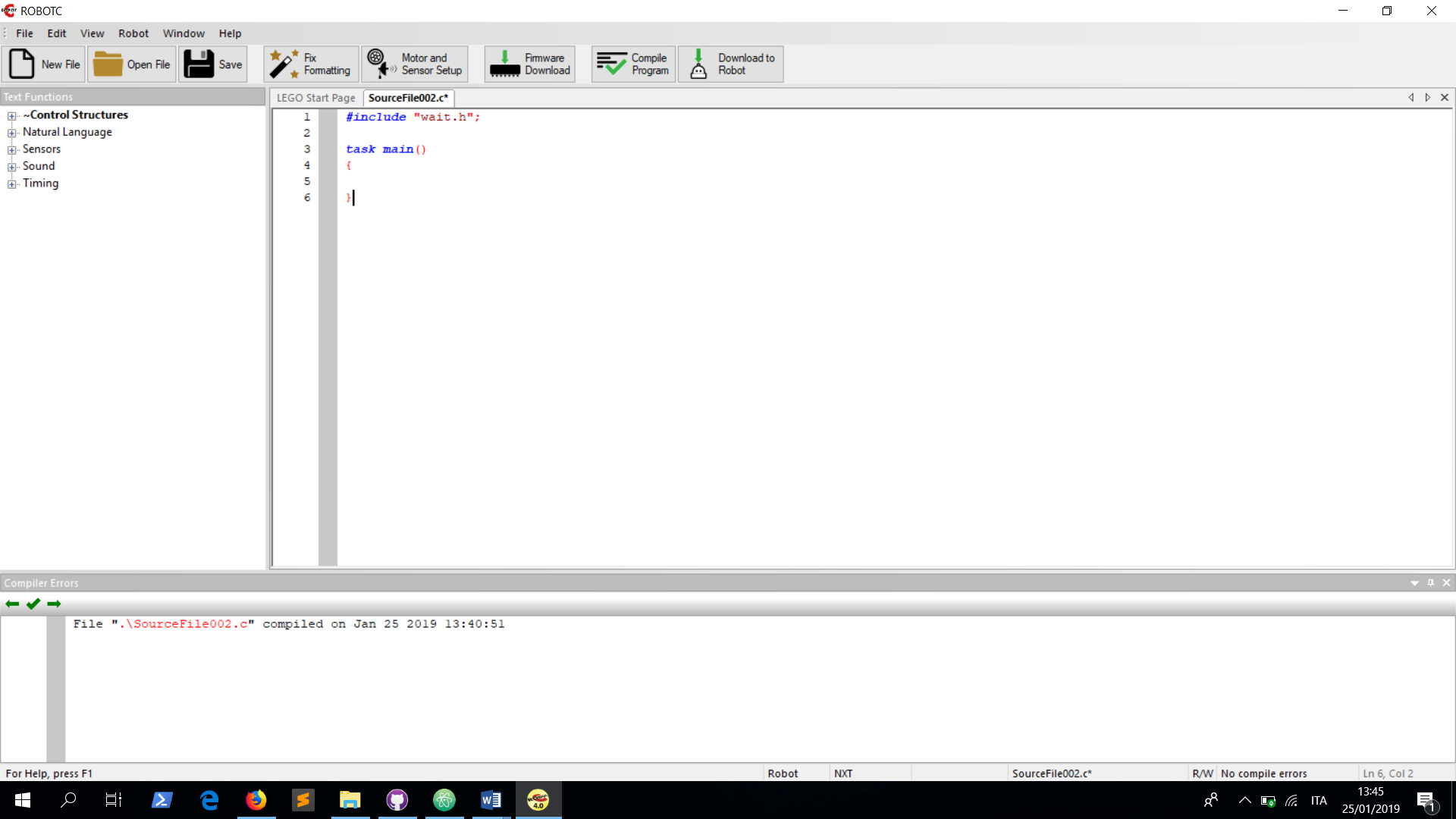


Figura 1 Includere la libreria wait.h

Nella figura 1 il file wait.h si trova nello stesso percorso del file appena creato quindi è abbastanza specificare il nome della libreria.

## Configurazione dei sensori

Per avere una lettura corretta dei valori da parte dei sensori è necessario configurarli. I passaggi sono semplici e veloci da effettuare:

Aprire la voce ***Motors and Sensor Setup*** nel menù ***Robot***

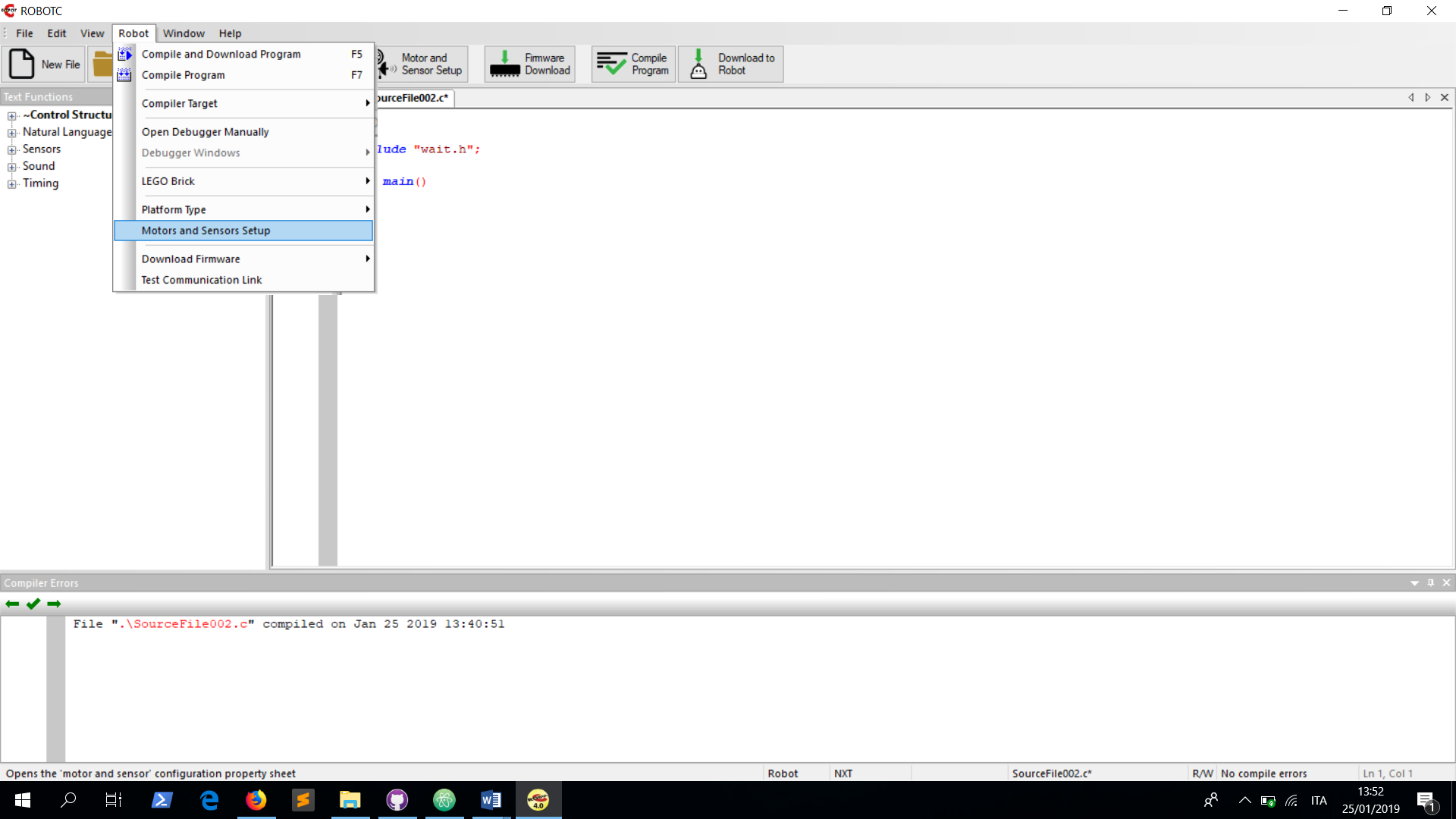


Figura 2 Motors and Sensor Setup

Successivamente bisogna assegnare un nome (auto esplicativo) al sensore scelto tramite il menù a tendina sulla destra.

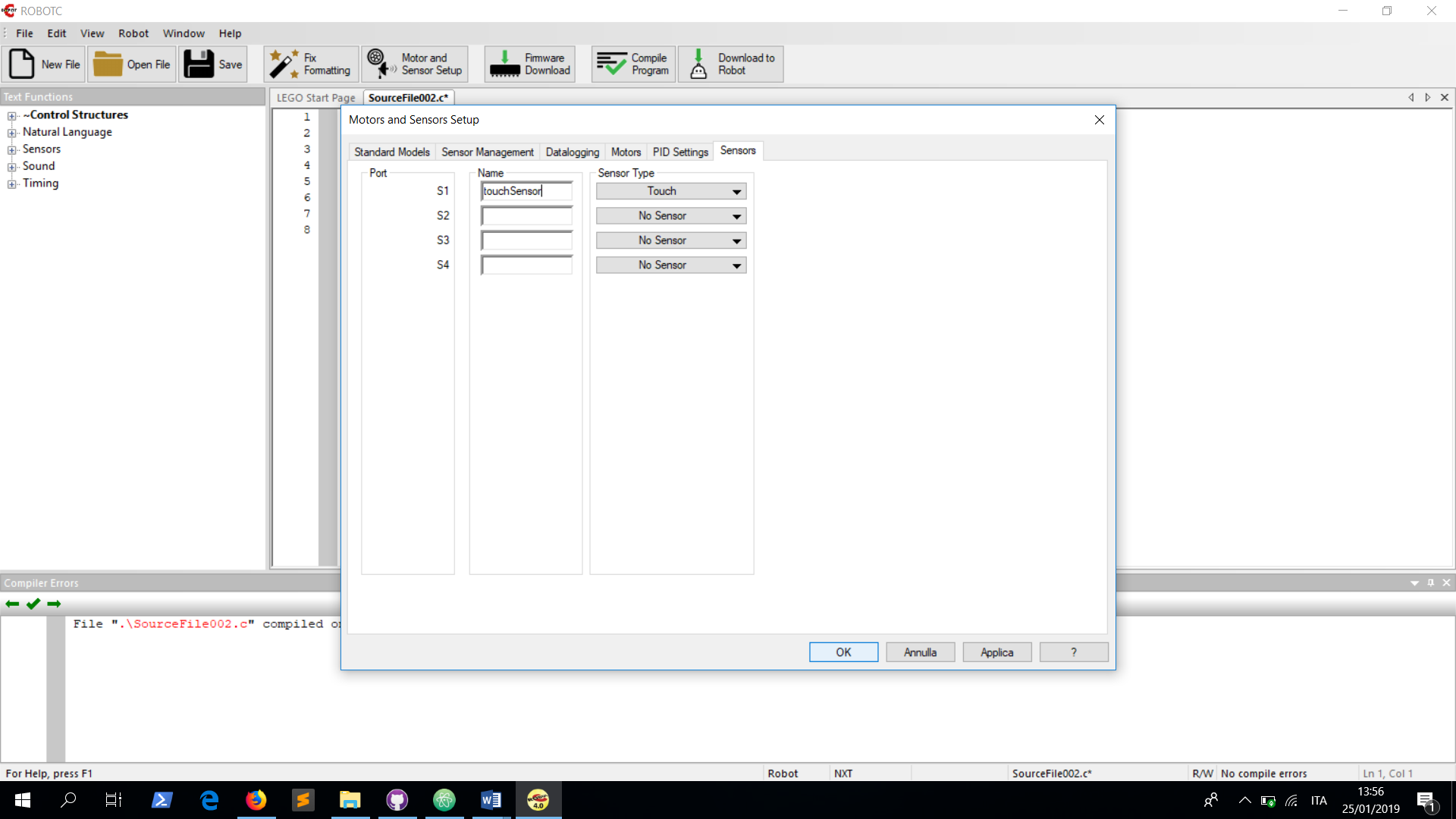


Figura 3 Configurazione sensore touch

Infine cliccare su applica e okay ed in automatico verrà generata la linea di codice contente la configurazione.

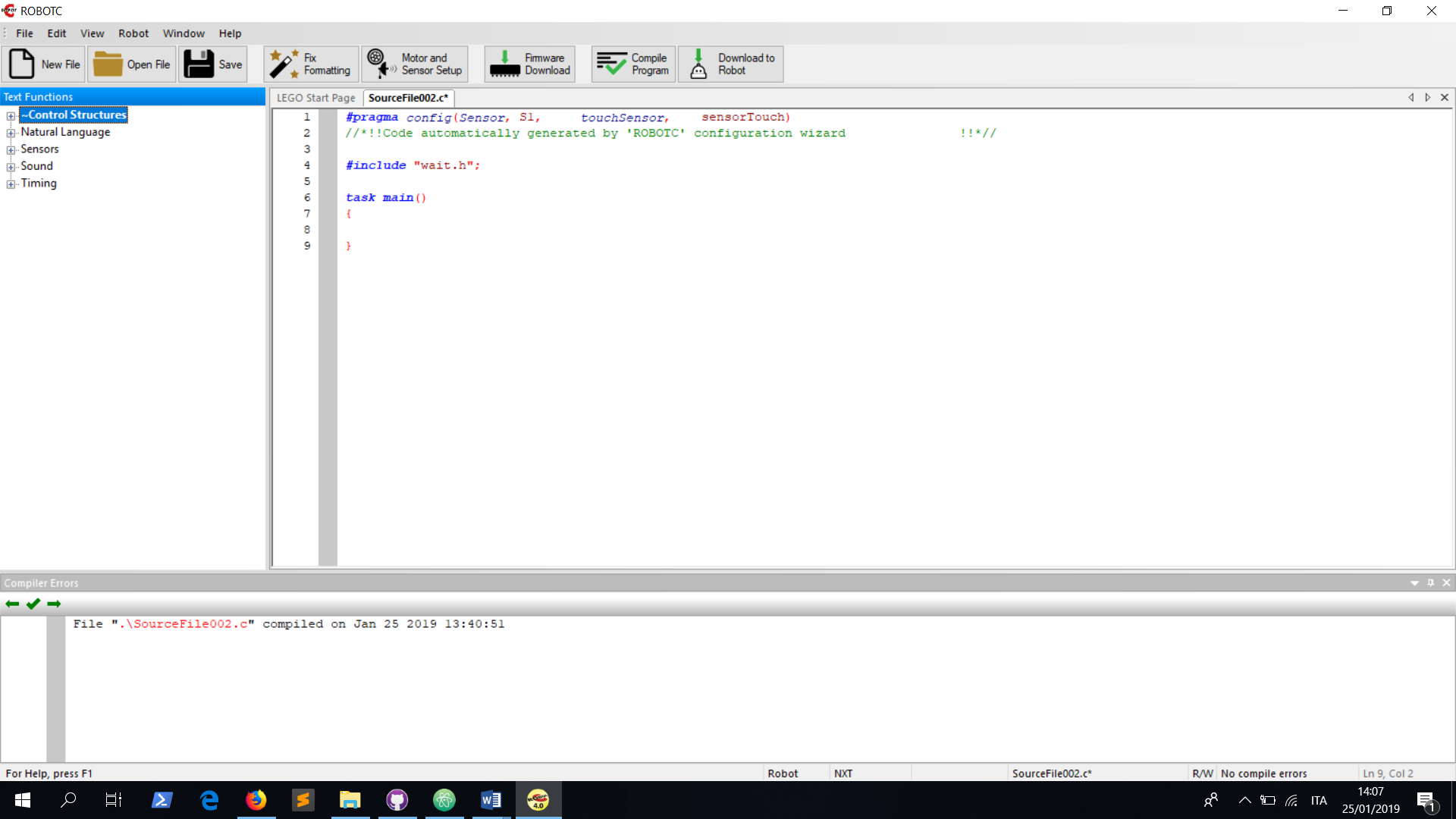


Figura 4 Codice configurazione sensore touch

## Utilizzo

A questo punto non resta che utilizzare i metodi della libreria wait.

In questo esempio il motore continua a muoversi a velocità 50 fino a quando il sensore touch viene premuto.

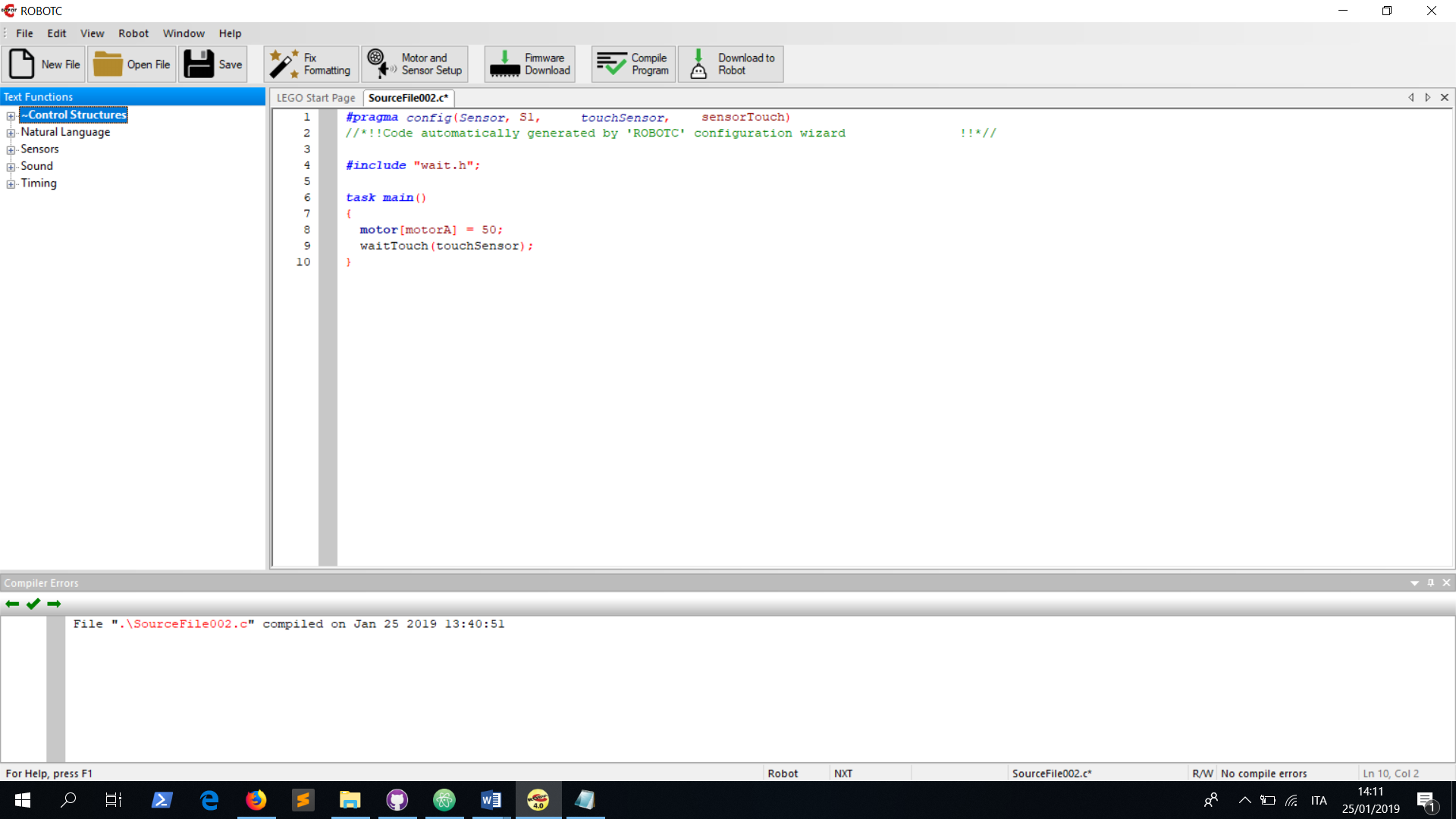


Figura 5 Esempio di utilizzo

## Metodi

### WaitTouch(int port)

**Parametri:**

* **int port**: numero di tipo intero che indica la porta a cui il sensore è collegato al robot, in questo caso si possono passare i valori S1, S2, S3 e S4.

**Esempio:**

waitTouch(S2);

### WaitLight(int port, int threshold, bool higher)

**Parametri:**

* **int port**: numero di tipo intero che indica la porta a cui il sensore è collegato al robot, in questo caso si possono passare i valori S1, S2, S3 e S4.
* **int threshold**: soglia di luce minima o massima (in base al valore di higher).
* **bool higher**: determina se il valore del sensore deve rilevare un valore superiore o inferiore alla soglia.

**Esempio:**

waitLight(S2, 50, true);

### WaitDistance(int port, int distance)

Solitamente utilizzata come condizione all’interno di un ciclo.

**Parametri:**

* **int port**: numero di tipo intero che indica la porta a cui il sensore è collegato al robot, in questo caso si possono passare i valori S1, S2, S3 e S4.
* **int distance:** distanza minima da un oggetto in centimetri.

**Return:** se la distanza è inferiore alla soglia ritorna **true** altrimenti **false**.

**Esempio:**

waitDistance(S2, 30);

### WaitSound(int port, int threshold)

**Parametri:**

* **int port**: numero di tipo intero che indica la porta a cui il sensore è collegato al robot, in questo caso si possono passare i valori S1, S2, S3 e S4.
* **int threshold**: soglia del suono.

**Esempio:**

waitSound(S2, 70);

### WaitTime(long millis)

**Parametri:**

* **int millis**: tempo in millisecondi.

**Esempio:**

waitTime(S2, 70);

### WaitRotations(int port, int times, int speed)

**Parametri:**

* **int port**: numero di tipo intero che indica la porta a cui il motore è collegato al robot, in questo si possono passare i valori S1, S2, S3 e S4.
* **int times** : numero intero di rotazioni che il motore deve compiere.
* **int speed**: velocità con cui vengono svolte le rotazioni.

**Esempio:**

waitRotations(S2, 5, 100);

### WaitDegrees(int port, int degree, int speed)

**Parametri:**

* **int port**: numero di tipo intero che indica la porta a cui il motore è collegato al robot, in questo si possono passare i valori S1, S2, S3 e S4.
* **int degree** : numero intero di gradi di cui deve girare il motore.
* **int speed**: velocità con cui vengono svolti i gradi di rotazione.

**Esempio:**

waitRotations(S2, 130, 100);