
UNIVERSITÀ DI PISA

Corso di Laurea Magistrale in Ingegneria Elettronica

# Progettazione di Sistemi Meccatronici

## Line follower and obstacle avoiding robot

Docenti:

Prof. Baronti Federico

Prof. Roncella Roberto

Ing. Di Rienzo Roberto

Studenti:

Monopoli Matteo

Passarelli Giusy Valeria

Anno Accademico 2021/2022

# Indice

# 1 Introduzione

L'obiettivo del nostro lavoro è stato quello di progettare e implementare, attraverso l'ambiente di sviluppo MATLAB/Simulink, un piccolo veicolo in grado di seguire un percorso tracciato ed evitare eventuali ostacoli. La progettazione è stata svolta seguendo un approccio di tipo Model-Based: la prima fase del flusso di progetto, infatti, ha riguardato la realizzazione di un modello funzionale sia per la parte di controllo che per i vari componenti del sistema; successivamente, in una seconda fase, ci siamo concentrati sull'implementazione hardware del controllo attraverso l'utilizzo della scheda NXP S32K142EVB e sul montaggio e la messa a punto delle restanti parti del sistema.

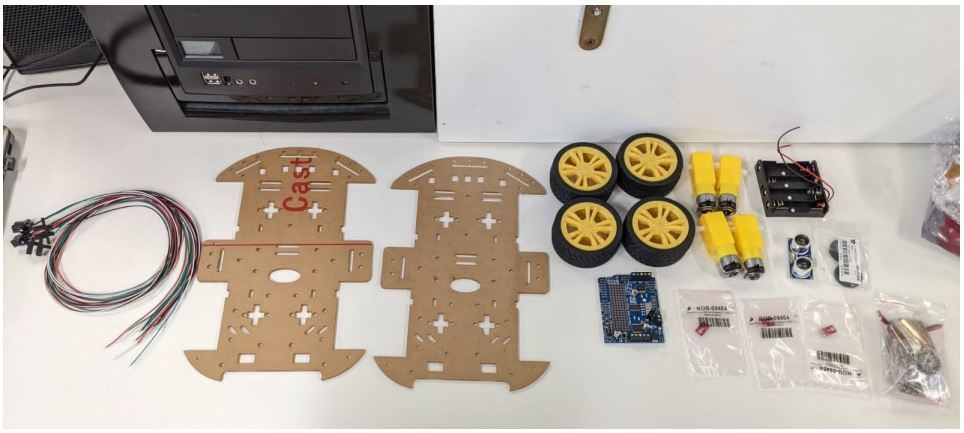
Non meno importante è stata la scelta dei componenti da adoperare per l'implementazione fisica del veicolo. Tale scelta è stata guidata soprattutto dall'esigenza di utilizzare dispositivi che fossero poco ingombranti, economici e adatti al nostro tipo di applicazione.

In questo report saranno, dunque, descritte e commentate in maniera dettagliata tutte le varie fasi del flusso di progetto.

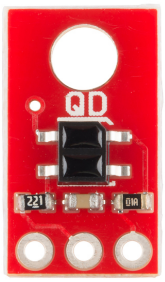
# 2 Componenti

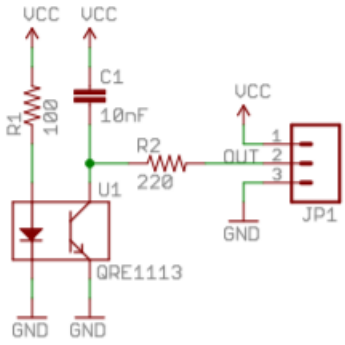
I componenti che si sono resi necessari per lo sviluppo del nostro progetto sono i seguenti:

- Sensori infrarossi - 3 x **ROB-09454**
- Sensori ottici - 4 x **OPB880T55Z**
- Sensore ad ultrasuoni - 1 x **HC-SR04**
- Scheda **S32K142EVB**
- Motori DC brushed - 4 x **BO geared motors** $(V_M = 3 \div 12\ \text{V})$
- Motor driver - 2 x **ROB-14450**
- Batterie NMC - 2 x **18650** $(V_{bat} = 3 \div 4.2\ \text{V})$
- 4 x Ruote
- 4 x Encoder
- 1 x Chassis in plexiglass



## 2.1 Sensori infrarossi

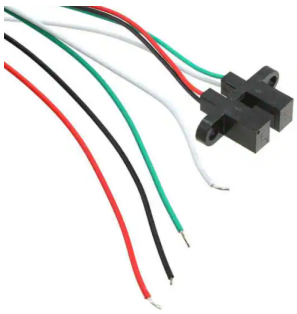


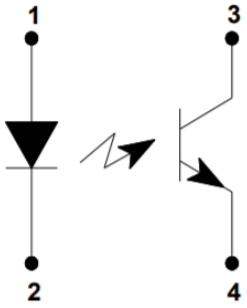




Il sensore infrarossi a cui abbiamo fatto riferimento è un sensore attivo, costituito da una coppia IR LED - fototransistor. Il LED emette una radiazione nel range dell'infrarosso ($\lambda = 1$ mm - 700 nm) che, una volta riflessa, manda in conduzione il fototransistor, producendo in uscita un valore di tensione assimilabile ad uno 0 logico. In caso contrario, se la luce non viene riflessa, il fototransistor non entrerà in conduzione e sarà schematizzabile con un circuito aperto; per leggere, quindi, un valore di tensione pari a $V_{CC}$ sarà necessario inserire una resistenza di pull-up sull'uscita.

In virtù del funzionamento appena descritto, possiamo facilmente dedurre che quando il sensore si trova al di sopra di una base riflettente, quale ad esempio un oggetto di colore bianco, darà in uscita un valore logico basso, mentre darà in uscita un valore logico alto esattamente nel caso contrario. Proprio per questo motivo, il percorso che è stato fatto seguire al veicolo è stato realizzato applicando direttamente del nastro isolante nero su una superficie di colore bianco.

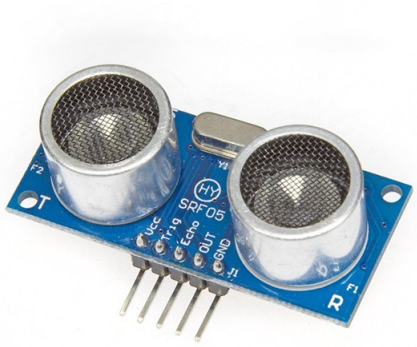
## 2.2 Sensori ottici e encoder

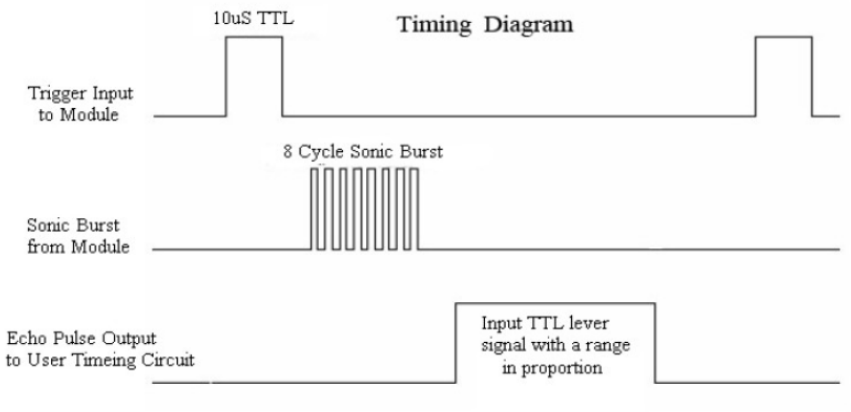





Il sensore ottico selezionato è costituito da una coppia fotoemettitore - fotorilevatore. Il fotoemettitore emette un segnale luminoso che, passando attraverso uno dei fori dell'encoder, manda in conduzione il fotorilevatore. Collegando i terminali 2 e 4 mostrati in figura a GND, il terminale 1 a $V_{CC}$ e il terminale 3 a $V_{CC}$ attraverso una resistenza di pull-up, in uscita (3) otterremo un valore logico basso quando il sensore si trova in corrispondenza di un foro dell'encoder (il fotorilevatore conduce ed è assimilabile ad un corto circuito) ed un valore logico alto in caso contrario. Pertanto, questo sensore, insieme all'encoder, potrà essere utilizzato per effettuare una stima della velocità del veicolo e implementare, di conseguenza, un semplice controllo della stessa.

## 2.3 Sensore ad ultrasuoni




Il sensore ad ultrasuoni serve per rilevare con affidabilità la presenza di un ostacolo, misurandone la distanza in modo continuo e preciso. In particolare, è possibile ricavare tale valore grazie all'emissione di onde ultrasoniche, che saranno successivamente riflesse da un potenziale bersaglio. Il funzionamento del dispositivo si basa sull'invio in ingresso di un segnale di trigger di durata pari a 10 $\mu$s, in seguito al quale verrà emesso dal sensore un burst di impulsi ultrasonici a frequenza 40 kHz. A questo punto, il pin di uscita (*ECHO*) andrà sul livello logico alto, rimanendovi fino all'istante in cui

l'impulso, riflesso dall'ostacolo, non sarà rilevato nuovamente dal sensore. L'intervallo di tempo in cui *ECHO* rimane ad 1 corrisponde esattamente al *time of fly*, ovvero al tempo di andata e ritorno dell'impulso ultrasonico, sulla base del quale è possibile risalire alla distanza tra il sensore e l'ostacolo:

$$2d = v \cdot t_{of} \tag{1}$$

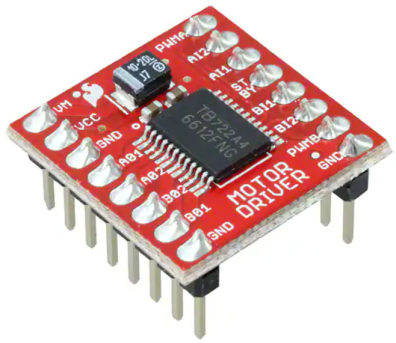
Dove $d$ è la distanza tra il sensore ad ultrasuoni e l'ostacolo, $v$ è la velocità del suono pari a 343 m/s e $t_{of}$ è il tempo di volo.
Rielaborando l'espressione riportata sopra, possiamo facilmente ricavare il tempo in $\mu$s:

$$t_{of} = 2d \cdot \frac{1}{v} = d[cm] \cdot 58 \left[\frac{\mu s}{cm}\right] \tag{2}$$

I vantaggi legati all'utilizzo di sensori a ultrasuoni sono molteplici:

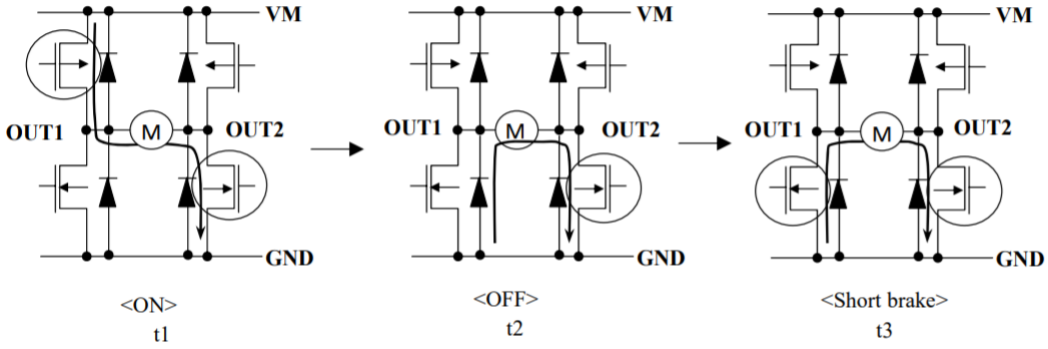
- Il colore e la trasparenza degli oggetti non influenzano in alcun modo la lettura del sensore;
- Possono essere utilizzati in ambienti poco illuminati, in quanto il buio non inficia le capacità di detezione del sensore;
- Rappresentano una soluzione a basso costo;
- Non sono influenzati dalla presenza di sporco, polvere o acqua (a meno che questi non siano presenti in quantità eccessive);
- Hanno un'elevata accuratezza e un'elevata sensibilità;
- Possono essere facilmente interfacciati con un microcontrollore.

## 2.4 Motor driver

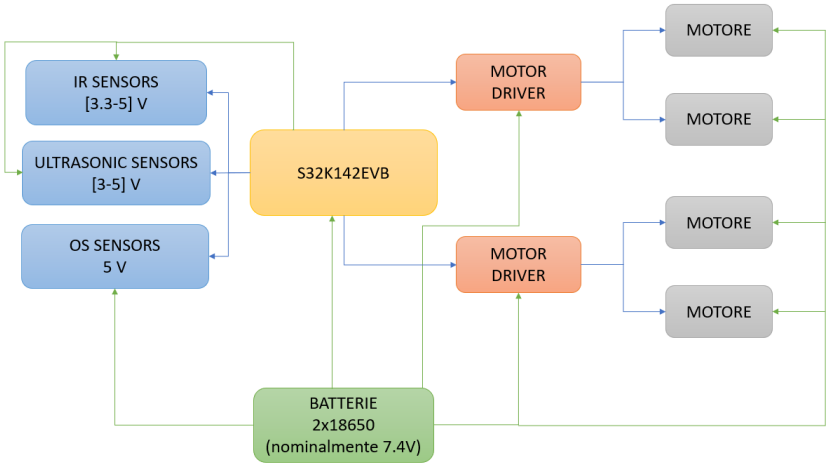



Il dispositivo considerato contiene un driver motore TB6612FNG, che può pilotare due motori DC attraverso due H-bridge. Gli ingressi *IN1* e *IN2* di ciascun ponte possono essere utilizzati per controllare il motore in una delle quattro modalità possibili: rotazione in senso orario ($CW$), rotazione in senso antiorario ($CCW$), stop ($STOP$) e frenata ($SHORT\text{-}BREAK$); la velocità di rotazione del motore, invece, potrà essere regolata attraverso l'ingresso di $PWM$. È presente, infine, un ingresso di stand-by ($STBY$) che dovrà essere tenuto sempre a $V_{CC}$ affinché il motore rimanga fuori da tale modalità. Di seguito riportiamo le uscite del driver e il conseguente pilotaggio del motore in corrispondenza delle varie combinazioni di ingressi:

| Input | | | | Output | | |
|---|---|---|---|---|---|---|
| IN1 | IN2 | PWM | STBY | OUT1 | OUT2 | Mode |
| H | H | H/L | H | L | L | Short brake |
| L | H | H | H | L | H | CCW |
| | | L | H | L | L | Short brake |
| H | L | H | H | H | L | CW |
| | | L | H | L | L | Short brake |
| L | L | H | H | OFF (High impedance) | | Stop |
| H/L | H/L | H/L | L | OFF (High impedance) | | Standby |



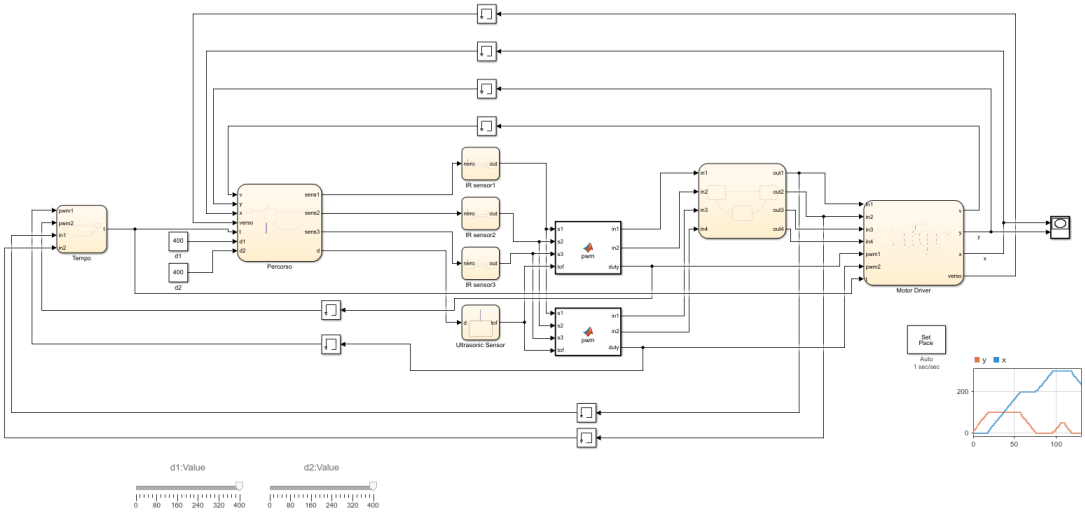

## 2.5 Schema a blocchi




Nello schema a blocchi in figura è stato riportato il modo in cui i singoli componenti sono stati collegati tra loro, indicando, in aggiunta, la tensione di alimentazione richiesta dai vari sensori. In particolare, dato che i sensori infrarossi (qui ne abbiamo utilizzati tre) e il sensore ad ultrasuoni vengono alimentati direttamente attraverso la $V_{DD}$ della scheda, si è reso necessario effettuare un controllo sulla massima corrente erogabile dalla parte di potenza della stessa. In questo caso non risultano esserci particolari pro-

blemi: la corrente complessivamente assorbita dai quattro sensori, infatti, è nominalmente pari a 90 mA (25 mA per ciascuno dei sensori infrarossi e 15 mA per il sensore ad ultrasuoni), pertanto ben al di sotto della massima corrente erogabile dalla $V_{DD}$, come è possibile verificare a partire da una breve analisi dello schematico e del datasheet della scheda.

# 3 Modello funzionale

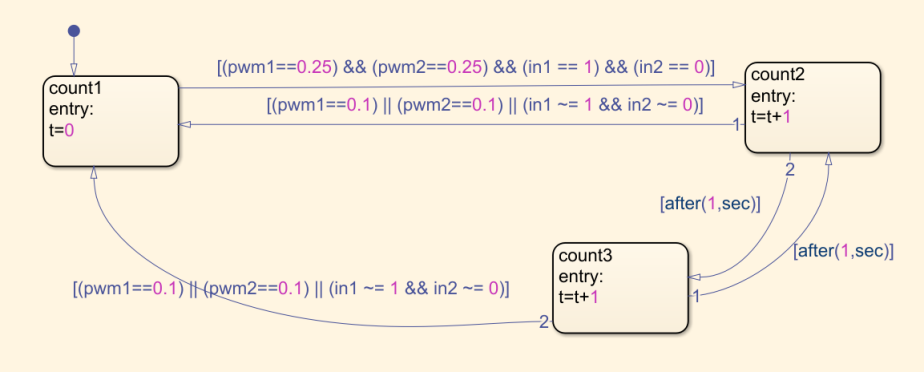



Il modello funzionale è costituito dai seguenti blocchi:

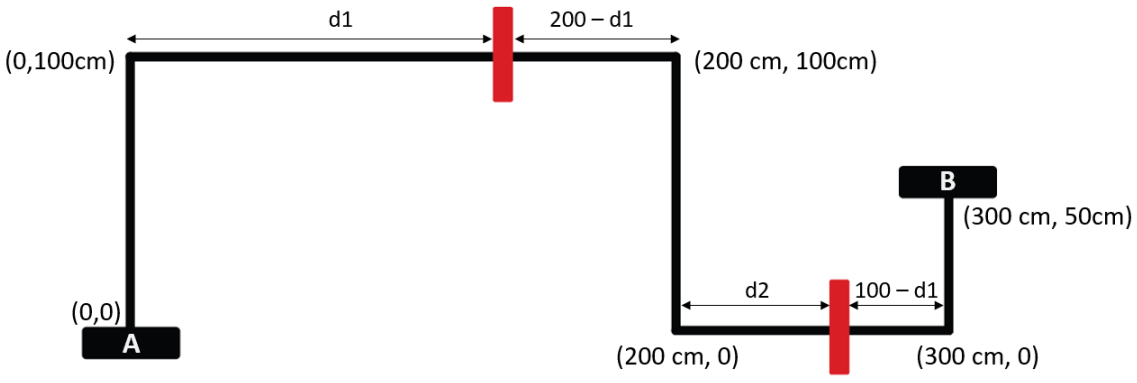
- Blocco di gestione del tempo;
- Generazione delle uscite dei sensori;
- Sensori infrarossi;
- Sensore ad ultrasuoni;
- Parte di controllo;
- Motor driver e percorso;
- Elementi di memoria per rendere possibile la retroazione.

## 3.1 Gestione del tempo

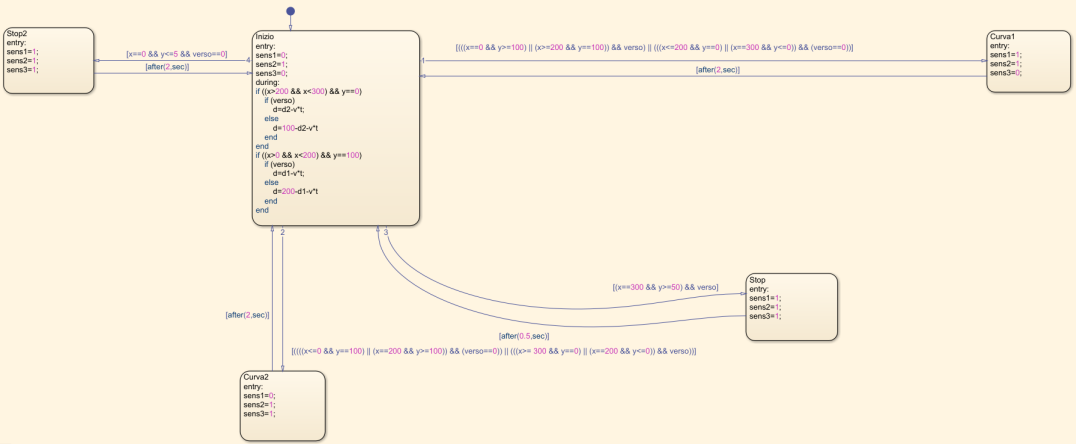
Questo blocco consente di simulare lo scorrere del tempo ed è stato modellato con una macchina a stati che, secondo dopo secondo, incrementa un contatore. In particolare, tenendo conto di come è stata realizzata la macchina a stati relativa al percorso, il conteggio viene azzerato in corrispondenza di ogni curva e di ogni rotazione di 180° del veicolo e viene poi fatto ripartire quando quest'ultimo torna a muoversi su una traiettoria rettilinea.




## 3.2 Generazione delle uscite dei sensori

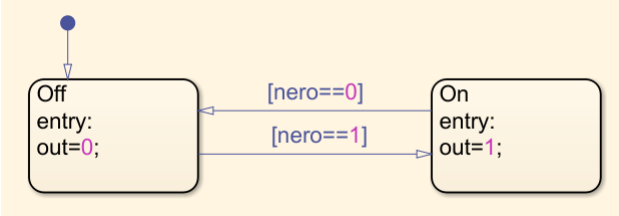


Per il modello funzionale abbiamo preso in considerazione come possibile percorso da far seguire al veicolo quello riportato in figura. Definito un sistema di coordinate spaziali, abbiamo realizzato una macchina a stati che, in funzione della posizione $(x,y)$ sul percorso e del verso di percorrenza (se *verso = 1* il veicolo si sposta dal punto A al punto B, viceversa se *verso = 0*), fornisce in uscita il colore rilevato dai sensori infrarossi e la distanza del veicolo da un potenziale ostacolo, posizionato in alcuni tratti specifici.

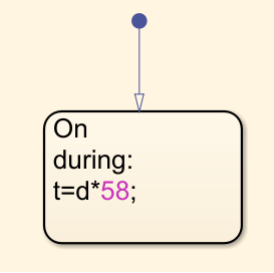



## 3.3 Modello per i sensori infrarossi e ad ultrasuoni

I sensori infrarossi sono stati modellati ricorrendo ad una semplice macchina a stati che restituisce in uscita 0 quando il sensore vede bianco e 1 quando il sensore vede nero.



Anche per il sensore ad ultrasuoni abbiamo definito una semplicissima macchina a stati che, sulla base del valore della distanza tra il sensore e l'ostacolo, restituisce in uscita il tempo di volo $t_{of}$.




Per semplicità, in questa fase abbiamo deciso di non realizzare un modello funzionale per i sensori ottici, assumendo costante la velocità del veicolo.

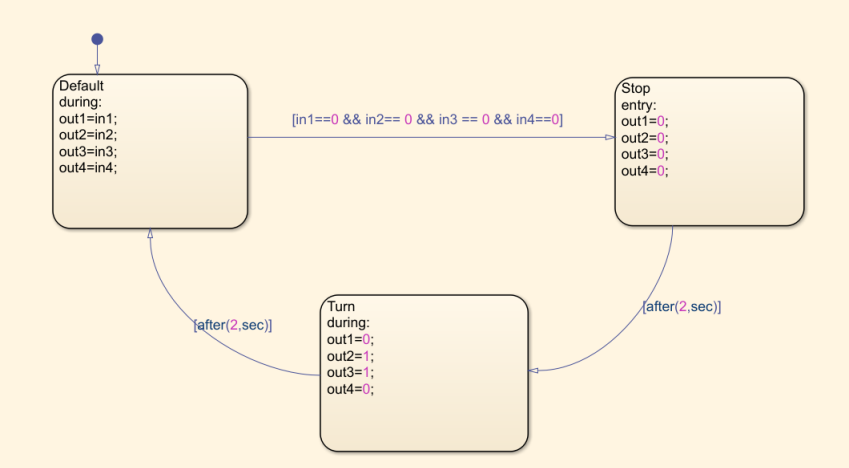
## 3.4 Modello per la parte di controllo

La parte di controllo è stata modellata con due funzioni MATLAB ed una macchina a stati, immaginando di dover controllare soltanto i motori delle due ruote anteriori. In particolare, quando il veicolo segue un percorso rettilineo abbiamo considerato il duty cycle del $PWM$ pari a 0.25 e abbiamo fissato gli ingressi *in1* e *in2* a 1 e a 0 rispettivamente (rotazione in senso orario). Nel momento in cui, invece, i sensori infrarossi rilevano la presenza di una curva, il duty cycle di uno dei due $PWM$ viene posto a 0.1, lasciando inalterati gli altri due ingressi del driver. Quando tutti i sensori infrarossi forniscono un'uscita pari a 1 o quando il sensore ad unltrasuoni rileva un tempo di volo inferiore a 290 $\mu$s (distanza dall'ostacolo inferiore a 5 cm), i motori vengono fermati ($in1 = in2 = 0$); una volta raggiunta la condizione di stop, il veicolo rimane fermo per due secondi, per poi effettuare una rotazione di 180° e ripartire seguendo il percorso nel verso contrario. La

rotazione è stata implementata invertendo gli ingressi di uno dei due driver, affinché una delle due ruote inizi a girare in senso antiorario; dopodiché, per semplicità, è stato fatto aspettare un tempo arbitrario prima di far ripartire il veicolo nella direzione opposta.

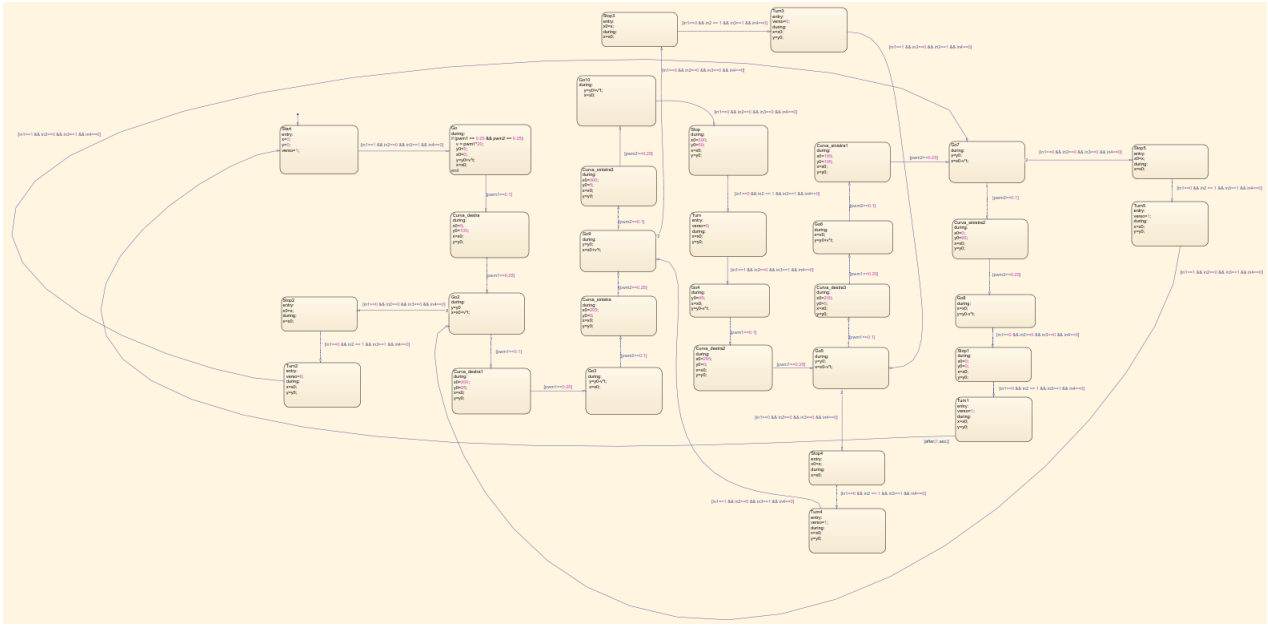
```
%pilotaggio della ruota anteriore destra
function [in1, in2, duty] = pwm(s1, s2, s3, tof)

%ruota in senso orario
in1 = 1;
in2 = 0;
duty = 0.25;

if (s1 == 0 && s2 == 1 && s3 == 0)
      duty = 0.25;
%se vi è una curva a destra, la ruota viene rallentata
elseif (s1 == 1 && s2 == 1 && s3 == 0)
      duty = 0.1;
end

%condizione di stop
if (s1 == 1 && s2 == 1 && s3 == 1)
      in1 = 0;
      in2 = 0;
end

%se il veicolo è ad una distanza <= 5 cm dall'ostacolo, si deve fermare
if (tof < 580 && tof > 0)
      in1 = 0;
      in2 = 0;
end
```

```
%pilotaggio ruota anteriore sinistra
function [in1, in2, duty] = pwm(s1, s2, s3, tof)

%ruota in senso orario
in1 = 1;
in2 = 0;
duty = 0.25;

if (s3 == 0 && s2 == 1 && s3 == 0)
      duty = 0.25;
%se vi è una curva a sinistra, la ruota viene rallentata
elseif (s3 == 1 && s2 == 1 && s1 == 0)
      duty = 0.1;
end

%condizione di stop
if (s1 == 1 && s2 == 1 && s3 == 1)
      in1 = 0;
      in2 = 0;
end

%se il veicolo è ad una distanza <= 5 cm dall'ostacolo, si deve fermare
if (tof < 580 && tof > 0)
      in1 = 0;
      in2 = 0;
end
```



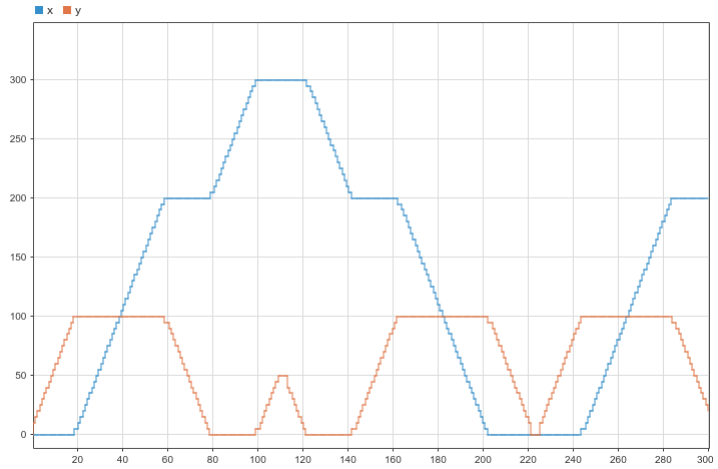
## 3.5 Modello del motor driver e del percorso

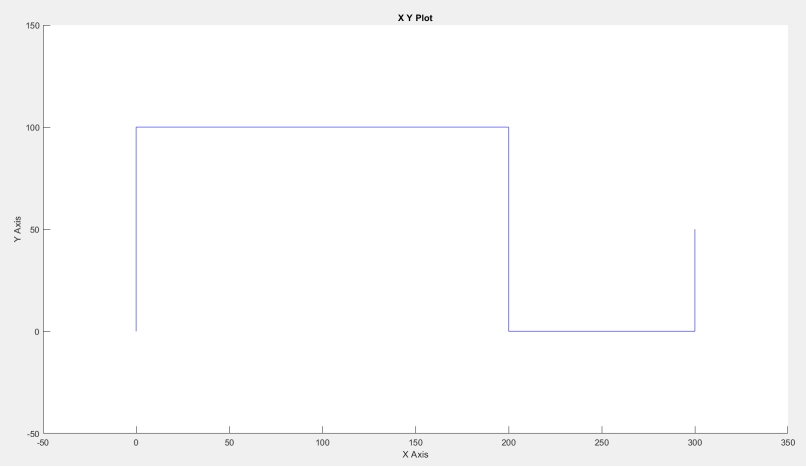
In questo caso è stato necessario mettere a punto una macchina a stati abbastanza complessa, al cui interno abbiamo inserito un blocco per ogni tratto rettilineo e uno per ogni curva. Dopodiché, per quanto riguarda gli ostacoli e i due punti di inizio e fine del percorso (A e B), abbiamo dovuto utilizzare altri blocchi per gestire le fasi di *stop* e di *turn* sia in un senso che nell'altro. La seguente macchina a stati, dunque, permette di aggiornare, al variare del tempo, la posizione $(x,y)$ del veicolo in funzione della velocità $v$ (assunta costante e pari a 5 $cm/s$) e della posizione iniziale $(x_0,y_0)$, che verrà opportunamente modificata ogni volta che il veicolo si ferma o esegue una curva.



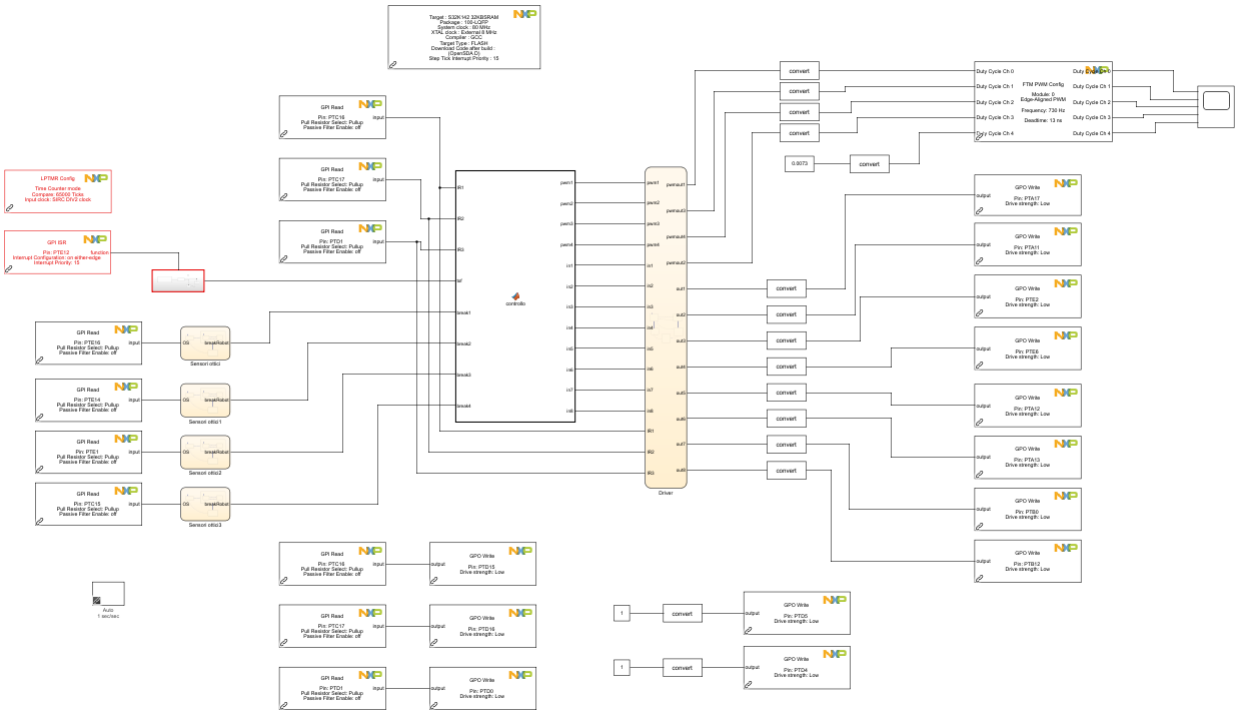
## 3.6 Simulazione

Per simulare il modello abbiamo collegato, attraverso la dashboard, degli slider alle variabili legate alla posizione degli ostacoli (*d1* e *d2*) e uno scope ai valori delle coordinate spaziali $x$ e $y$. In aggiunta, per visualizzare in maniera più esplicita il percorso seguito dal veicolo, abbiamo inserito un *plot xy*, che ci fornisce l'andamento dell'ordinata ($y$) in funzione dell'ascissa ($x$) al variare del tempo. I risultati sono quelli mostrati qua sotto:



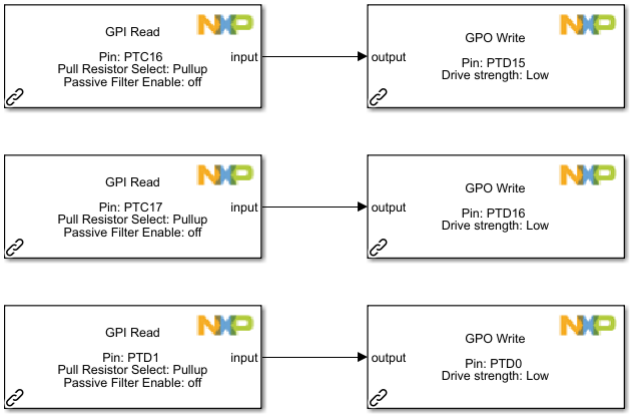
# 4 Implementazione HW



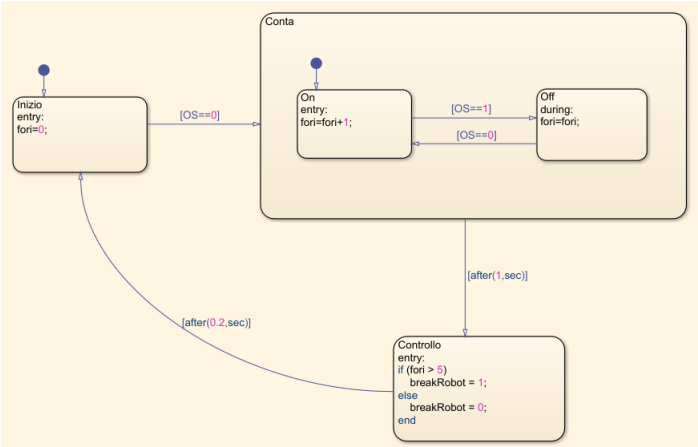
Con l'ausilio della scheda NXP, l'implementazione fisica del sistema è avvenuta attraverso la realizzazione di un modello Simulink dedito principalmente alla gestione della parte di controllo. La scheda, infatti, leggendo i segnali provenienti dai vari sensori (IR, ultrasuoni e sensori ottici), è in grado di scrivere il valore logico opportuno sui vari pin collegati ai driver motore. I componenti utilizzati sono stati, inoltre, testati singolarmente per verificarne la corretta operatività. Visti i tempi abbastanza ristretti, l'implementazione hardware è stata effettuata senza ricorrere all'utilizzo del sensore ad ultrasuoni (quindi senza la parte di riconoscimento degli ostacoli), ma verificando soltanto che il veicolo seguisse correttamente il percorso tracciato.

## 4.1 Sensori

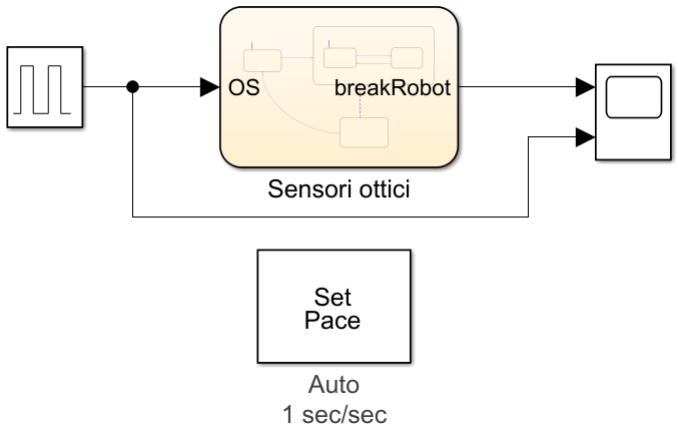
Il corretto funzionamento dei sensori infrarossi, collegati ai pin PTC16, PTC17 e PTD1 della scheda attraverso una resistenza di pull-up, è stato testato collegando tali pin ai tre LED RGB (PTD0, PTD15, PTD16), in modo da verificare che il LED corrispondente si accendesse in caso di rilevazione del colore nero da parte del sensore.

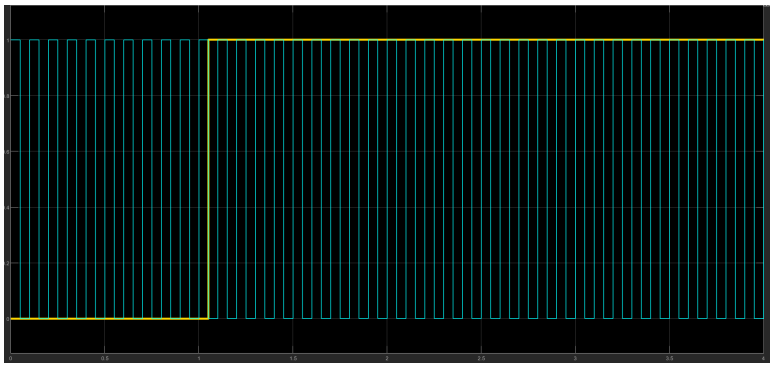


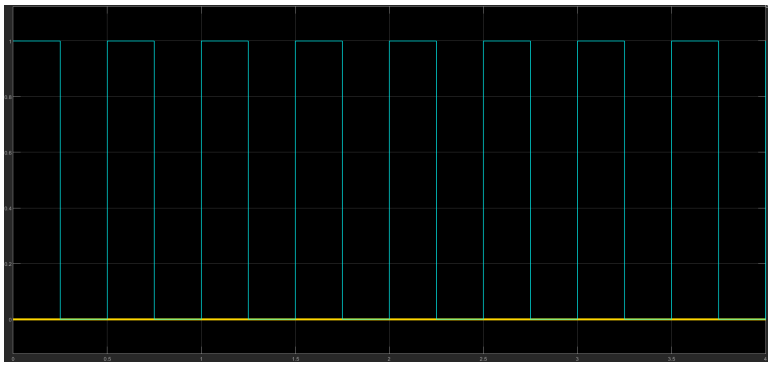

I sensori ottici, invece, sono stati utilizzati per implementare un semplice controllo della velocità attraverso la seguente macchina a stati:



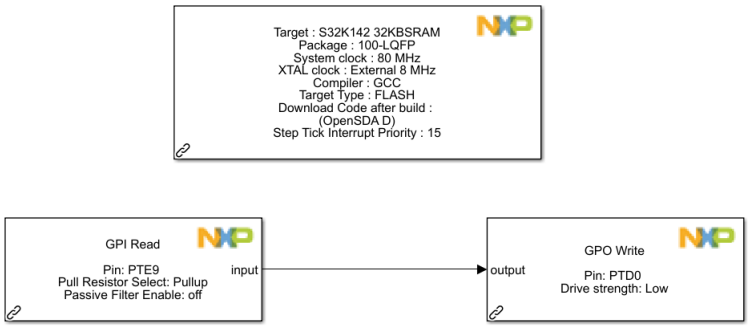
In particolare, ogni volta che il sensore rileva il passaggio del segnale luminoso attraverso un foro (uscita $OS = 0$), viene incrementata la variabile *fori*, effettuando, dopo ogni secondo, una semplice verifica: se il numero di fori è maggiore di 5, allora vuol dire che il veicolo si sta muovendo ad una velocità più alta rispetto a quella da noi prevista; di conseguenza, verrà implementata una frenata settando a 1 il valore della variabile *breakRobot*. Il controllo è stato realizzato considerando un numero di fori pari a 5 in quanto, tenuto conto del numero di fori da cui è costituito ciascun encoder (20) e tenuto conto delle dimensioni della ruota (66mm di diametro), se vogliamo che il veicolo non superi una velocità di 5 *cm/s*, il segnale luminoso non dovrà attraversare più di 5 fori al secondo.
Per verificare il funzionamento della macchina a stati, le abbiamo fornito in ingresso un segnale pulsato, controllando che, al di sopra di una certa frequenza, l'uscita venisse settata correttamente ad 1.



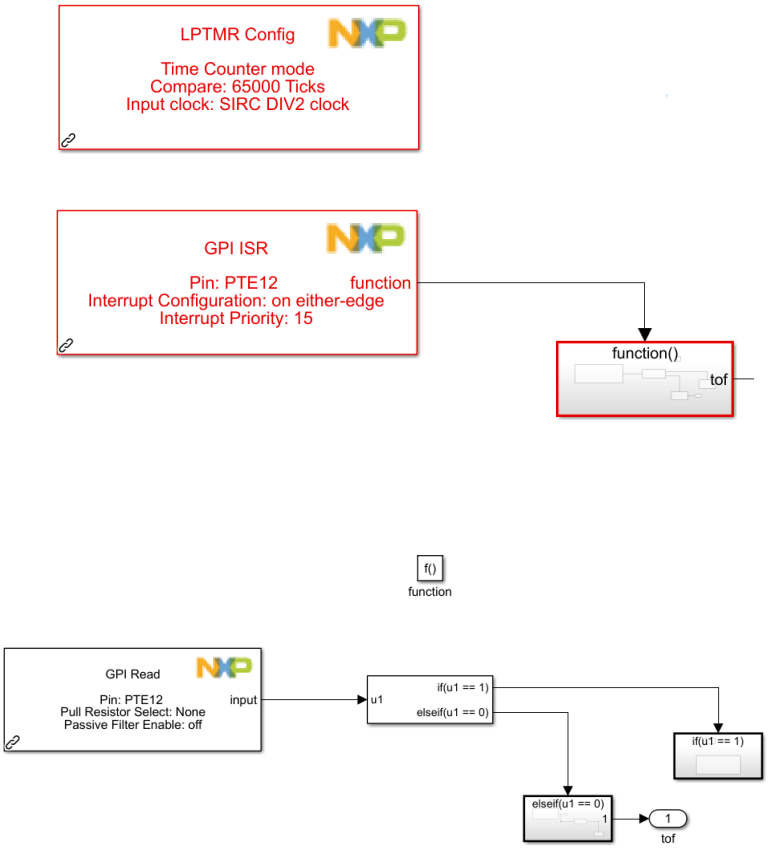


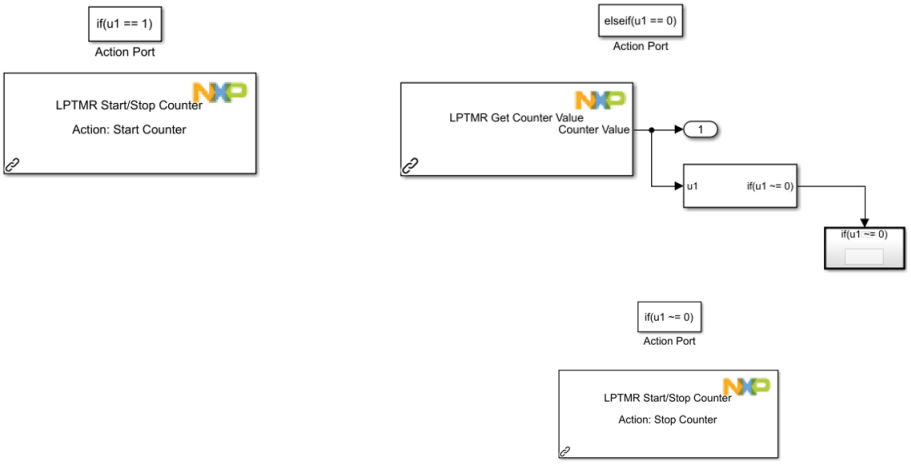


Infine, come già fatto con i sensori infrarossi, anche i sensori ottici sono stati testati collegandone l'uscita ad uno dei LED della scheda e controllando che quest'ultimo si accendesse correttamente una volta messa in movimento la ruota corrispondente.

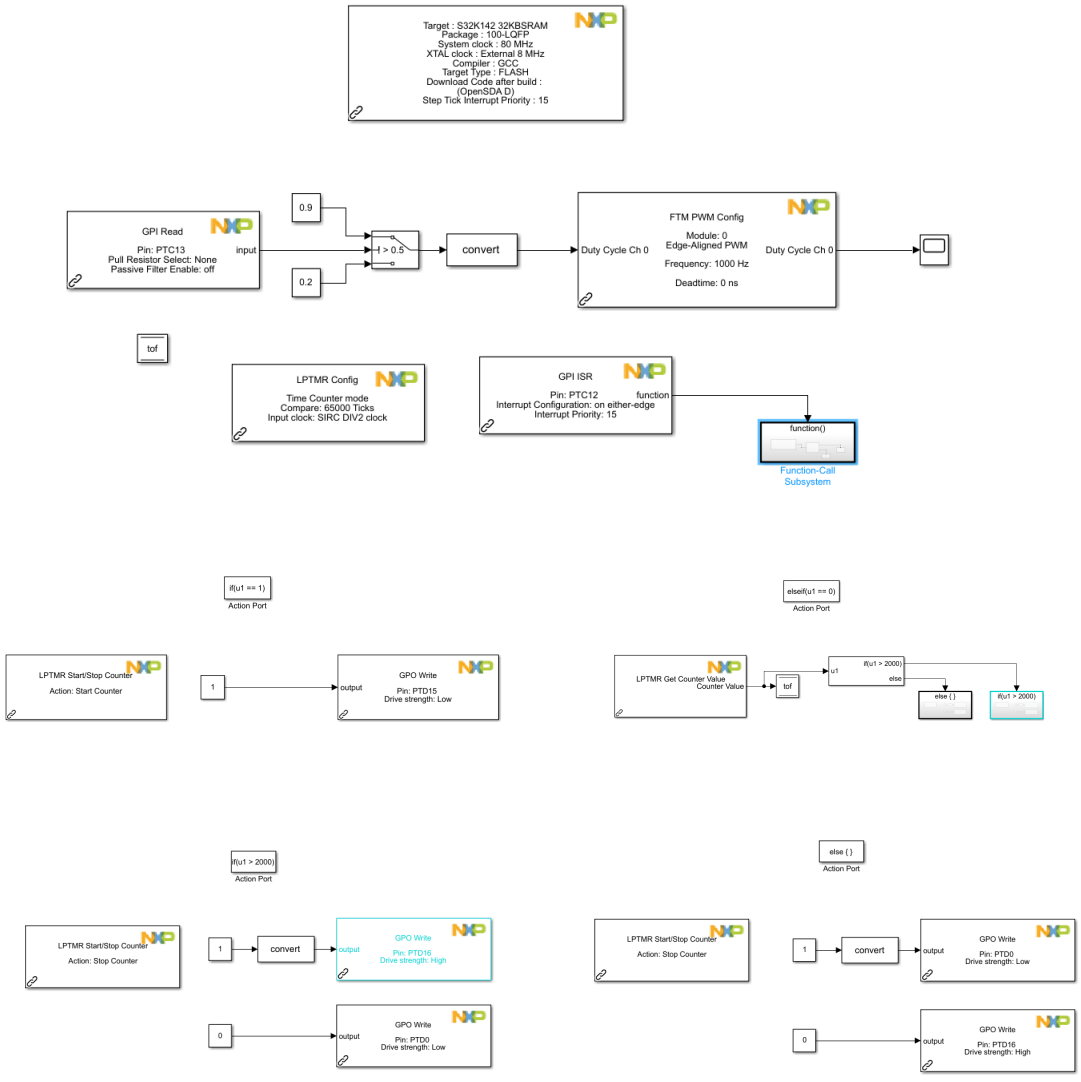


Per quanto riguarda il sensore ad ultrasuoni, invece, il $t_{of}$ è stato ricavato utilizzando il blocco *LPTMR*. L'uscita *ECHO* del sensore è stata collegata al pin PTE12, sul quale viene generato un segnale di *interrupt*, sia in presenza di un fronte in salita che in presenza di un fronte in discesa. La funzione chiamata dall'*interrupt* permette di far partire il timer quando viene letto un valore logico alto sul pin ($ECHO = 1$) e di leggere il valore del contatore in caso contrario ($ECHO = 0$), resettando successivamente il conteggio.

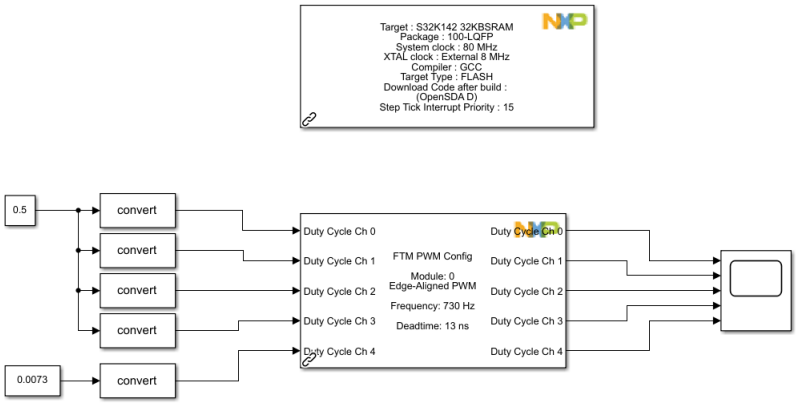





La funzionalità del timer è stata verificata generando un segnale *PWM* di frequenza pari a 1KHz e $\delta = 20\%$ o $\delta = 90\%$ a seconda della pressione o meno del tasto collegato a PTC13; in questo modo, il segnale rimarrà alto per 200 $\mu$s oppure per 900 $\mu$s. Il blocco *LPTMR* lavora con una frequenza di clock pari a 4MHz (e quindi con un periodo di clock di 0.25 $\mu$s) e fornisce il risultato del conteggio in *clock ticks*; pertanto, se il segnale rimane alto per 200 $\mu$s, ci aspettiamo che il timer restituisca un valore pari a $\frac{200}{0.25} = 800$ *clock ticks*, mentre se il segnale rimane alto per 900 $\mu$s, ci aspettiamo che restituisca invece un valore pari a $\frac{900}{0.25} = 3600$ *clock ticks*. Per assicurarci che il sistema funzioni correttamente, quindi, abbiamo fatto in modo che nel momento in cui parte il conteggio si accenda il LED rosso, effettuando poi un controllo sul valore restituito dal timer: nel caso in cui tale valore sia maggiore di 2000 si accenderà il LED verde, mentre si accenderà il LED blu in caso contrario.




Il segnale di trigger fornito in ingresso al sensore è stato generato con un *PWM* a frequenza 730 Hz e duty cycle pari a 0.0073. In questa maniera, il segnale rimarrà alto esattamente per 10 $\mu$s e garantirà una frequenza di ripetizione opportuna del burst di impulsi ultrasonici. La corretta generazione dell'impulso è stata, inoltre, verificata sperimentalmente collegando il pin corrispondente sulla scheda alla sonda di un oscilloscopio.




## 4.2 Controllo

Il controllo è stato implementato facendo ricorso ad una funzione MATLAB e ad una macchina a stati che, a partire dalle uscite fornite dai vari sensori, permettono di pilotare in maniera opportuna i quattro motori del veicolo. Quando il veicolo percorre una traiettoria rettilinea, i due ingressi di ciascun ponte ad H sono stati configurati in modo che tutte le ruote girino in senso orario, mentre il duty cycle associato ai segnali $PWM$ è stato assunto pari a 0.5. Unica eccezione riguarda la ruota posteriore sinistra, per la quale è stato considerato un valore di duty cycle pari a 0.75, vista la minore velocità di rotazione osservata sperimentalmente. Quando i due sensori ad infrarossi posti ai lati rilevano bianco e quello al centro rileva nero, il veicolo prosegue nella sua traiettoria rettilinea; nel momento in cui, invece, uno dei due sensori laterali rileva nero, il veicolo inizia a curvare. In tal caso, il $PWM$ delle due ruote di destra (curva a destra) o il $PWM$ delle due ruote di sinistra (curva a sinistra) viene portato a 0, mentre quello delle altre due viene lasciato inalterato. Quando tutti e tre i sensori forniscono 1 in uscita o

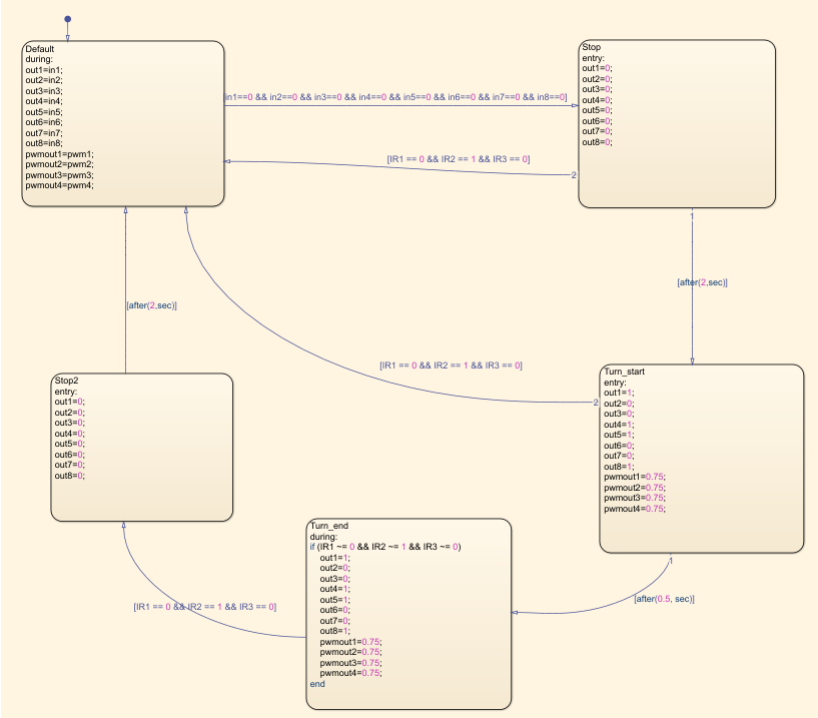
quando il tempo di volo misurato in *clock ticks* risulta essere inferiore a 1160 (290 $\mu$s), i motori vengono stoppati, azzerando tutti gli ingressi dei quattro ponti ad H. Il segnale *breakRobot* descritto precedentemente, invece, viene utilizzato per ridurre il duty cycle del $PWM$ nel caso in cui il veicolo si stia muovendo troppo velocemente.

La macchina a stati a valle della funzione gestisce la rotazione di 180° del veicolo dopo lo stop. Passati due secondi, due delle ruote vengono fatte ruotare in senso orario e le altre due in senso antiorario; nel momento in cui i sensori ad infrarossi rilevano nuovamente bianco, nero e bianco rispettivamente, il veicolo torna a muoversi seguendo una traiettoria rettilinea sul percorso.

```
function [pwm1,pwm2,pwm3,pwm4,in1,in2,in3,in4,in5,in6,in7,in8] =
controllo(IR1,IR2,IR3,tof,break1,break2,break3,break4)
in1 = 1;
in2 = 0;
in3 = 1;
in4 = 0;
in5 = 1;
in6 = 0;
in7 = 1;
in8 = 0;
pwm1 = 0.50;
pwm2 = 0.50;
pwm3 = 0.75;
pwm4 = 0.50;

%dritto
if (IR1 == 0 && IR2 == 1 && IR3 == 0) % 0 se vede bianco, 1 se vede nero
    pwm1 = 0.50;
    pwm2 = 0.50;
    pwm3 = 0.75;
    pwm4 = 0.50;
end
```

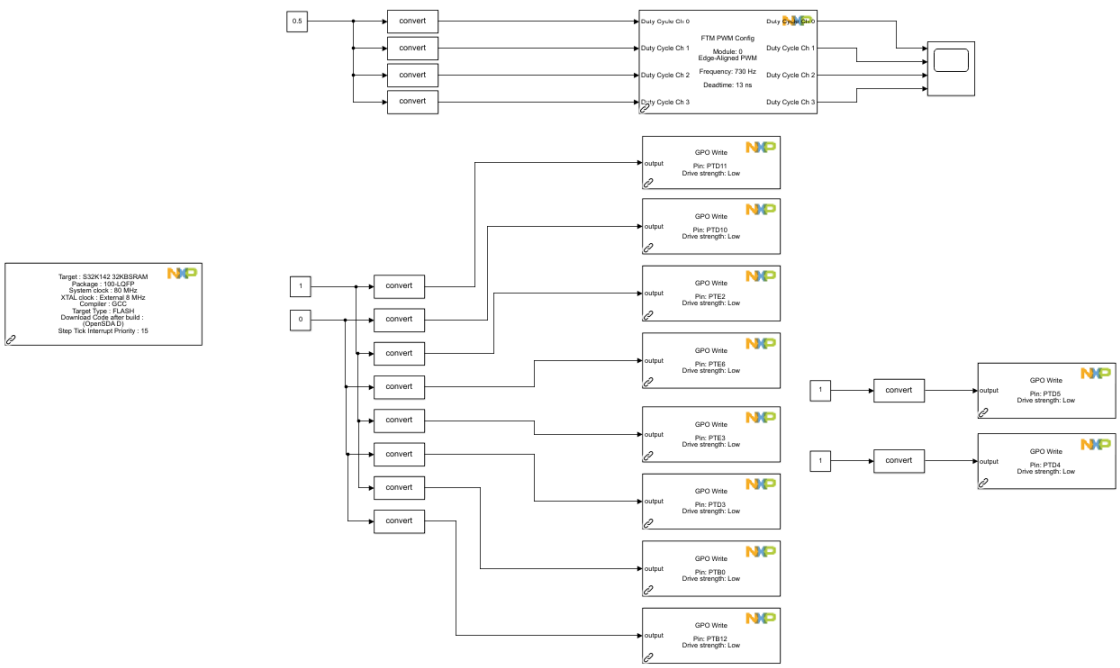
```
%curva a destra
if (IR1 == 1 && IR2 == 1 && IR3 == 0 || IR1 == 1 && IR2 == 0 && IR3 == 0)
    pwm2 = 0;
    pwm4 = 0;
end

%curva a sinistra
if (IR1 == 0 && IR2 == 1 && IR3 == 1 || IR1 == 0 && IR2 == 0 && IR3 == 1 )
    pwm1 = 0;
    pwm3 = 0;
end

%stop
if (IR1 == 1 && IR2 == 1 && IR3 == 1)
    in1 = 0;
    in2 = 0;
    in3 = 0;
    in4 = 0;
    in5 = 0;
    in6 = 0;
    in7 = 0;
    in8 = 0;
end

%ostacolo, stop
if (tof > 0 && tof < 1160) %1160 ticks corrispondente a 290us considerando una f=4MHz
    in1 = 0;
    in2 = 0;
    in3 = 0;
    in4 = 0;
    in5 = 0;
    in6 = 0;
    in7 = 0;
    in8 = 0;
end

if (break1)
    pwm1 = pwm1 - 0.1;
```

```
end

if (break2)
    pwm2 = pwm2 - 0.1;
end

if (break3)
    pwm3 = pwm3 - 0.1;
end

if (break4)
    pwm4 = pwm4 - 0.1;
end
```

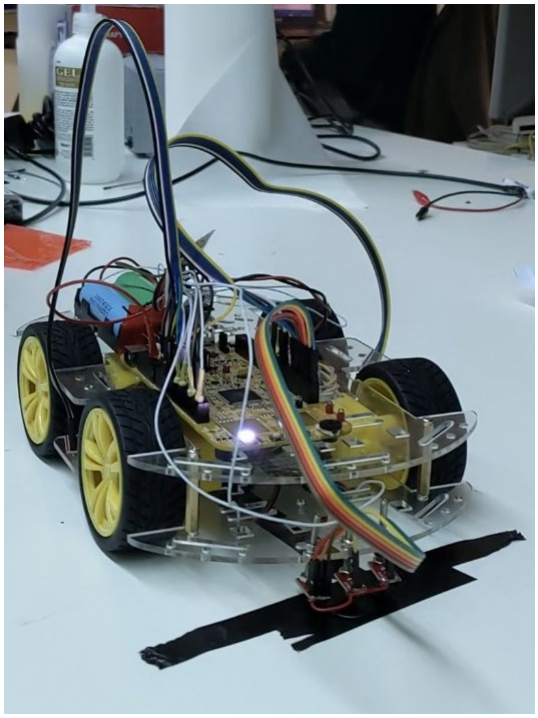


## 4.3 Motor driver

La macchina a stati descritta precedentemente produce dodici uscite: gli otto ingressi dei ponti ad H e i quattro segnali di *PWM* utilizzati per regolare la velocità di rotazione dei motori. Ciascuna di queste uscite è stata, quindi, opportunamente connessa ad uno dei pin della scheda, a cui è stato poi collegato, in fase di montaggio, il pin corrispondente del driver. I segnali di *STBY* sono stati settati ad un valore costante pari a 1, mentre i segnali di *PWM* sono stati generati sfruttando uno degli appositi blocchi NXP per il controllo motore e verificati in laboratorio con l'ausilio di un oscilloscopio. A valle della fase di montaggio è stato possibile, inoltre, testare la configurazione dei motor driver assegnando dei valori costanti ai vari ingressi e verificando, di conseguenza, che i motori ruotassero correttamente.



# 5 Risultato finale e conclusioni



In conclusione, possiamo affermare che l'obiettivo di base sia stato pienamente raggiunto; il veicolo, infatti, esegue correttamente sia i tratti rettilinei che le curve ed è in grado di arrestarsi, ruotare e percorrere il tracciato nel senso contrario. Si potrebbe ovviamente pensare, come ulteriore funzionalità, di utilizzare anche il sensore ad ultrasuoni, eventualmente con l'ausilio di un servomotore che permetta al veicolo di ispezionare l'ambiente circostante ed evitare possibili ostacoli presenti in qualsiasi direzione.