

1222 · 2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Progetto prova finale 2023/24:

**Realizzazione di un
“SELF-BALANCING ROBOT”**

1222 · 2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Proposta di progetto 3 (Marzo 2024):

Componenti del gruppo:

Osello Alberto
Parolo Jacopo
Pauro Alex
Pizzolato Matteo
Ragogna Davide
Salvador Moreno
Tocchet Alessandro

Relatore:
Antonello Riccardo

Correlatore:
Michieletto Giulia

Il progetto consiste nella creazione di un robot autobilanciante, un veicolo robotico capace di mantenersi in equilibrio su due ruote utilizzando sensori di movimento e algoritmi di controllo. Questo robot verrà utilizzato per insegnare concetti fondamentali della teoria del controllo automatico, come la stabilizzazione di un sistema in equilibrio instabile e la struttura classica di un sistema di controllo per dispositivi mecatronici. Il telaio del robot ospiterà un'unità di controllo basata su una scheda a microcontrollore, che sarà interfacciata con i sensori di movimento. Inoltre, questa unità di controllo dovrà essere in grado di trasmettere dati telemetrici e ricevere comandi da una stazione di monitoraggio remota tramite un canale di comunicazione wireless. Per lo sviluppo del software di controllo è prevista l'applicazione di moderne metodologie basate sul paradigma della prototipazione rapida di sistemi di controllo (rapid control prototyping).

- ✓ Re-design del telaio
- ✓ Disegno CAD 3D del robot
- ✓ Comando di motori in corrente continua
- ✓ Controllo retroazionato
- × Controllo remoto (componente modulare)
- ✓ Realizzazione del prototipo
- ✓ Sostituzione scheda di controllo
- ✓ Interfacciamento con la sensoristica
- ✓ Stima via software di dati utili al controllo

1222 · 2022
800
ANNI

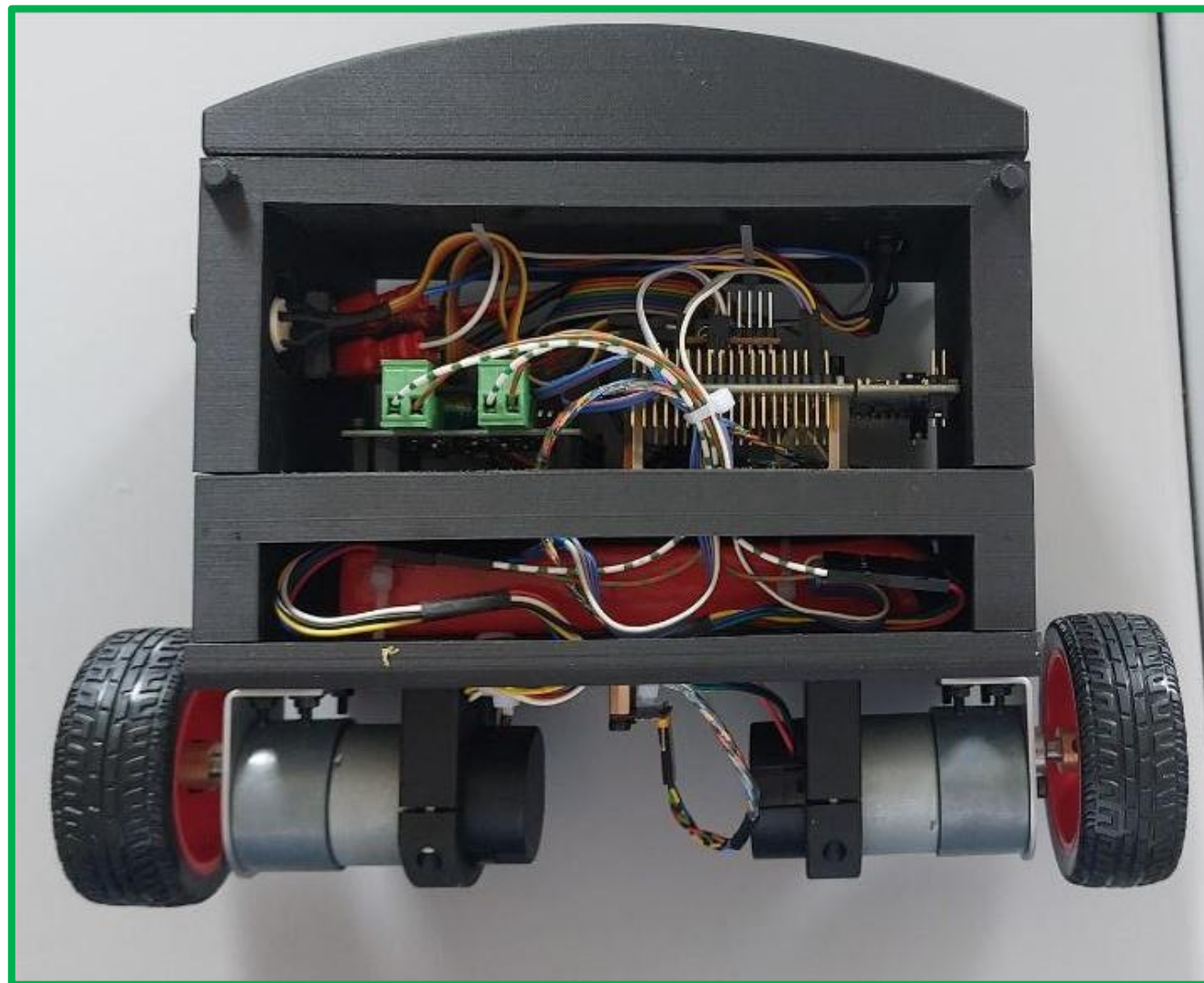
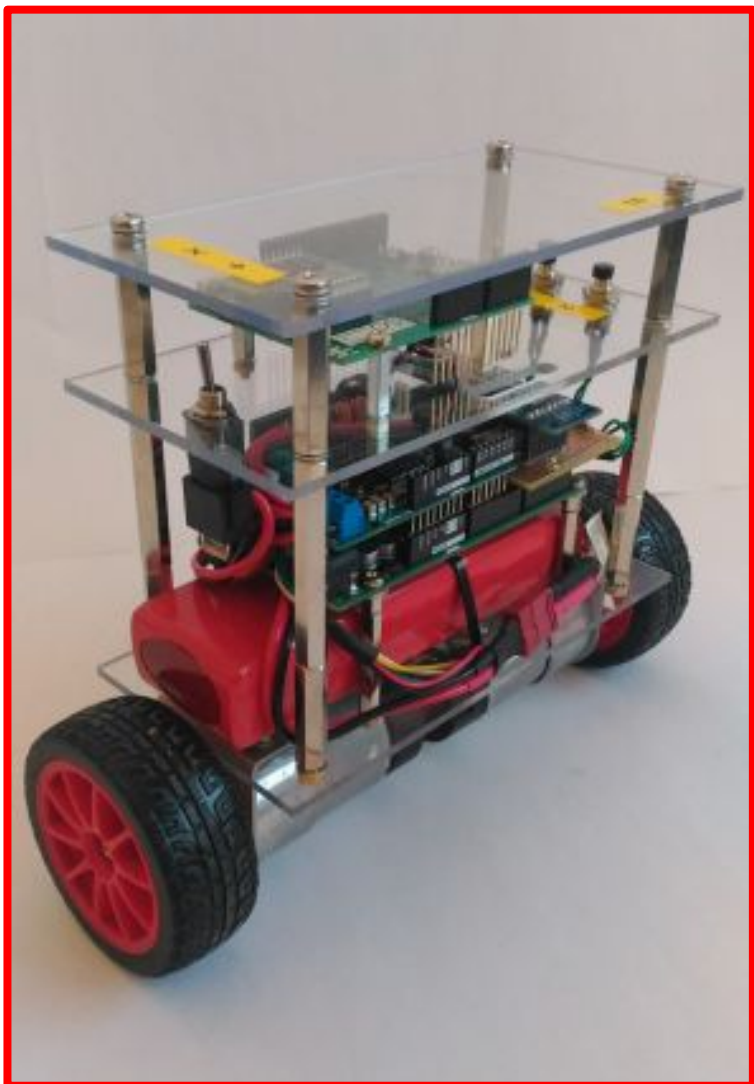


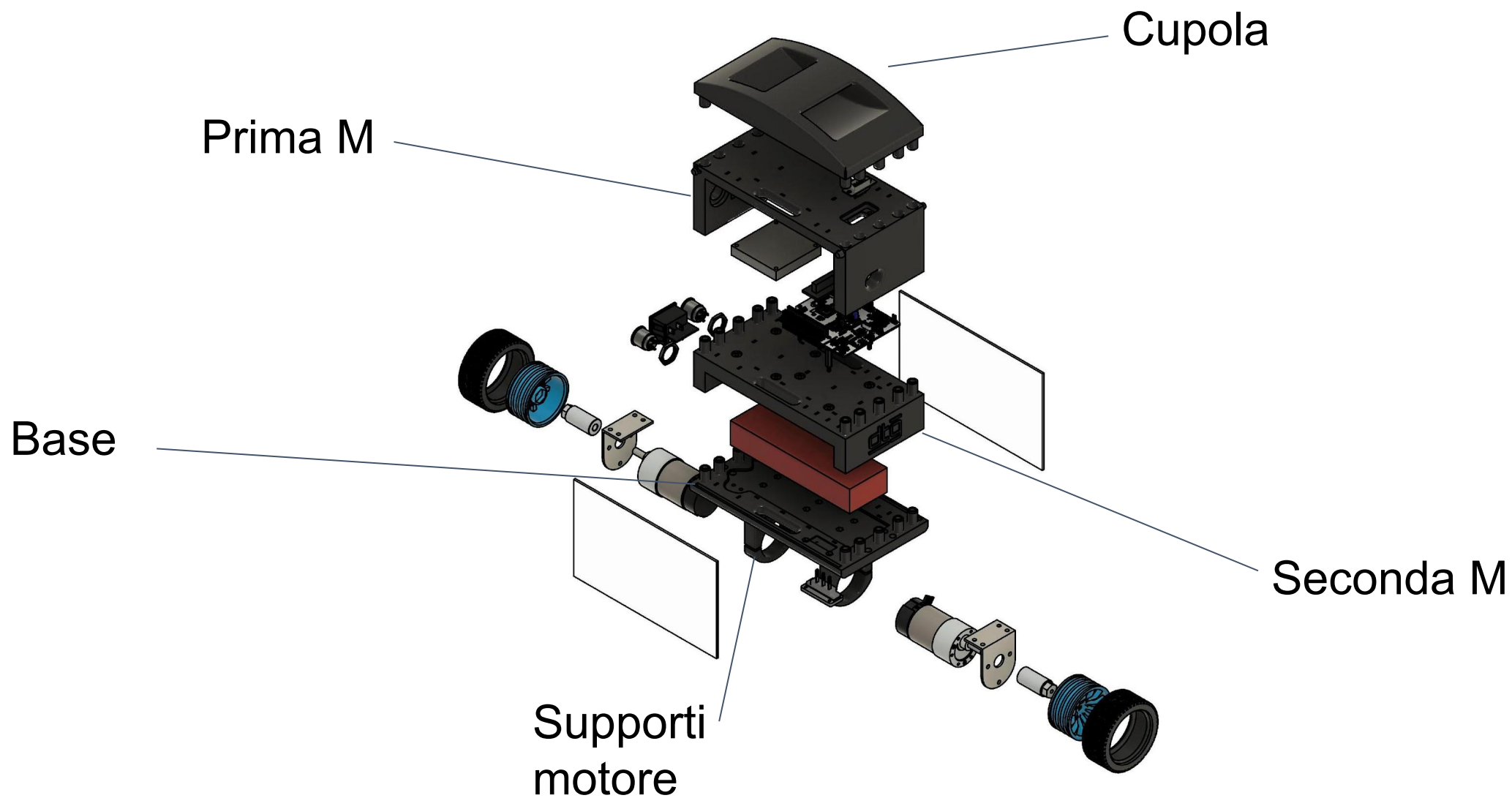
UNIVERSITÀ
DEGLI STUDI
DI PADOVA

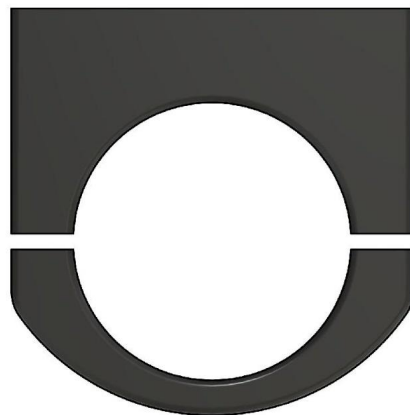


PROGETTAZIONE E REALIZZAZIONE MECCANICA

Confronto tra Vecchio e Nuovo

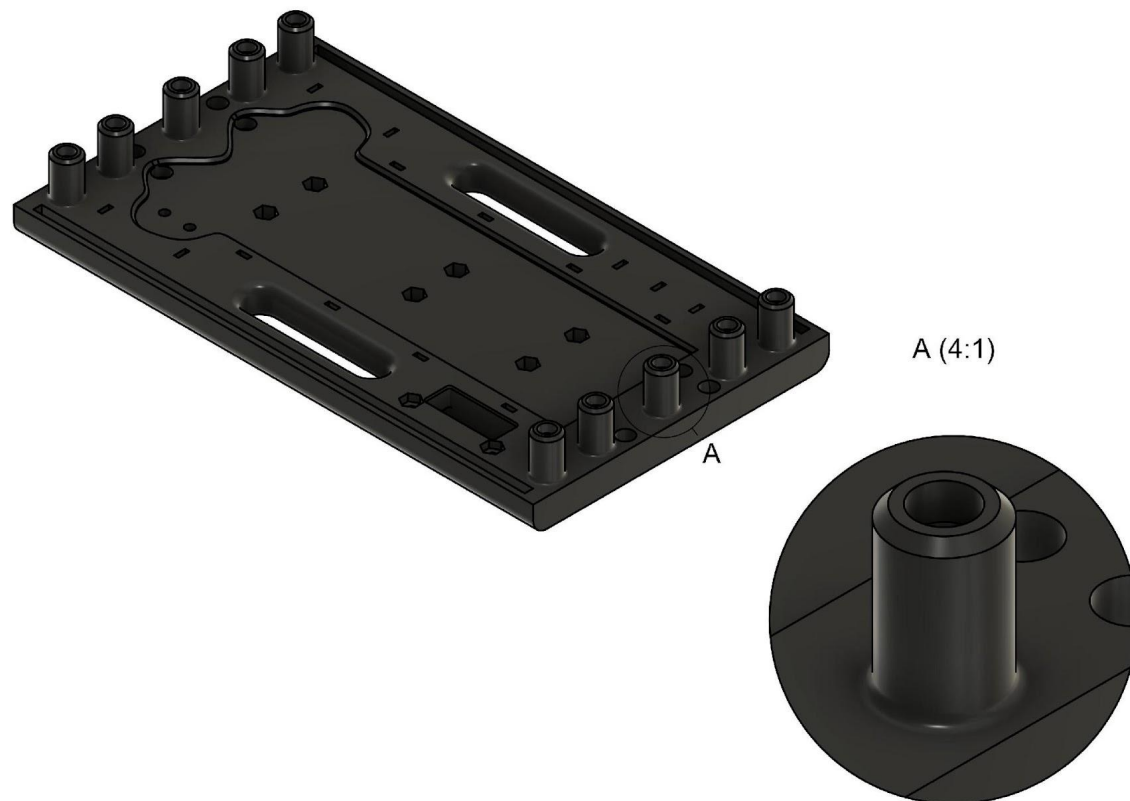


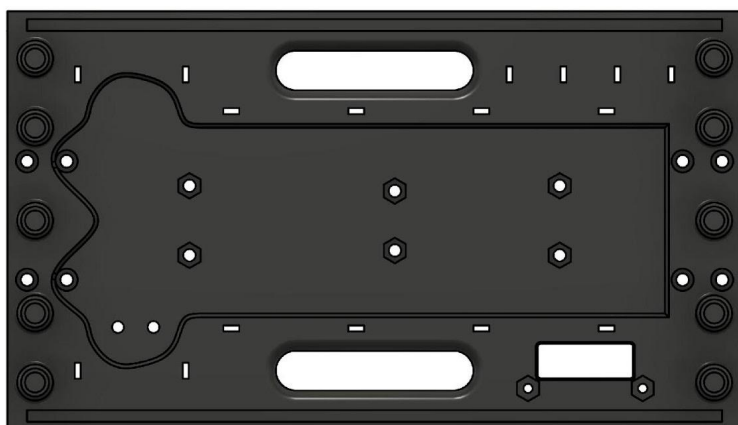
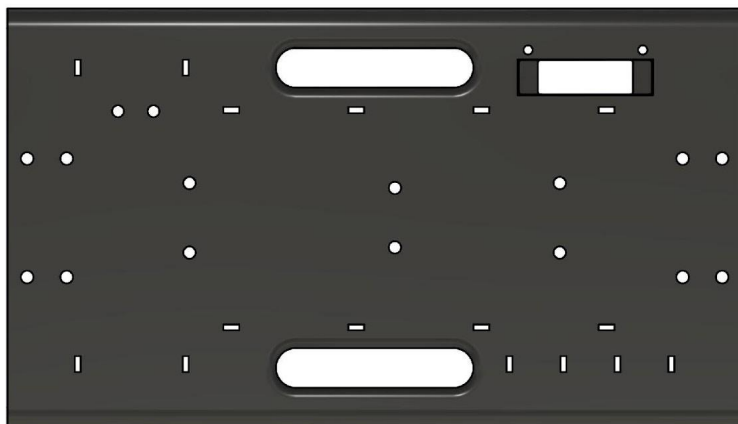




- Supporto ulteriore per il motore

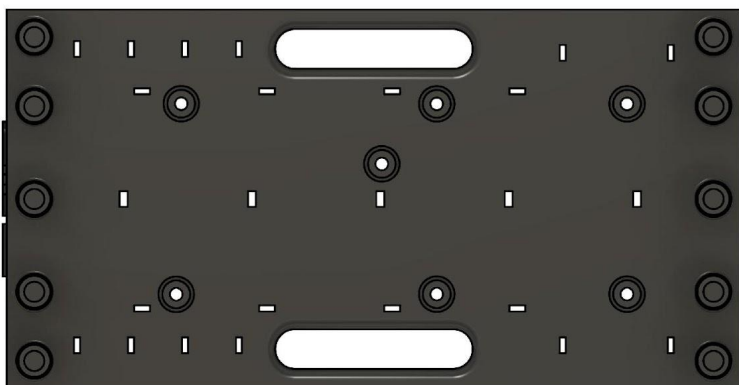
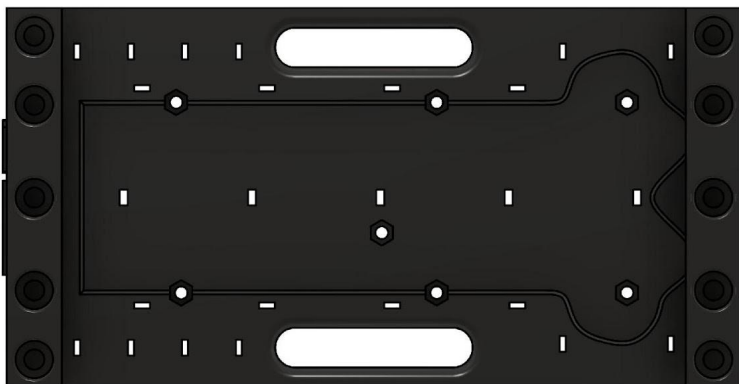
- Elemento di accoppiamento: Spina





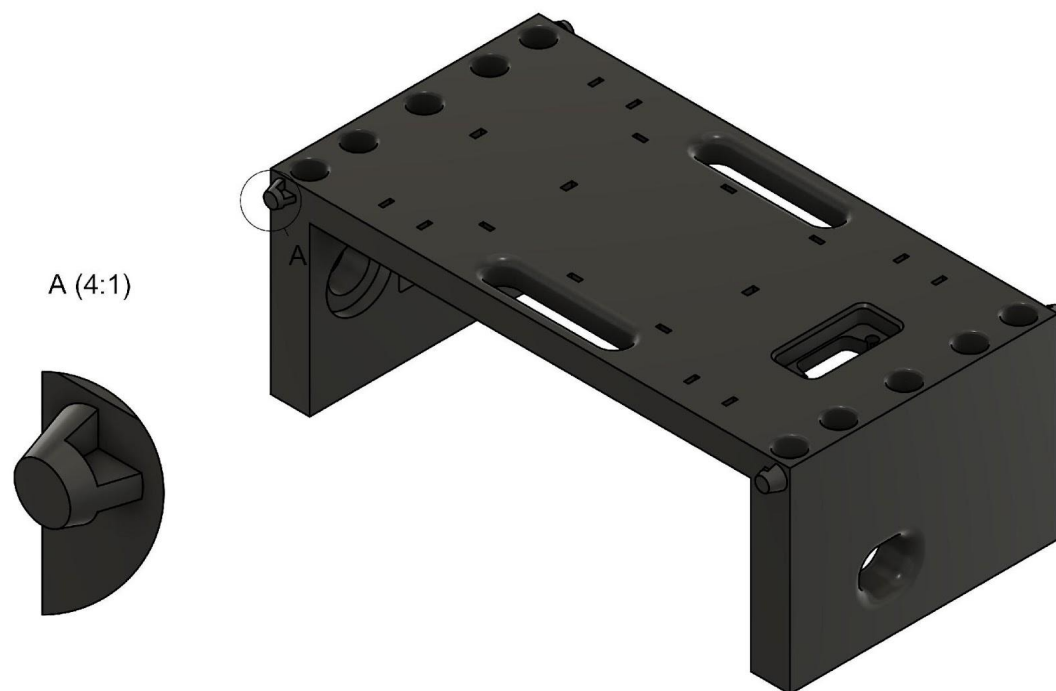
Ospita:

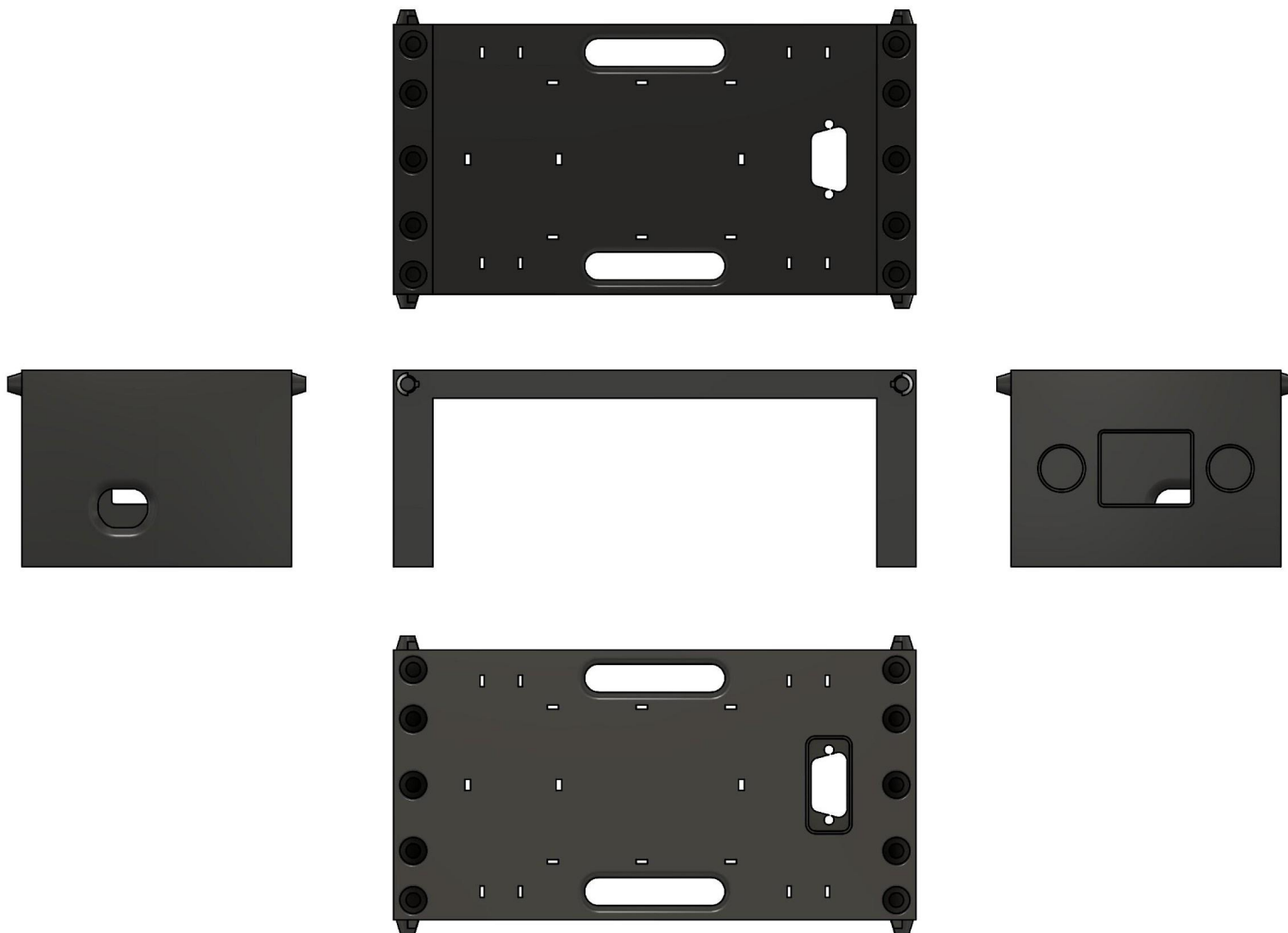
- Batteria
- Connettore di ricarica
- Imu
- Supporti motori



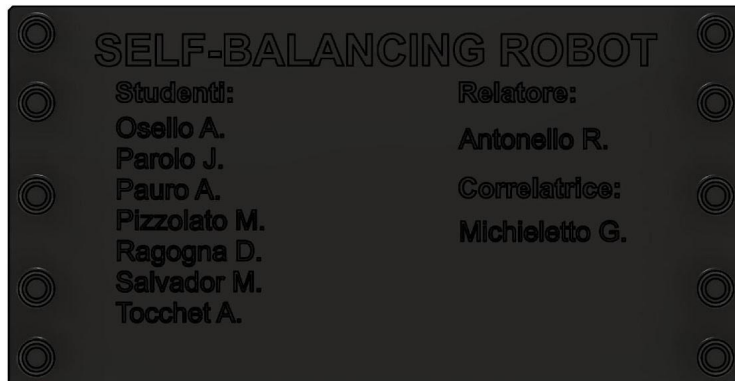
- Sopra ospita le schede
- Sotto chiude la batteria

- Elemento particolare:
Tampone





- Ospita:
 - Interruttori
 - Connettore per l'aggiunta di un ulteriore piano
- Permette la connessione diretta con la scheda



- Funge da maniglia

2 tipologie principali:

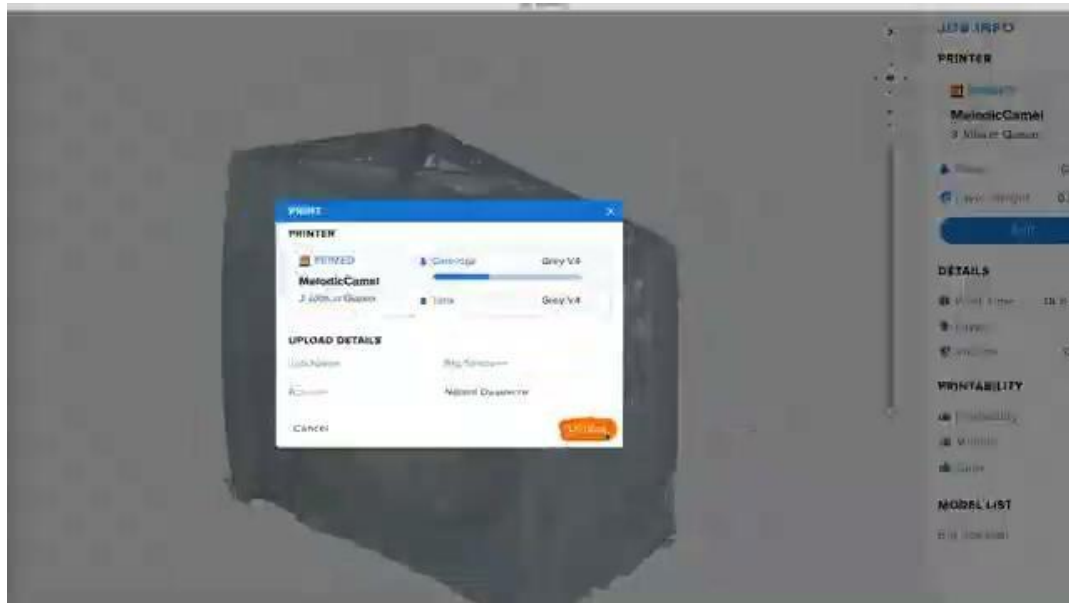


SLA: stampante a resina

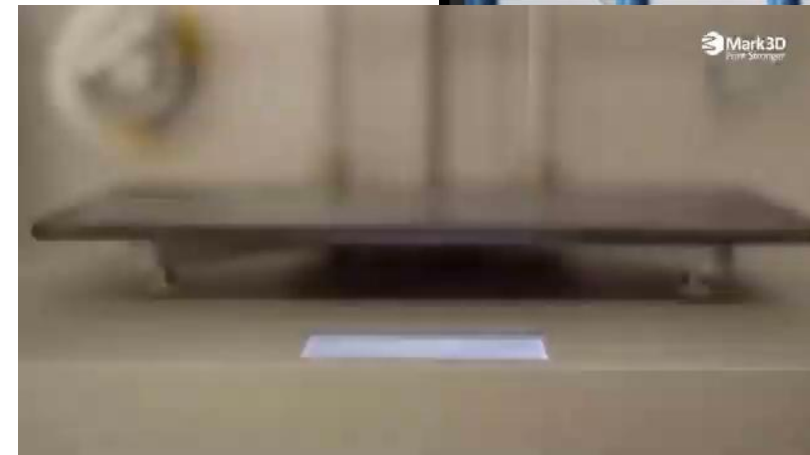
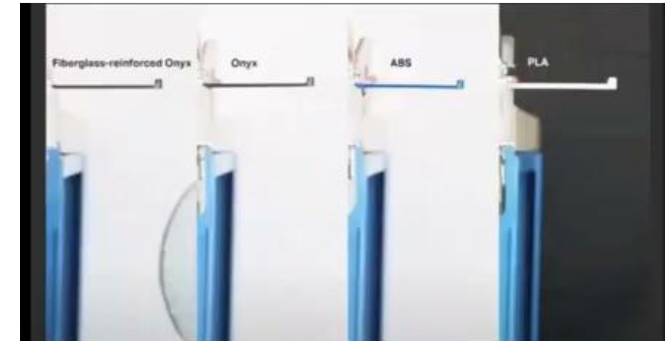


FDM: stampante a filamento

2 tipologie principali:



SLA: stampante a resina



FDM: stampante a filamento

2 tipologie principali:

SLA: stampante a resina

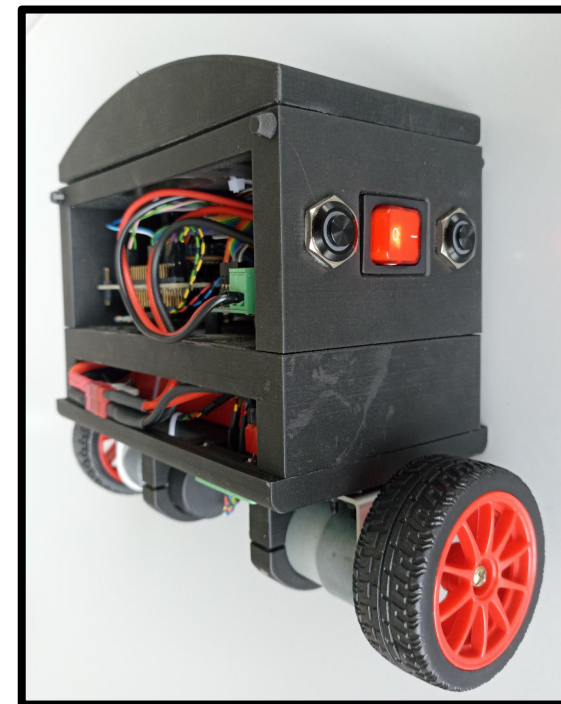
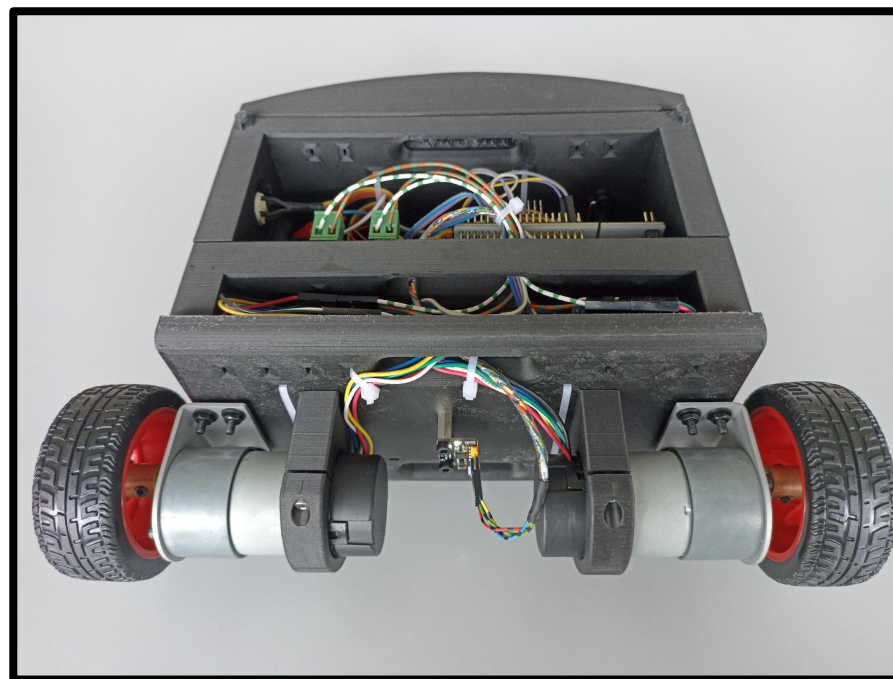
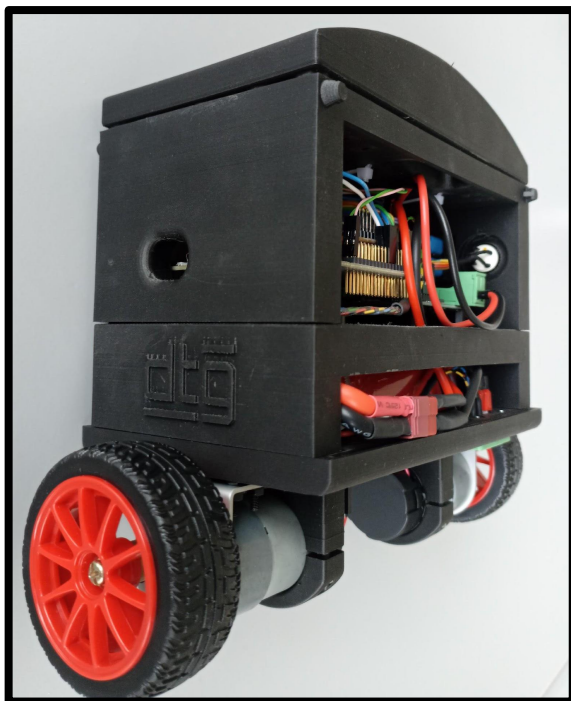
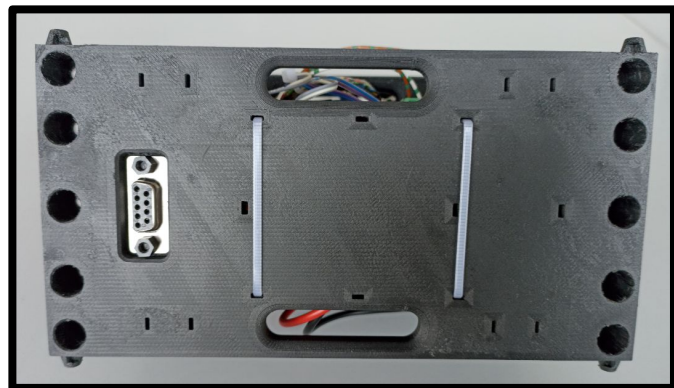
- ✓ Molteplicità di materiali disponibili
- ✓ Velocità di stampa
- ✓ Qualità di stampa
- × Componente più massiccio
- × Costosa
- × Processo più laborioso

FDM: stampante a filamento

- ✓ Economica
- ✓ Componenti svuotati
- ✓ Semplicità di stampa
- × Componente anisotropo
- × Rimozione supporti
- × Qualità di stampa

DATO	VALORE NUMERICO
Massa totale del robot	1,71 <i>kg</i>
Volume	1,582 <i>dm</i> ³
Distanza del baricentro dall'asse delle ruote	51,6 <i>mm</i>
Inerzia rispetto all'asse motore	0,0105 <i>kg m</i> ²

Montaggio e risoluzione dei problemi



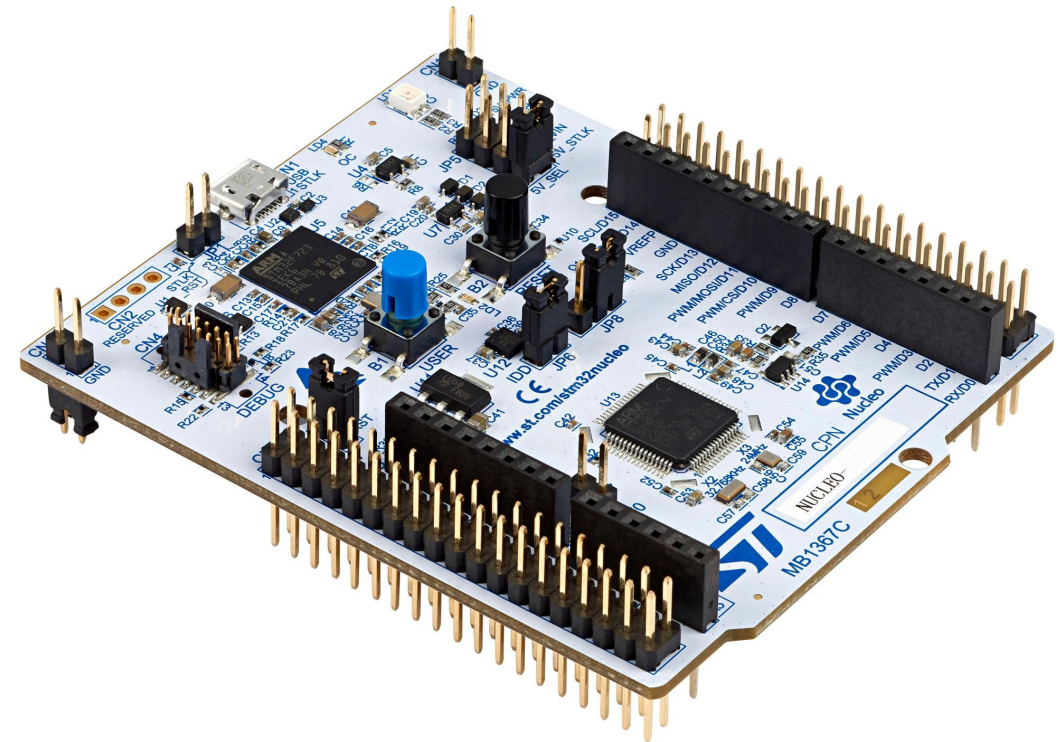
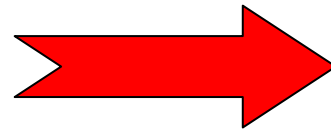
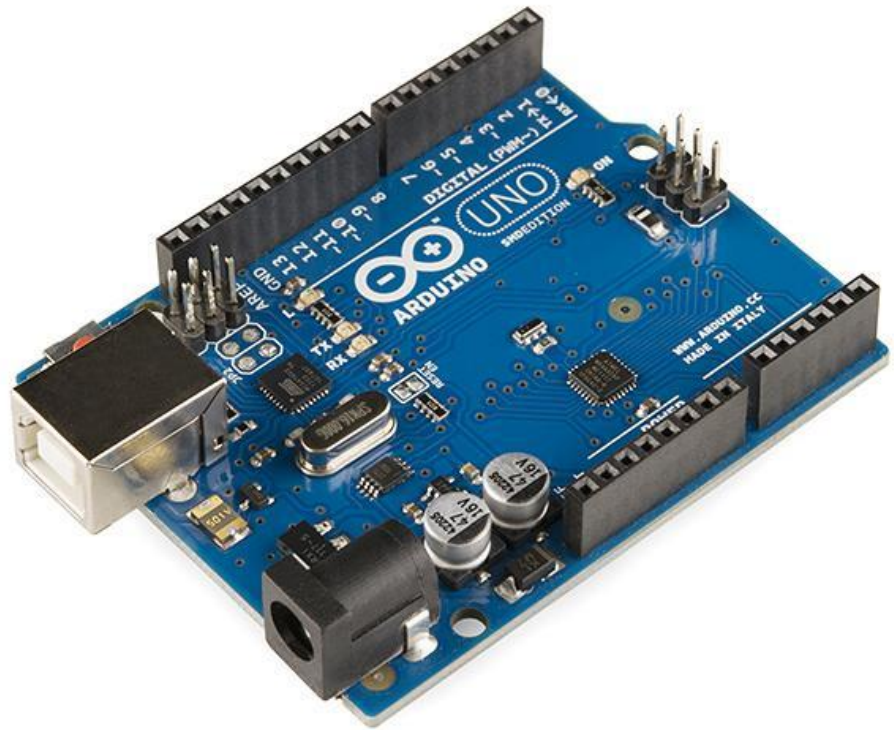
1222 · 2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



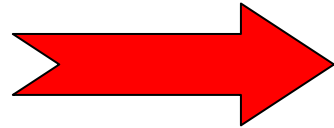
ASSEMBLAGGIO E TESTING DEI COMPONENTI ELETTRONICI



Caratteristiche

ARDUINO UNO:

- CPU ATmega328P
- 32KB Flash
- 2KB RAM
- Frequenza massima CPU 16MHz
- 1KB EEPROM
- Periferiche limitate o assenti
- Costo contenuto



Caratteristiche

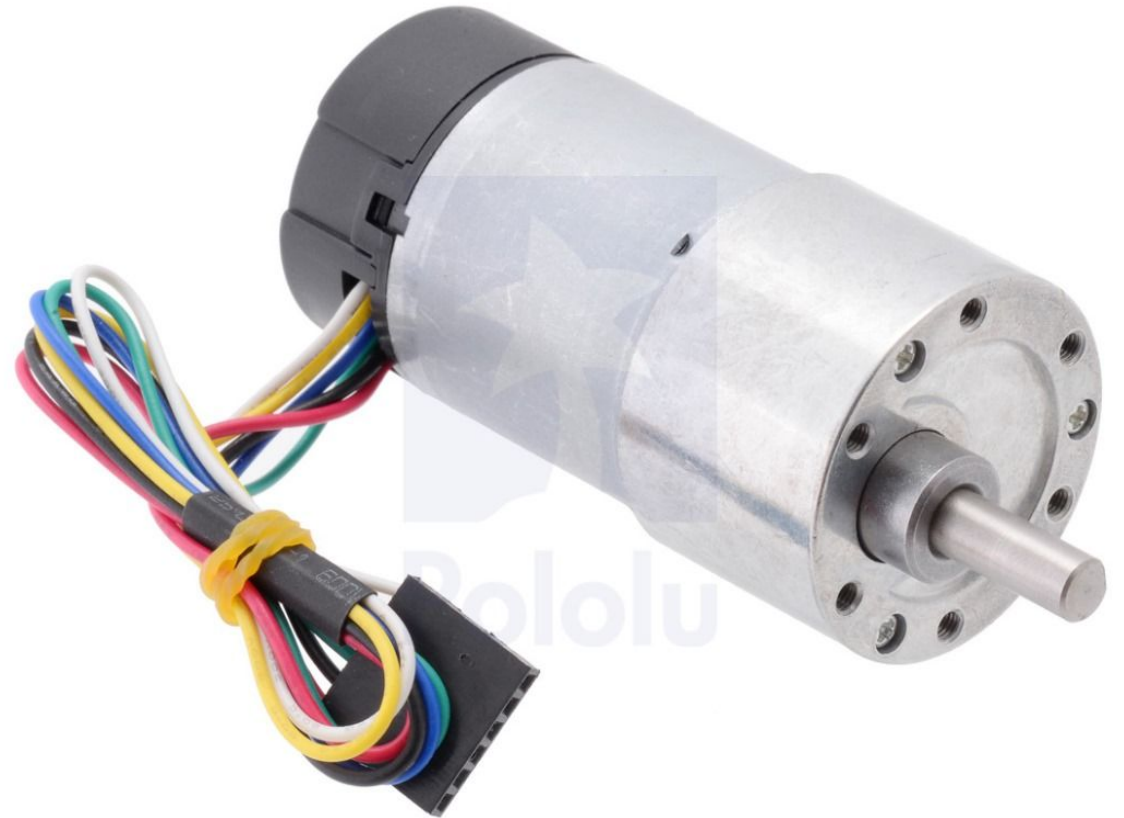
STM32 Nucleo64 L476RG:

- CPU ARM32-bit Cortex M4
- 1MB Flash
- 128KB SRAM
- Frequenza massima CPU 80MHz
- Numero di periferiche maggiore
- Costo contenuto
- Miglior rapporto prestazioni/dimensioni (tra le schede Nucleo)

- Scheda principale STM32 Nucleo64 L476RG
- Scheda driver motori DFRobot DRI0041
- Scheda custom applicata alla scheda principale
- 2 sistemi motore-riduttore-encoder Pololu 12V 37Dx64L mm
- Ruote e distanziali DFRobot diametro 65 mm
- MPU 6050 6-DoF Adafruit
- Pacco batteria agli ioni di litio 12 V da 3300 mAh
- 2 pulsanti monostabili retroilluminati
- 1 interruttore bistabile a bilanciere retroilluminato
- Connettore DB9 femmina
- Tutto il necessario per l'assemblaggio (cavi, connettori ecc..)

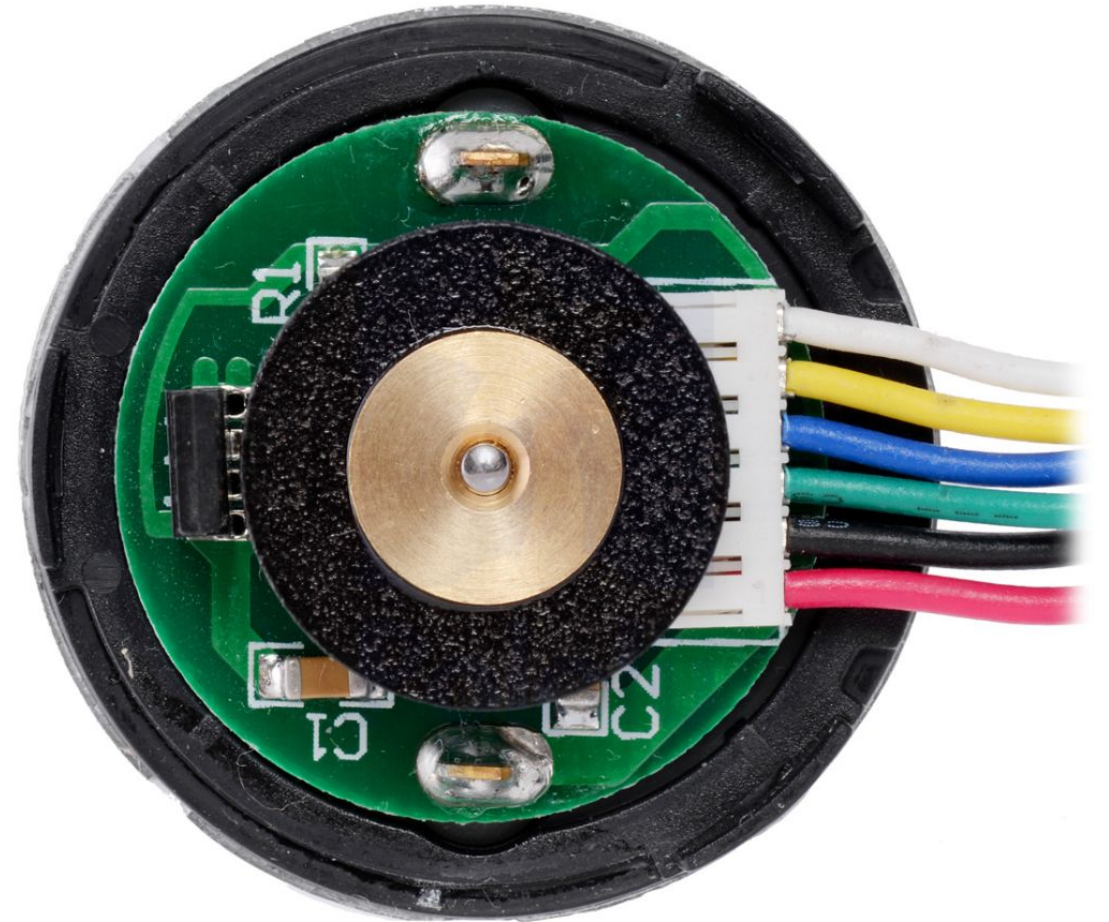
Caratteristiche:

- Alimentazione 12 V
- Corrente di stallo 5.5 A
- Corrente a vuoto 0.2 A
- Rapporto riduzione 30:1
- Velocità a vuoto 330 rpm
- Coppia di stallo 14 kg*cm
- Potenza massima 14 W
- Massa 215g



Caratteristiche:

- Risoluzione 64 conteggi per giro (1920 post gearbox)
- Alimentazione 3.5-20 V
- 2 cavi per alimentazione encoder
- 2 cavi segnale output
- 2 cavi alimentazione motore



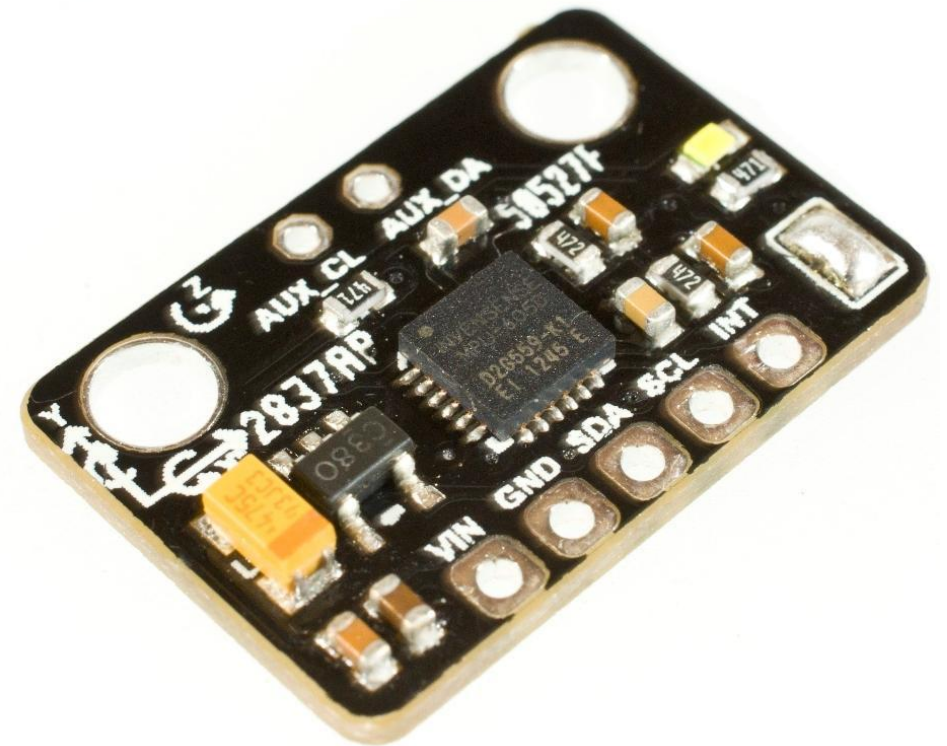
Caratteristiche:

- Alimentazione 7-24 V
- Corrente nominale 7A
- Corrente di picco 50A
- Minimo impulso rilevabile 5 μ s
- Temperatura funzionamento 25-85 °C
- Dimensioni 55x55 mm
- Massa 32g

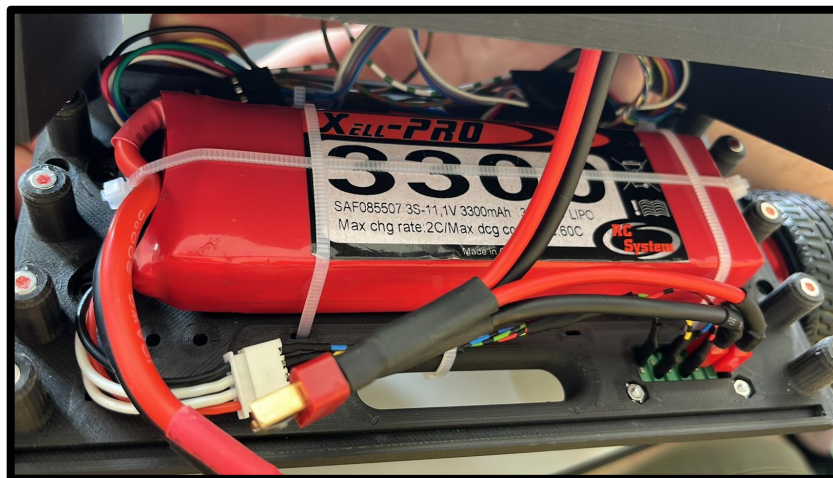
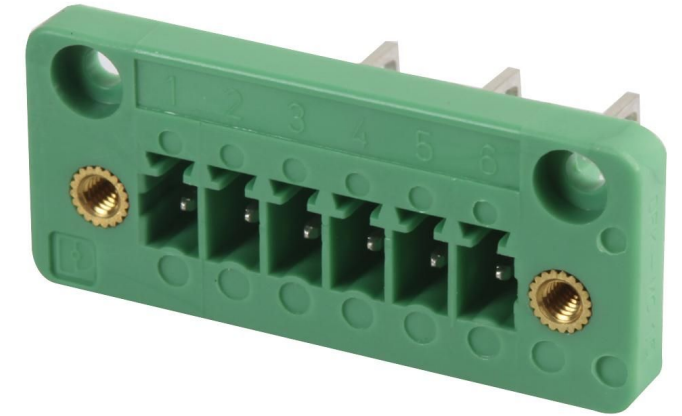


Caratteristiche:

- Accelerometro e giroscopio (MEMS) a 3 assi
- Output a 16 bit con scala variabile
- Comunica attraverso la I2C
- Dimensione 26x17.8x4.6 mm
- Massa 1.8 g
- Alimentazione a 3.3V



Ulteriori componenti



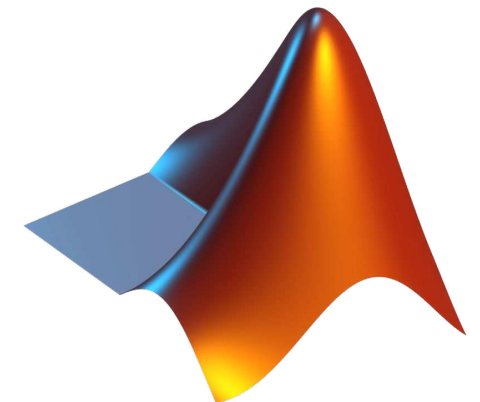
Fase di testing:

- STM32CubeMX
- Keil uVision5



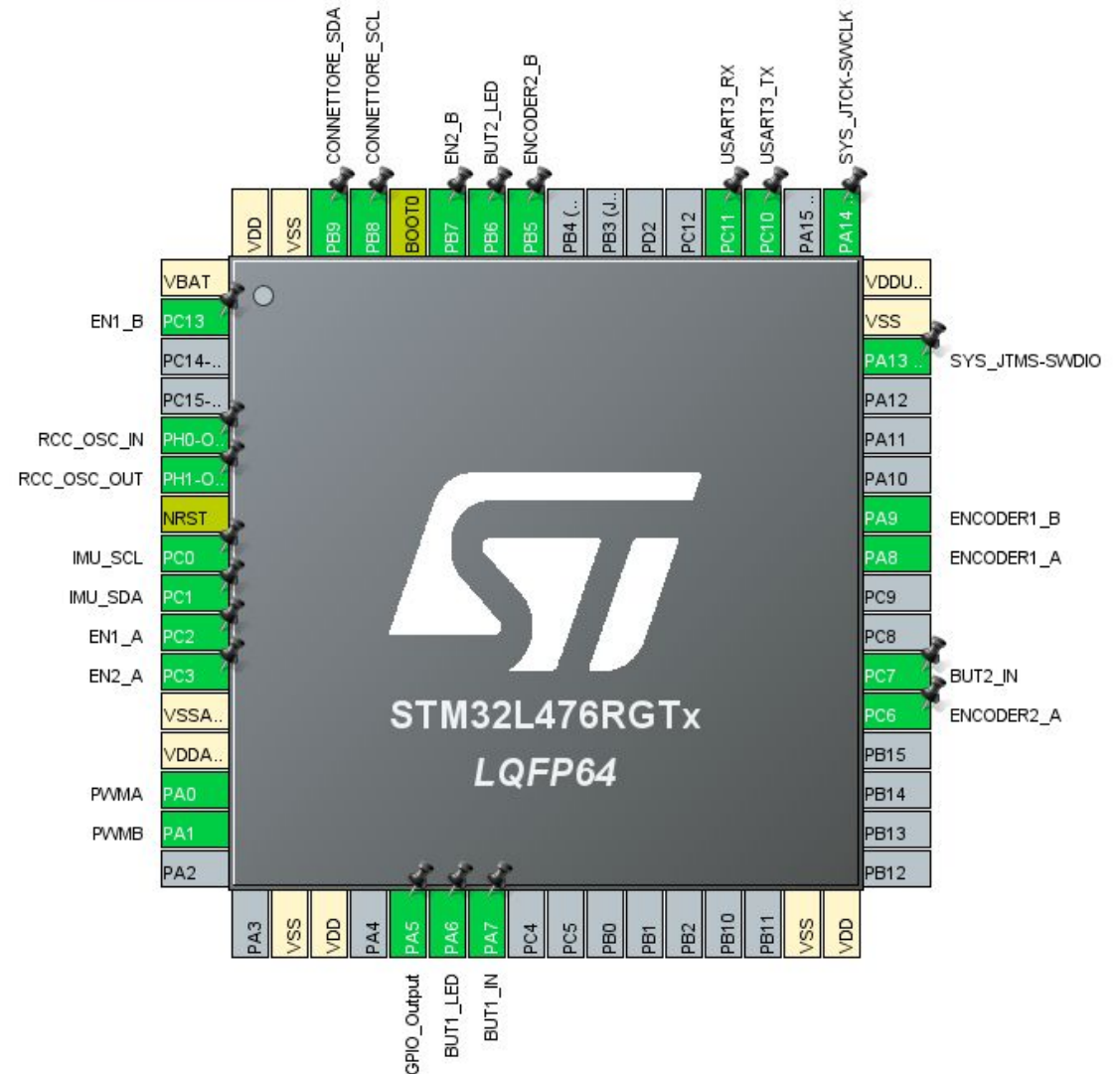
Fase di programmazione:

- STM32CubeProgrammer
- Matlab simulink



Configurazione STM32L407RG

- TIM1 e TIM3: Encoder
- TIM2: PWM
- TIM7 : Richiama la ISR di controllo con una frequenza di 1kHz
- 2 connessioni seriali I2C
- 1 connessione seriale USART
- 4 porte GPIO per i pulsanti
- 4 porte GPIO per gli enable dei motori



TIM1 e TIM3: Encoder A e Encoder B

Mode	
Channel1	Disable
Channel2	Disable
Channel3	Disable
Channel4	Disable
Channel5	Disable
Channel6	Disable
Combined Channels	Encoder Mode

Parameter Settings	User Constants	NVIC Settings	DMA Settings	GPIO Settings
NVIC Interrupt Table				
Enabled	Preemption Priority	Sub Priority		
<input type="checkbox"/>	0	0	TIM1 break interrupt and TIM15 global interrupt	
<input checked="" type="checkbox"/>	0	0	TIM1 update interrupt and TIM16 global interrupt	
<input type="checkbox"/>	0	0	TIM1 trigger and commutation interrupts and TIM17 global interrupt	
<input type="checkbox"/>	0	0	TIM1 capture compare interrupt	

Search (Ctrl+F)	
auto-reload preload	Disable
▼ Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection TRGO	Reset (UG bit from TIMx_EGR)
Trigger Event Selection TRGO2	Reset (UG bit from TIMx_EGR)
▼ Encoder	
Encoder Mode	Encoder Mode T11 and T12
____ Parameters for Channel 1 ____	
Polarity	Falling Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter	0
____ Parameters for Channel 2 ____	
Polarity	Rising Edge
IC Selection	Direct
Prescaler Division Ratio	No division
Input Filter	0

TIM2: PWM

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

Configure the below parameters :

Search (Ctrl+F) Acquisizione dati

Parameter	Value
Prescaler (PSC - 16 bits value)	77
Counter Mode	Up
Counter Period (AutoReload Register - 32 bit...)	1023
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
Trigger Output (TRGO) Parameters	
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection TRGO	Reset (UG bit from TIMx_EGR)
Clear Input	
Clear Input Source	Disable
PWM Generation Channel 1	
Mode	PWM mode 1
Pulse (32 bits value)	0
Output compare preload	Enable
Fast Mode	Disable
CH Polarity	High

TIM7: Richiama la ISR alla frequenza di 1 kHz

Configuration

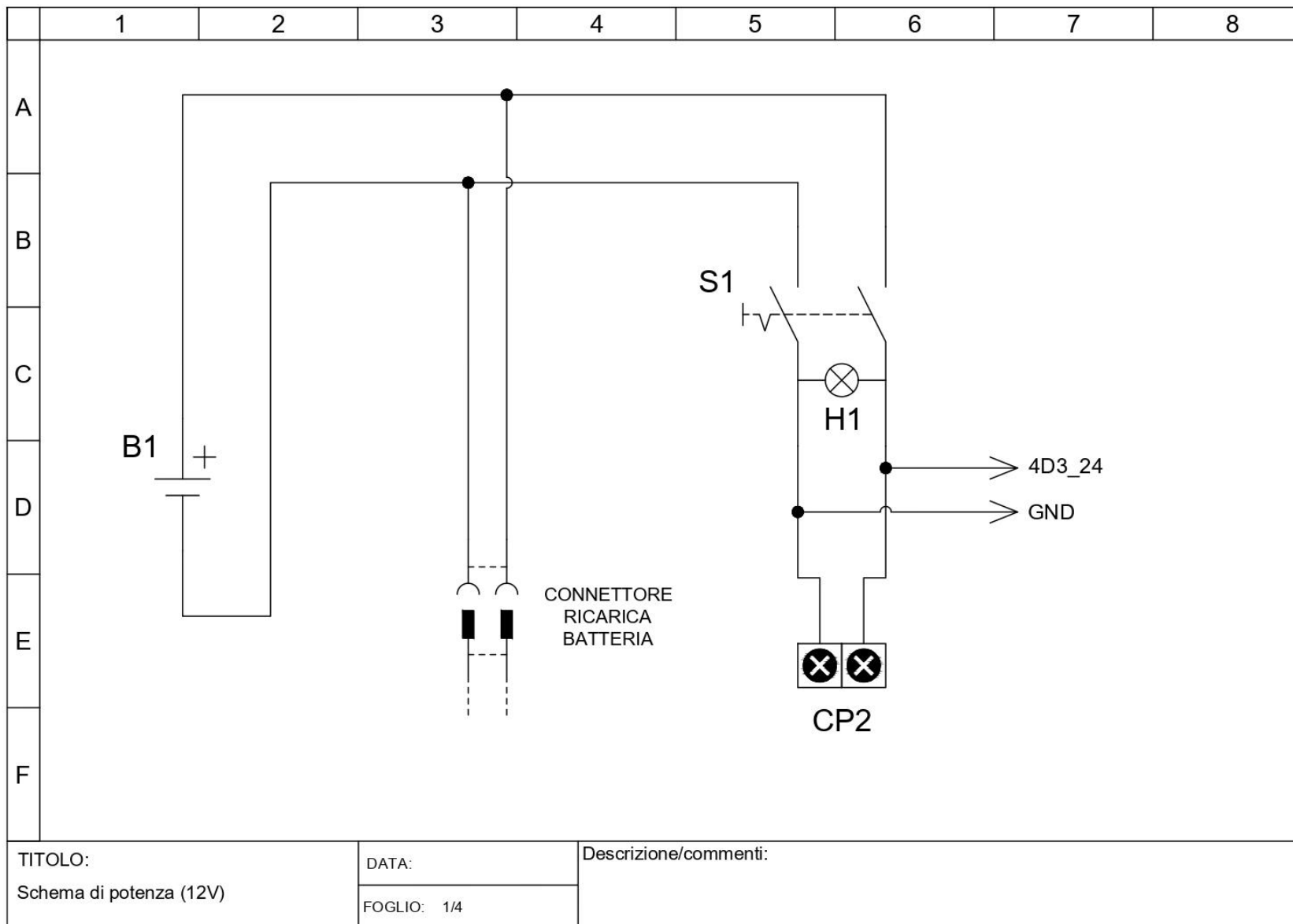
Reset Configuration

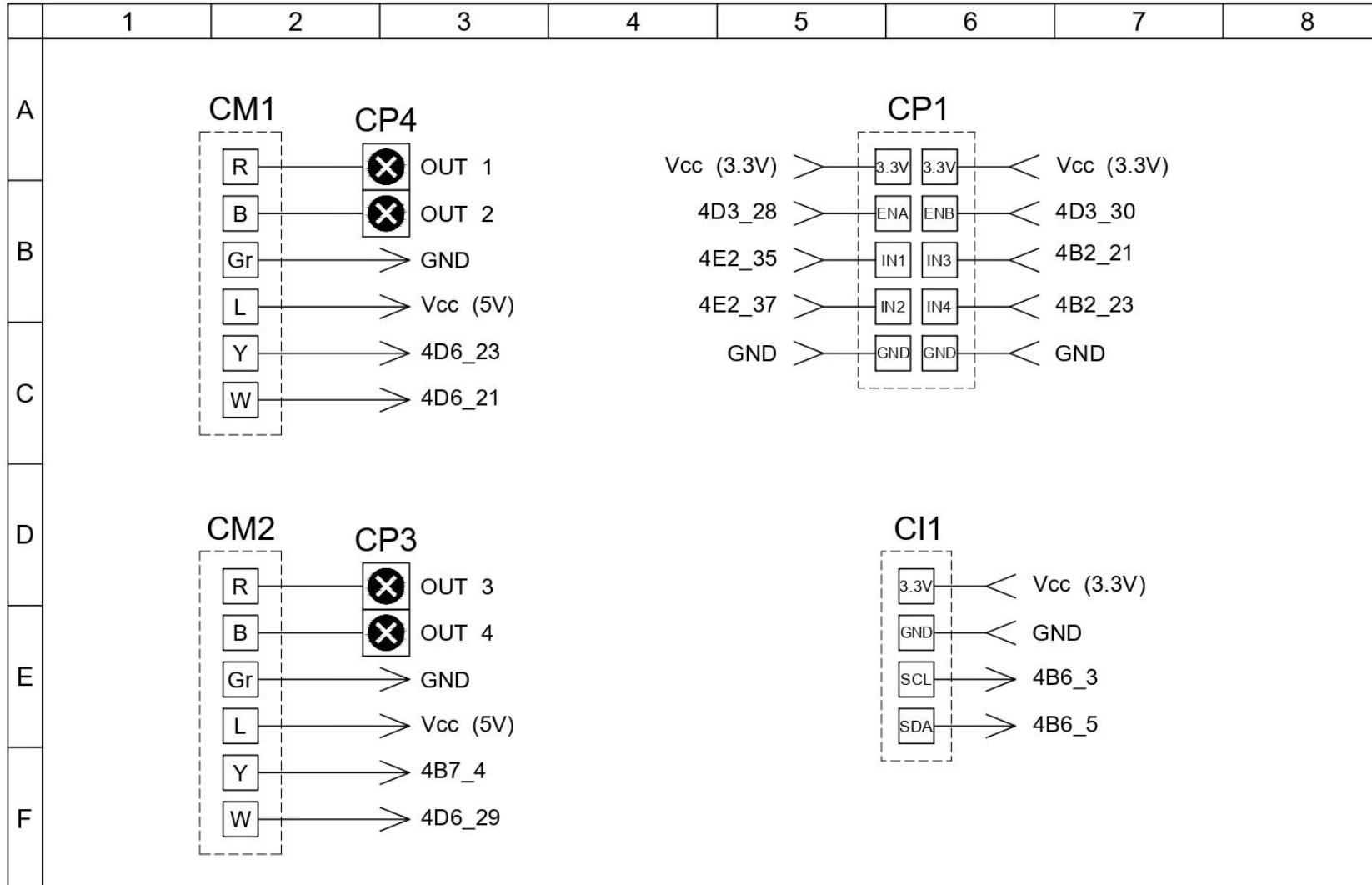
Parameter Settings User Constants NVIC Settings DMA Settings

Configure the below parameters :

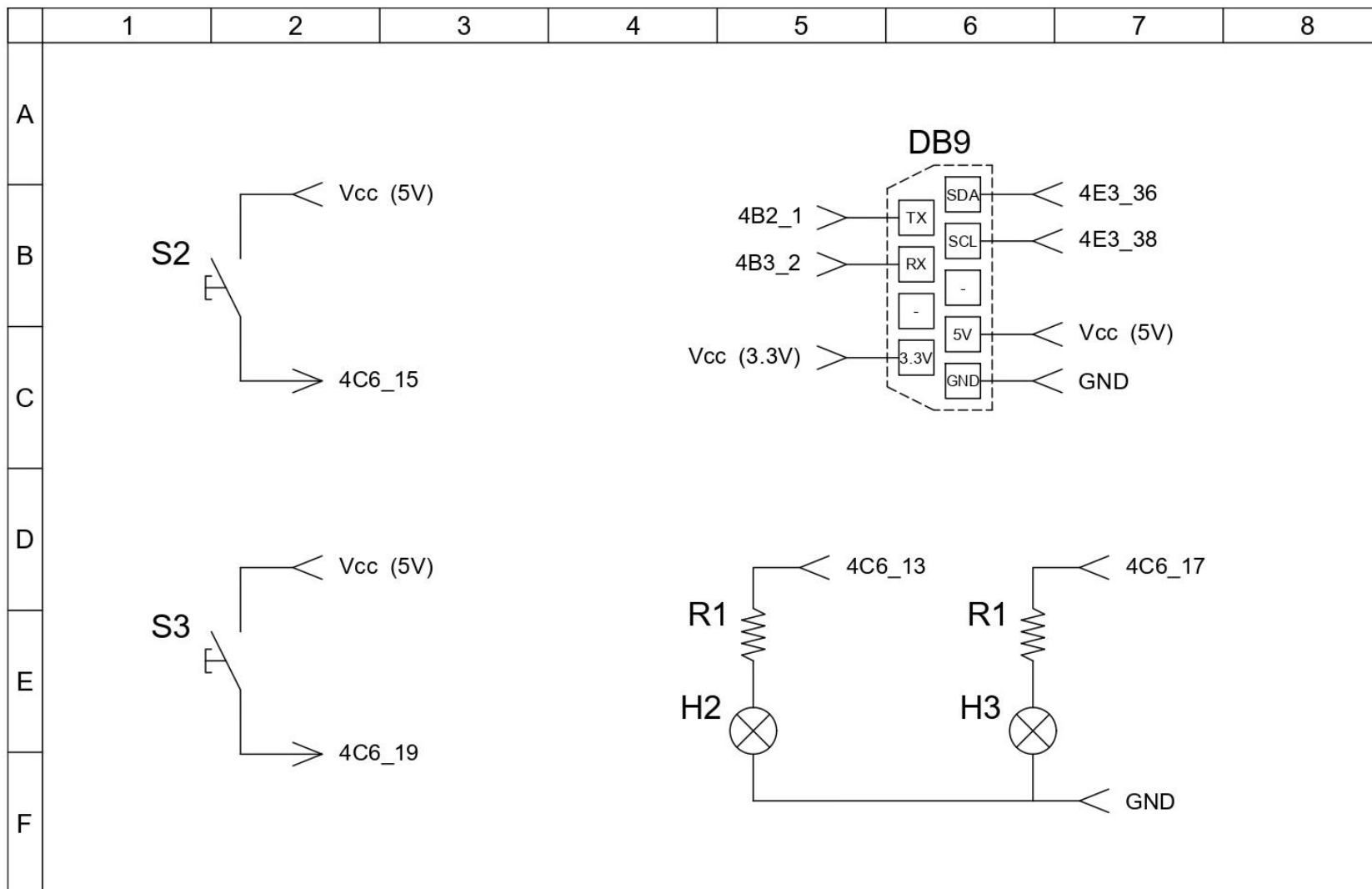
Search (Ctrl+F) ⏪ ⏩ ⓘ

- Counter Settings
 - Prescaler (PSC - 16 bits value) 1
 - Counter Mode Up
 - Counter Period (AutoReload Register - 16 bits... 39999
 - auto-reload preload Disable
- Trigger Output (TRGO) Parameters
 - Trigger Event Selection Reset (UG bit from TIMx_EGR)

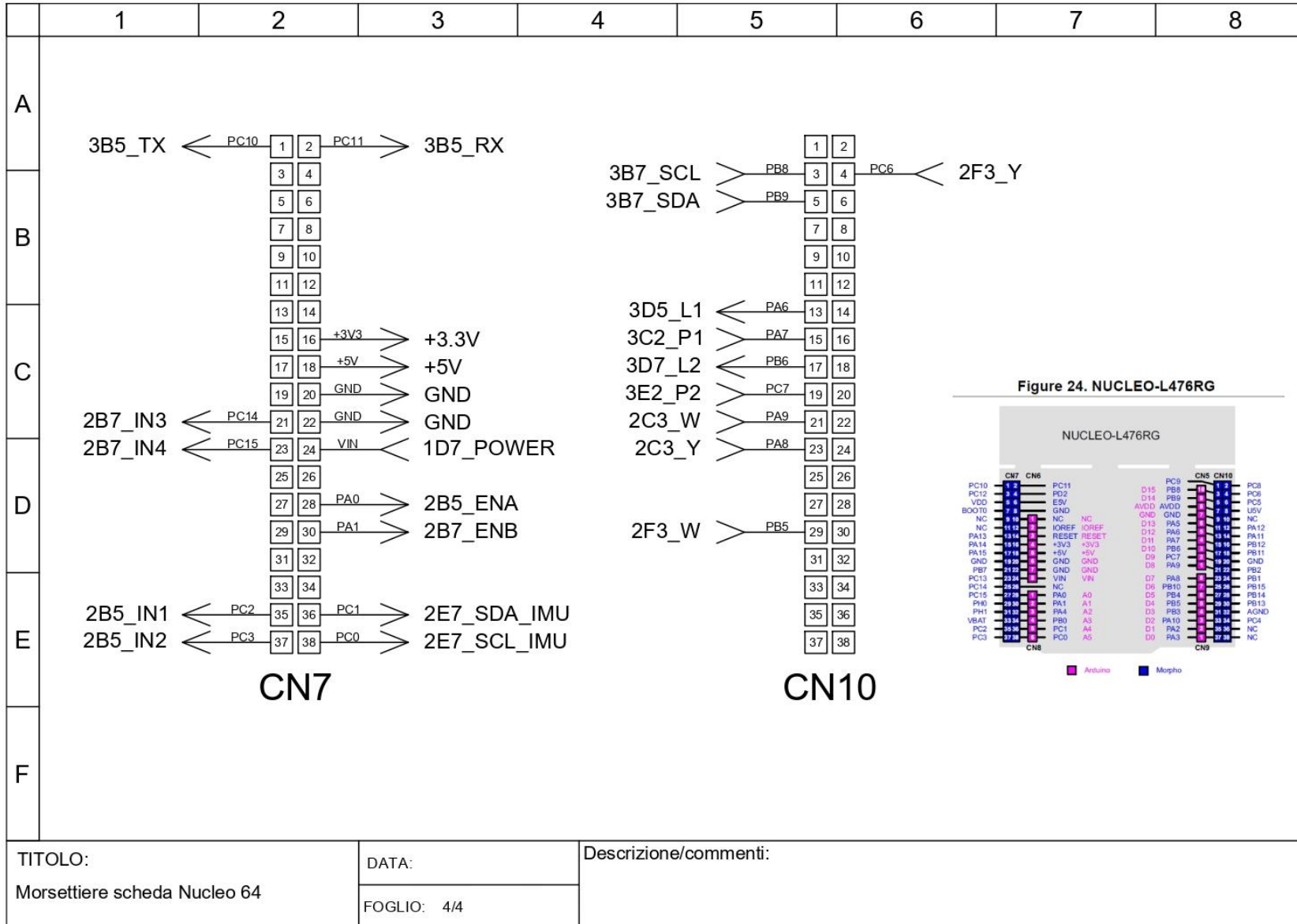


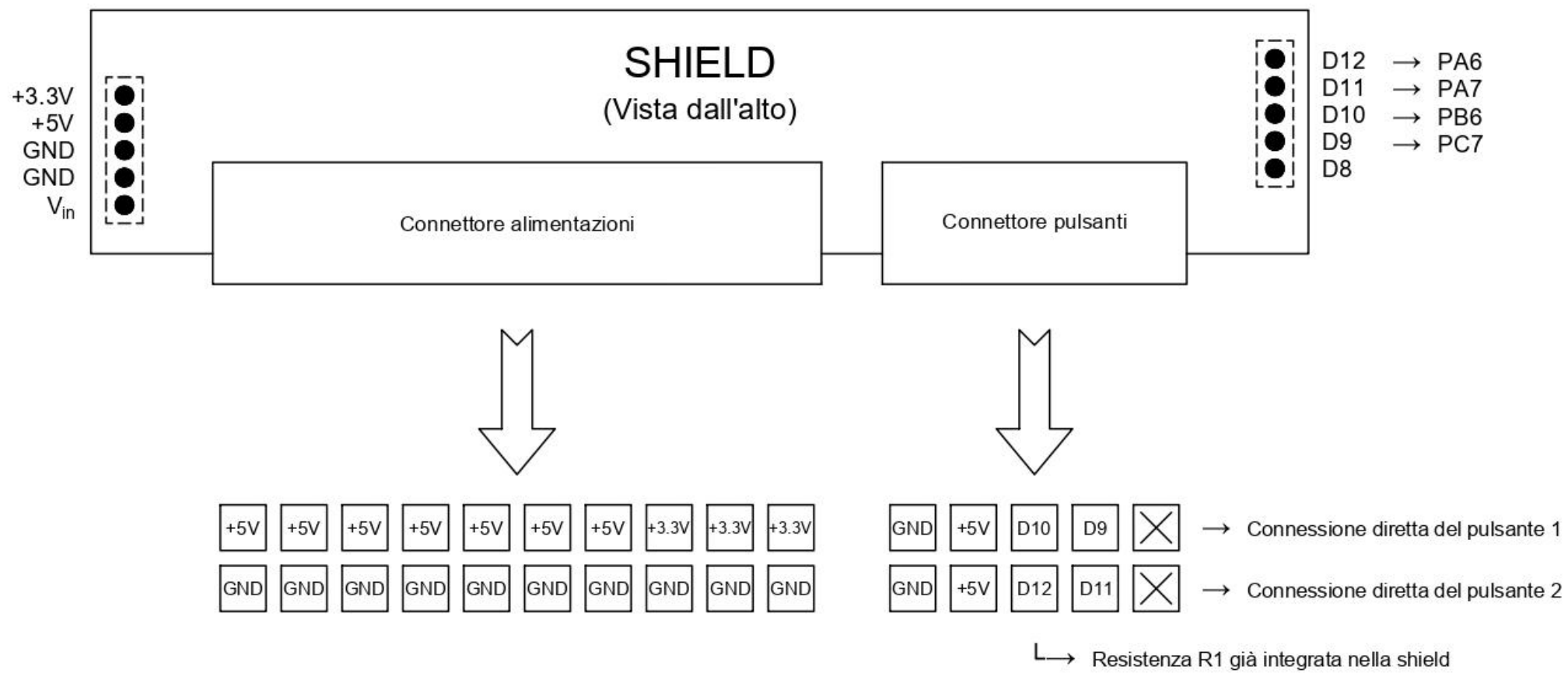


TITOLO: Morsettiere motori - driver - IMU	DATA:	Descrizione/commenti: CM: Connettore motori CP: Connettore driver motori CI: Connettore IMU
	FOGLIO: 2/4	

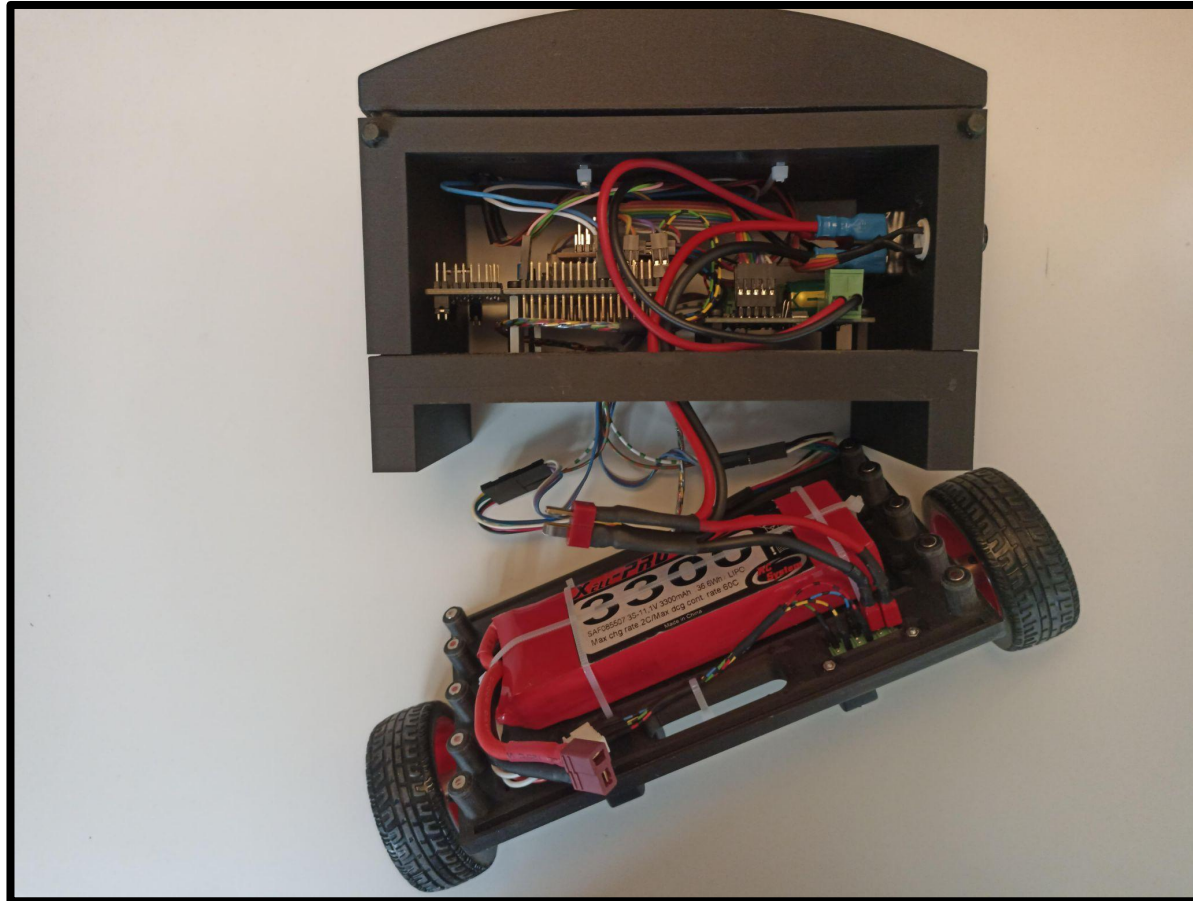


TITOLO: Connettore DB9 e pulsanti	DATA:	Descrizione/commenti:
	FOGLIO: 3/4	

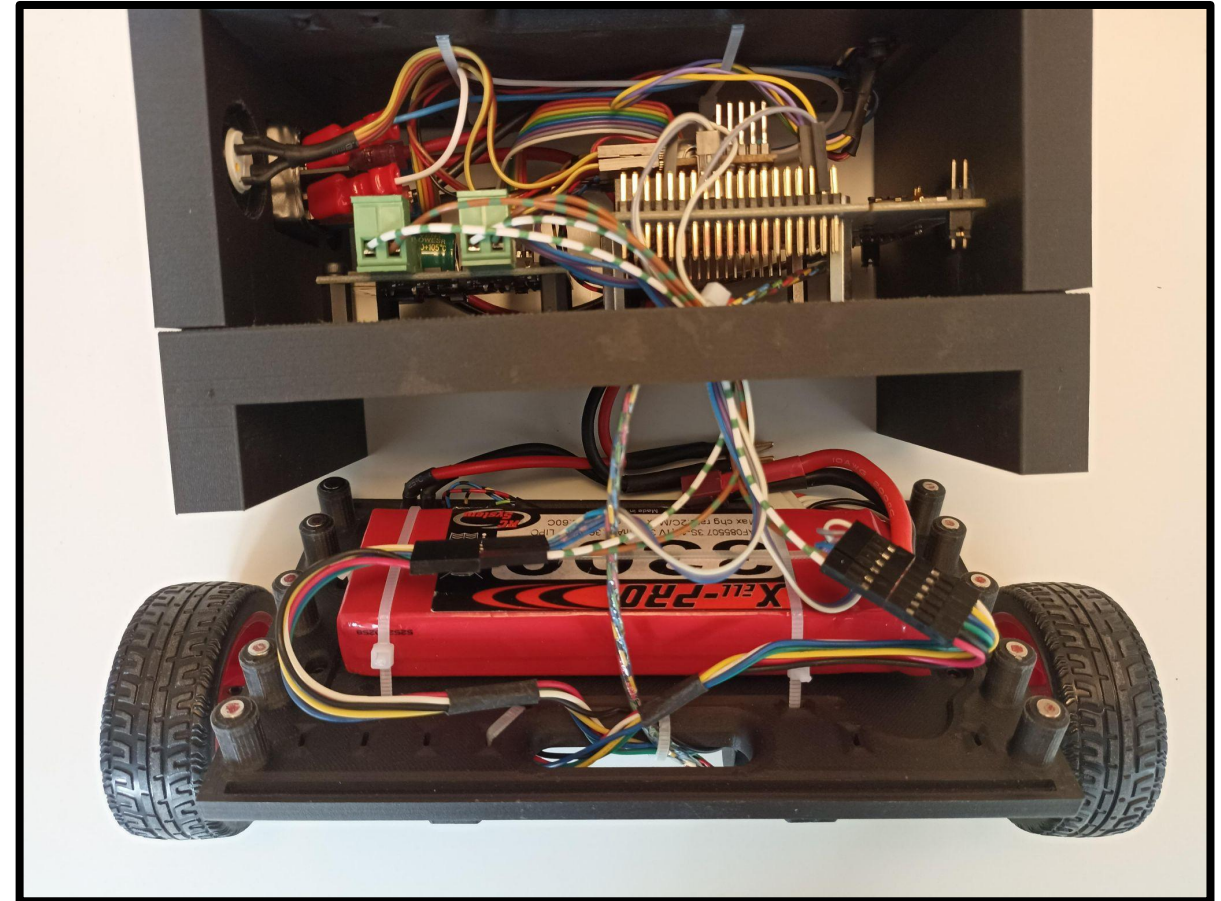




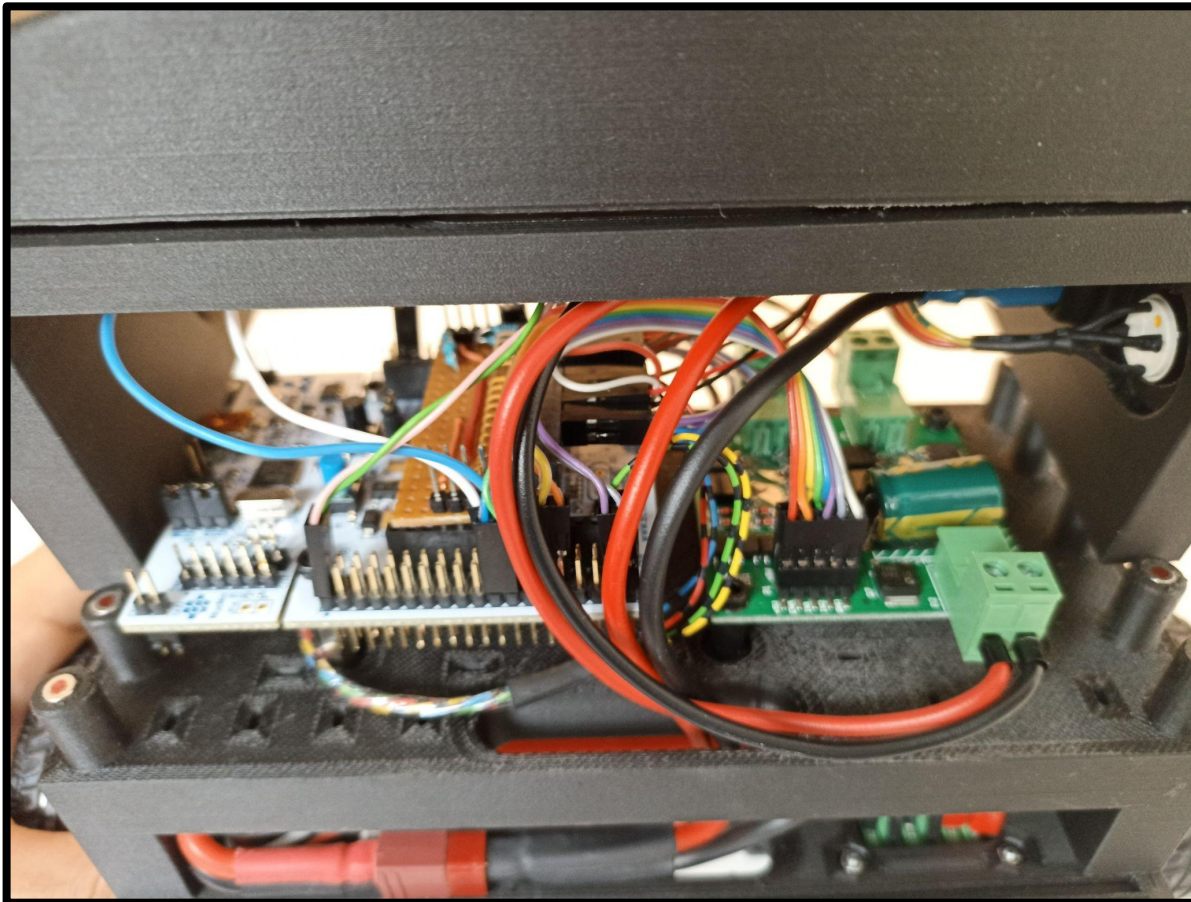
Lato A:



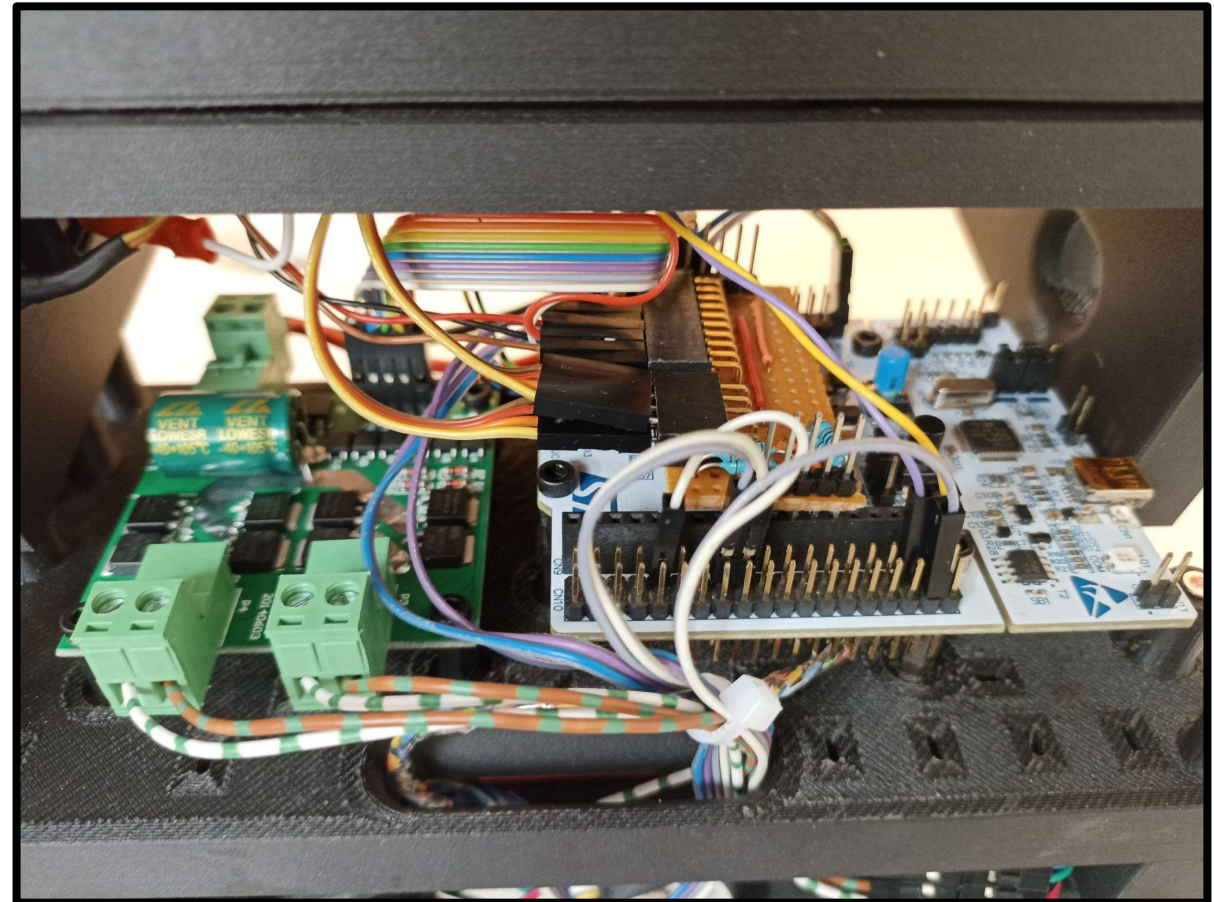
Lato B:



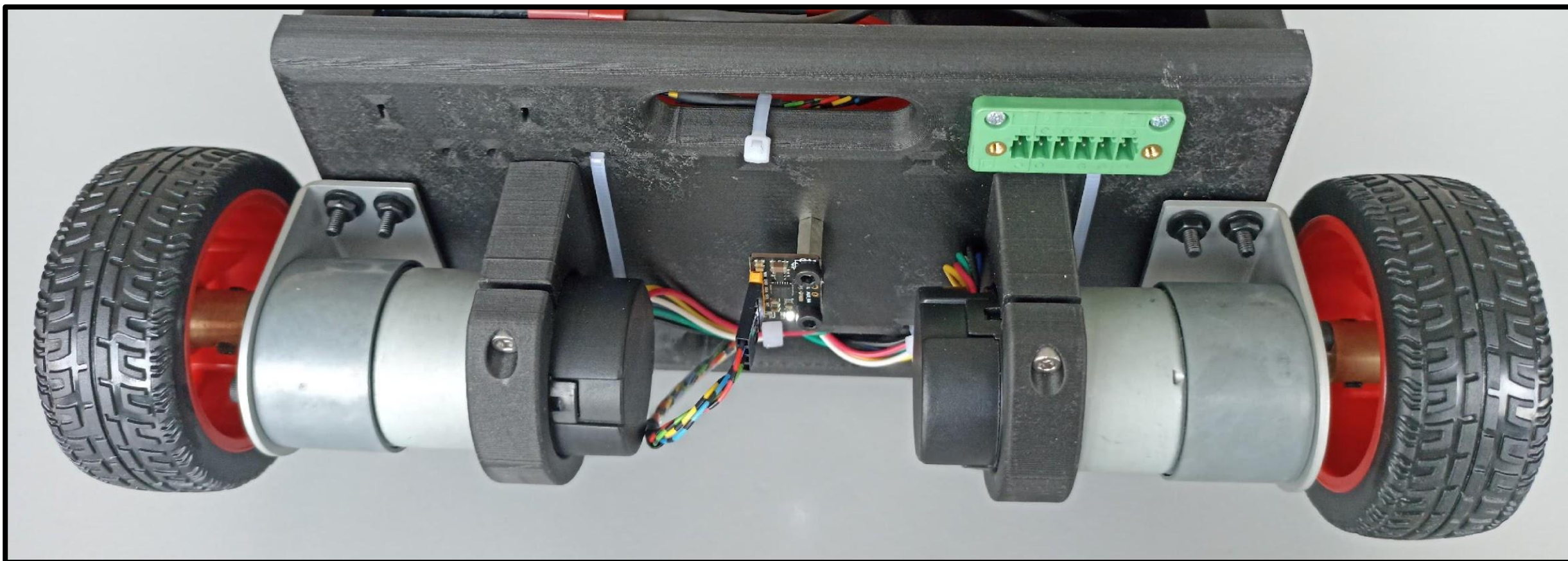
Lato A:



Lato B:



Parte sottostante:



main.c

```
//Inizializzazione degli interrupt globali dei timer
HAL_TIM_Base_Start_IT(&htim1);
HAL_TIM_Base_Start_IT(&htim3);
HAL_TIM_Base_Start_IT(&htim7);

//Inizializzazione del PWM
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1); //Inizializzazione PWM_A timer 2 sul canale 1
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_2); //Inizializzazione PWM_B timer 2 sul canale 2

// controllo se la periferica è collegata correttamente
HAL_I2C_Mem_Read (&hi2c3, MPU6050_ADDR,WHO_AM_I_REG,1, &check, 1, 1000);

// se MPU è collegata correttamente
if (check ==104){

    // uscita dalla sleep mode
    data=0;
    HAL_I2C_Mem_Write(&hi2c3, MPU6050_ADDR,0x6B,1, &data, 1, 1000);

    // impostazione del divider del sample rate dell'accelerometro e giroscopio
    data =7;
    HAL_I2C_Mem_Write(&hi2c3, MPU6050_ADDR,0x19,1, &data, 1, 1000);

    // impostazione del range a +-2g
    data =0x00;
    HAL_I2C_Mem_Write(&hi2c3, MPU6050_ADDR,0x1B,1, &data, 1, 1000);

    // impostazione del range a +-250deg/s
    data =0x00;
    HAL_I2C_Mem_Write(&hi2c3, MPU6050_ADDR,0x1C,1, &data, 1, 1000);
```

stm32l4xx_it.c

```
/* USER CODE BEGIN TIM1_UP_TIM16_IRQn 1 */  
  
//Nel caso di overflow del timer 1 (associato al motore 1)  
//Verifico registro CR1, bit DIR, per vedere se contatore contava in su o in giu'  
//cosi' da riconoscere se ho un OVERFLOW o un DOWNFLOW  
if((TIM1->CR1 & TIM_CR1_DIR) == TIM_CR1_DIR)  
{  
    over_A--;  
}  
else  
{  
    over_A++;  
}  
  
/* USER CODE END TIM1_UP_TIM16_IRQn 1 */  
}
```


stm32l4xx_it.c

```
/* USER CODE BEGIN TIM3_IRQn 1 */  
  
//Nel caso di overflow del timer 3 (associato al motore 2)  
//Verifico registro CR1, bit DIR, per vedere se contatore contava in su o in giu'  
//cosi' da riconoscere se ho un OVERFLOW o un DOWNFLOW  
if((TIM3->CR1 & TIM_CR1_DIR) == TIM_CR1_DIR)  
{  
    over_B--;  
}  
else  
{  
    over_B++;  
}  
/* USER CODE END TIM3_IRQn 1 */
```

stm32l4xx_it.c

```
/* USER CODE BEGIN TIM7_IRQn 1 */

//INTERRUPT DEL TIMER 7 A 1KHz

count = count + 1;

//SEZIONE DI CAMPIONAMENTO DEI DATI

//lettura stato pulsanti
but1_value = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_7);
but2_value = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_7);

//lettura acc
HAL_I2C_Mem_Read (&hi2c3, MPU6050_ADDR, 0x3B, 1, Rec_Data, 6, 1000);
Accel_X_RAW = (int16_t) (Rec_Data[0] << 8 | Rec_Data [1]);
Accel_Y_RAW = (int16_t) (Rec_Data[2] << 8 | Rec_Data [3]);
Accel_Z_RAW = (int16_t) (Rec_Data[4] << 8 | Rec_Data [5]);

//lettura Gyro
HAL_I2C_Mem_Read (&hi2c3, MPU6050_ADDR, 0x43, 1, Rec_Data, 6, 1000);
Gyro_X_RAW = (int16_t) (Rec_Data[0] << 8 | Rec_Data [1]);
Gyro_Y_RAW = (int16_t) (Rec_Data[2] << 8 | Rec_Data [3]);
Gyro_Z_RAW = (int16_t) (Rec_Data[4] << 8 | Rec_Data [5]);
```

stm32l4xx_it.c

```
//conversione accelerazioni
Ax = Accel_X_RAW/16384.0;
Ay = Accel_Y_RAW/16384.0;
Az = Accel_Z_RAW/16384.0;

//conversione giroscopio
Gx = Gyro_X_RAW/131.0;
Gy = Gyro_Y_RAW/131.0;
Gz = Gyro_Z_RAW/131.0;

//acquisizione/calcolo posizione motori (numero impulsi encoder)
pos_A = TIM1->CNT + 1920*over_A;
pos_B = TIM3->CNT + 1920*over_B;
```

stm32l4xx_it.c

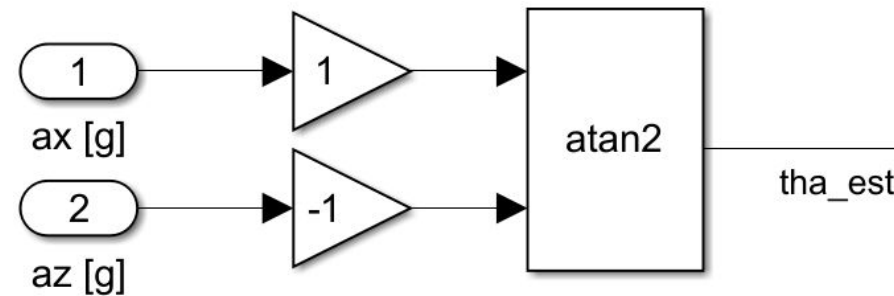
```
//SEZIONE DI SCRITTURA DELLE USCITE AL TERMINE DEL CALCOLO

//scrittura led
HAL_GPIO_WritePin(GPIOA,GPIO_PIN_6,but1_led);
HAL_GPIO_WritePin(GPIOB,GPIO_PIN_6,but2_led);

//enable direzione motori
HAL_GPIO_WritePin(GPIOC,GPIO_PIN_2,EN1_A);
HAL_GPIO_WritePin(GPIOC,GPIO_PIN_3,EN2_A);
HAL_GPIO_WritePin(GPIOC,GPIO_PIN_13,EN1_B);
HAL_GPIO_WritePin(GPIOB,GPIO_PIN_7,EN2_B);

//scrivo nuovo duty cycle
TIM2->CCR1 = DUTYA;
TIM2->CCR2 = DUTYB;
```

Accelerometro: $\theta_a = \text{atan2}(a_x, -a_z)$

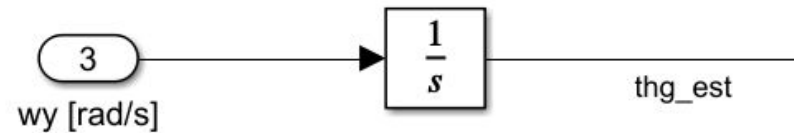


Difetti:

- Sensibile alle accelerazioni della struttura del robot
- Stima corrotta in alta frequenza da disturbi interni

Giroscopio:

$$\theta_g = \theta_g(0) + \int_0^t w_y(\tau) d\tau$$



Difetti:

- Necessità dell'inizializzazione di $\theta_g(0) = 0$;
- Stima è affetta da un errore costante e uno a rampa lineare

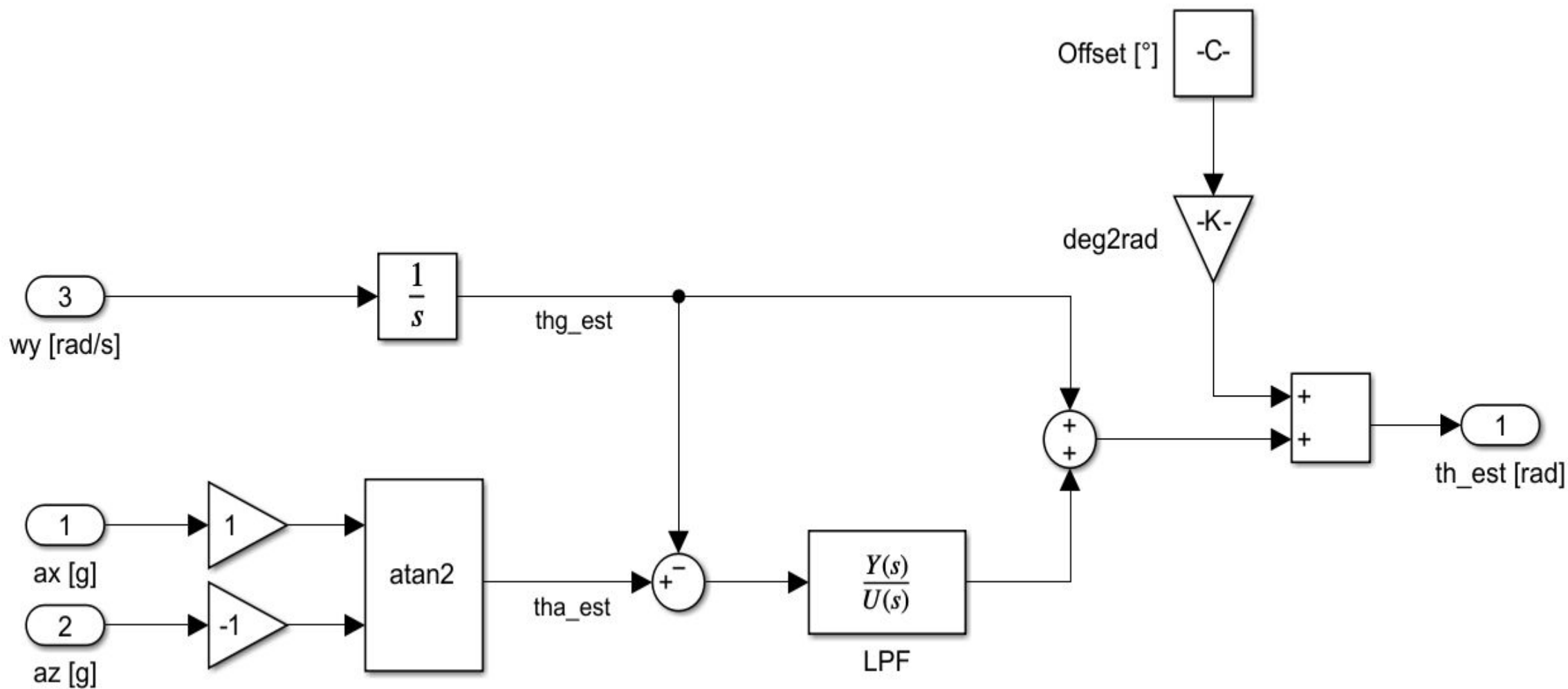
Si combinano le due stime attraverso un filtro complementare:

$$H(s) = \frac{1}{\tau s + 1} \quad 1 - H(s) = \frac{\tau s}{\tau s + 1}$$

$$\theta = \theta_a H(s) + \theta_g [1 - H(s)]$$

In questo modo abbiamo una stima dell'angolo accurata sia in bassa frequenza (accelerometro) sia in alta frequenza (giroscopio).

Implementazione filtro complementare



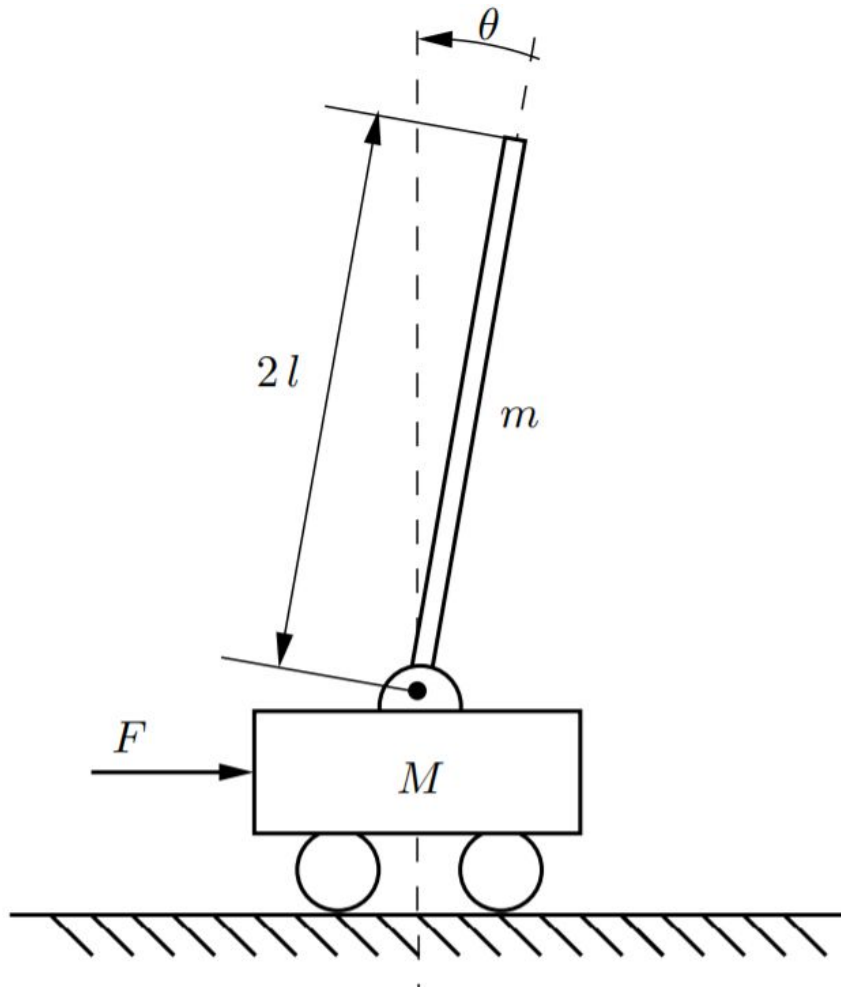
1222 · 2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DEFINIZIONE DEL MODELLO E SISTEMA DI CONTROLLO



Analisi Newtoniana:

$$\begin{cases} I_e \ddot{\theta} - ml \cos \theta \ddot{x} - mgl \sin \theta = 0 \\ -ml \cos \theta \ddot{\theta} + M_e \ddot{x} + b\dot{x} + ml \sin \theta \dot{\theta}^2 = F \end{cases}$$

dove I_e è l'inerzia del pendolo rispetto alla sua base e $M_e = M + m$

Linearizzazione attorno alla posizione di equilibrio e analisi nel dominio di Laplace:

- $P_{\theta \rightarrow x} = \frac{X(s)}{\Theta(s)} = \frac{I_e s^2 - mgl}{m l s^2}$
- $G_{F \rightarrow \theta} = \frac{\Theta(s)}{F(s)} = \frac{m l s^2}{s[(M_e I_e - m^2 l^2) s^3 + (b I_e) s^2 - (M_e m g l) s - (b m g l)]}$

Noti i valori di massa e di inerzia dei componenti del robot si identificano numericamente i parametri del modello del pendolo inverso, per poi descrivere il comportamento dei motori (non previsti nel modello del pendolo inverso).

La relazione tra la coppia prodotta dai motori e l'angolo di inclinazione si trova:

$$G_{\tau_m \rightarrow \theta} = G_{F \rightarrow \theta} \cdot \frac{N_{rid}}{r_r}$$

La relazione tra la tensione fornita ai motori (effettivo ingresso del sistema fisico) e la coppia prodotta dai motori, è data da:

$$\tau_m = \frac{N_{rid} k_t}{R_a} u_a - \frac{N_{rid}^2 k_t k_e}{R_a} (\dot{\gamma} - \dot{\theta})$$

dove $\gamma = \frac{x}{r_r}$

Si ricava così la relazione finale $P_{u_a \rightarrow \theta}$

I valori delle costanti di coppia e di fcem k_t e k_e sono stati ricavati rispettivamente dai valori di stallo e a vuoto nel datasheet dei motori.

Stima dei parametri del pendolo

Nel modello del pendolo inverso si incontrano i parametri meccanici da adattare a quelli del robot, non tutti stimabili dal programma di CAD:

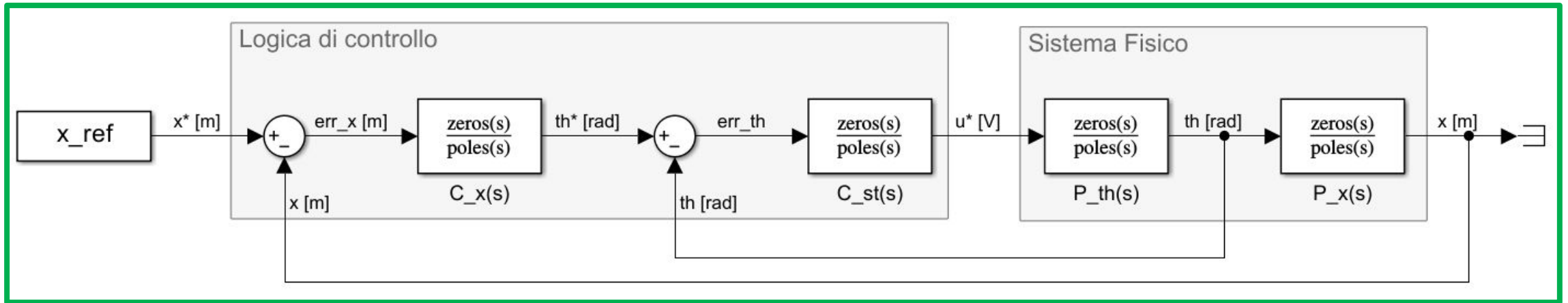
- l è la distanza del centro di massa dall'asse di rotazione delle ruote
- I_e è l'inerzia del robot rispetto all'asse di rotazione delle ruote
- m è la massa rotante, calcolata come la massa totale del robot sottratta della massa delle ruote e dei rotori dei motori (hp $m_{rot} \approx 35\% m_{mot}$): $m = M_{robot} - 2(m_{rot} + m_r) = 1,495\text{kg}$
- M è la massa equivalente traslante, e rappresenta il comportamento di inerzia lineare opposto dai rotori e dalle ruote. Le masse di questi e l'inerzia delle ruote sono trascurabili rispetto all'inerzia del rotore ridotta all'albero lento: $M = 2 \left(I_{rot} \frac{N_{rid}^2}{r_r^2} \right) \approx 15\text{kg}$

Si ottengono quindi le due fdt:

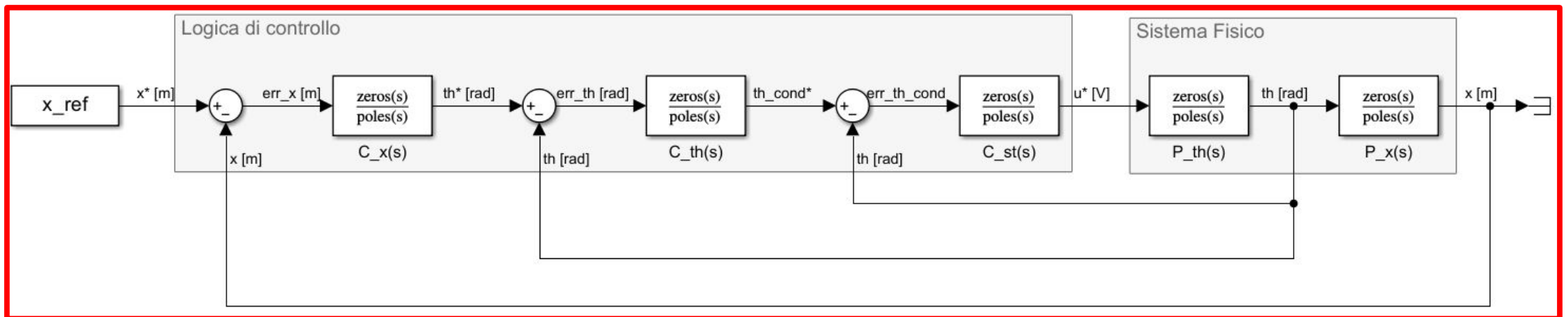
$$P_{\theta \rightarrow x} = \frac{0,13595(s-8,495)(s+8,495)}{s^2}$$

$$e \quad P_{u \rightarrow \theta} = \frac{27,71s}{(s+26,37)(s+10,35)(s-9,468)}$$

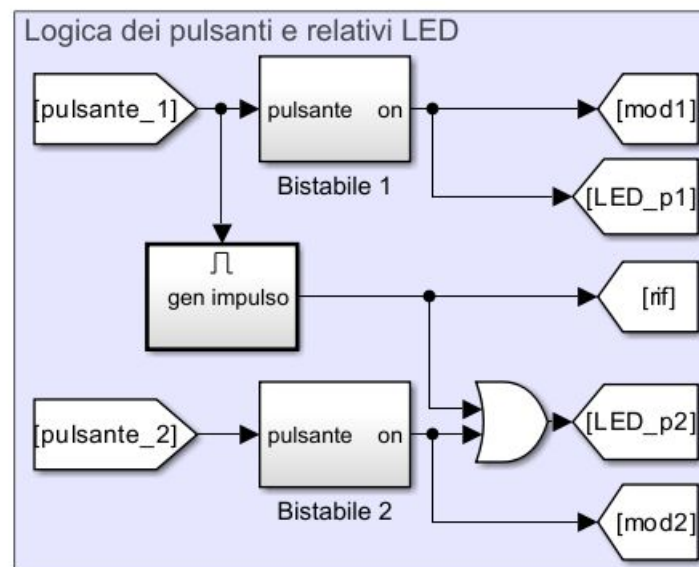
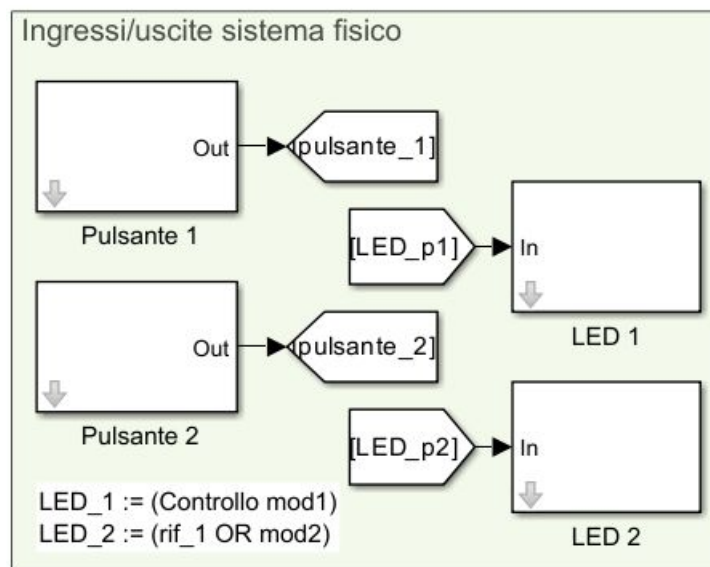
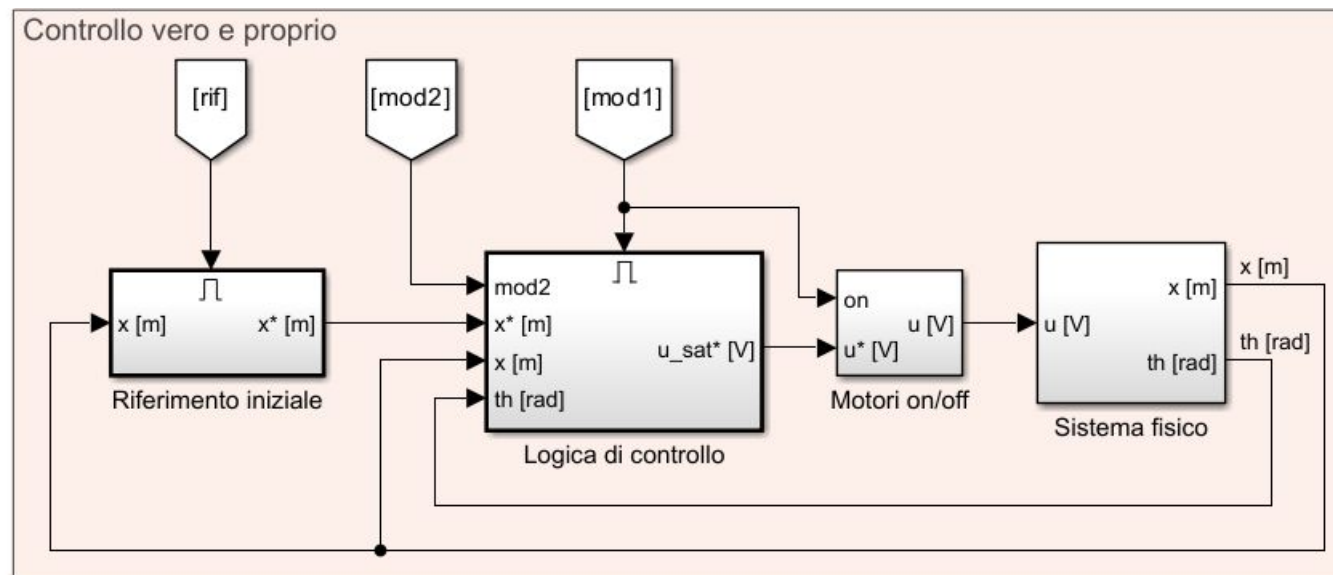
Schema della struttura del controllo, poi utilizzato per la programmazione:

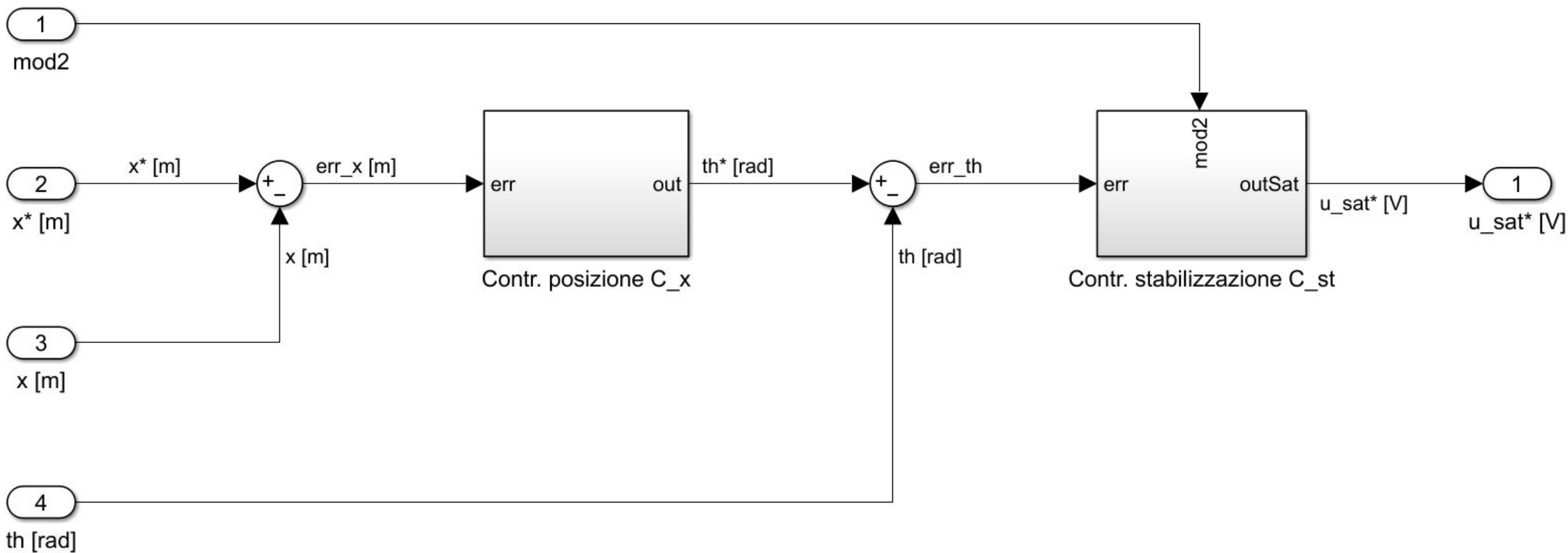


Schema dimensionato per il controllo, poi scartato sperimentalmente:

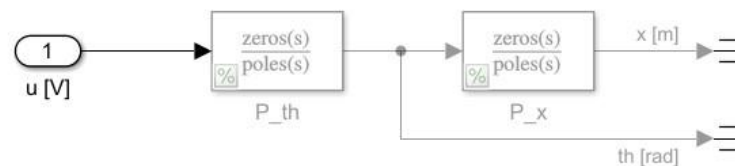
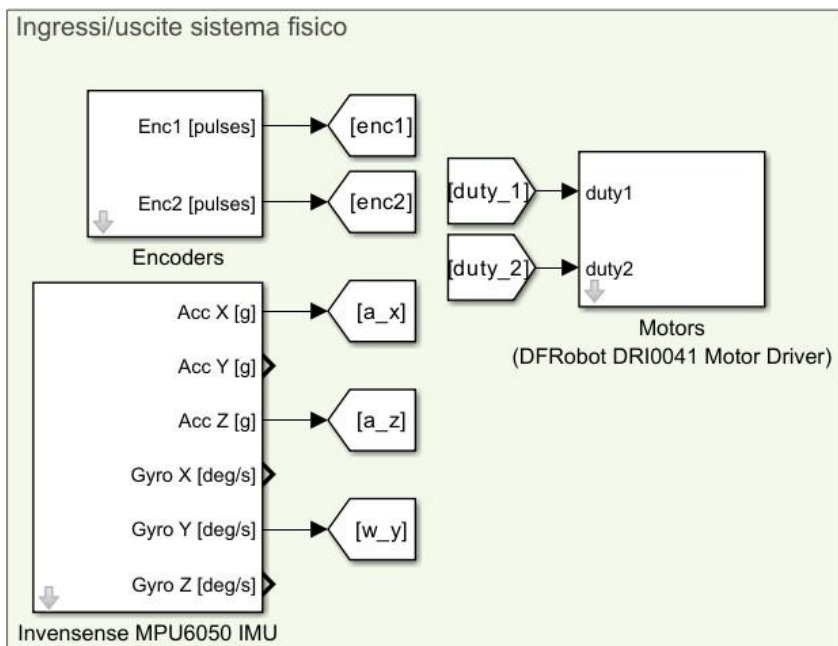
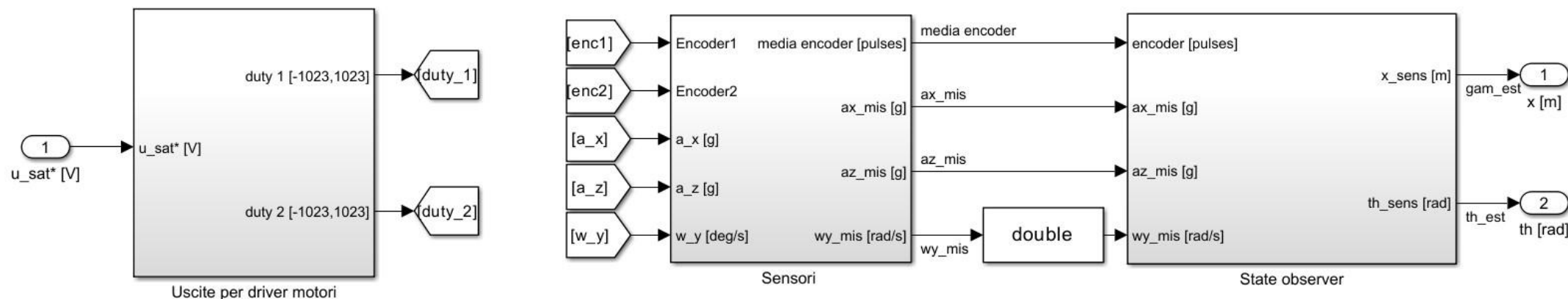


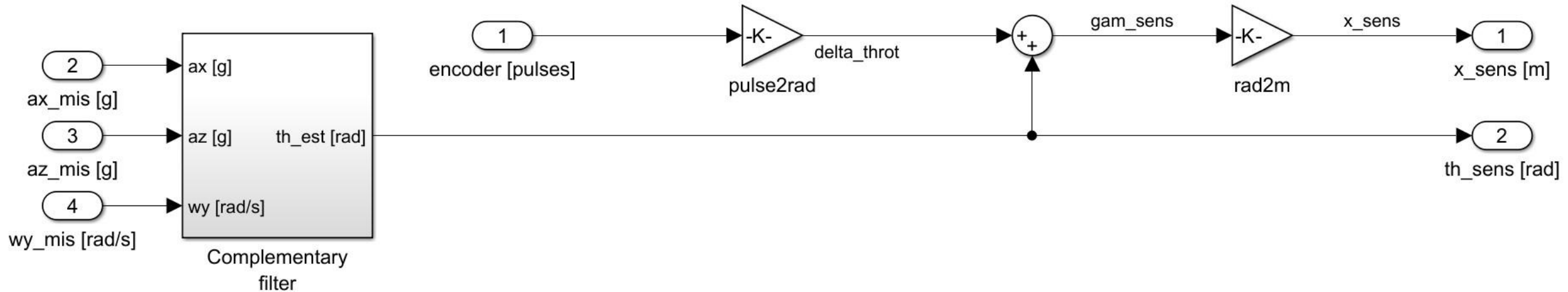
Simulink - schema generale





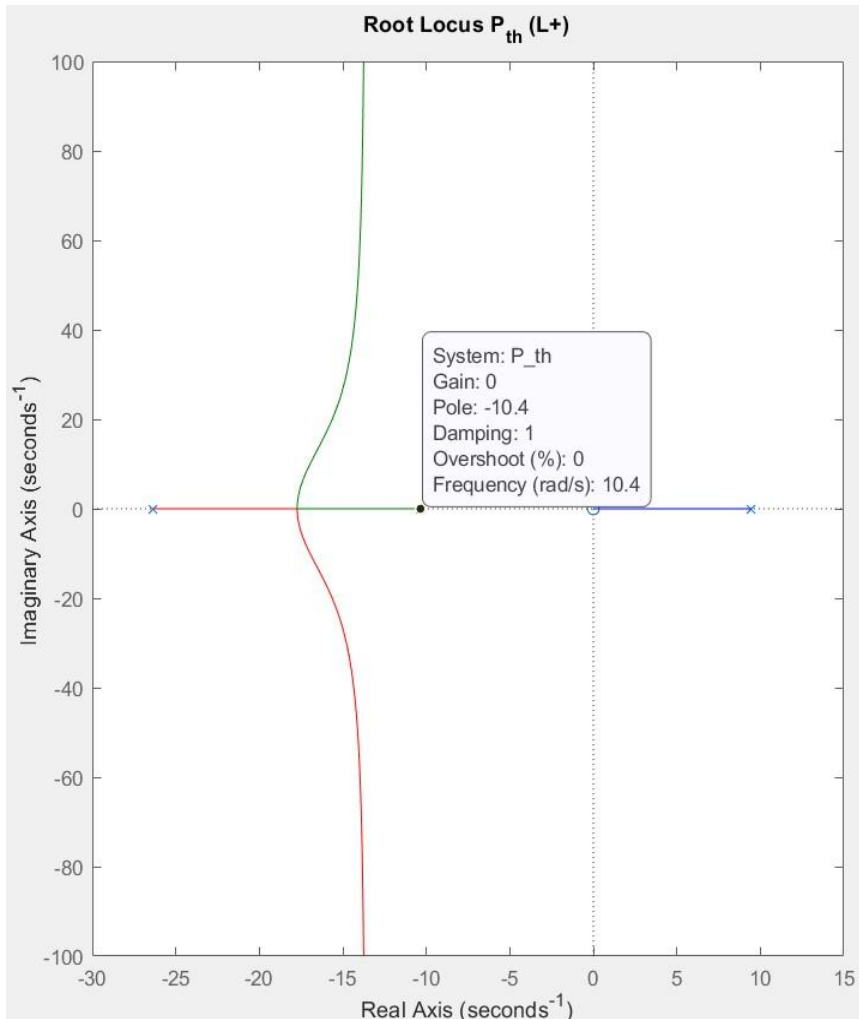
Simulink - sistema fisico





L'angolo gamma che rappresenta la "posizione angolare" delle ruote rispetto alla verticale gravitazionale, si ricava dagli encoder (**delta_throt**) che misurano lo stesso angolo rispetto alla struttura del robot, sommato all'inclinazione theta della struttura stessa del robot: $\gamma = \delta + \theta$

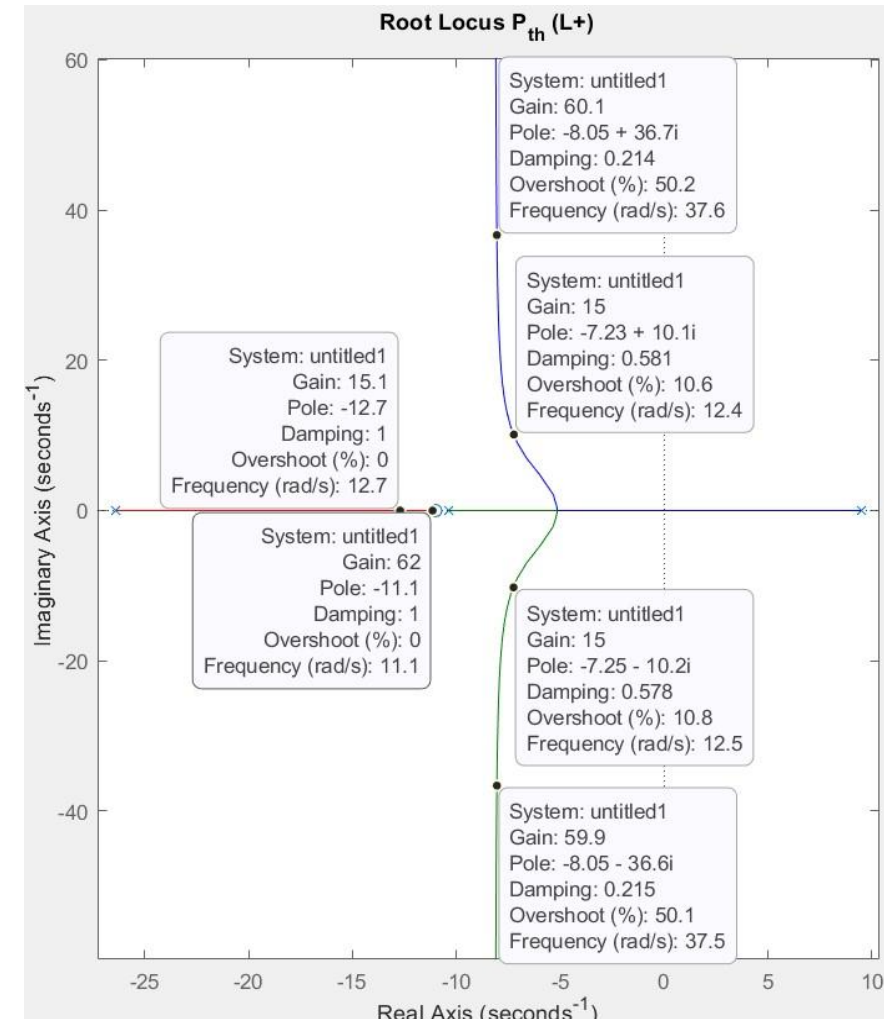
$$P_{\theta} = \frac{27,71s}{(s + 26,37)(s + 10,35)(s - 9,468)}$$

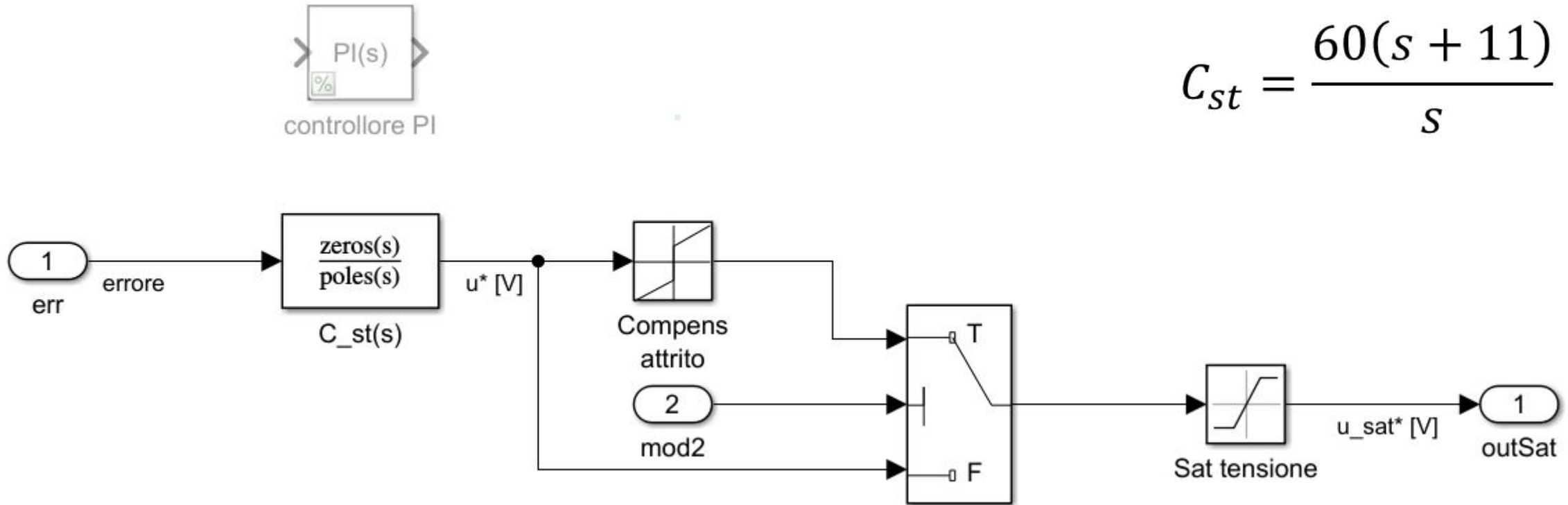


La fdt da $U(s)$ a $TH(s)$ è instabile, e va quindi stabilizzata.

Usiamo il luogo delle radici dopo aver inserito un controllore di tipo proporzionale-integrale:

$$C_{st} = \frac{60(s + 11)}{s}$$

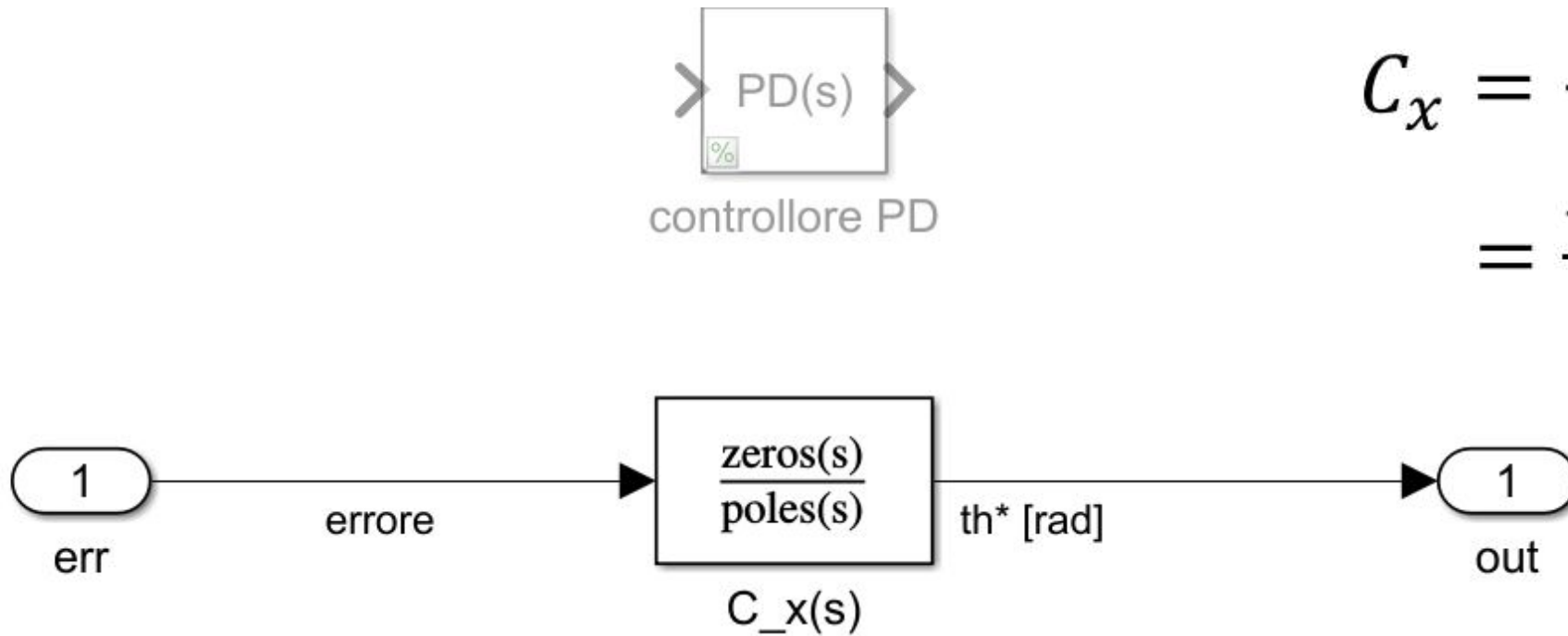




$$C_{st} = \frac{60(s + 11)}{s}$$

Il controllo di stabilizzazione interna include:

- il controllore $C_{st}(s)$, calcolato come una fdt con i parametri di un PI
- una saturazione dell'uscita in tensione
- un offset attorno allo 0V per ridurre le ondulazioni date dall'attrito statico e dal gioco che hanno i riduttori nel cambio di direzione, attivabile con la modalità 2 di funzionamento (premendo il pulsante 2)



$$C_x = -0,2 - \frac{3s}{s+100}$$

$$= \frac{-3,2s-20}{s+100}$$

Il controllo di posizione esterna include:

- il controllore $C_x(s)$, inizialmente ipotizzato come un controllore puramente derivativo, per poi tararlo manualmente con prove sperimentali e l'aggiunta di una componente proporzionale

1222 · 2022
800
ANNI



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

dtg