

Relazione finale

I semafori smart rappresentano un pilastro fondamentale della mobilità sostenibile, mirando a ridurre l'impatto ambientale dei trasporti e migliorando l'efficienza e la qualità della vita urbana.

La partecipazione ai Campionati di Problem Solving ha fornito l'occasione di approfondire e sperimentare tale tematica, con l'obiettivo di conoscere le potenzialità delle tecnologie innovative ai fini dello sviluppo sostenibile.

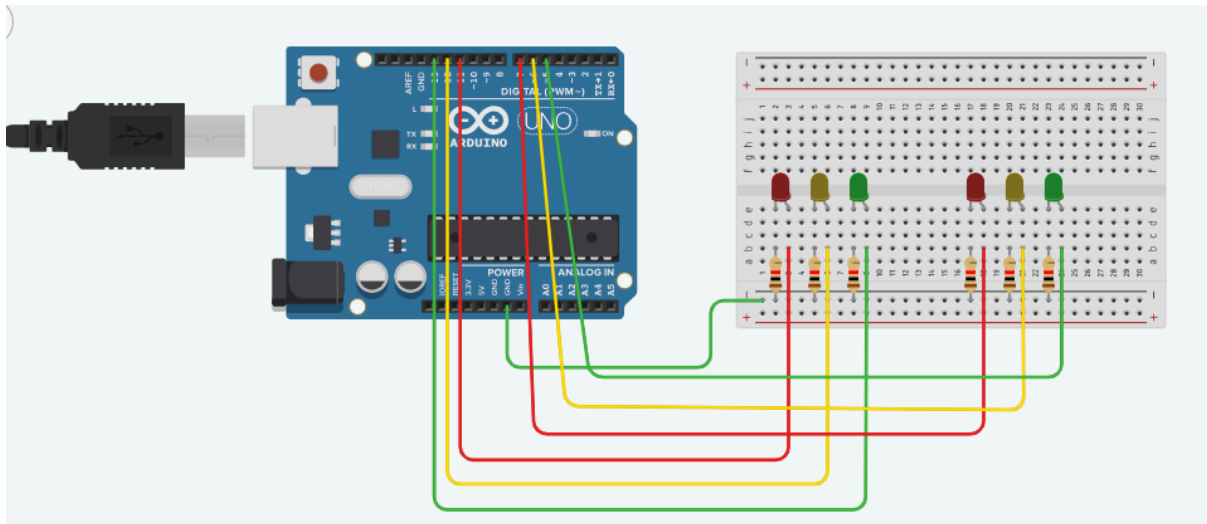
La sperimentazione in oggetto ha consentito di sviluppare competenze di progettazione e risoluzione di problemi, nel tentativo di concretizzare l'idea progettuale.

I discenti hanno simulato con Arduino il ciclo semaforico sincronizzato con timer e l'attivazione di semafori smart con la rilevazione dei veicoli attraverso l'uso di sensori ad ultrasuoni.

Semafori sincronizzati con timer

In un primo step è stato analizzato il **ciclo semaforico sincronizzato con timer** (semafori tradizionali), con l'obiettivo di comprenderne il principio di funzionamento. Sono state ipotizzate due direzioni. Lo stesso è stato riprodotto con Arduino Uno, prima con simulazione con Tinkercad Circuits (1) e, successivamente, con la realizzazione attraverso la scheda Arduino Uno, led e resistenze (2). La scheda Arduino è stata programmata con linguaggio a blocchi.

1.Simulazione semafori sincronizzati con Tinkercad Circuits



In allegato il codice (All.1 semafori_sincronizzati1.ino)

2. Semafori sincronizzati con Scheda Arduino Uno programmato con mBlock5-Codice sorgente

```
// generated by mBlock5 for <your product>  
// codes make you happy
```

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

void _delay(float seconds) {
  long endTime = millis() + seconds * 1000;
  while(millis() < endTime) _loop();
}

void setup() {
  pinMode(9,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(5,OUTPUT);
  while(1) {
    digitalWrite(9,0);
    digitalWrite(3,0);
    digitalWrite(8,0);
    digitalWrite(4,0);
    digitalWrite(11,1);
    digitalWrite(5,1);
    _delay(18);
    digitalWrite(11,1);
    digitalWrite(5,0);
    digitalWrite(4,1);
    _delay(3);
    digitalWrite(11,1);
    digitalWrite(3,1);
    digitalWrite(4,0);
    _delay(2);
    digitalWrite(11,0);
    digitalWrite(9,1);
    digitalWrite(3,1);
    _delay(20);
    digitalWrite(3,1);
    digitalWrite(8,1);
    digitalWrite(9,0);
    _delay(3);

    _loop();
  }
}
```

```

}

void _loop() {
}

void loop() {
  _loop();
}

```

In allegato il codice (All.2 Semafori sincronizzati-def.mblock).

Semafori con sensori ad ultrasuoni

In seguito si è passati a sperimentare l'uso **dei sensori ad ultrasuoni per simulare il comportamento dei semafori smart con Scheda Arduino Uno programmato con mblock5 (3)**. Sono state ipotizzate due direzioni. Entrambi i semafori sono rossi all'avvio della scheda; il verde scatta in una direzione solo se in quella direzione viene rilevata la presenza del veicolo dal sensore.

3. Semafori smart con Scheda Arduino Uno programmato con mblock5-Codice sorgente

```

// generated by mBlock5 for <your product>
// codes make you happy

#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

#define echoPin2 2
#define trigPin2 6
#define echoPin1 10
#define trigPin1 13
long duration1;
long distancel;
long duration2;
long distance2;
void Direzione_1(){
  _delay(3);
  digitalWrite(3,0);
  digitalWrite(5,1);
  _delay(15);
  digitalWrite(3,0);
  digitalWrite(5,0);
}

```

```

digitalWrite(4,1);
_delay(3);
digitalWrite(3,1);
digitalWrite(5,0);
digitalWrite(4,0);

}

void Direzione_2(){
  _delay(3);
digitalWrite(11,0);
digitalWrite(9,1);
_delay(15);
digitalWrite(8,1);
digitalWrite(9,0);
_delay(3);
digitalWrite(8,0);
digitalWrite(11,1);
digitalWrite(3,1);

}

void _delay(float seconds) {
  long endTime = millis() + seconds * 1000;
  while(millis() < endTime) _loop();
}

void setup() {
  pinMode(3,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(echoPin2,INPUT);
  pinMode(trigPin2,OUTPUT);
  pinMode(echoPin1,INPUT);
  pinMode(trigPin1,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(8,OUTPUT);
  digitalWrite(3,1);
  digitalWrite(11,1);
  while(1) {
    digitalWrite(trigPin1, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin1, HIGH);

```

```

delayMicroseconds(10);
digitalWrite(trigPin1, LOW);
duration1 = pulseIn(echoPin1, HIGH);
distance1 = duration1/58.2;
delay(50);
digitalWrite(trigPin2, LOW);
delayMicroseconds(2);
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distance2 = duration2/58.2;
delay(50);
if((distance1 < 10)  &&  (distance2 > 10)){
  Direzione_1();
  if((distance2 < 10)  &&  (distance1 > 10)){
    Direzione_2();

  }else{
    digitalWrite(3,1);
    digitalWrite(11,1);

  }

}

if((distance2 < 10)  &&  (distance1 > 10)){
  Direzione_2();
  if((distance1 < 10)  &&  (distance2 > 10)){
    Direzione_1();

  }else{
    digitalWrite(3,1);
    digitalWrite(11,1);

  }

}

_delay(3);

_loop();
}
}

```

```
void _loop() {  
}
```

```
void loop() {  
  _loop();  
}
```

In allegato il codice sorgente con linguaggio a blocchi (**A11.3 OPS-SmartTrafficLight.mblock**)

Il passo successivo della sperimentazione prevede l'applicazione di sensori e segnalatori acustici (buzzer) che consentano di simulare semafori smart accessibili ad ipovedenti.