

Arduino e la Mobilità: Sistema Intelligente per Pista Ciclabile nella Città del Futuro

Inquadramento del Progetto

Il progetto si inserisce nel tema del concorso “*Arduino e la mobilità: progetti per la città del futuro*”, proponendo una soluzione innovativa per migliorare la sicurezza e la gestione delle infrastrutture dedicate alla mobilità sostenibile.

L’idea sviluppata consiste nella progettazione e realizzazione di un modellino di pista ciclabile intelligente, all’interno del quale è integrato un sistema automatizzato di controllo degli accessi (*tool gate*) basato su tecnologia Arduino. Questo approccio consente di rappresentare in scala ridotta un sistema reale, evidenziando come le tecnologie elettroniche possano contribuire alla costruzione di città più sicure, efficienti e intelligenti.

Descrizione del Sistema e del Modellino

Il progetto prevede la realizzazione di un modellino fisico che riproduce una pista ciclabile urbana. All’interno di questo ambiente simulato è stato inserito un punto di accesso controllato costituito da una sbarra automatica.

Il modellino non ha solo una funzione dimostrativa, ma rappresenta una vera e propria simulazione di un’infrastruttura intelligente, in cui diversi componenti elettronici cooperano per garantire un utilizzo corretto dello spazio pubblico.

Elemento centrale del sistema è il cosiddetto *tool gate*, ovvero un meccanismo di accesso intelligente che consente il passaggio solo agli utenti che soddisfano determinati requisiti di sicurezza, in questo caso l’utilizzo del casco.

Architettura Tecnologica del Sistema

Dal punto di vista tecnico, il sistema è strutturato secondo un’architettura modulare basata su una scheda Arduino, che svolge il ruolo di unità di controllo centrale.

Arduino è collegato a:

- un modulo ricevitore utilizzato per la lettura dell’identificatore;
- un servo motore, responsabile del movimento meccanico della sbarra;
- un caschetto dotato di identificatore univoco, che funge da chiave di accesso al sistema.

Il caschetto rappresenta un elemento attivo: esso trasmette un codice identificativo che viene rilevato dal sistema quando il ciclista si avvicina alla sbarra.

Funzionamento del Sistema

Il funzionamento del sistema si basa su una sequenza automatizzata di operazioni.

Quando il ciclista si avvicina al punto di accesso, il modulo ricevitore rileva il codice identificativo associato al caschetto. Questo dato viene trasmesso ad Arduino, che esegue un confronto con una lista di codici autorizzati precedentemente memorizzati.

Il processo decisionale può essere descritto nel seguente modo:

- se il codice è valido, Arduino invia un segnale di controllo al servo motore, che provoca l'apertura della sbarra;
- dopo un intervallo di tempo prestabilito, la sbarra viene automaticamente richiusa;
- se il codice non è valido o non viene rilevato, il sistema mantiene la sbarra in posizione chiusa.

Questo meccanismo consente di automatizzare completamente il controllo degli accessi, eliminando la necessità di intervento umano.

Aspetti Tecnici e Implementativi

Dal punto di vista implementativo, Arduino gestisce:

- l'acquisizione dei dati tramite comunicazione seriale con il modulo ricevitore;
- l'elaborazione delle informazioni mediante un semplice algoritmo di confronto;
- l'attuazione del sistema tramite segnali PWM inviati al servo motore.

L'utilizzo di un identificatore univoco garantisce un buon livello di sicurezza, mentre l'automazione del processo aumenta l'affidabilità e riduce la possibilità di errore.

Il sistema è inoltre facilmente scalabile e modificabile, grazie alla flessibilità della piattaforma Arduino.

Ruolo nel Contesto della Smart City

Nel contesto della città del futuro, il sistema proposto rappresenta un esempio concreto di infrastruttura intelligente applicata alla mobilità sostenibile.

L'integrazione tra dispositivi elettronici, sensori e attuatori consente di:

- migliorare la sicurezza dei ciclisti;
- incentivare comportamenti corretti, come l'utilizzo del casco;
- ottimizzare la gestione degli accessi alle infrastrutture urbane.

In una visione più ampia, questo tipo di sistema potrebbe essere integrato in reti urbane intelligenti (Smart City), comunicando con altri dispositivi e contribuendo alla raccolta di dati sul traffico ciclabile.

Valore Innovativo e Possibili Sviluppi Futuri

Il valore innovativo del progetto risiede nella combinazione tra modellazione fisica e tecnologie digitali. Il modellino consente infatti di visualizzare concretamente il funzionamento di un sistema reale, rendendo più immediata la comprensione dei principi di automazione e controllo.

Tra i possibili sviluppi futuri si possono includere:

- integrazione con sistemi IoT per il monitoraggio remoto;
 - utilizzo di tecniche di crittografia per aumentare la sicurezza dei dati;
 - gestione tramite applicazioni mobili;
 - raccolta e analisi dei dati per lo studio dei flussi di traffico ciclabile.
-

Conclusioni

Il progetto dimostra come Arduino possa essere utilizzato per sviluppare soluzioni innovative nel campo della mobilità urbana.

Attraverso la realizzazione di un modellino di pista ciclabile intelligente con tool gate integrato, è stato possibile trasformare un'idea in un sistema concreto e funzionante, evidenziando il ruolo fondamentale della tecnologia nella costruzione di città più sicure, sostenibili e intelligenti.

In conclusione, questa esperienza ha permesso di approfondire concetti chiave come il controllo automatico, la comunicazione tra dispositivi e l'integrazione tra hardware e software, dimostrando il grande potenziale delle piattaforme embedded nello sviluppo delle infrastrutture del futuro.

Codice Sorgente

```
#include <SPI.h>

#include <MFRC522.h>

#include <Servo.h>

// ricevitore Pins

#define SS_PIN 10

#define RST_PIN 9

// LEDs

#define LED_ROSSO 6

#define LED_VERDE 7

// Servo motore

#define SERVO_PIN 5

Servo servoMotoreSbarra;

MFRC522 ricevitore(SS_PIN, RST_PIN);

// Posizioni servo motore

const int SBARRA_APERTA = 90;

const int SBARRA_CHIUSA = 180

        ;

// ---- Chiavi abilitate----

byte chiaveAbilitata[4] = {0xFB, 0x09, 0xAA, 0x02};

void setup() {

    SPI.begin();

    ricevitore.PCD_Init();

    delay(4);

    ricevitore.PCD_DumpVersionToSerial();

    pinMode(LED_ROSSO, OUTPUT);
```

```

pinMode(LED_VERDE, OUTPUT);
servoMotoreSbarra.attach(SERVO_PIN);
servoMotoreSbarra.write(SBARRA_CHIUSA);

//All' inizio il LED ROSSO è acceso
digitalWrite(LED_ROSSO, HIGH);
//All' inizio il LED VERDE è spento
digitalWrite(LED_VERDE, LOW);

}

bool confrontaChiaveDigitale(byte *a, byte *b) {
    for (byte i = 0; i < 4; i++)
        if (a[i] != b[i]) return false;
    return true;
}

void attendiLaRimozioneDellaCard() {
    while (ricevitore.PICC_IsNewCardPresent() && ricevitore.PICC_ReadCardSerial()) {
        delay(50);
    }
}

//Automa
void loop() {

    //Si accende il led rosso
    digitalWrite(LED_ROSSO, HIGH);

    //Si spegne il led verde
    digitalWrite(LED_VERDE, LOW);

    //si chiude la sbarra
    servoMotoreSbarra.write(SBARRA_CHIUSA);

    while(true){

```

```

//Attende che venga avvicinata una card al lettore
if (!ricevitore.PICC_IsNewCardPresent() || !ricevitore.PICC_ReadCardSerial()) {
    delay(100);
    continue;
}

//Si verifica se la chiave digitale letta è autorizzata
if (confrontaChiaveDigitale(ricevitore.uid.uidByte, chiaveAbilitata)) {
    apriSbarra();
} else {
    Serial.println("Unknown Card -- Access Denied");
    negaAccesso();
}

// Si aspetta la rimozione della card
attendiLaRimozioneDellaCard();
ricevitore.PICC_HaltA();
ricevitore.PCD_StopCrypto1();
delay(200);
}
}

void apriSbarra() {

//Si spegne il led rosso
digitalWrite(LED_ROSSO, LOW);

//Si accende il led verde
digitalWrite(LED_VERDE, HIGH);

//si apre la sbarra
servoMotoreSbarra.write(SBARRA_APERTA);

// si aspetta 5 secondi per il passaggio della bicicletta
delay(5000);

```

```
//si chiuda la sbarra
servoMotoreSbarra.write(SBARRA_CHIUSA);

//Si spegne la luce verde
digitalWrite(LED_VERDE, LOW);

//Si accende la luce rossa
digitalWrite(LED_ROSSO, HIGH);

}

void negaAccesso() {
    //Viene fatto lampeggiare il led rosso 3 volte per indicare che l'accesso è negato
    for (int i = 0; i < 3; i++) {
        digitalWrite(LED_ROSSO, LOW);
        delay(150);
        digitalWrite(LED_VERDE, HIGH);
        delay(150);
    }
}
```