SPI LED RING – MASTER

Sommario

[1 Configurazione del sistema 1](#_Toc515978993)

[1.1 Collegamento per SPI 1](#_Toc515978994)

[1.2 Tabella dei comandi inviati allo Slave 1](#_Toc515978995)

[1.3 Componenti utilizzati 1](#_Toc515978996)

[1.4 Passaggi configurazione software 2](#_Toc515978997)

[2 Implementazione del codice 2](#_Toc515978998)

[3 Problemi riscontrati 4](#_Toc515978999)

[4 Guide 4](#_Toc515979000)

# Configurazione del sistema

Il dispositivo Master si interfaccia con lo Slave tramite il protocollo SPI. Il suo ruolo è quello di inviare allo Slave gli effetti di luce da riprodurre, tramite un comando dedicato per ciascuno di essi che consiste in un Byte (seguire la tabella per i codici dei comandi). Quando il Master deve inviare allo Slave il tempo della sessione, e solo in questo caso, invia due Byte (uno dopo l’altro) che rappresentano le ore e i minuti di durata della sessione.

## Collegamento per SPI

Per far comunicare Master e Slave con il protocollo SPI servono cinque collegamenti tra i pin:

1. SCK: clock in comune tra i due dispositivi, emesso dal Master;
2. MISO: Master Input – Slave Output;
3. MOSI: Master Output – Slave Input;
4. SS: segnale usato per abilitare lo Slave; è attivo basso quindi va mantenuto a 0 Volt per abilitarlo e a 5 Volt per disabilitarlo;
5. GND: deve essere in comune.

Tutti i pin devono corrispondere alla stessa posizione tra i due dispositivi, esempio MISO del Master va collegato al MISO dello Slave, e lo stesso vale per gli altri pin del protocollo SPI.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **SCK** | **MISO** | **MOSI** | **SS** |
| Arduino Board  (Master) | Pin 13 | Pin 12 | Pin 11 | Pin 10 |
| Scheda con LED  (Slave) | Pin 17 (**PB5**) | Pin 16 (**PB4**) | Pin 15 (**PB3**) | Pin 14 (**PB2**) |

## Tabella dei comandi inviati allo Slave

Per evitare desincronizzazioni, la durata della sessione viene fornita dal Master il quale invia due byte dopo aver inviato SetBooked: il primo valore indica il numero di ore che la scrivania sarà occupata, mentre il secondo indica i minuti.

|  |  |  |
| --- | --- | --- |
| **Funzionalità** | **Codice** | **Descrizione** |
| Setup | 1 | Master si collega allo Slave, il quale risponde con lo stesso valore per confermare |
| DefaultState | - | Accensione dei LED nella configurazione iniziale |
| SetFree | 2 | LED accesi in colore verde in quanto la scrivania è libera |
| SetBooked | 3 | LED accesi in colore blu in quanto la scrivania è stata prenotata con successo |
| SetDenied | 4 | LED accesi in colore rosso in quanto la scrivania è occupata, o il tentativo di prenotazione non è valido |

## Componenti utilizzati

1. Modulo **Arduino ISP** per il caricamento del bootloader e dello sketch sullo Slave;
2. Collegamenti da usare per il protocollo SPI, ossia cinque cavi: MISO, MOSI, SS, CLK e GND;
3. Alimentazione esterna dello Slave.

## Passaggi configurazione software

1. Caricamento dello sketch su dispositivo Master (tramite Arduino IDE);
2. Preparazione di Arduino IDE al caricamento di dati su un microcontrollore diverso da ATmega328P:
   1. Cercare il file *avrdude.conf* nelle cartelle temporanee del sistema:
      1. In caso si utilizzasse Windows, andare nella cartella AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino8\etc\;
      2. In caso si utilizzasse Linux, andare nella propria Home e successivamente nella cartella nascosta .arduino15/packages/arduino/tools/avrdude/6.3.0-arduino8/etc/.
   2. Aprire tale file, cercare la stringa “m328p” e poche righe sotto sarà presente:

signature = 0x1e 0x95 0x0F;

* 1. Sostituire i due byte finali con “14”, in modo da avere come risultato la riga:

signature = 0x1e 0x95 0x14.

1. Caricamento del bootloader sullo Slave (in caso il microcontrollore ne fosse sprovvisto):
   1. Collegare Arduino ISP a Slave e alla porta del PC;
      1. In caso si utilizzasse Windows, installare i driver scaricabili dal sito ufficiale di Arduino (<https://store.arduino.cc/arduino-isp> -> sezione “Documentation” a metà pagina);
      2. In caso si utilizzasse Linux, potrebbe essere necessario abilitare i privilegi di lettura, scrittura ed esecuzione al dispositivo.
   2. Selezionare ArduinoISP come Programmatore nella sezione Tools dell’IDE;
   3. Cliccare su “Scrivi il bootloader”.
2. Caricamento dello sketch sullo Slave tramite la funzione “Carica tramite un programmatore” di ArduinoIDE, a cui si accede premendo contemporaneamente il tasto SHIFT e il bottone “Carica”.
3. Collegare Master e Slave seguendo il protocollo SPI.

# Implementazione del codice

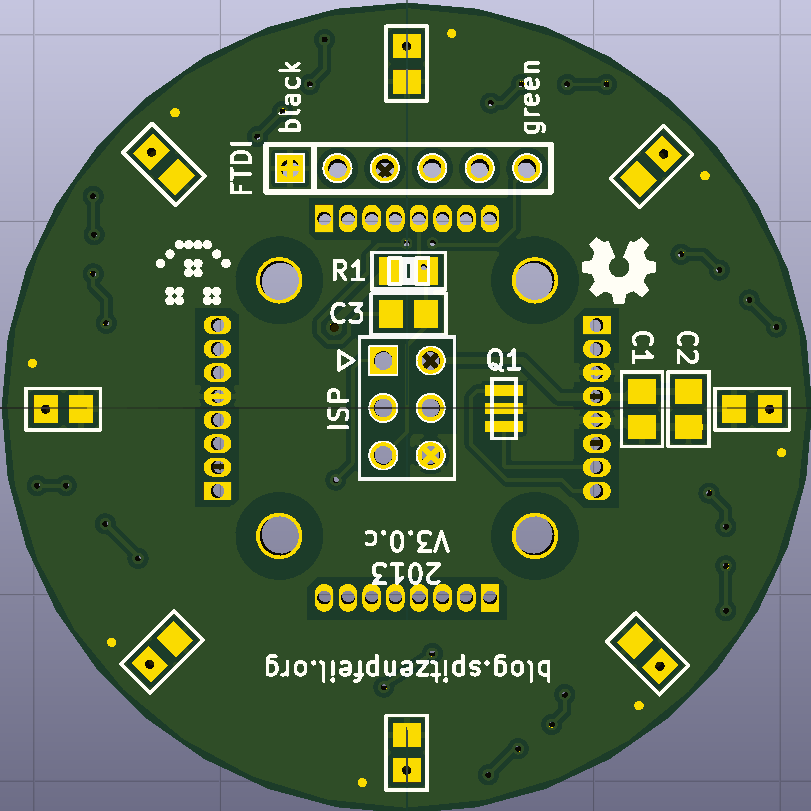
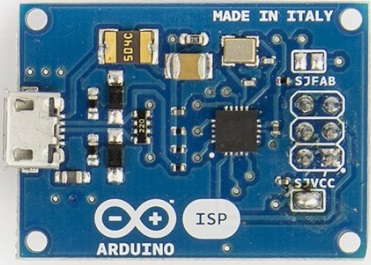
La prima fase è quella di Handshake tra Master e Slave. Una volta effettuata, il Master attende un input dall’utente che vuole autenticarsi (il tentativo di autenticazione nel nostro caso consiste nella pressione di un bottone, perciò non è stato implementato alcun controllo dell’identità) e autentica l’utente inizializzando la sessione; invia eventuali codici SetDenied durante la sessione e una volta terminata libera la scrivania.

Il seguente schema illustra il flusso completo di esecuzione del Master: 

# Problemi riscontrati

1. **Alimentazione dello Slave**: è necessario utilizzare un’alimentazione esterna, in quanto quella fornita dal Master non è sufficiente ad alimentare tutti i LED montati sulla scheda;
2. **Caricamento di codice sullo Slave**: è necessario utilizzare Arduino ISP perché la versione di ATmega dello Slave (328-AU) non è perfettamente compatibile con la versione montata sul Master (328p-PU), mentre con Arduino ISP non ci sono problemi di compatibilità.

Inoltre il connettore IPS presente sul retro della scheda non è speculare rispetto allo standard, di conseguenza è necessario utilizzare dei cavi per poter collegare il programmatore allo Slave tramite ISP:



Il pin MOSI nel programmatore, secondo la foto sopra, è quello in basso a destra (è segnato dal piccolo pallino bianco), mentre nella scheda è quello segnato dalla freccetta triangolare;

1. **Sincronizzazione SPI**: affinché la comunicazione tra i due dispositivi vada a buon fine, è necessario aggiungere dei ritardi (delay) di 15µs nel codice del Master.

# Guide

1. RGB LED Ring V3: <http://blog.spitzenpfeil.org/wordpress/2013/10/19/ws2812b-atmega168-breakout-a-k-a-rgb-led-ring-v3/>
2. Workaround caricamento bootloader su ATmega328-AU: <http://www.crash-bang.com/resource/bootload-atmega328/>
3. Video dimostrativo di Master e Slave: <https://drive.google.com/file/d/1mbogf5BUdaXdClg92zYNF4ls8EPiSI5C/view?usp=sharing>