



Università degli Studi di Bologna
Scuola di Ingegneria e Architettura

Corso di Laurea in Ingegneria Informatica

Sviluppo di progetti per fini didattici con Arduino

Giovanni Bonaccio

Anno accademico 2017/2018

Relatore:

Stefano Mattoccia

Obiettivi

Principale:

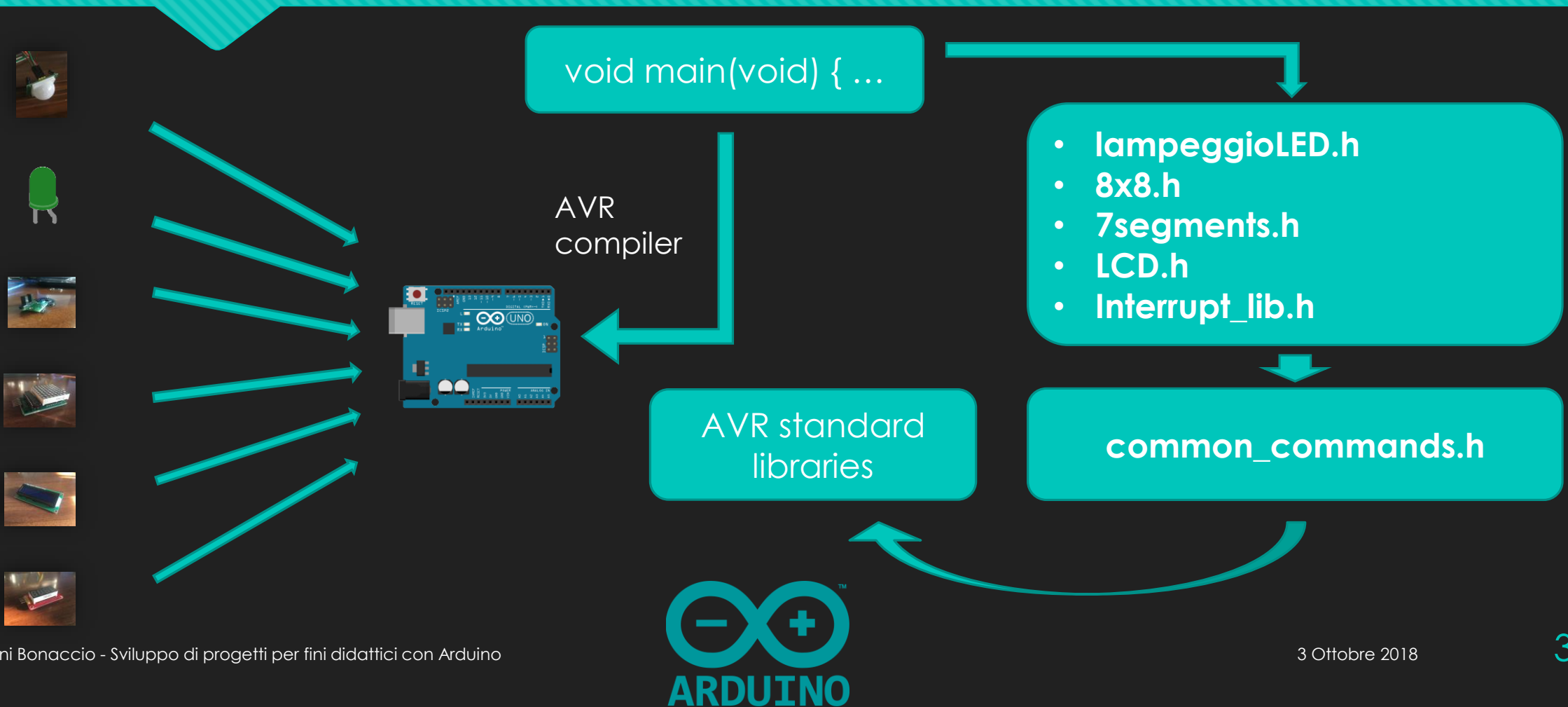
- Sviluppare piccoli progetti al fine di agevolare la comprensione del processo di scambio di dati tra dispositivi

Come?

- Interfacciandosi direttamente con i registri
- Attraverso la gestione degli Interrupt



Panoramica



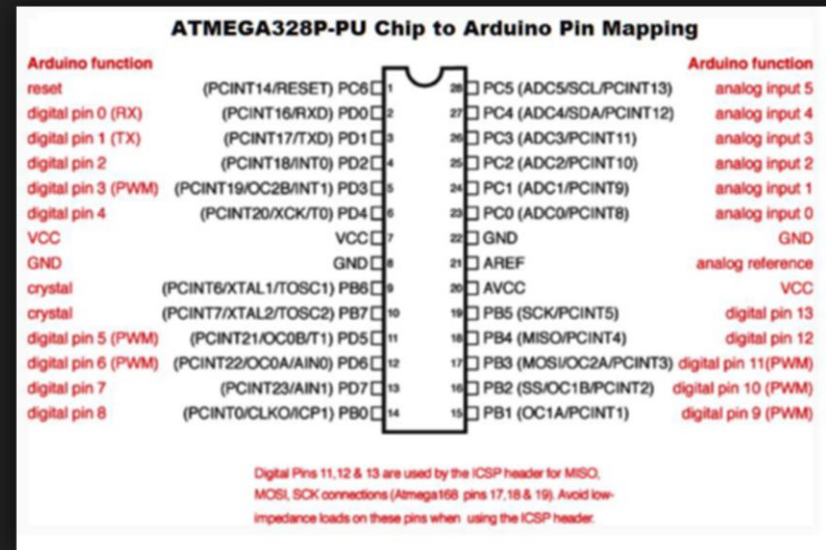
Pin mapping e interazioni con i registri

Microcontrollore ATMEGA:

- Porta B
- Porta C
- Porta D

Registri per ogni porta:

- DDRB
- PORTB
- PINB

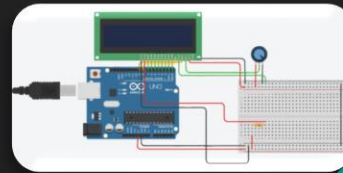


```
1 void main (void)
2 {
3     /* settiamo il pin13 (PB5) come output*/
4     DDRB = DDRB | 0b00100000;
5
6     while(1) {
7         /* accendi il led */
8         PORTB = PORTB | 0b00100000;
9         _delay_ms(500);
10
11        /* spegni il led */
12        PORTB = PORTB & 0b11011111;
13        _delay_ms(500);
14    }
15
16 }
```

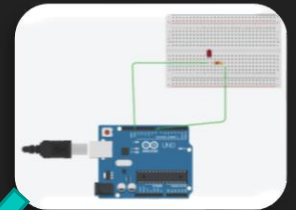


common_commands.h

- Include le librerie AVR necessarie per le operazioni I/O e gestione interrupt
- Funzioni comuni a tutti gli esperimenti che permettono di assegnare il valore o la direzione dei dati di un pin
- Funzioni per il settaggio degli interrupt

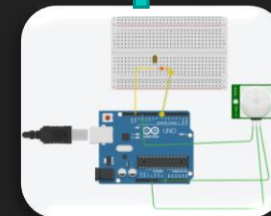


port_set_high_bits_port_D(BYTE)

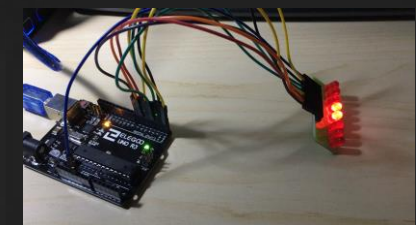
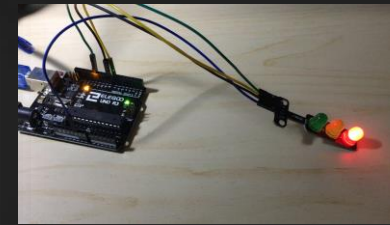
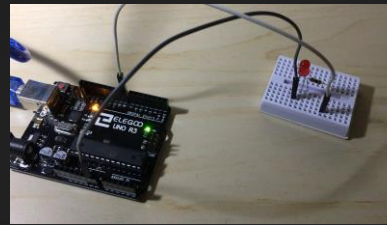


common_commands.h

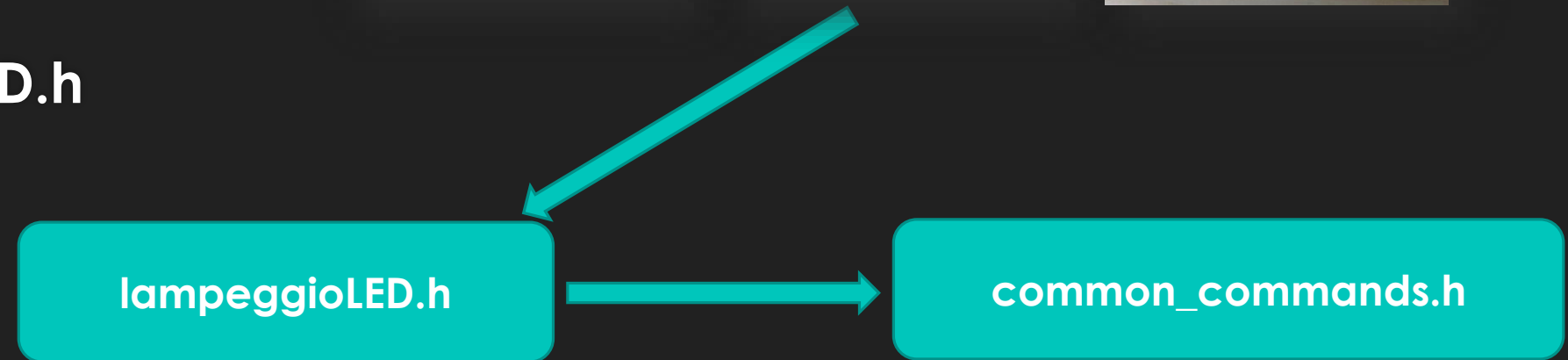
port_setup_set_data_direction_output_bits_port_B(BYTE)



Realizzazione dei progetti: lampeggio di led



Libreria lampeggioLED.h



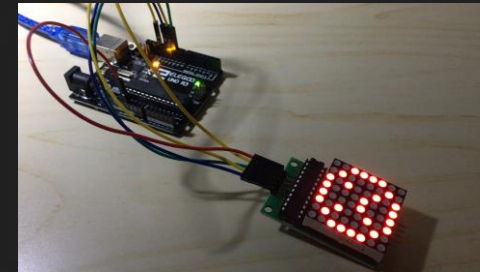
Realizzazione dei progetti: matrice di led 8x8

Libreria 8x8.h

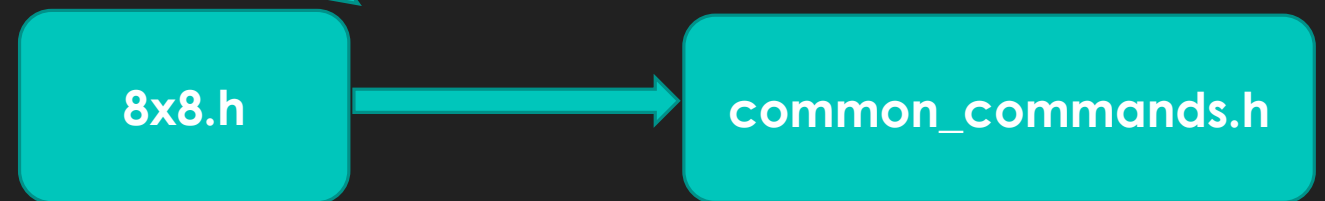
- Definisce una funzione per la scrittura dei dati
- Definisce una funzione per il settaggio delle impostazioni di default

Caratteristiche del componente:

- Monta un microcontrollore
- Si accede ai led attraverso il numero della colonna
- DIN e CLK



`led_matrix_8x8_write_data_output(data, address)`



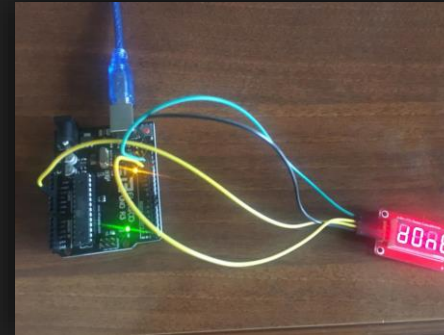
Realizzazione dei progetti: display a 7 segmenti

Libreria 7segments.h:

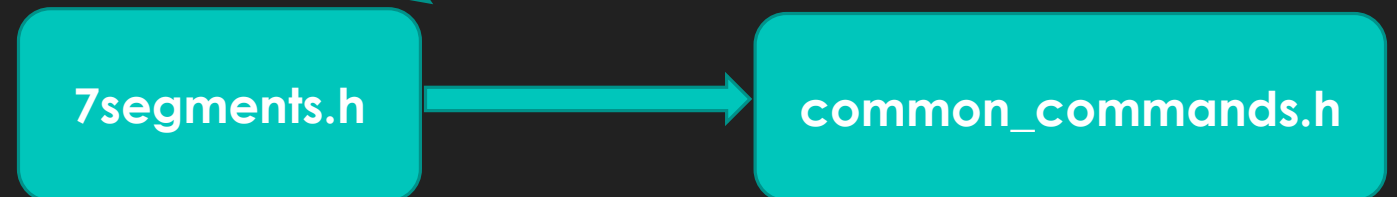
- Definisce funzioni per la scrittura delle cifre

Caratteristiche del componente:

- Monta un microcontrollore
- Tre comandi ad ogni trasferimento
- Ogni cifra ha un indirizzo d'accesso
- DIO e CLK



display_7segments_set_segments(data, address)



Realizzazione dei progetti: display LCD

Libreria LCD.h:

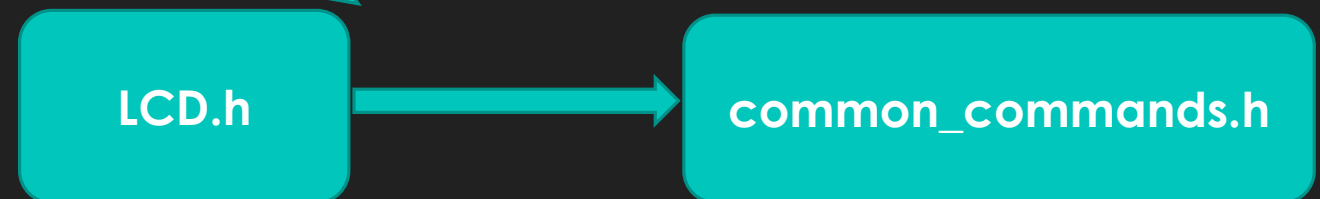
- Definisce funzioni per la scrittura dei caratteri e dei comandi
- Definisce funzione per il settaggio delle impostazioni di default

Caratteristiche del componente:

- Interfacciamento diretto con tutti i pin
- Richiede l'utilizzo di un potenziometro
- Basta assegnare un carattere alla porta per scrivere sul display



`lcd_display_write_character(BYTE)`



Gestione degli Interrupt

- Funzioni per la gestione nella libreria `common_commands.h`

Registri per il settaggio:

- `PCMSK0`, `PCMSK1`, `PCMSK2`
- `PCICR`

- Handler definito in `avr/interrupt.h` utilizzando la macro:

```
#define ISR(vector, ...) extern "C" void vector(void) __attribute__((signal, __INTR_ATTRS)) __VA_ARGS__; \ void vector(void)
```

- Macro ridefinita come `interrupt_handler(vector)` per rendere il codice più leggibile

Realizzazione dei progetti: sensore PIR, encoder

Libreria Interrupt_lib.h

- Definisce l'handler

Dispositivi utilizzati:

- Display a 7 segmenti/LCD/Led
- Sensore PIR / Encoder come generatore di interrupt



Interrupt_handler(PCINTX_vect)

Interrupt_lib.h

common_commands.h



Conclusione

- Codice semplice ed intuitivo
- Incrocio tra concetti chiave della programmazione e dell'elettronica

Per eventuali sviluppi futuri

- Implementazione di nuove librerie
- Utilizzo di nuovi dispositivi
- Realizzazione di progetti più complessi evitando troppe astrazioni



Grazie per l'attenzione...

