Università degli Studi di Brescia
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Elettronica e delle Telecomunicazioni

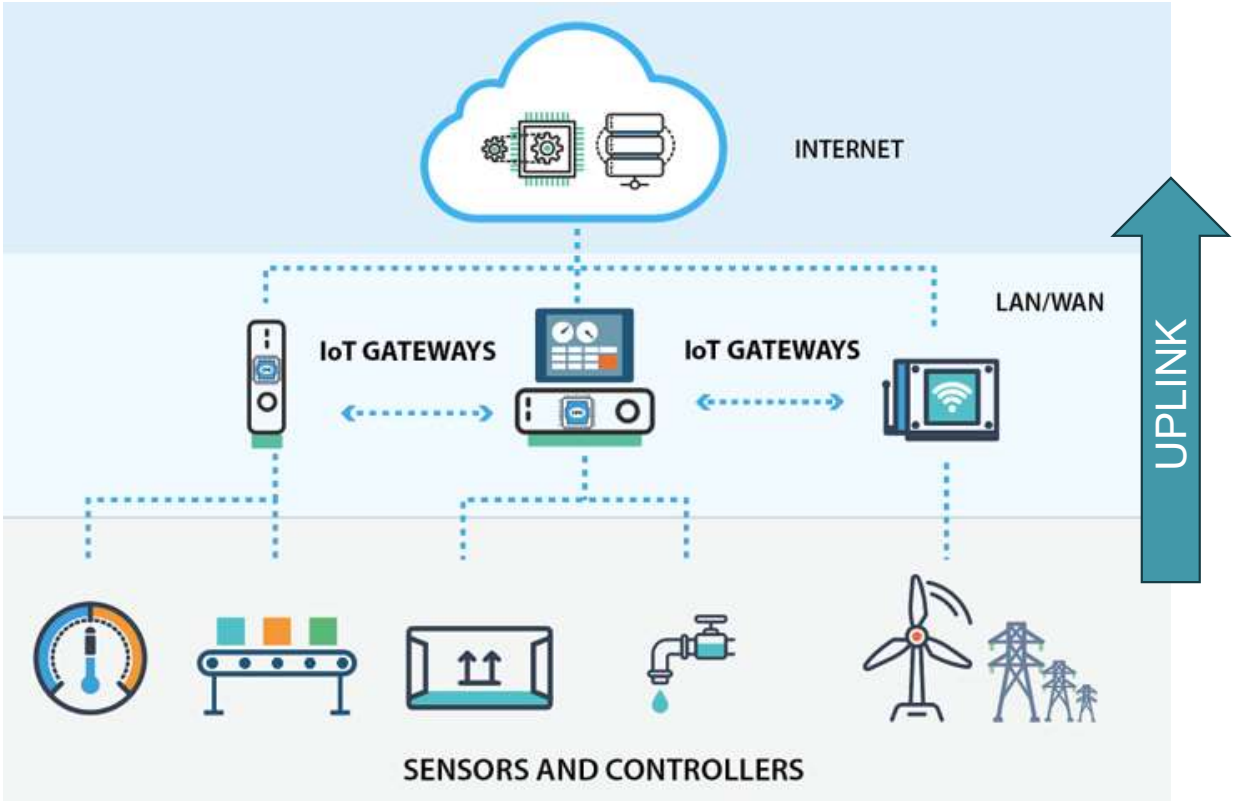
Anno accademico 2024/2025

Sistema di inserimento numerico tramite tastiera su LCD16x2 connesso a ThingSpeak

Stefano Molari 727197
Daniil Kretinin 729256
Prof. Emiliano Sisinni

INTERNET OF THINGS - IoT

- Smart things: offrono capacità di elaborazione e di comunicazione
- Campi di impiego
  - ✓ Industria (IIoT)
  - ✓ Agricoltura
  - ✓ Domotica
- Architettura a tre livelli
- Flusso bidirezionale (uplink direzione preferenziale)
- Eterogeneità delle applicazioni e dei protocolli di comunicazione a livello di campo



OBIETTIVI

Conoscere

- Circuiti digitali e IOT
- FPGA, linguaggio VHDL, linguaggio C
- Funzionamento di un LCD

Progettare

- Modelsim
- Intel Quartus II
- Programmazione HPS in C
- Realizzazione circuito elettrico

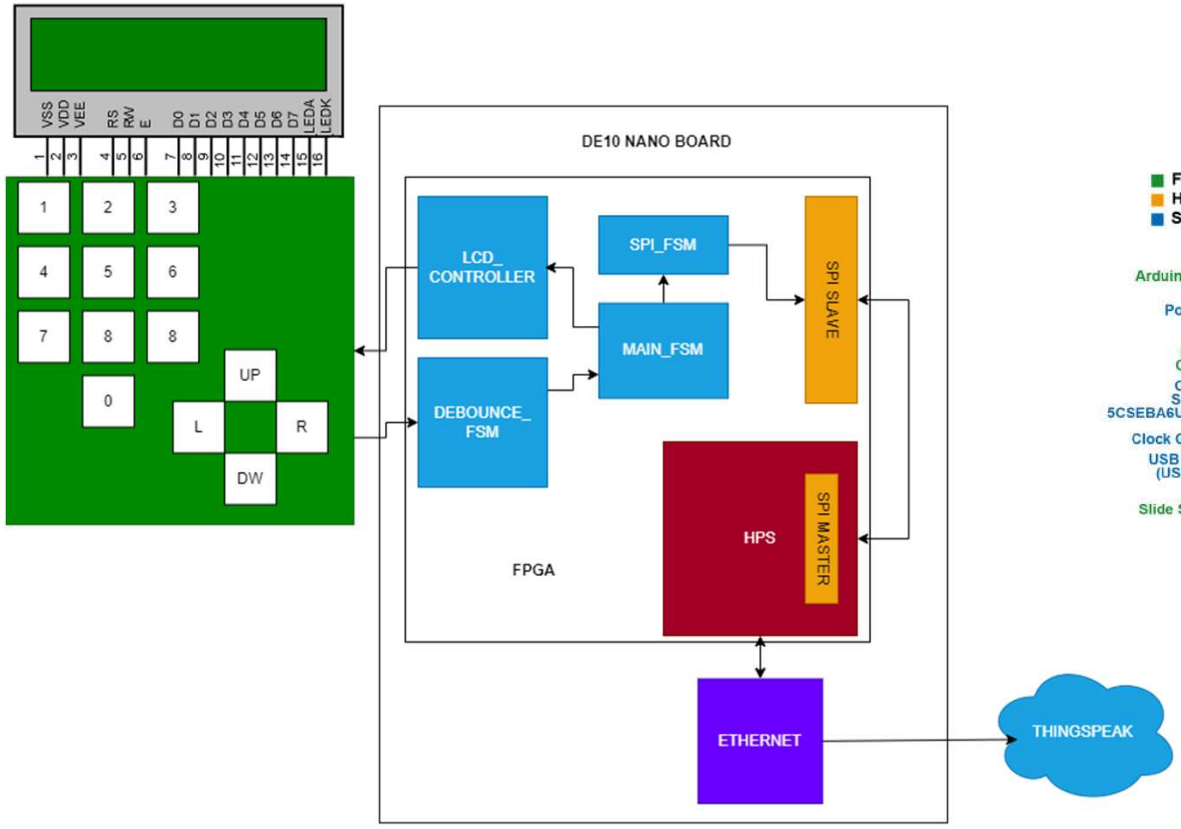
Verificare

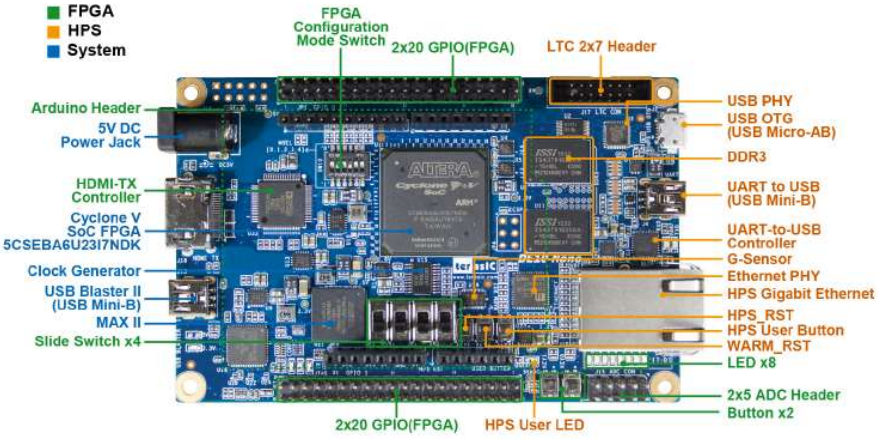
- Simulazioni
- Test su FPGA con analizzatore logico
- Verifica funzionamento complessivo

FLUSSO DEL PROGETTO

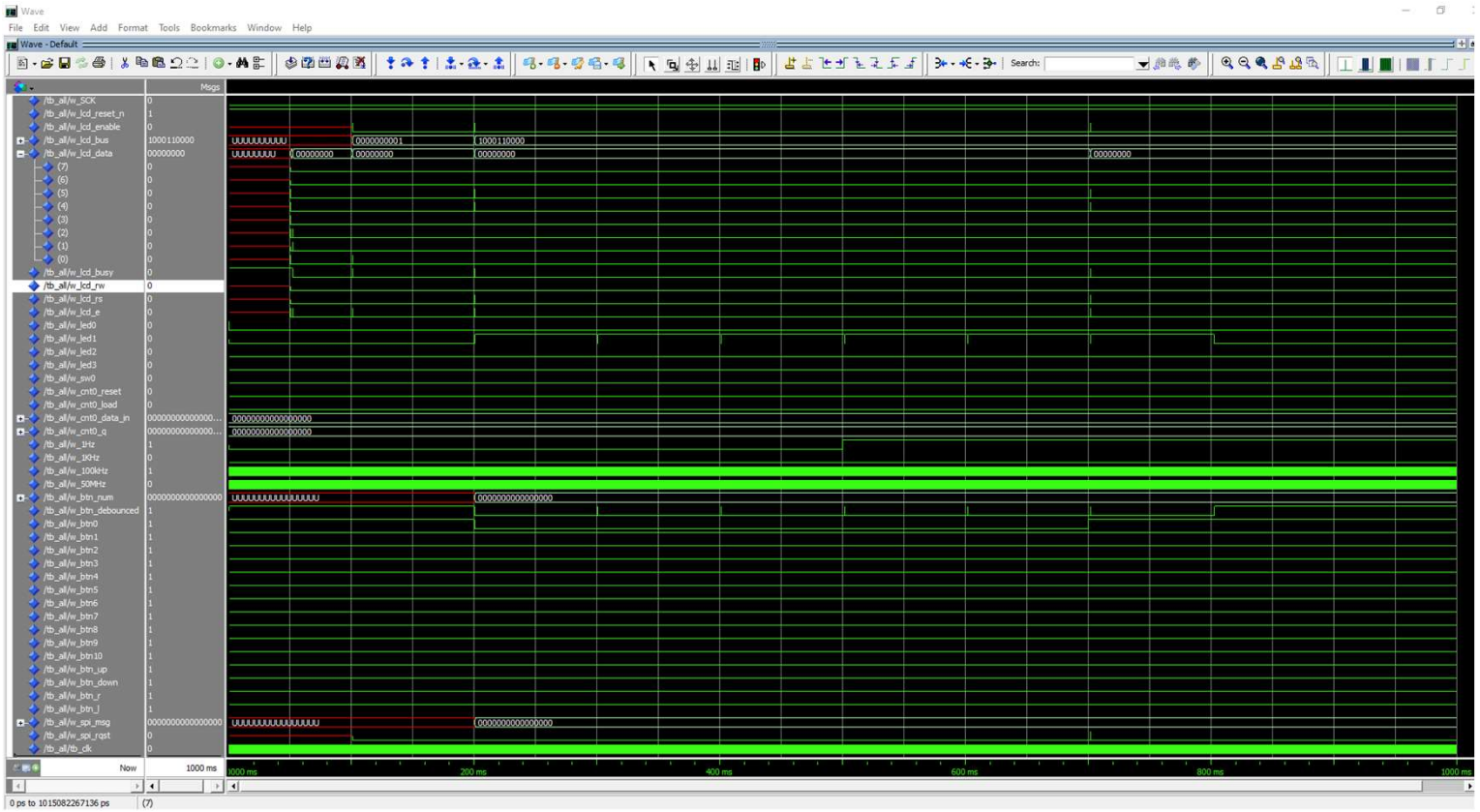
- Definizione delle specifiche
- Realizzazione di schemi a blocchi
- Scrittura codice VHDL per FPGA
- Simulazione con Modelsim
- Sintesi tramite Quartus
- Scrittura codice C per HPS
- Montaggio circuito fisico
- Verifica programma e circuito tramite analizzatore logico e ThingSpeak
- Stesura relazione e presentazione

SCHEMA A BLOCCHI DI PRINCIPIO

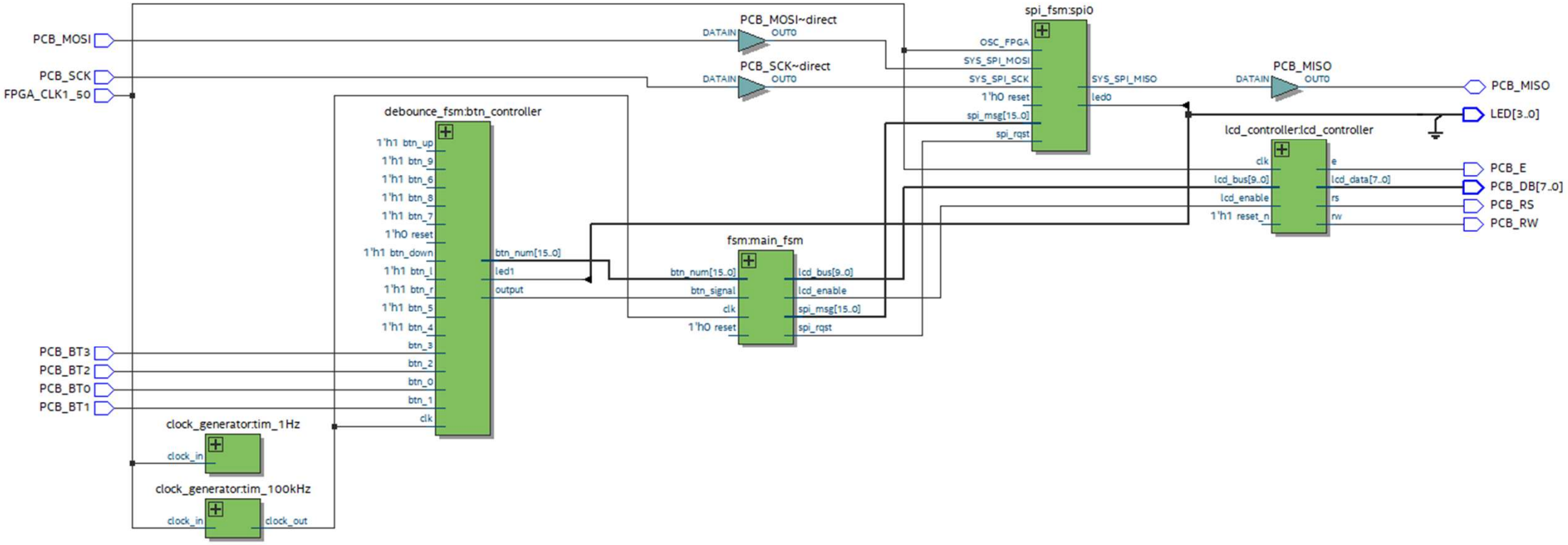




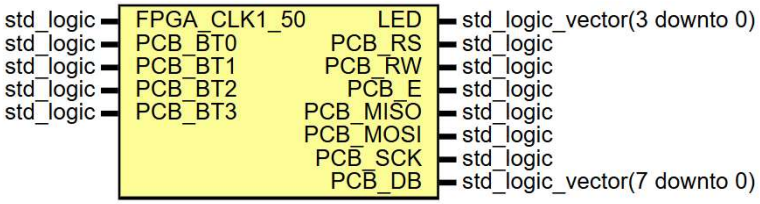
SIMULAZIONE MODELSIM

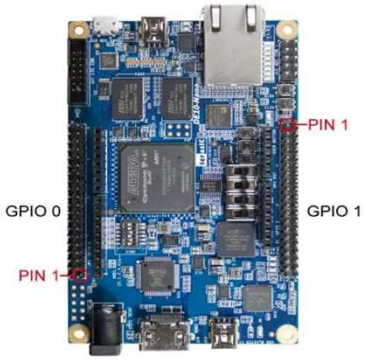


SINTESI E VISTA RTL



TOP



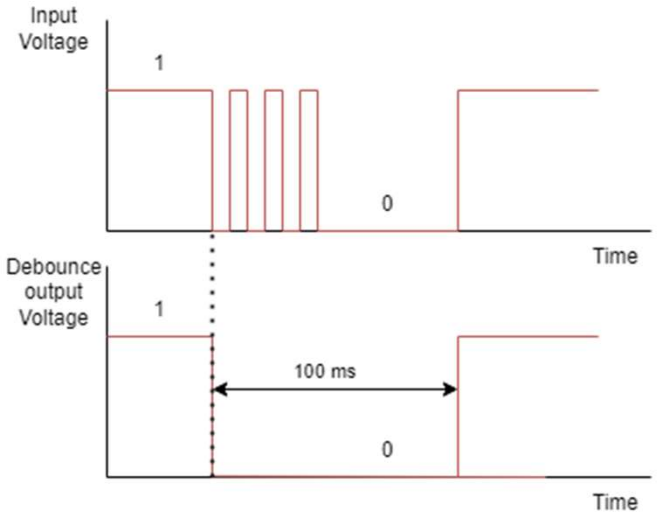


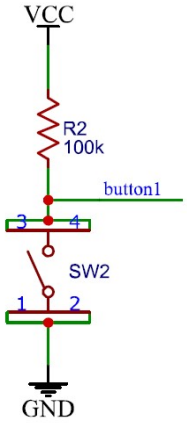
GPIO 0 (JP1)

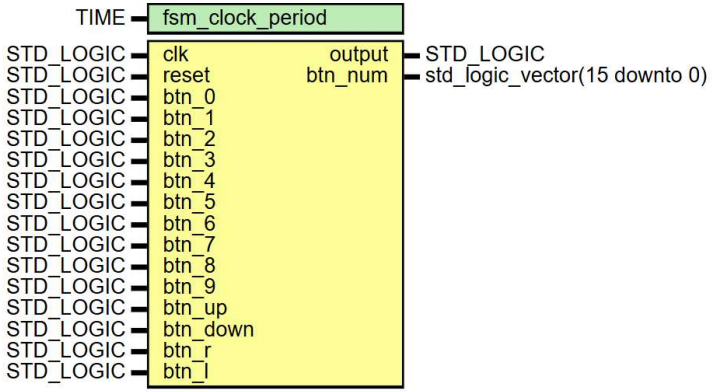
RIGHT	GPIO_0[0]	1		2	GPIO_0[1]	PIN_E8	DB7	GPIO_1[0]	PIN_AC24
UP	GPIO_0[2]	3		4	GPIO_0[3]	PIN_D11	DB6	GPIO_1[2]	PIN_AD26
BT9	GPIO_0[4]	5		6	GPIO_0[5]	PIN_AH13	DB5	GPIO_1[4]	PIN_AF28
BT6	GPIO_0[6]	7		8	GPIO_0[7]	PIN_AH14	DB4	GPIO_1[6]	PIN_AF27
BT3	GPIO_0[8]	9		10	GPIO_0[9]	PIN_AH3	DB3	GPIO_1[8]	PIN_AH27
5V	5V	11		12	GND	Gnd	5V	5V	
BT0	GPIO_0[10]	13		14	GPIO_0[11]	PIN_AG14	DB2	GPIO_1[10]	PIN_AH26
PIN_AE23	GPIO_0[12]	15		16	GPIO_0[13]	PIN_AE6	PIN_AH24	GPIO_1[12]	PIN_AF25
PIN_AD23	GPIO_0[14]	17		18	GPIO_0[15]	PIN_AE24	PIN_AG23	GPIO_1[14]	PIN_AF23
PIN_D12	GPIO_0[16]	19		20	GPIO_0[17]	PIN_AD20	PIN_AG24	GPIO_1[16]	PIN_AH22
PIN_C12	GPIO_0[18]	21		22	GPIO_0[19]	PIN_AD17	PIN_AH21	GPIO_1[18]	PIN_AG21
PIN_AC23	GPIO_0[20]	23		24	GPIO_0[21]	PIN_AC22	BT1	GPIO_1[20]	PIN_AA20
PIN_Y19	GPIO_0[22]	25		26	GPIO_0[23]	PIN_AB23	BT2	GPIO_1[22]	PIN_AE22
PIN_AA19	GPIO_0[24]	27		28	GPIO_0[25]	PIN_W11	BT4	GPIO_1[24]	PIN_AF21
3V3	3.3V	29		30	GND	Gnd		3.3V	29	30	GND	
DB1	GPIO_0[26]	31		32	GPIO_0[27]	PIN_W14	BT5	GPIO_1[26]	PIN_AH19
DB0	GPIO_0[28]	33		34	GPIO_0[29]	PIN_Y17	BT7	GPIO_1[28]	PIN_AH18
E	GPIO_0[30]	35		36	GPIO_0[31]	SCK	BT8	GPIO_1[30]	PIN_AF20
RW	GPIO_0[32]	37		38	GPIO_0[33]	MOSI	LEFT	GPIO_1[32]	PIN_AE20
RS	GPIO_0[34]	39		40	GPIO_0[35]	MISO	DOWN	GPIO_1[34]	PIN_AE17

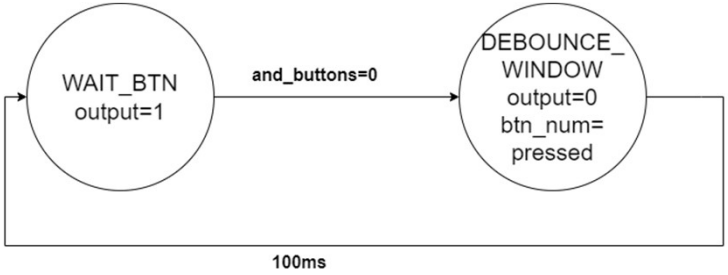
GPIO 1 (JP7)

DEBOUNCE FSM





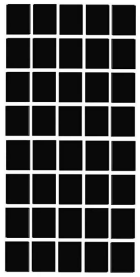




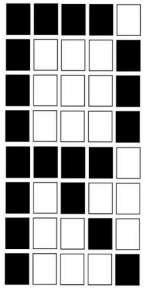
Funzionamento LCD

No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	V0	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

5 x 8
dot matrix display



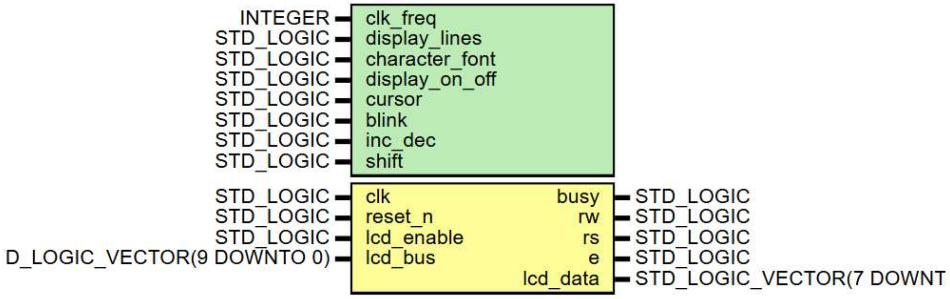
If we want to print
character
“R”

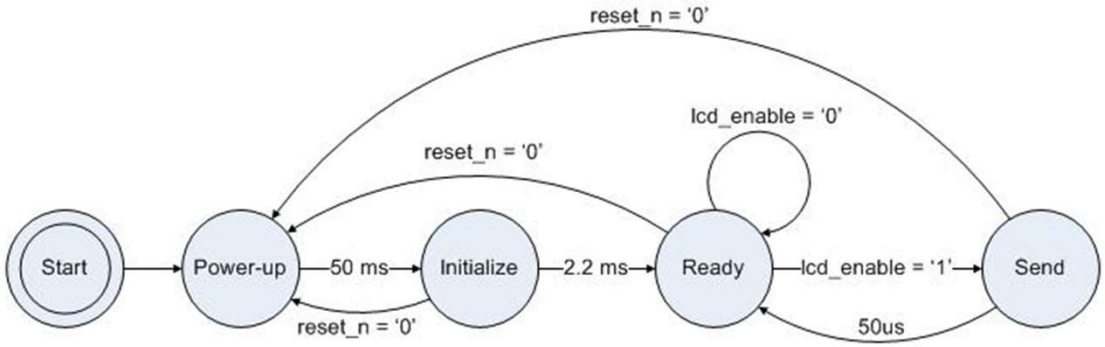


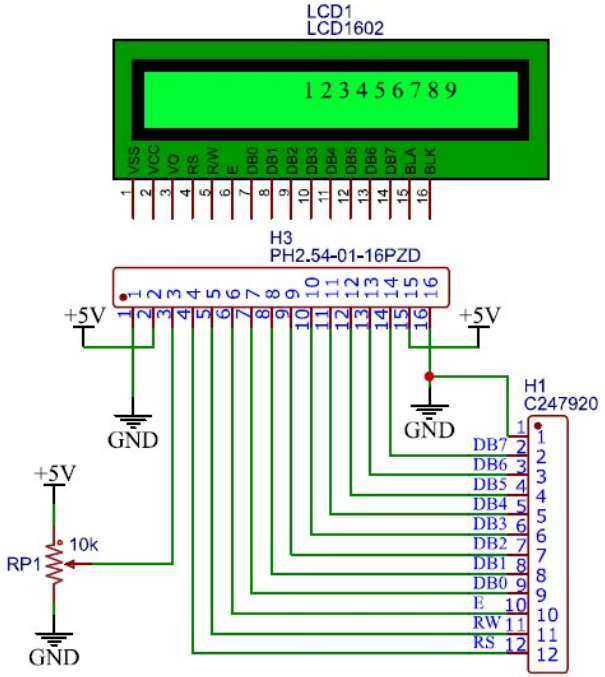
Sr. No.	Command in Hex	Operation
1	01	Clear display
2	02	Return home
3	04	Shift cursor to left
4	05	Shift display to right
5	06	Shift cursor to right
6	07	Shift display to left
7	08	Display off, Cursor off
8	0A	Display off, Cursor on
9	0C	Display on, Cursor off
10	0E	Display on, Cursor on
11	0F	Display on, Cursor blinking
12	10	Shift cursor to left
13	14	Shift cursor to right

Sr. No.	Command in Hex	Operation
14	80	Cursor at first line, first character
15	C0	Cursor at second line, first character
16	38	2 Lines, 5x7 crystal matrix and 8 bit mode
17	28	2 Lines, 5x7 crystal matrix and 4 bit mode

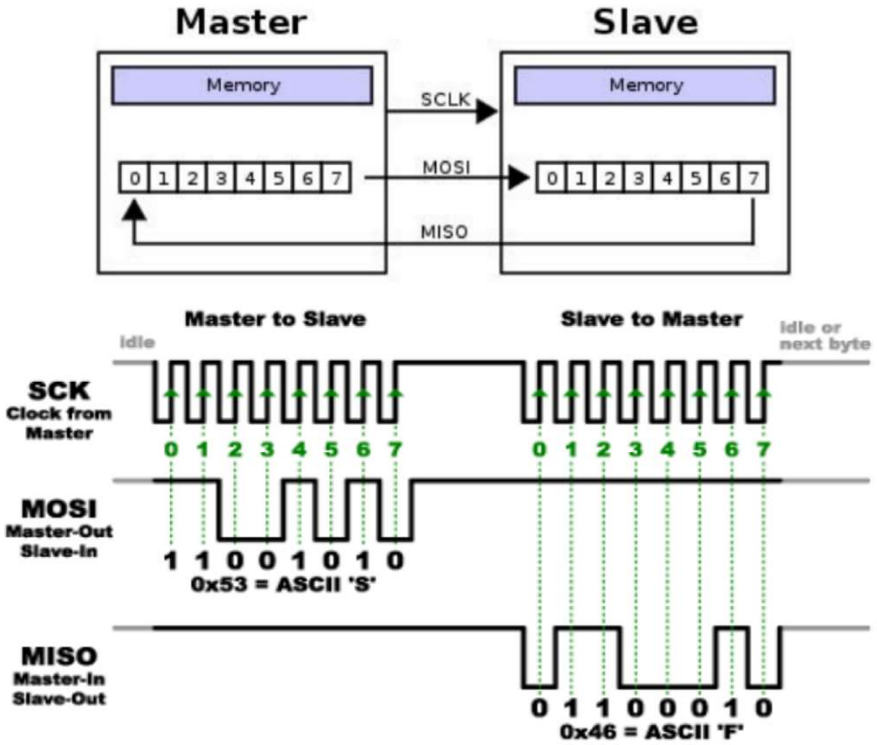
LCD CONTROLLER

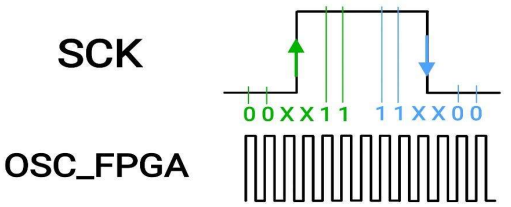




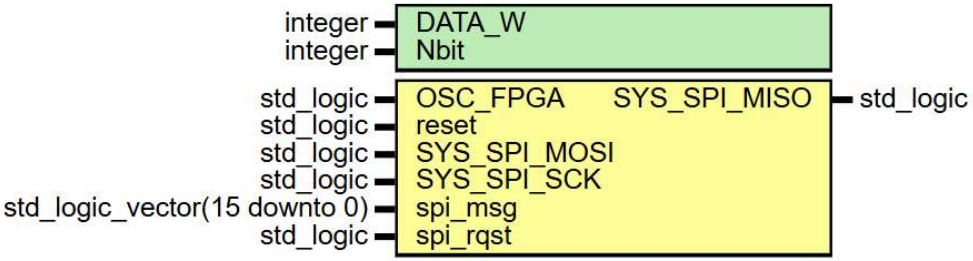


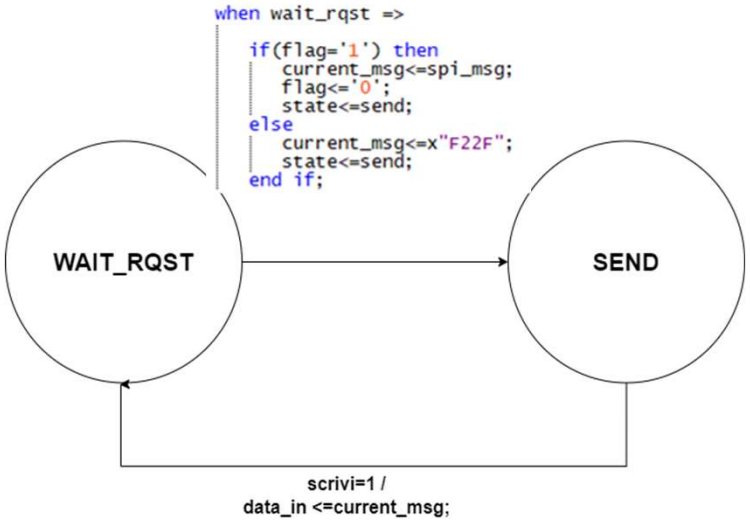
FUNZIONAMENTO SPI

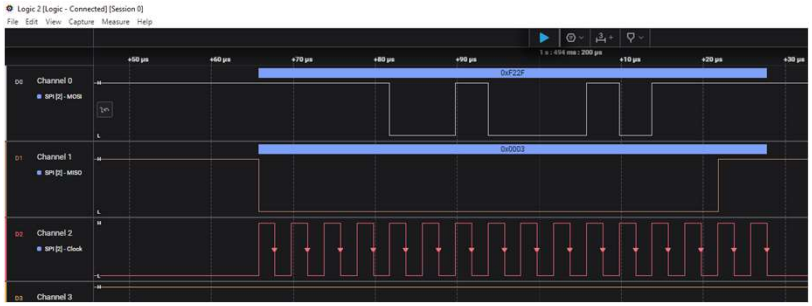


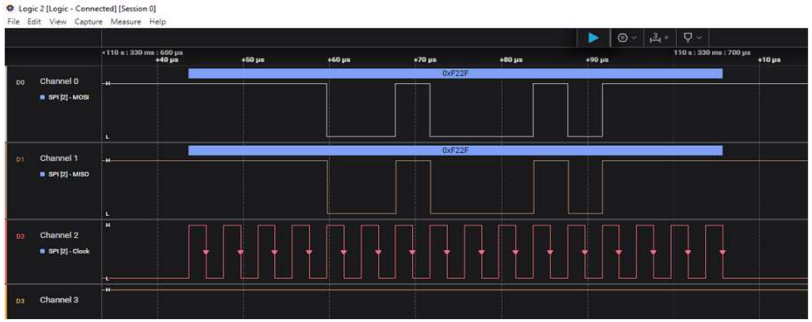


SPI FSM

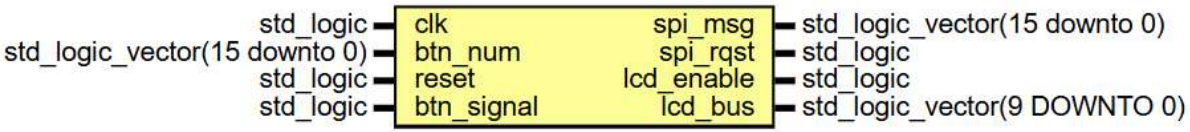

integer
integer
DATA_W
Nbit
std_logic
std_logic
std_logic
std_logic
std_logic_vector(15 downto 0)
std_logic
OSC_FPGA
SYS_SPI_MISO
reset
SYS_SPI_MOSI
SYS_SPI_SCK
spi_msg
spi_rqst
std_logic

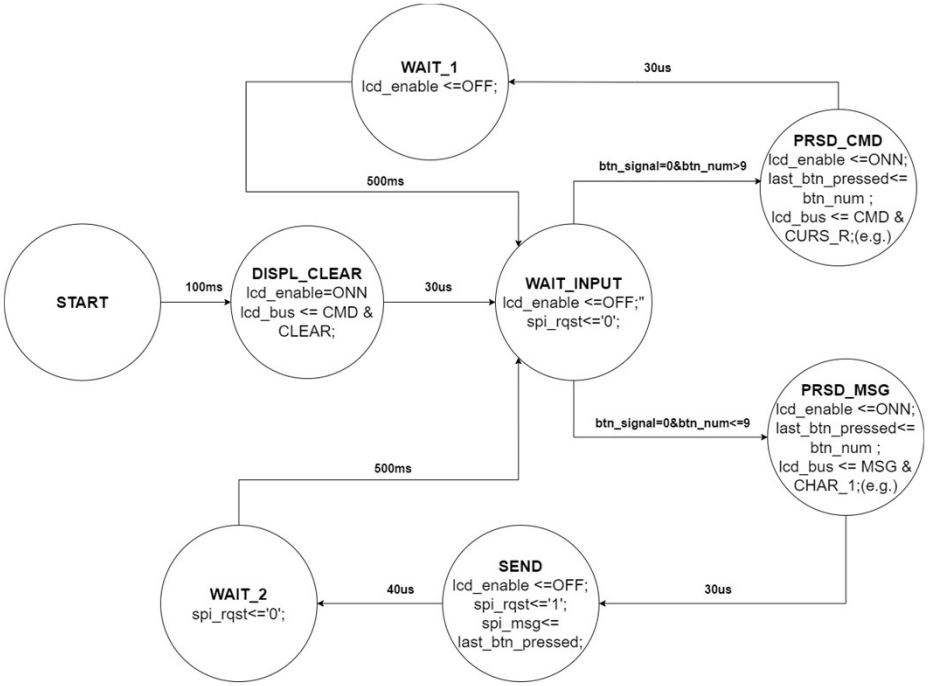

when wait_rqst =>
if(flag='1') then
current_msg<=spi_msg;
flag<='0';
state<=send;
else
current_msg<=x"F22F";
state<=send;
end if;
WAIT_RQST
SEND
scrivi=1 /
data_in <=current_msg;


Channel 0
Channel 1
Channel 2
Channel 3
0xF22F
0x0003


Channel 0
Channel 1
Channel 2
Channel 3
0xF22F
0xF22F

MAIN FSM

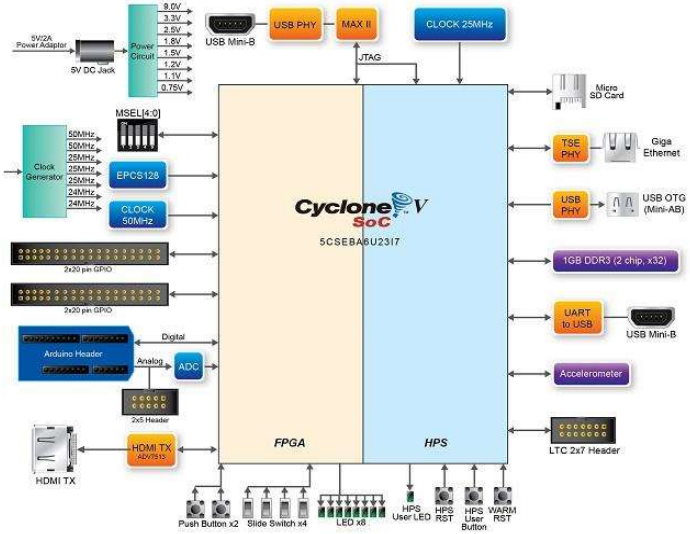

std_logic
std_logic_vector(15 downto 0)
std_logic
std_logic
clk
btn_num
reset
btn_signal
spi_msg
spi_rqst
lcd_enable
lcd_bus
std_logic_vector(15 downto 0)
std_logic
std_logic
std_logic_vector(9 DOWNTO 0)


WAIT_1
lcd_enable <=OFF;
30us
PRSD_CMD
lcd_enable <=ONN;
last_btn_pressed<=
btn_num ;
lcd_bus <= CMD &
CURS_R;(e.g.)
btn_signal=0&btn_num>9
500ms
START
100ms
DISPL_CLEAR
lcd_enable=ONN
lcd_bus <= CMD &
CLEAR;
30us
WAIT_INPUT
lcd_enable <=OFF;"
spi_rqst<='0';
PRSD_MSG
lcd_enable <=ONN;
last_btn_pressed<=
btn_num ;
lcd_bus <= MSG &
CHAR_1;(e.g.)
btn_signal=0&btn_num<=9
500ms
WAIT_2
spi_rqst<='0';
40us
SEND
lcd_enable <=OFF;
spi_rqst<='1';
spi_msg<=
last_btn_pressed;
30us

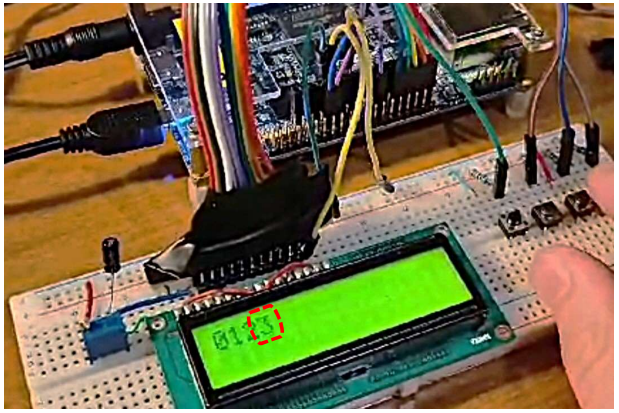
HPS

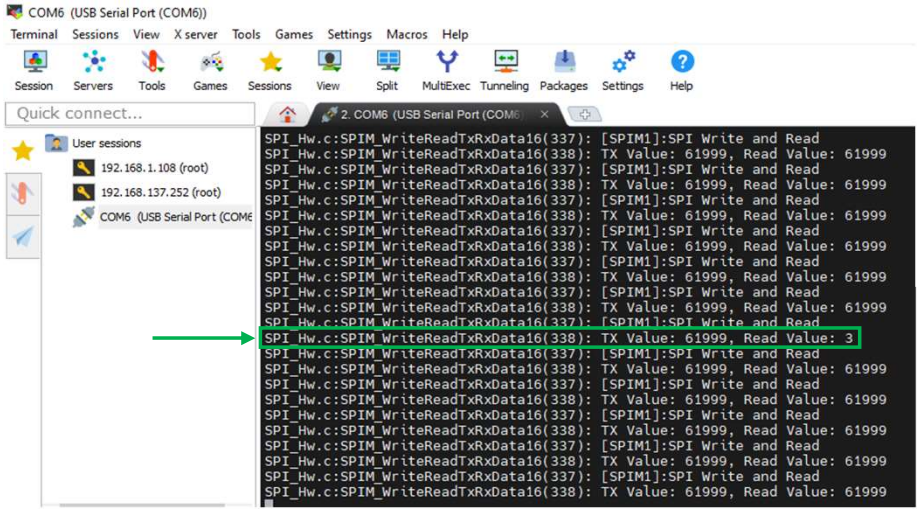
```c
printf("LCD VISUALIZER\n");
int16_t received = 0xF22F; //data received from slave on MISO, 16 bit
char bufferurl [200];
int to_send = 0;           //data to send on ThingSpeak
while(1){ // infinite loop

    received = SPIM_WriteReadTxRxData16(0xF22F); //sending dummy data on MOSI (hex F22F) and read data from MISO and store in "received"

    //check if SLAVE has sent back F22F or the BUTTON DATA
    if(received == 0x0000 ||received == 0x0001||received == 0x0002||received == 0x0003||received == 0x0004||received == 0x0005||received == 0x0006|
        switch(received){//conversion from 16 bit to int, in base of the API %d
            case 0x0000:
                to_send=0;
                break;
            case 0x0001:
                to_send=1;
                break;
```

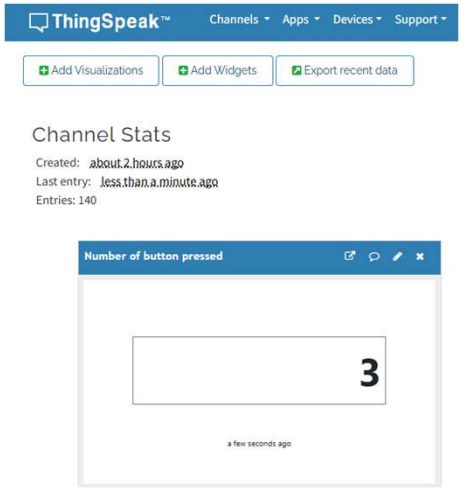
```c
        }
        // url per la scrittura di dati del canale thingspeak
        sprintf(bufferurl, "https://api.thingspeak.com/update?api_key=OP8H82Y91IYI1FYB&field1=%d",to_send);
        dl_curl_easy_setopt(curl, CURLOPT_URL, bufferurl);
        res = dl_curl_easy_perform(curl);

        // Verifica se la richiesta è stata eseguita con successo
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                    dl_curl_easy_strerror(res));
        } else {
            printf("\r");
            fflush(stdout);
        }
        to_send=15;// setting a not pressed char to not reenter the loop

        sleep(15);// time interval to refresh ThingSpeak
    }
    sleep(0.1);//time interval between 2 data send of MOSI
}
```



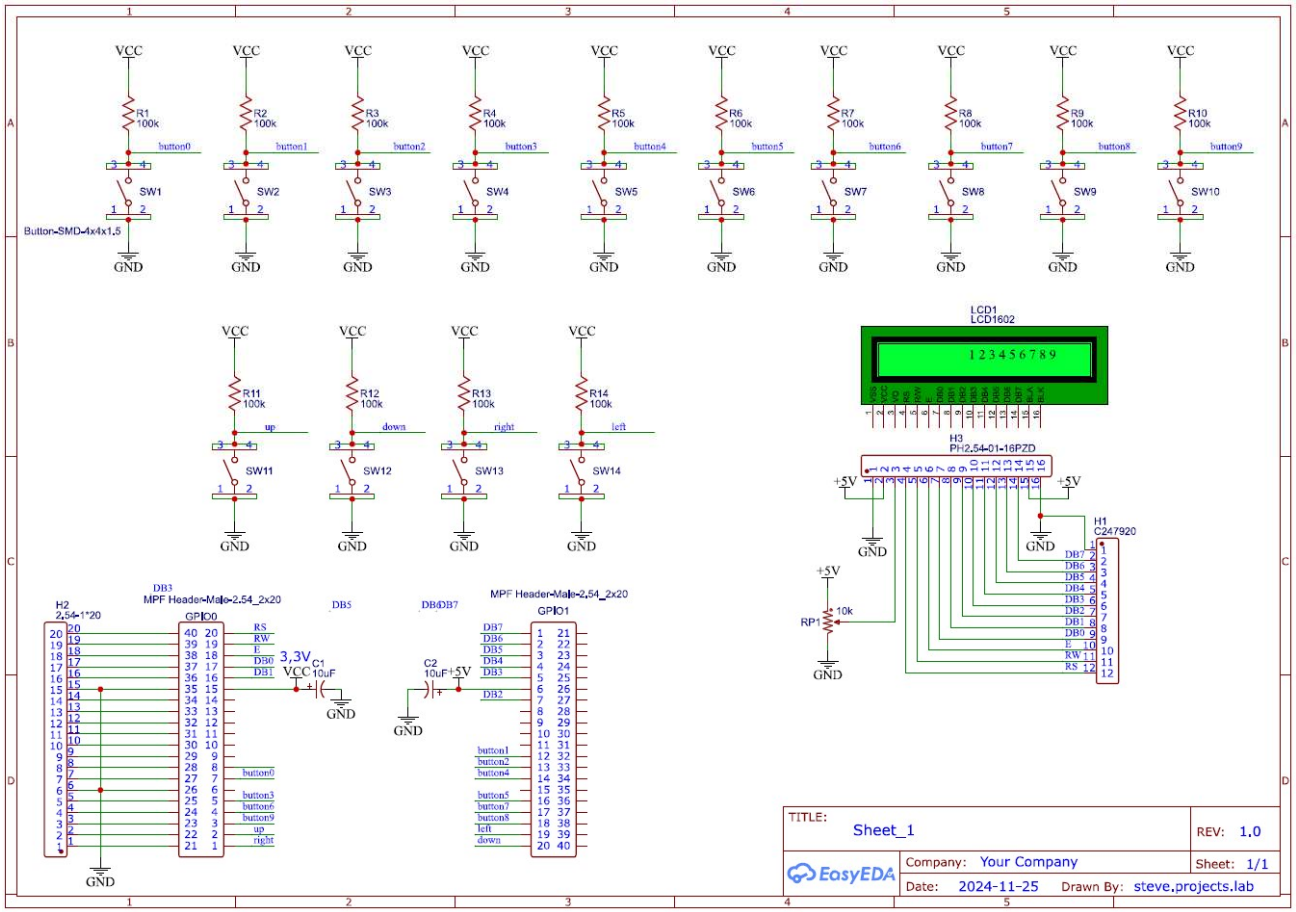
THINGSPEAK

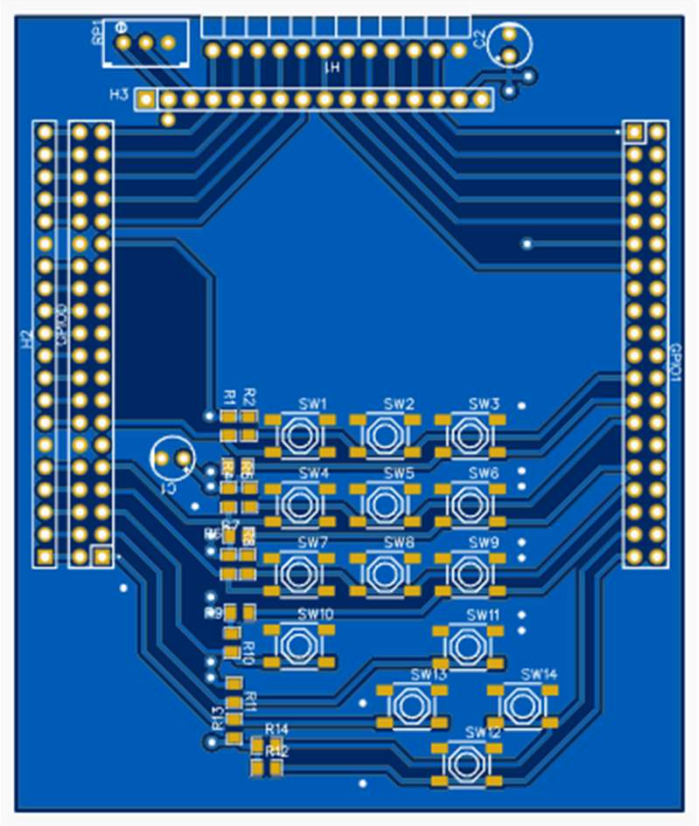



COM6 (USB Serial Port (COM6))
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
User sessions
192.168.1.108 (root)
192.168.137.252 (root)
COM6 (USB Serial Port (COM6
2. COM6 (USB Serial Port (COM6
SPI_Hw.c:SPIM_WriteReadTxRxData16(337): [SPIM1]:SPI Write and Read
SPI_Hw.c:SPIM_WriteReadTxRxData16(338): TX Value: 61999, Read Value: 61999
SPI_Hw.c:SPIM_WriteReadTxRxData16(338): TX Value: 61999, Read Value: 3


ThingSpeak™
Channels Apps Devices Support
Add Visualizations
Add Widgets
Export recent data
Channel Stats
Created: about 2 hours ago
Last entry: less than a minute ago
Entries: 140
Number of button pressed
3
a few seconds ago

SCHEMA ELETTRICO E LAYOUT





CONCLUSIONI E SVILUPPI FUTURI

- Compreso gli argomenti
  - ✓ Tematica IOT e FPGA
  - ✓ Funzionamento LCD
  - ✓ Funzionamento SPI
- Simulazione Modelsim
- Programmazione FPGA in VHDL
- Programmazione HPS in C
- Utilizzo piattaforma ThingSpeak
- Sviluppi futuri
  - ✓ Completamento codice VHDL
  - ✓ Arricchimento funzionalità main_fsm
  - ✓ Modulo custom ThingSpeak con MATLAB