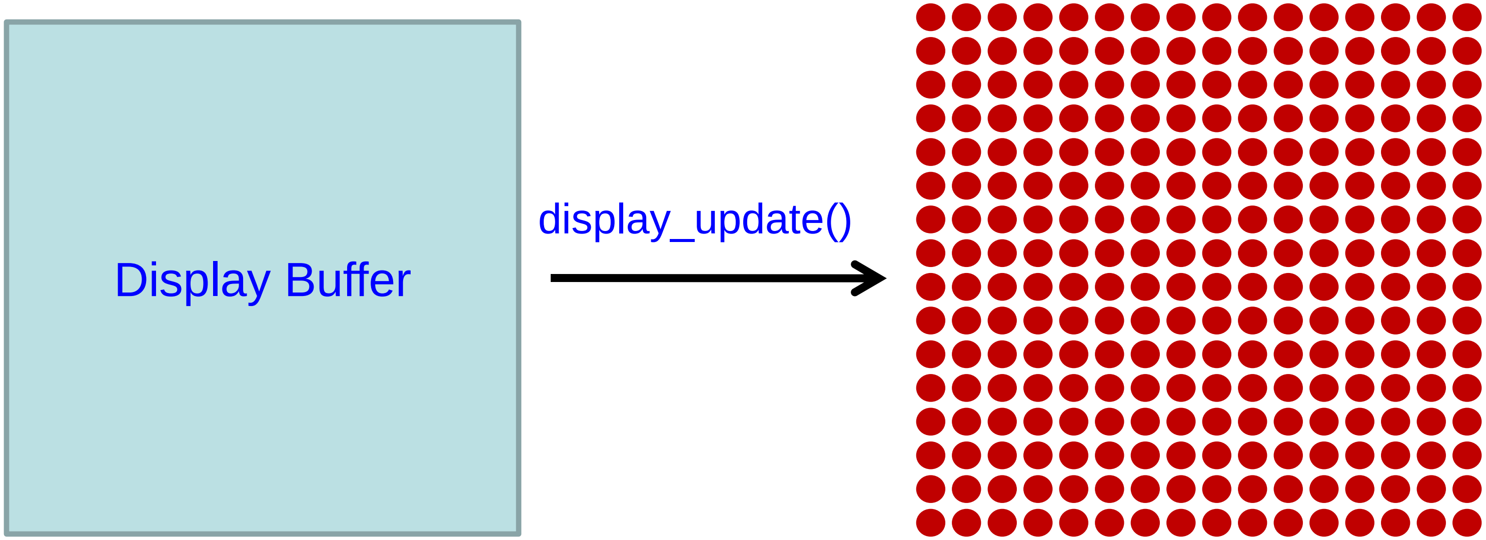
Overview

1. **Library description**
2. **Copying files and folders to and from PI**
3. **How enable SPI driver on Pi Board**
4. **What's in the bag? – Student Folder**
5. **Procedure for compiling files (Student)**
6. **What's in the bag? – Teacher Folder**
7. **Procedure for compiling files (Teacher , Instructor)**
8. **How deliver objet code to students**

Display Driver Library Services

The display driver has a display buffer in ram. In order to write to the display user must first write to this buffer. After writing to this buffer display_update() function **must** be called . The display buffer is organized as a 16 x 16 Matrix.



Display Buffer

display_update()

Display Driver Library Services

```
/*
 * void display_write(uint8_t x,uint8_t y, dlevel_t val)
 * This function writes to display buffer
 *
 * The display buffer is a square matrix of 16x16 pixels
 * A pixel can be turned on or off
 * The pixel coordinates are the first two parameters of the function
 * x: 0 to 15
 * y: 0 to 15
 *
 * The last parameter is the pixel Value: D_ON or D_OFF
 *
 * Important Writing to the display buffer does no actually be the same as writing
 * to display. To write to display we need to call display_update() function
 * which transfers the entire display buffer to the display
 *
 * Example display_write(2,3,D_ON) Turns on the pixel located at 3rd row and 2nd columm
 *
 *
 */
```

Display Driver Library Services

```
/* void Send_2_display(dcoord_t point, dlevel_t val)
 *
 * This function writes to display buffer
 *
 * The display buffer is a square matrix of 16x16 pixels
 * A pixel can be turned on or off
 * The first parameter of the function is an structure containing the pixel coordinates row (x) ,column (y)
 *  typedef struct {
 *                  uint8_t x;
 *                  uint8_t y;
 *                        } dcoord_t;
 *
 * x: 0 to 15
 * y: 0 to 15
 *
 * The last parameter is the pixel Value: D_ON or D_OFF
 *
 * Important Writing to the display buffer does no actually be the same as writing
 * to display. To write to display we need to call display_update() function
 * which transfers the entire display buffer to the display
 *
 * Example
 * ddcoord_t point1={3,2};
 * Send_2_display((dcoord_t){6,3},D_ON);              //  Turns on pixel 6,3
 * Send_2_display(set_coord(7,2),D_ON);               //  Turns on pixel 6,3
 * Send_2_display(point1,D_ON);        //  Turns on pixel at point1
 *
 */
```

Display Driver Library Services

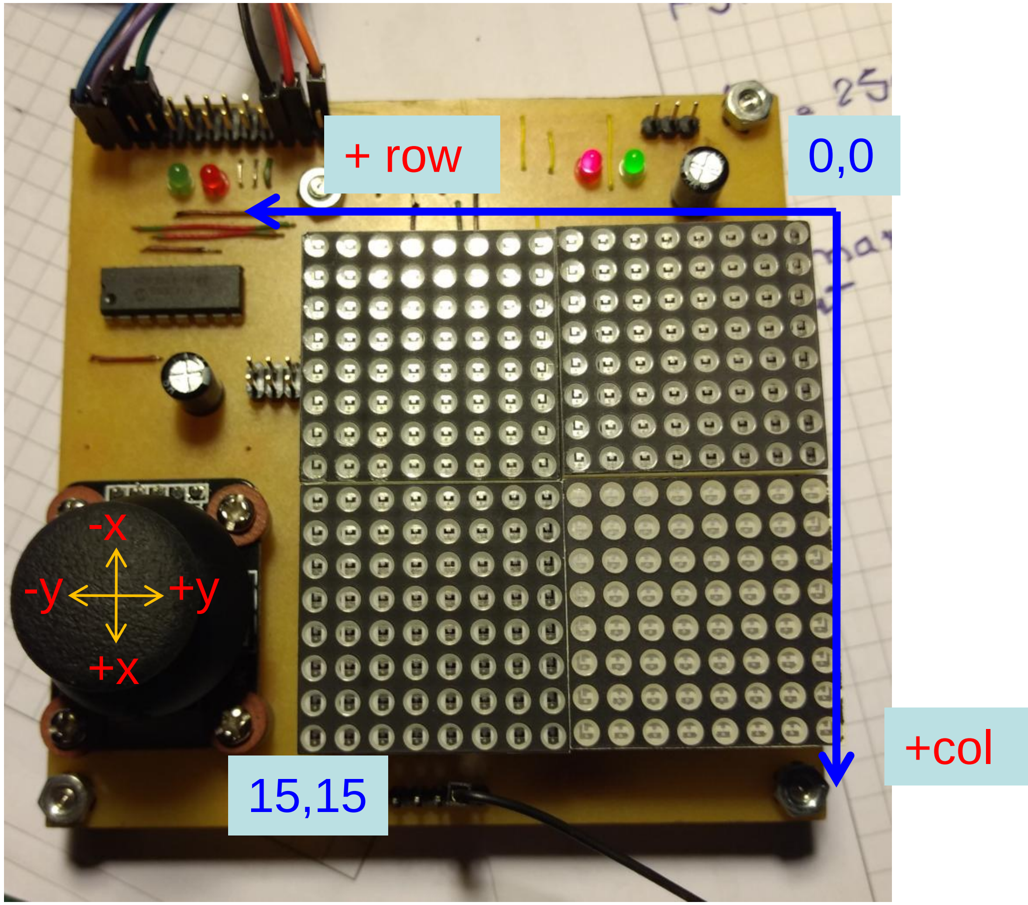
```
/*
 * display_init();
 * This function initialize MAX7219 internal registers
 * It must be called once at the very begining of the Aplication
 *
 * Example: display_init();
 *
 */

/* display_clear();
 * This function clears the entire display and the display buffer
 *
 *
 * Example: display_clear();
 *
 */


/* void display_update(void)
 * This function updates the the entire display
 * with the display buffer contents
 *
 *
 * Example: display_update();
 *
 */
```

Display Driver Library Services

```
/* void set_display_axis(uint8_t orientation)
 *
 * This function sets display orientation
 *
 * It recieves one parameter (orientation) wich defines the display axis orientation
 * Posible values NORMAL or ROTATE
 *
 * NORMAL Joystick on the bottom of display ( Right Handed)
 * ROTATE Joystick on the LEFT of display ( Left Handed)
 *
 *
 * Example: set_display_axis(ROTATE);
 *
 *
 */
```

See next page …

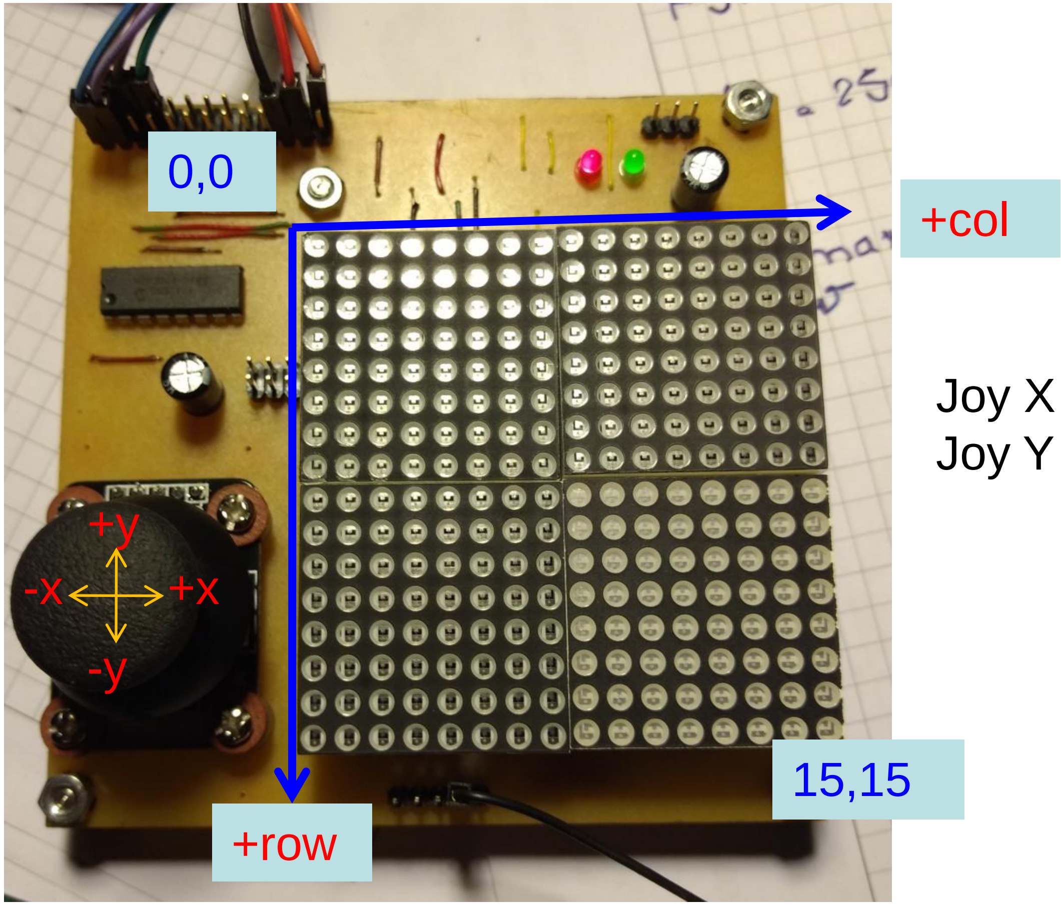
OPCION 1 (NORMAL)



User Position mano
Deecha

Joy X -128 to +127
Joy Y -128 to +127

OPCION 2 (ROTATE)

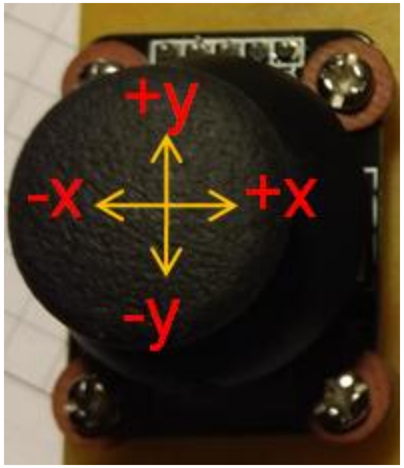


Joy X -128 to +127
Joy Y -128 to +127

User Position Mano Izquierda

Joystick Driver Library Services

The joystick driver report the angle of the x and y joystick direction as well the state of its switch. In order to read from the driver user **must** first call joystick_update() function. Then we can get the x,y angles and the switch state.



Joystick Driver Library Services

```
/* ---------------------------------------------------------------------
* jcoord_t  joystick_get_coord(void)
*
* Get Joystick Coordinates: returns a structure containing x and y angles
*
* typedef struct {
*                int16_t x;
*                int16_t y;
*
* } jcoord_t;
*
* Example :
* *
* jcoord_t joy_coordinates1;
* int16_t xangle,yangle;
*
* joystick_update();
* joy_coordinates1=joystick_get_coord();
*
* xangle=joy_coordinates1.x
* yangle=joy_coordinates1.y
*
*
----------------------------------------------*/
```

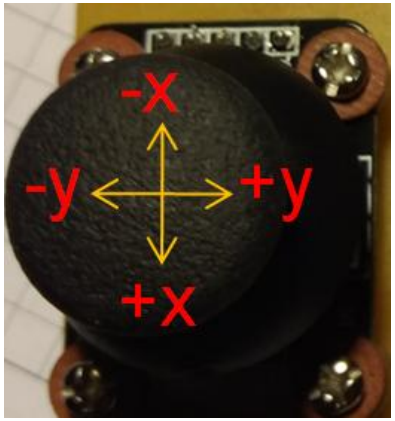
Joystick Driver Library Services

```
/*
* jswitch_t joystick_get_switch_value(void)
*
* Get Joystick Switch state
*
* Returns J_NOPRESS ,J_PRESS
*
* Example:
*
* jswitch_t joy_switch1;
*
*                  joystick_update();  // Read joystick Hardware
*                  joy_switch1=joystick_get_switch_value(); // And get switch value
* /


/*  void joy_init(void)
*   Initialize Joystick System:
*   Call this function only ONCE
*   at he very beginning of your Application
*
*   Example: joy_init();
*/
```

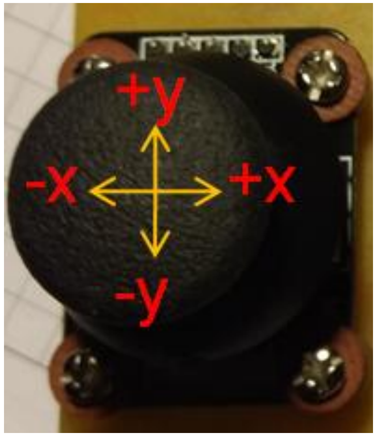
Joystick Driver Library Services

```
/* void set_joy_axis(uint8_t orientation)
 * This function sets Joystick orientation
 *
 * It receives one parameter (orientation) wich defines the Joystick axis orientation
 * Possible values NORMAL or ROTATE
 *
 * NORMAL Joystick on the bottom of display ( Right Handed)
 * ROTATE Joystick on the LEFT of display ( Left Handed)
 *
 *
 * Example: set_joy_axis(ROTATE);
 *
 *
 */
```

NORMAL

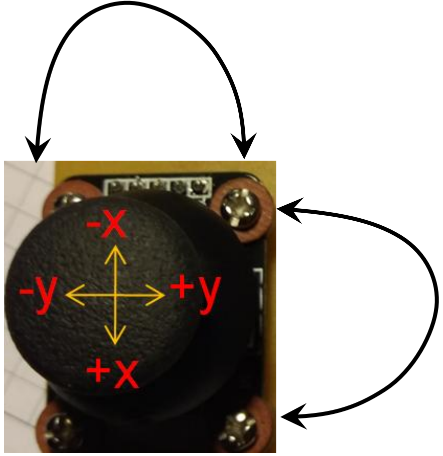


ROTATE



Joystick Driver Library Services

```
/* void set_joy_direction(int8_t dir_x,int8_t dir_y)
 * This function inverts Joystick axis
 *
 * It receives one parameter (orientation_x,orientation_y) wich defines
 * the joystick x and/or y axis orientation
 * Posible values J_INV_FALSE or J_INV_TRUE
 *
 * J_INV_FALSE Joystick axis default direction
 * J_INV_TRUE Joystick axis direction is inverted i.e.
 *
 * if orientation_x=J_INV_TRUE +x <-> -x
 * if orientation_y=J_INV_TRUE +y <-> -y
 *
 *
 *
 * Example: set_joy_direction(J_INV_TRUE,J_INV_TRUE); Inverts both axix directions
 *
 *
 */
```

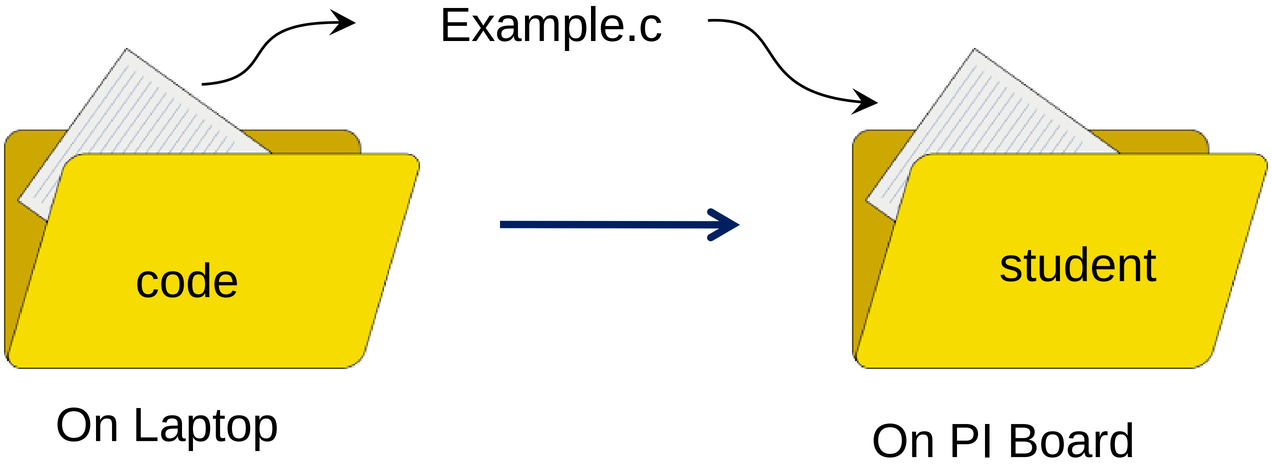


A1-How Copy a single file to the Pi Board folder

1- On your laptop go to folder (e.g. : code) where the file to be copied resides (e.g. : example.c)
2- Execute the following command on terminal

```
scp example.c  pi@10.42.0.70:/home/pi/student
```

The file in folder code (example.c) will be copied to destination folder student on PI board. The file will be created if it doesn't exist. If the file exist it will be overwritten.



Example.c

code

student

On Laptop

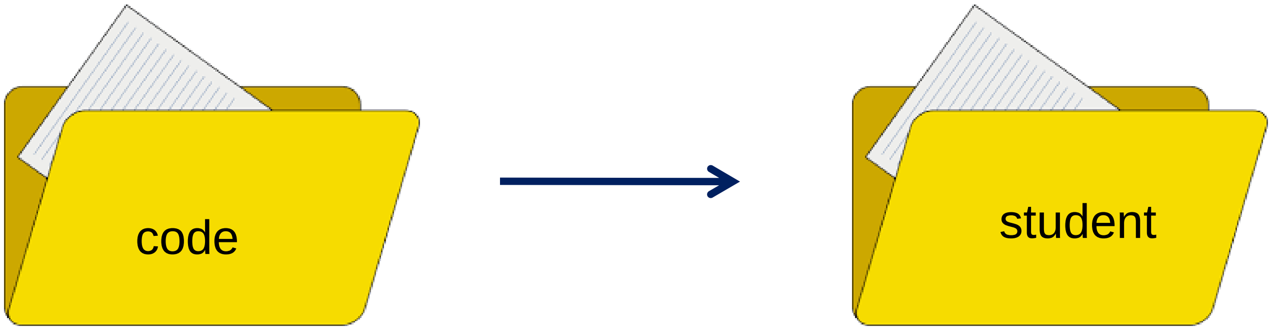
On PI Board

A2-How Copy a Folder to the Pi Board

1- On your laptop go to folder to be copied (e.g. : code)
2- Execute the following command on terminal

```
scp -r ../code/.     pi@10.42.0.70:/home/pi/student
```

All files in folder code will be copied to PI board
The destination folder student will be created if it doesn’t exist.
If exist it will be overwritten.



code

student

On Laptop

On PI Board

A3-How Copy a Folder from the Pi Board

1- On your laptop go to a folder where pi folder will be copied (e.g. : backup)
2- Execute the following command on terminal

```
scp -r pi@10.42.0.70:/home/pi/student/ .
```

The folder student on PI board will be copied to the backup folder.The destination folder backup will be created if it doesn’t exist. If exist it will be overwritten.

backup
student

On Laptop

On PI Board

A4-How enable SPI driver on Pi Board

Overview

To enable SPI driver we need to modify 3 system files and then reboot the PI board.

The system files are:

1- /etc/modules
2- /etc/modprobe.d/raspi-blacklist.conf
3- /boot/config.txt

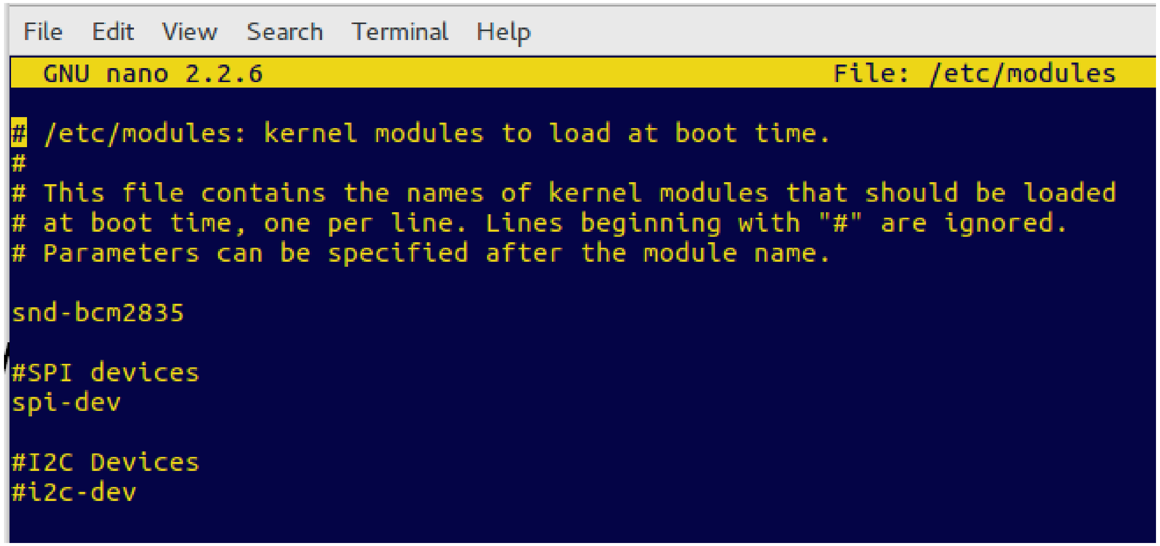
To edit these files we MUST do it as super user (sudo)

On next pages the procedure will de explained in detail:

How enable SPI driver on Pi Board

A - Configure /etc/modules

1-Edit the file /etc/modules → sudo nano /etc/modules



```
  GNU nano 2.2.6                                            File: /etc/modules

# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835

#SPI devices
spi-dev

#I2C Devices
#i2c-dev
```

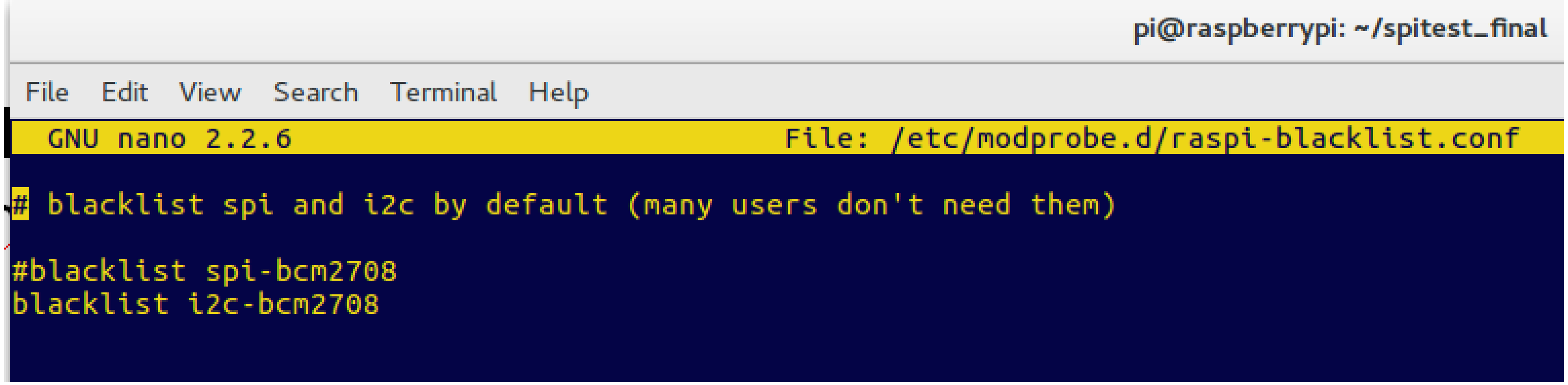
2-Make sure that the line spi-dev is present
and uncommented (no # in front of it).

3-Save the file: ctrl-o then ctrl-x

How enable SPI driver on Pi Board

B - Configure /etc/modprobe.d/raspi-blacklist.conf

1-Edit the file /etc/modprobe.d/raspi-blacklist.conf
→ sudo nano /etc/modprobe.d/raspi-blacklist.conf


pi@raspberrypi: ~/spitest_final
File Edit View Search Terminal Help

```
  GNU nano 2.2.6                          File: /etc/modprobe.d/raspi-blacklist.conf

# blacklist spi and i2c by default (many users don't need them)

#blacklist spi-bcm2708
blacklist i2c-bcm2708
```

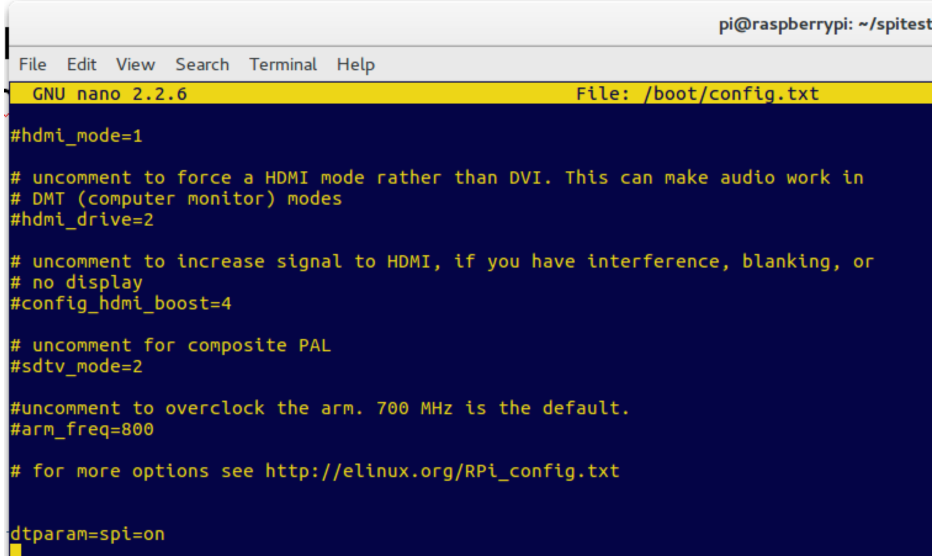
2-Make sure that the line blacklist spi-bcm2708 is present and commented (# in front of it).

3-Save the file: ctrl-o then ctrl-x

How enable SPI driver on Pi Board

C - Configure /boot/config.txt

1-Edit the file /boot/config.txt
→ sudo nano /boot/config.txt



pi@raspberrypi: ~/spitest
File Edit View Search Terminal Help

```
  GNU nano 2.2.6                                   File: /boot/config.txt

#hdmi_mode=1

# uncomment to force a HDMI mode rather than DVI. This can make audio work in
# DMT (computer monitor) modes
#hdmi_drive=2

# uncomment to increase signal to HDMI, if you have interference, blanking, or
# no display
#config_hdmi_boost=4

# uncomment for composite PAL
#sdtv_mode=2

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# for more options see http://elinux.org/RPi_config.txt


dtparam=spi=on
```

2- This file can be quite long so, go to the end of the file and write the following: **dtparam=spi=on**.

3-Save the file: ctrl-o then ctrl-x

How enable SPI driver on Pi Board

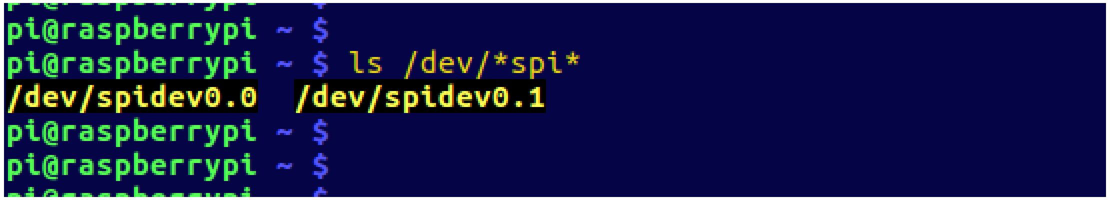
D – Reboot

1- sudo reboot (this takes some time !! Be patient)

2- login again

3- Check if SPI driver is loaded

Enter the following command: ls /dev/*spi* and check that spidev0.0 and spidev0.1 are present.



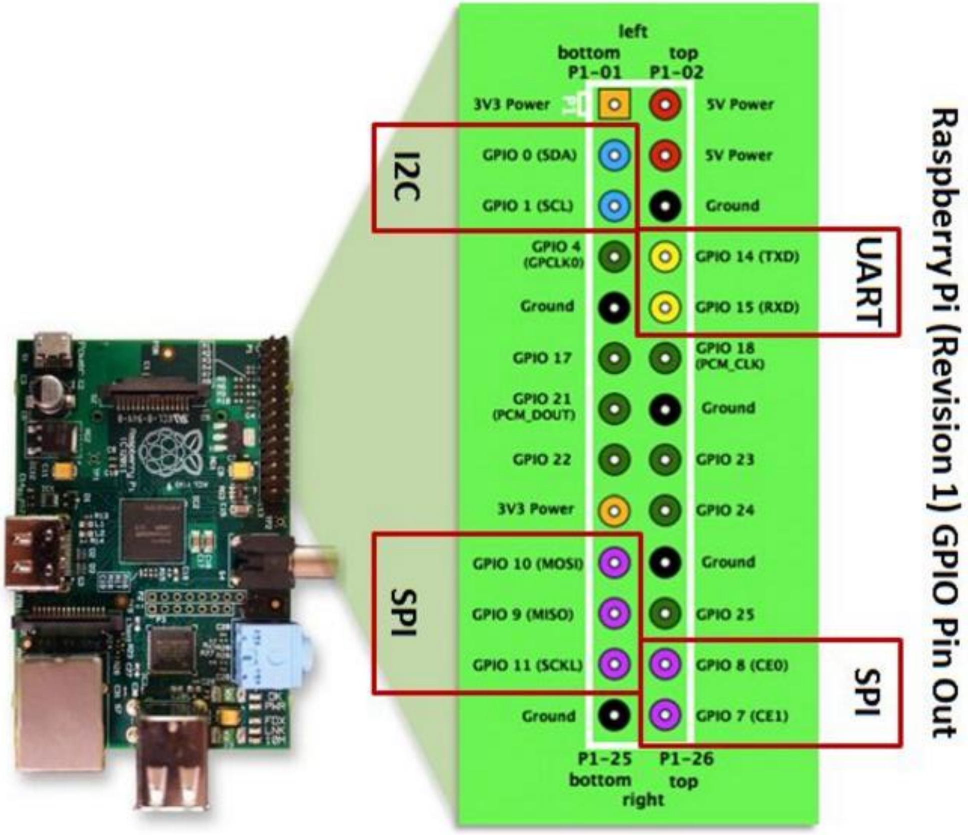
```
pi@raspberrypi ~ $
pi@raspberrypi ~ $ ls /dev/*spi*
/dev/spidev0.0  /dev/spidev0.1
pi@raspberrypi ~ $
pi@raspberrypi ~ $
```

4- You are done !!!!!

How enable SPI driver on Pi Board

E – Test SPI (Optional)

1- Connect a scope probe to the SPI Clock (GPIO 11)



How enable SPI driver on Pi Board

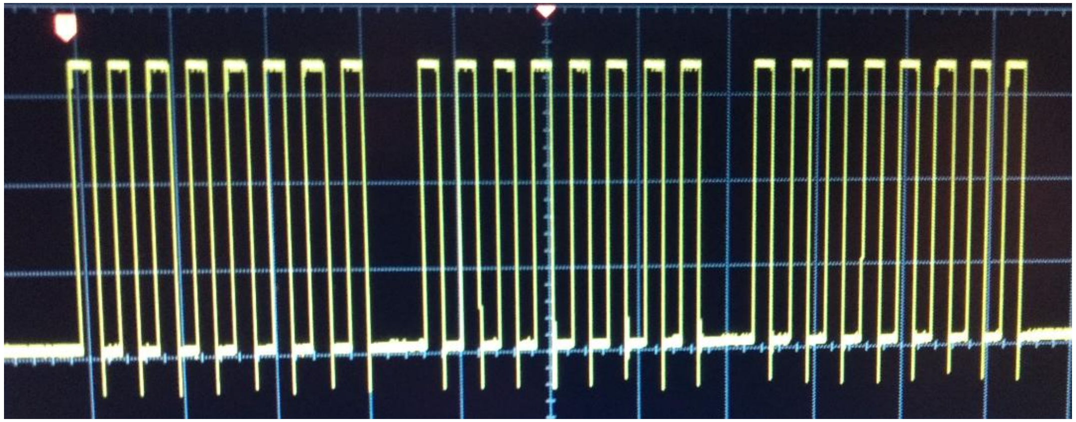
E – Test SPI (Optional)

2- Log as Superuser sudo su

3- Run the following command:

```
sudo echo -ne "\x01\x02\x03" > /dev/spidev0.0
```

4- If all is ok you will see somethink like this:



A5.1- Student folder contents

disdrv.o	Display Library (PI object file)
disdrv.h	Display Library header
joydrv.o	Joystick Library (PI object file)
joydrv.h	Joystick Library header
termlib.o	Terminal utilities Library (PI object file)
termlib.h	Terminal utilities header
Example.c	Sample Code

A5.2-How Compile the example file on the Pi Board (Student)

1-cd to student folder then copy the folder Student to the PI board:

```
scp -r ../Student/.     pi@10.42.0.70:/home/pi/Student
```

2- Log in, go to folder and compile Example.c

```
gcc Example.c joydrv.o disdrv.o  termlib.o -o Example
```

Run example.c

```
sudo ./Example
```

Once the folder was transferred to PI board we just need to send Example.c if we make changes to it. (object files already are on the folder).

A6.1-Teacher folder contents

disdrv.c	Display Library source code
disdrv.h	Display Library header
joydrv.c	Joystick Library source code
joydrv.h	Joystick Library header
termlib.c	Terminal utilities source code
termlib.h	Terminal utilities header
MAX7219.h	Display controller definitions
MCP3008.h	ADC definitions
Example.c	Sample Code

A6.2-How Compile the example file on the Pi Board (Teacher)

1-Copy the folder Teacher to the PI board (see Apendix A2)

2- Log in, go to folder and Compile drivers

```
gcc -c disdrv.c
gcc -c joydrv.c
gcc -c termlib.c
```

Compile Example.c

```
gcc Example.c joydrv.o disdrv.o  termlib.o -o Example
```

Run example.c

```
sudo ./Example
```

Once libraries were compiled on the PI board we just need to send Example.c any time we make changes to it.

A6.3-How deliver objet code (compiled libraries) to the Pi Board (Teacher)

1 - On laptop create a folder called Piobj and cd to it.

2 - Copy only object files from PI board to this folder

```
scp  pi@10.42.0.70:/home/pi/Teacher/*.o  .
```

3 –Additionally we can get the compiled Example.c file

```
scp  pi@10.42.0.70:/home/pi/Teacher/Example .
```

So we end up with the following files:

```
/PIobj$  ls
/PIobj$  disdrv.o  Example  joydrv.o  termlib.o
```

Now we can distribute them to students.