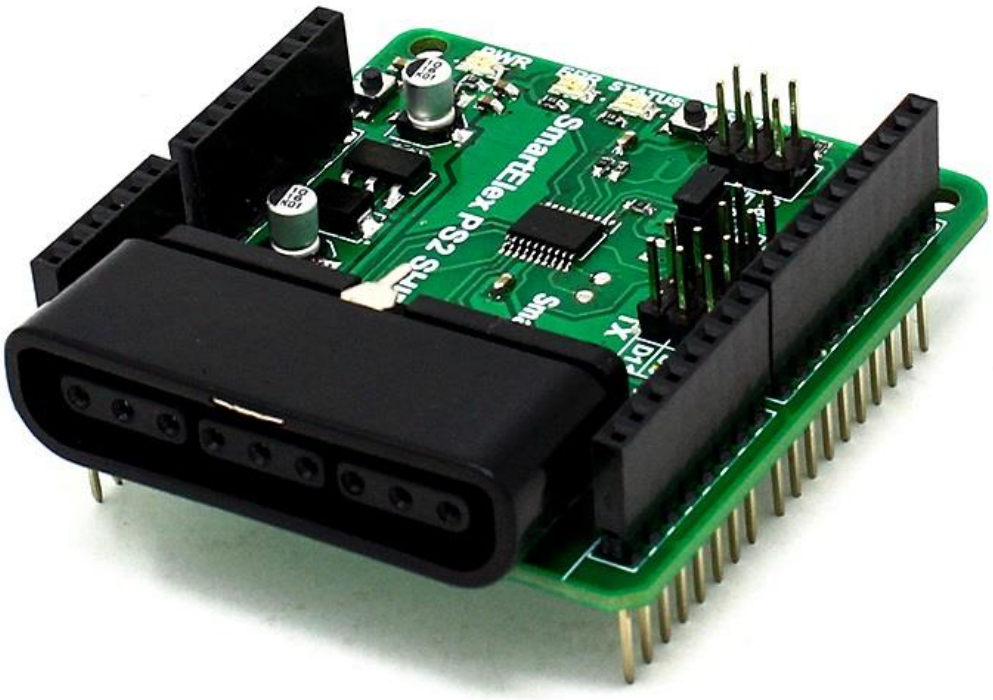
SmartElex PS2 Shield.



Index

1) Introduction and overview.

2) Board and product layout.

3) Button and joystick status.

4) Supported Ps2 controllers.

5) Library Functions.

1. INTRODUCTION AND OVERVIEW

SmartElex PS2 Shield (SHIELD-PS2) is an Arduino compatible shield which is compatible with Arduino UNO, Arduino Mega and Arduino Leonardo. SmartElex PS2 Shield offers a compact yet reliable PS2 Controller Converter for user. SmartElex PS2 Shield is powered from Arduino main board.

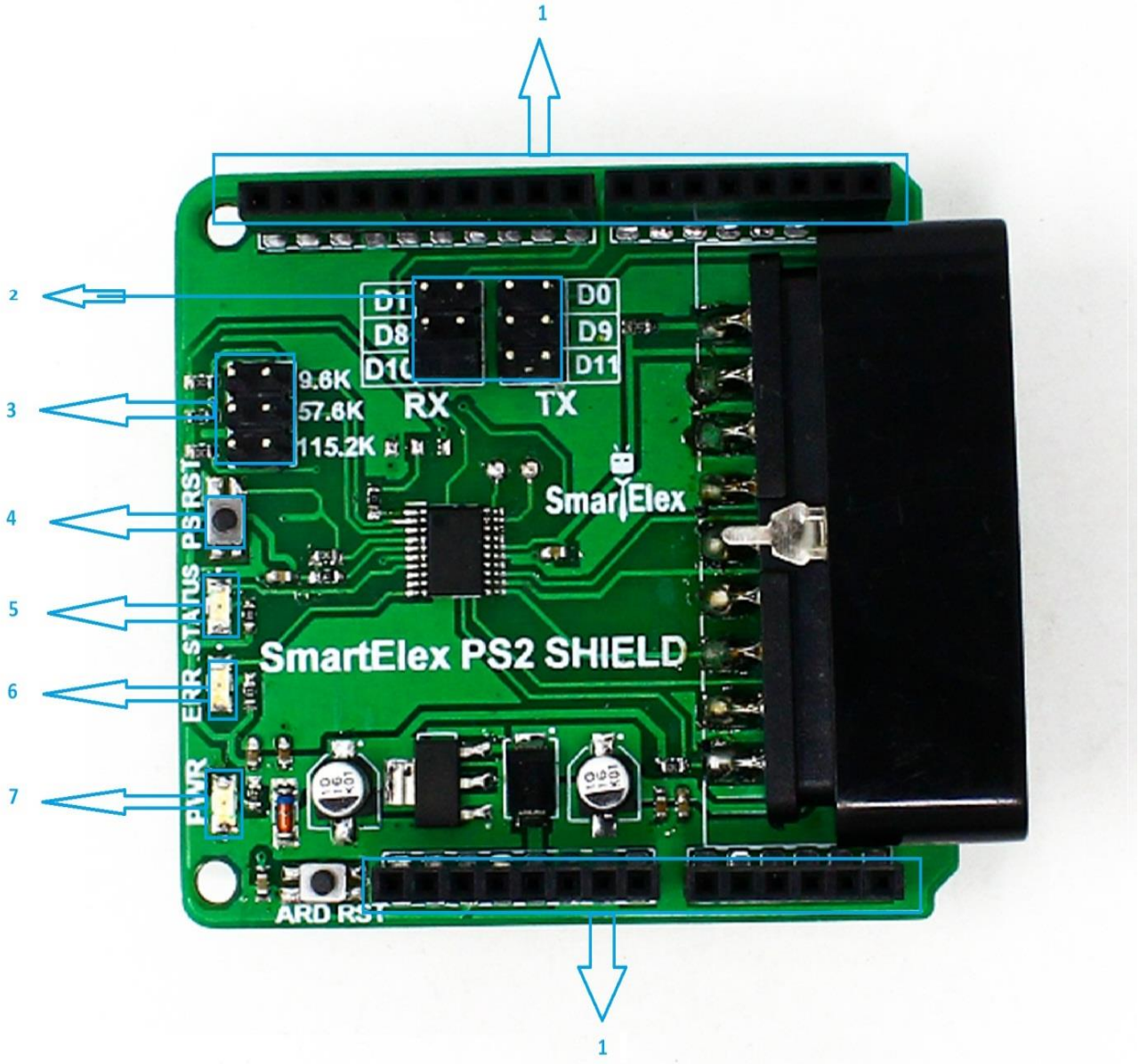
With SmartElex PS2 Shield Reading Joy-stick and button's state of PS2 controller will be easy. It offers a standard connector for SONY PS2 controller to plug-in, either wired or wireless.

SmartElex Shield-PS2 has stackable side headers which allows for more Arduino shields to be stacked on top of it. Besides, user has option to use either hardware **or software UART of** Arduino's main board to communicate with PS2 controller. SmartElex PS2 Shield can be reset with "PS2 RST" button provided on SmartElex PS2 Shield.

Features

1. 5 V powered, low current consumption. Controller / pull up operating voltage 3.3V .
2. Simple to use UART protocol
3. Vibrator motor on PS2 is controllable.
4. Wired and Wireless PS2 controller is supported.
5. PS2 Controller will automatically operate in analog mode. (Default)
6. A status LED
7. Jumper selector to select different UART Baud Rate (74880 (Default), 9600, 57600, 115200). Default: - when no jumper is attached.
8. Jumper selectors to select different digital pin as UART - TX and RX pin.
9. Shield can put Ps2 controller in Digital mode or Analog mode dynamically.
10. Hot plug in.

2. PRODUCT LAYOUT



1) Stackable Digital I/O Headers

a. Digital I/O pins stacked to the Arduino main board.

2) RX Pin Selector / TX Pin Selector

a. User may select D0, D9 or D11 as the RX pin of arduino with the mini jumper. If Arduino UNO or Mega is used, recommended to move the TX pin selector to D9, or D11 as D0 is used for bootloader (loading program).

b. User may select D1, D8 or D10 as the TX pin of arduino with the mini jumper. If Arduino UNO or Mega is used, recommended to move the RX pin selector to D8 or D10 as D1 is used for bootloader (loading program)

3) Baudrate select

a. To select the preferable UART baud rate For SmartElex PS2 Shield, to obtain latest baud rate from selector, Shield-PS2 need to be reset to update the board's baud rate.

4) On board Smart PS2 Shield reset button

a. PS2 RST button will only reset the SHIELD-PS2 but not the Arduino Main board.

5) Status indicator LED

a. **ON state :** Ps2 controller is in analog mode.
b. **OFF state :** Ps2 controller is in digital mode.
c. **Flashing & OFF :** Ps2 Shield has detect some error. (when controller is not plugged into SmartElex PS2 sheild or due to some communication error or when the controller is faulty.)
d. **Flashing :** Digital button on PS2 controller is pressed.

6) Error Led

a. **Flashing & OFF :** Ps2 Shield has detect some error. (when controller is not plugged into SmartElex PS2 sheild or due to some communication error or when the controller is faulty.)

7) Power indicator LED

a. Indicator to shows that power is supplied to SmartElex PS2 Shield

3. Button and Joystick status

Column1	Column2	Column3
Button	Ideal	Pressed
SELECT	1	0
L3	1	0
R3	1	0
START	1	0
UP	1	0
RIGHT	1	0
DOWN	1	0
LEFT	1	0
L2	1	0
R2	1	0
L1	1	0
R1	1	0
TRIANGLE	1	0
CIRCLE	1	0
CROSS	1	0
SQUARE	1	0
RIGHT_X_AXIS	127	0 - 255
RIGHT_Y_AXIS	128	0 - 255
LEFT_X_AXIS	127	0 - 255
LEFT_Y_AXIS	128	0 - 255
L3_Pressure	0	0 - 255
R3_Pressure	0	0 - 255
UP_Pressure	0	0 - 255
RIGHT_Pressure	0	0 - 255
DOWN_Pressure	0	0 - 255
LEFT_Pressure	0	0 - 255
L2_Pressure	0	0 - 255
R2_Pressure	0	0 - 255
L1_Pressure	0	0 - 255
R1_Pressure	0	0 - 255
TRIANGLE_Pressure	0	0 - 255
CIRCLE_Pressure	0	0 - 255
CROSS_Pressure	0	0 - 255
SQUARE_Pressure	0	0 - 255

Refer the sample code provided with product.

4. Supported PS2 controllers

There are many types of PS2 controller in the market with different sensitivity. User are free to choose any type of them. However, it is advised to use the Sony PS2 controller. we do not guarantee the compatibility for all PS2 controllers from other sources.

5. Library functions

1) **Include the header file.**

#include <SmartElexPs2Shield.h>

2) **Create the object to access member variables and member functions**.

SmartElexPs2Shield PS2 (0 , 1) ; // Rx of Arduino // Tx of Arduino

3) **Accessing member functions . Consider the example given below.**

PS2.begin(9600); // here " begin" is member function with "9600" as argument passed to it as baudrate.

4) **Other Member functions and Variables.**

PS2.VibrateMotors(A,B); // A = LeftMotorValue (0 or 255) , B = RightMotorValue (0 to 255)

Recommended minimum **RightMotorValue** > 70 to make right side motor rotate.

Left side motor can either be in ON state or in OFF state.

LeftMotorValue = 0 = OFF state.

LeftMotorValue = 255 = ON state.

PS2.SetController (Aargument);

Here "argument" should be 1,2,3 or 4.

Argument = 1 → This will set controller in digital mode. In digital mode controller can switch modes with button provided on controller itself.

Argument = 2 → This will set controller in fixed digital mode. In fixed digital mode controller **can not** switch its mode with button provided on controller itself**.**

Argument = 3 → This will set controller in analog mode. In analog mode controller can switch modes with button provided on controller itself.

Argument = 4 → This will set controller in fixed analog mode. In fixed analog mode controller **can not** switch its mode with button provided on controller itself.

Argument = other → This will set controller in analog mode. In analog mode controller **can** switch its mode with button provided on controller itself.

NOTE: Once the controller is dragged into fixed mode (fixed digital or fixed analog),and than setting controller back to normal mode i.e to digital or analog mode, disables the controller ability to switch modes with button provided on it.

To regain the ability to switch modes with button provided on controller one must unplugged the controller and plugged it back.

PS2.ReadControllerButtons( ); // This function will assign controller status to predefined variable declared in SmartElexPS2 library.

Example : PS2.SELECT , PS2.TRIANGLE , PS2. CIRCLE , , PS2. R1_Pressure, etc (refer library for more predefined variables.)