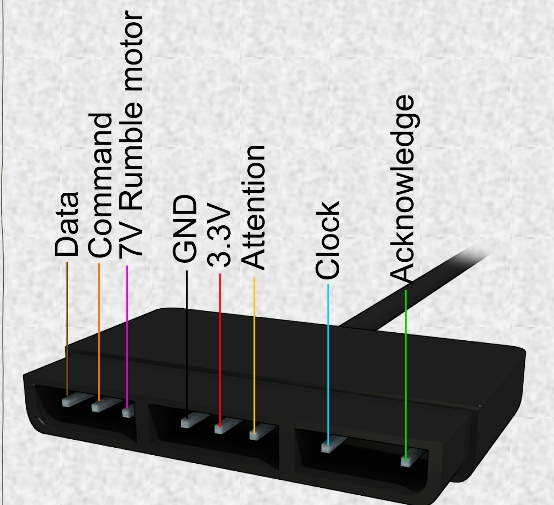
**Technocrats Task Report for Week 3 and Week 4**

**PS2 controller and Arduino:**

The PS2 controller has 8 pins. They are Data, Command, Rumble motor pin, Gnd, 3.3V, Attention, Clock and Acknowledge pin.

If the rumble feedback is required the pin has to be connected to 7-9V.

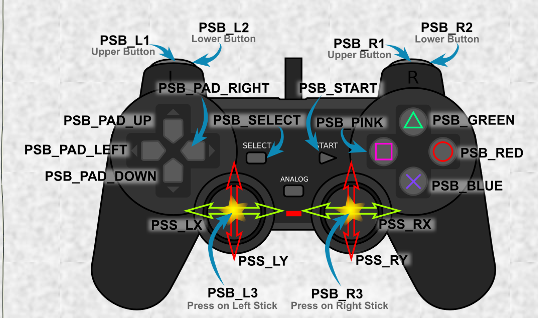


The library that is used - "PS2X\_lib.h"

The command to initialise the setup is [gamepad(clock, command, attention, data, pressure sensitivity enabled, rumble enabled)]

For E.g.:"ps2x.config\_gamepad(5,4,3,2, false, false);" provided that the pins are connected to digital pins 5(clock in PS2 controller), 4(command in PS2 controller), 3(Attention in PS2 controller), 2(Data in PS2 controller) respectively.

If the pressure sensitivity and rumble feature has to be enabled the respective field should be switched to TRUE state.



The setup portion is done now the loop portion is considered.

The above figure shows the control of the PS2 controller. If the pressure sensitivity has to be considered the string 'PSB' has to be changed to 'PSAB'.

For every single button there are three different actions in which it can be operated.

It can be pressed, pressed and held and pressed with varying pressure. Every action can be programmed.They are Button, ButtonPressed and Analog respectively.

some of the examples:

ps2x.ButtonPressed(PSB\_RED) is for a simple press of the red circle button.

ps2x.Button(PSB\_PAD\_DOWN) is for the down button on the pad being pressed and held.

ps2x.Analog(PSAB\_CROSS) is the command for measuring the pressure applied to the “X” button, if pressure sensitivity has been enabled. As you can see the “PSAB” form of the name has been used for the button.

ps2x.Analog(PSS\_RY) is the command to obtaining readings off the right analogue stick in the vertical direction.

To define a function for a button:

if(ps2x.ButtonPressed(PSB\_GREEN)) //Triangle pressed

{

LEDHdlts = !LEDHdlts; //Toggle the LED light flag

}