

see also [resistive vs capacitive touchscreen](#), [make your own pad](#)

## Currently Figuring Out:

- ✓ ~~reading the analog input to XADC on FPGA and show it with LED somehow~~ ✓  
~~2024-12-18~~
  - ✓ ~~XADC demo code runs fine, but ours doesn't~~
- ✓ ~~用電錶讀觸控版的電壓~~ ✓ ~~2024-12-06~~
  - ✓ ~~搞懂原理了，電阻觸控版內會形成一個分壓電路~~
- ✓ ~~分別以 FPGA 供電給 X 從 Y 讀 X 的讀數；再供電給 Y 從 X 讀 Y 的讀數~~ ✓ ~~2024-12-11~~
  - ~~電阻 Y- 跟電阻 X+ 並聯，所以從 Y- 端讀，另一軸同理~~
  - ~~use PWM to lower the 3.3V output to 1V (remember to account for the resistance of the panel), so that we can read it back to FPGA safely~~
    - **this doesn't work.** though the average voltage is 1v, the maximum voltage pulse is still 3v3, so it cannot be parsed by the XADC ports
- ✓ ~~PWM 行不通，改串電阻降壓~~ ✓ ~~2024-12-13~~
  - ✓ ~~降壓電路，抓個電阻比  $R_1 : R_2$  來得到適當的電壓 ( $< 1V$ )~~
- ✓ ~~read with FPGA XADC~~ ✓ ~~2024-12-20~~
  - ✓ ~~demo code works but our code doesn't, perhaps is the configuring issue?~~
  - ✓ ~~移植整個專案，改用 demo code 當 I/O 模板開始改~~
  - ✓ ~~多讀幾次取平均 (can be done by the IP module)~~
- ✓ ~~電壓讀數轉座標 or 直接當座標用？~~
  - ✓ ~~這樣會讓 PID 更難設計嗎 don't care~~
- ✓ ~~1D-PID~~
  - ✓ ~~PD controller~~
- ✓ ~~PID 調參數~~
  - ✓ ~~說不定 PD 控制就夠了~~
- ✓ ~~輪流讀~~ ✓ ~~2024-12-24~~
  - ~~[verilog inout?](#)~~
  - ✓ ~~[Verilog 中 inout 的用法 \(二\) verilog inout CSDN 博客](#)~~
  - ✓ ~~[Verilog inout 端口使用詳解 verilog 怎么写 inout 口 CSDN 博客](#)~~
  - ✓ ~~use switch to change state~~ ✓ ~~2024-12-24~~
  - ✓ ~~done by a bunch of [Transistor](#)~~
  - ☐ ~~use FSM to automatically change state after every read~~
    - ☐ ~~maybe swapped too fast~~
  - ~~5 wire panel is much more intuitive~~
  - ✓ ~~不然只能一次 I/O 一維，很遜~~
- ☐ jittering nightmare

```

module dual_port ( .... inout_pin, .... );
inout inout_pin;
wire inout_pin;
wire input_of_inout;
wire output_of_inout;
wire out_en;
assign input_of_inout = inout_pin;
assign inout_pin = out_en ? output_of_inout : 1'bz;
//this will be implemented by a tristate buffer
endmodule

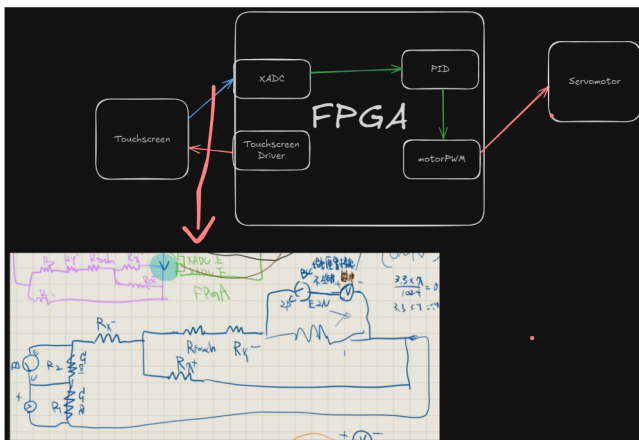
```

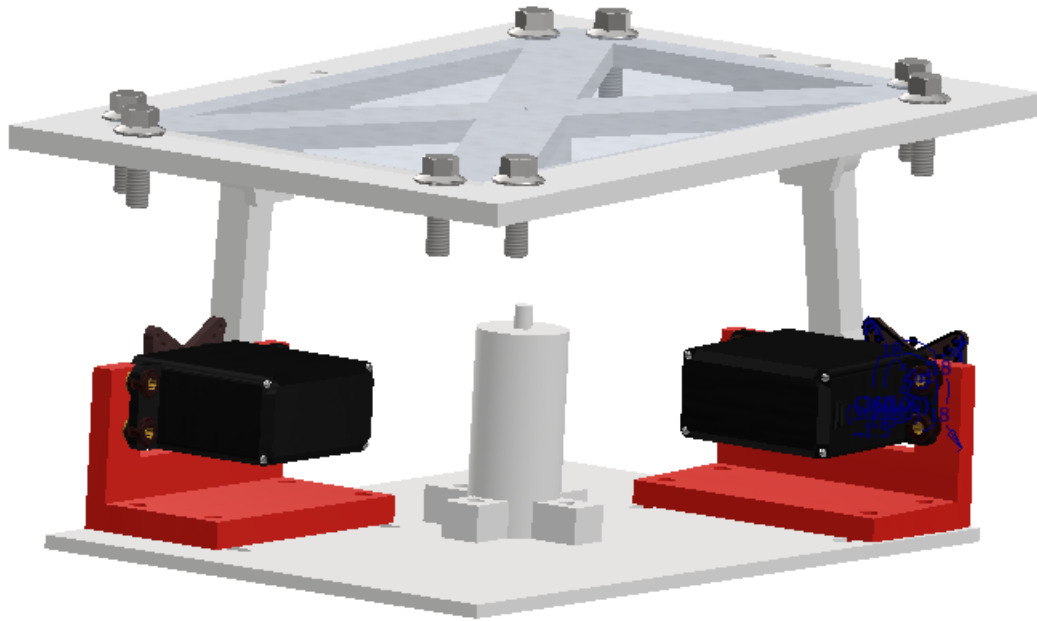
Basys3 has XADC ports

- input can be up to 1V, with 10-bit resolution
- [Basys 3 XADC Demo - Digilent Reference](#)
- [Basys-3-HW/src/hdl/XADCdemo.v at 332cd56de58e3897ee6ea089a1651c7c2431b150 · Digilent/Basys-3-HW · GitHub](#)

## Overview

[Balancer - plate balancing a ball with PID controller, resistive panel and servos, arduino - YouTube](#)





## Touchscreen (Resistive Panel)

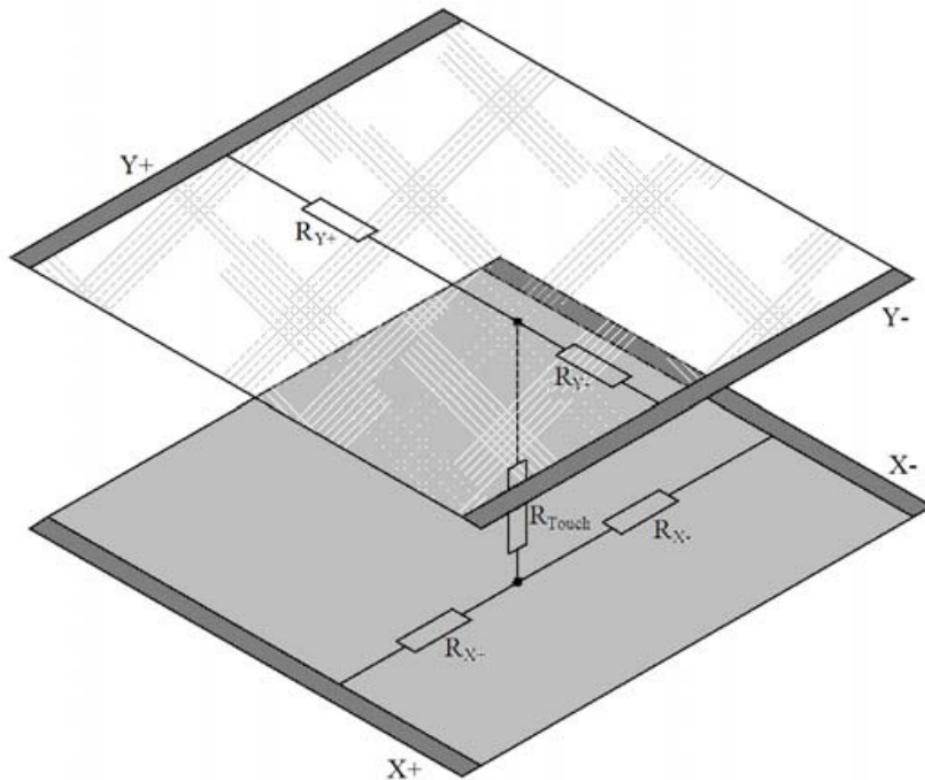
how people use it with Arduino

- [How to use a 4 wire resistive touchscreen with an Arduino - YouTube](#)
- [Arduino 5-wire Resistive Touchscreen Panel - YouTube](#)
- **manual**, how to design a controller for a touchscreen: [ww1.microchip.com/downloads/en/AppNotes/doc8091.pdf](http://ww1.microchip.com/downloads/en/AppNotes/doc8091.pdf)

電阻  $Y^-$  跟電阻  $X^+$  並聯，所以從  $Y^-$  端讀，另一軸同理

**Table 2-1.** 4-wire touch screens scanning

	$X^{+}_{Excite}$	$X^{-}_{Excite}$	$Y^{+}_{Excite}$	$Y^{-}_{Excite}$
<b>Standby</b>	Gnd	Hi-Z	Hi-Z	Pull up / Int
<b>X-Coordinate</b>	Gnd	Vcc	Hi-Z	Hi-Z / ADC
<b>Y-Coordinate</b>	Hi-Z	Hi-Z / ADC	Gnd	Vcc

**Figure 2-3.** “Schematic” of a 4-wire touch screen when pressure is applied.

the voltage gradient must be applied once in vertical and then in horizontal direction: first a supply voltage must be applied to one layer and a measurement of the voltage across the other layer is performed, next the supply is instead connected to the other layer and the opposite layer voltage is measured.

the output of the touchscreen is an analog signal:  $[0, input] v$ , something like that

## Servomotor

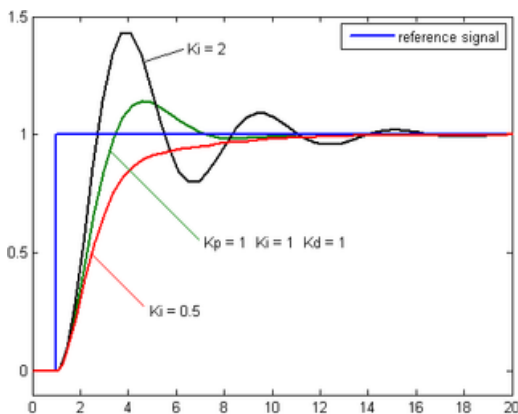
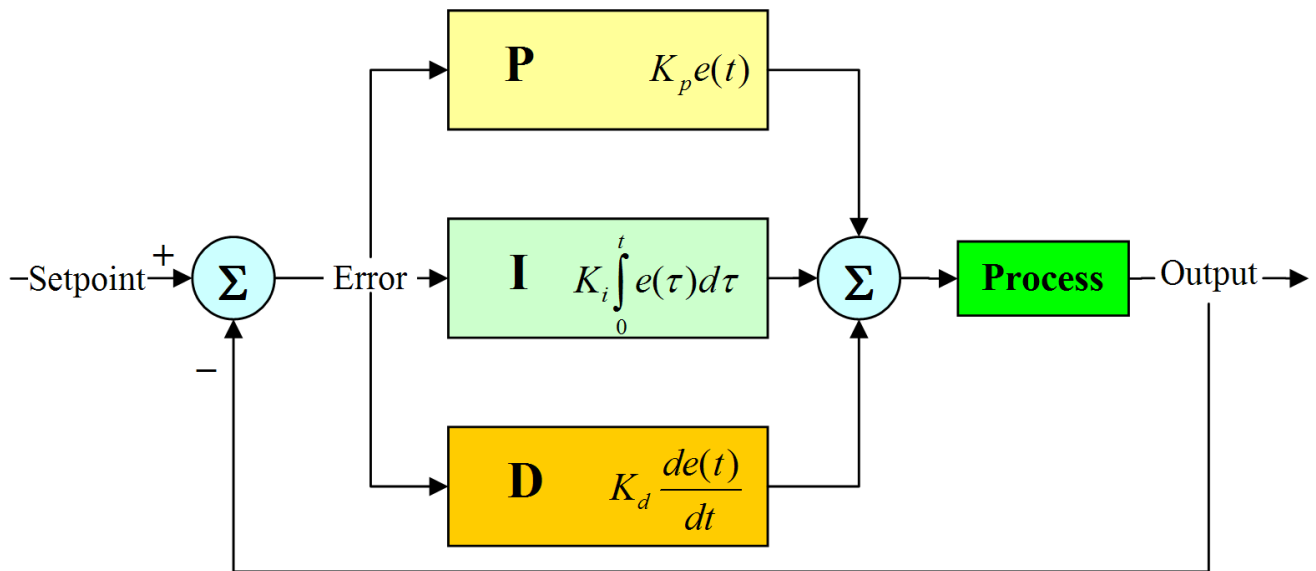


- ✓ ~~3-wire: ground, power, PWM?~~
  - ~~pulse 50Hz, or 20 ms~~
  - ~~set duty as 2.5/20 ~ 12.5/20 to get output of 0 ~ 180 degree rotation~~
  - ~~2.5 ms → 0°, 12.5 ms → 180°~~
- ✓ ~~broke one, buy another~~ ✓ 2024-12-06
- ✓ 萬向接頭

## PID Controller

basic idea

- P: reflects current error
- I: handle small error
- D: predict future error



- implementation
  - quite simple actually
- ☐ fine-tuning
  - ☒ manual
  - auto
  -

## Report & Demo

- 1 late-day left to spare
- [如何準備演講、報告、發表](#)

## Reference

- [Control Engineers / Balancer / Balancer · GitLab](#)
  - they use Arduino as controller and interface the hardware and software with Robot Operating System (ROS)
- [ATMEL, AVR341: Four and five-wire Touch Screen Controller](#)

- XADC Wizard v3.3 LogiCORE IP Product Guide (PG091)
- [7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter User Guide \(UG480\)](#)
  - this illustrates how to use resistive touchscreens