

Welcome to Phys 319  
Happy New Year

## PHYS 319 Electrical Laboratory:

A project-oriented course introducing the design, construction and programming of microprocessor-controlled devices.

# Course outline:

## **First 6 weeks:**

Crash course in microprocessor programming in C and incorporating external hardware.

Working on example programs and explaining how hardware and software works

**Next 6 weeks:** projects of your choice

Old projects are on the web page.

**Instructor: Dr. Andrzej Kotlicki**

**Office – Hennings 260**

**Email: kotlicki@physics.ubc.ca**

**Technician: Sing Chow**

**Office: Hebb 412**

**e-mail: scho@physics.ubc.ca;**

**phone 604 822 2588**

# TA's :

Johann Drayne [johanndrayne@phas.ubc.ca](mailto:johanndrayne@phas.ubc.ca) T  
Ailar Mahdizadeh [ailar.mahdizadeh@ubc.ca](mailto:ailar.mahdizadeh@ubc.ca) T, Piazza, Lecture  
Phillip Bement [pbement@phas.ubc.ca](mailto:pbement@phas.ubc.ca) Th  
Hassan Talaeian [htalaeia@ece.ubc.ca](mailto:htalaeia@ece.ubc.ca) W

- Web Page:

[https://phas.ubc.ca/~kotlicki/Physics\\_319/index.html](https://phas.ubc.ca/~kotlicki/Physics_319/index.html)

**Labs:** T, W, Th 2-6 pm Hebb 416

**Lecture:** Th 11:30-12:30 Hebb 116

# Past Projects

PID control for Segway

Weather station

[3 axis robot arm](#) by Nicholas di Lello

[Gesture controlled calculator with light strip readout](#) by Cassidy Donaldson

Digital inclinometer

Alarm Clock with music

Anchor lowering machine

Violin tuner

[Robotic Arm](#) by Aiden Smith

Car following black line

Pulse rate monitor

Internet connected personal fall detector

2 output metronome

Bike computer

[Automatic string tuner](#) by Paul Froese

# Why do we teach this course?

- Microprocessors are everywhere – maybe it is worth knowing something about them.
- Understanding microprocessor programming  
Including external hardware
- Learn how to quickly and cheaply build an instrument, a part of an industrial system or an appliance which will save a lot of yours or company time and money.
- Learn how to extract useful information from datasheets
- It is good experience for a summer job or grad school





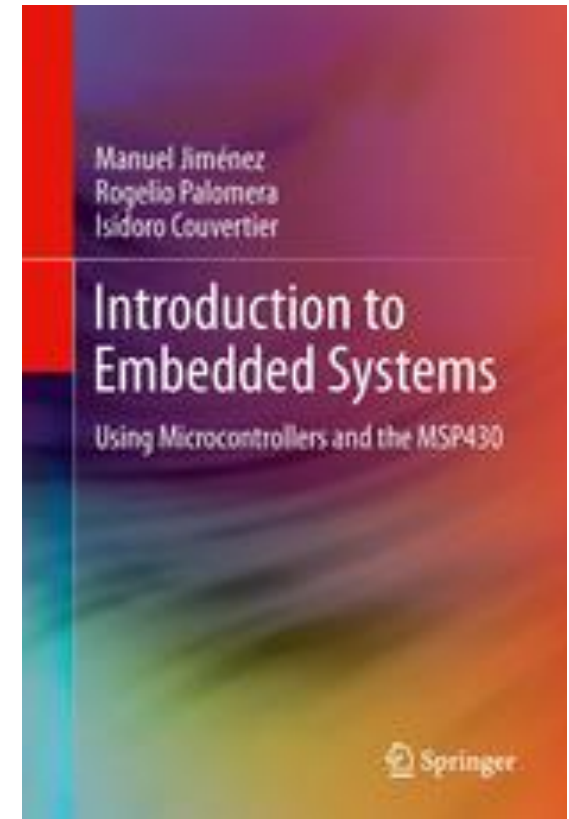
## Reference Text

Introduction to Embedded Systems Using  
Microcontrollers and the MSP430.

Available for download from  
the UBC library at

<http://webcat2.library.ubc.ca/vwebv/holdingsInfo?bibId=7372090>.

We won't follow this text, but its a great  
reference.



# Library of the example codes

[https://dev.ti.com/tirex/explore/node?node=A\\_\\_A  
CtifcK4C02IASwKY7wrRA\\_\\_msp430ware\\_\\_IOG  
qZri\\_\\_LATEST](https://dev.ti.com/tirex/explore/node?node=A__A<br/>CtifcK4C02IASwKY7wrRA__msp430ware__IOG<br/>qZri__LATEST)

# Lab procedures:

- Individual project work
- Laptops – PC or Mac
- Notes (electronic only) in any format convertible to pdf
- Submitting lab reports and programs: on Canvas pdf format only
- Lab manual and Technical manuals (on line, frequent changes and updates!)
- Lab checks

# Marking:

To pass the course you have to:

1. Submit all three reports and check all the required parts (show them working to the TAs or instructor)
2. Present the final project and submit the final report.

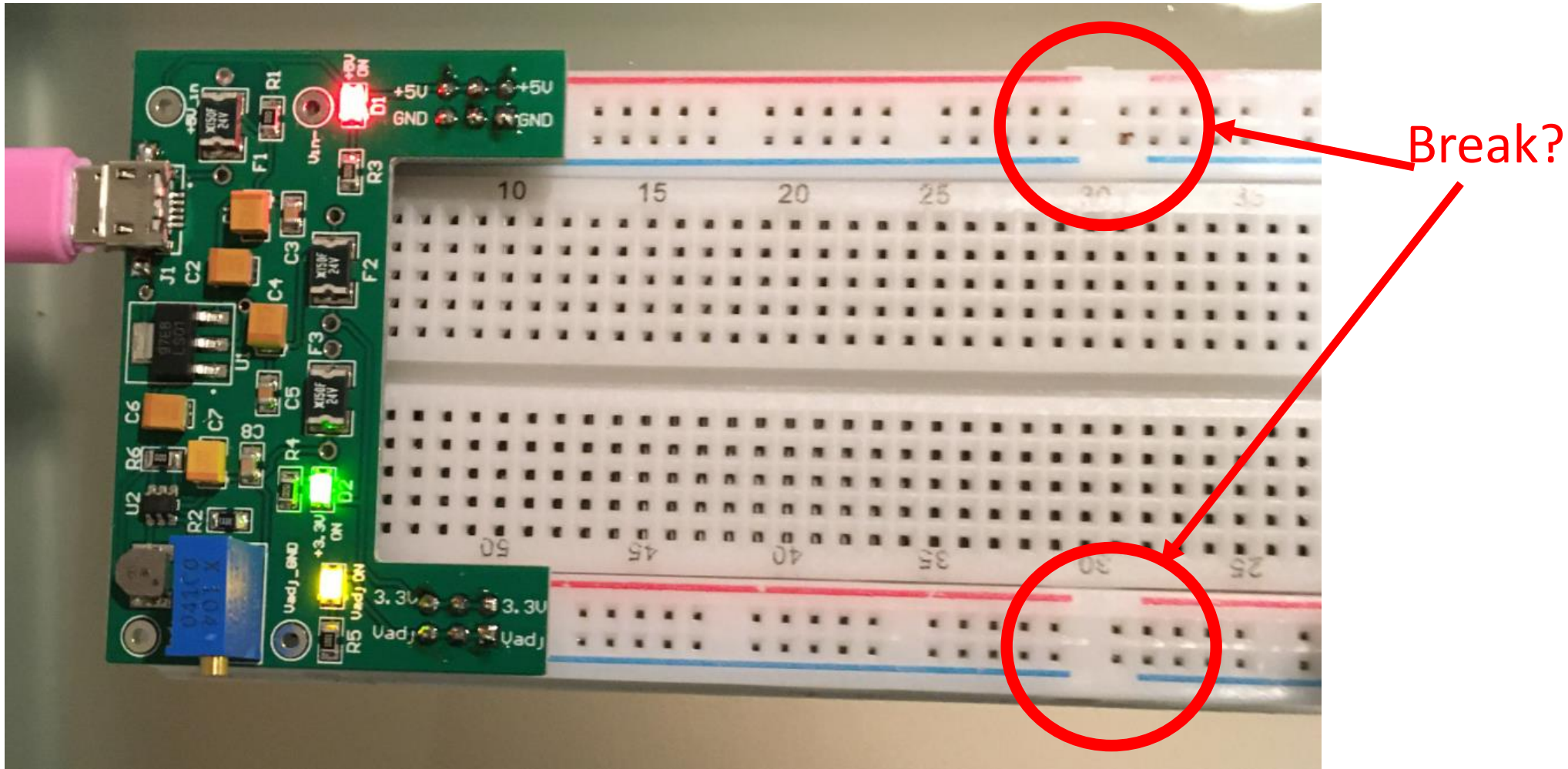
If any of these elements are missing, your grade will be lower of 45% or total of the points.

Marking:

- A Lecture test 20%
- Activities 5%
- Programs and lab reports in first 6 weeks 20%
- Project proposal 3%
- Status report 2%
- Project quality and functionality 20%
- Presentation 10%
- Final report 20%

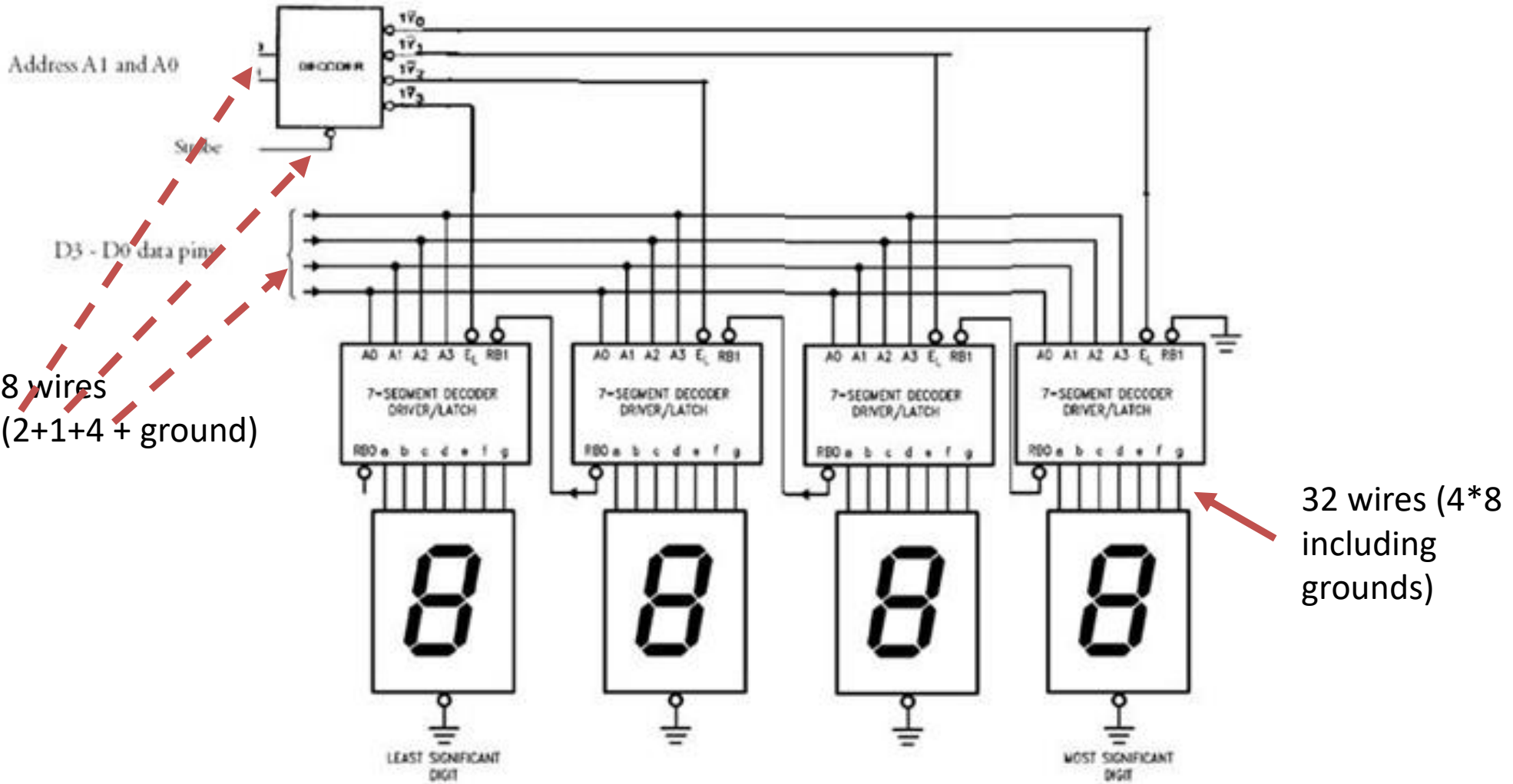
Late report submissions - 10% of the grade will be subtracted per day down to 50%. We often wave the subtractions for good reasons (not for.....)

# Board with power supply

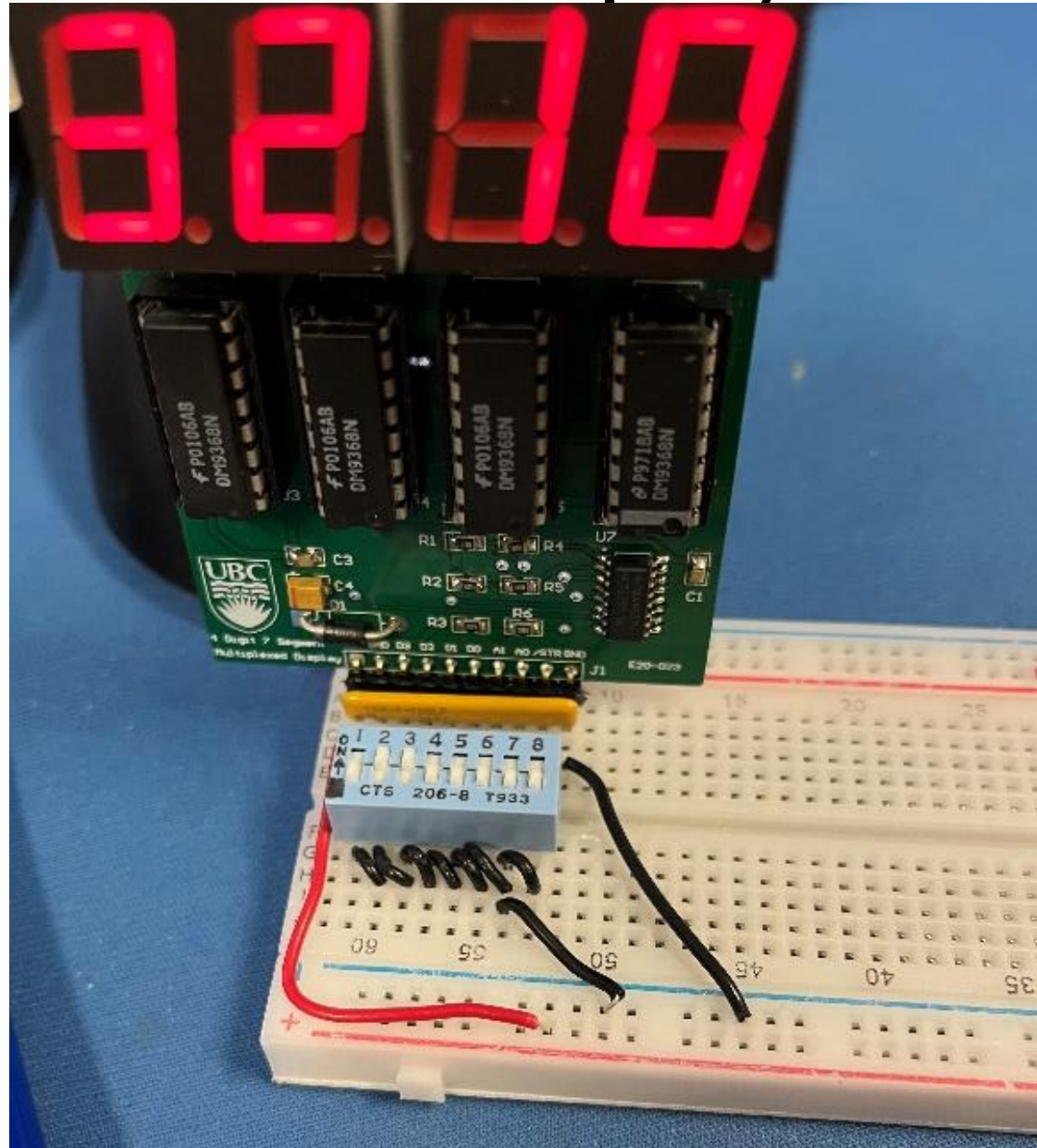


Notice 5V and 3.3V power supply, different than for 219  
Adjustable voltage on the power supply ( $V_{adj}$ ): Do not use it  
to power anything! It is only a low current reference voltage  
and it does not go to 0.

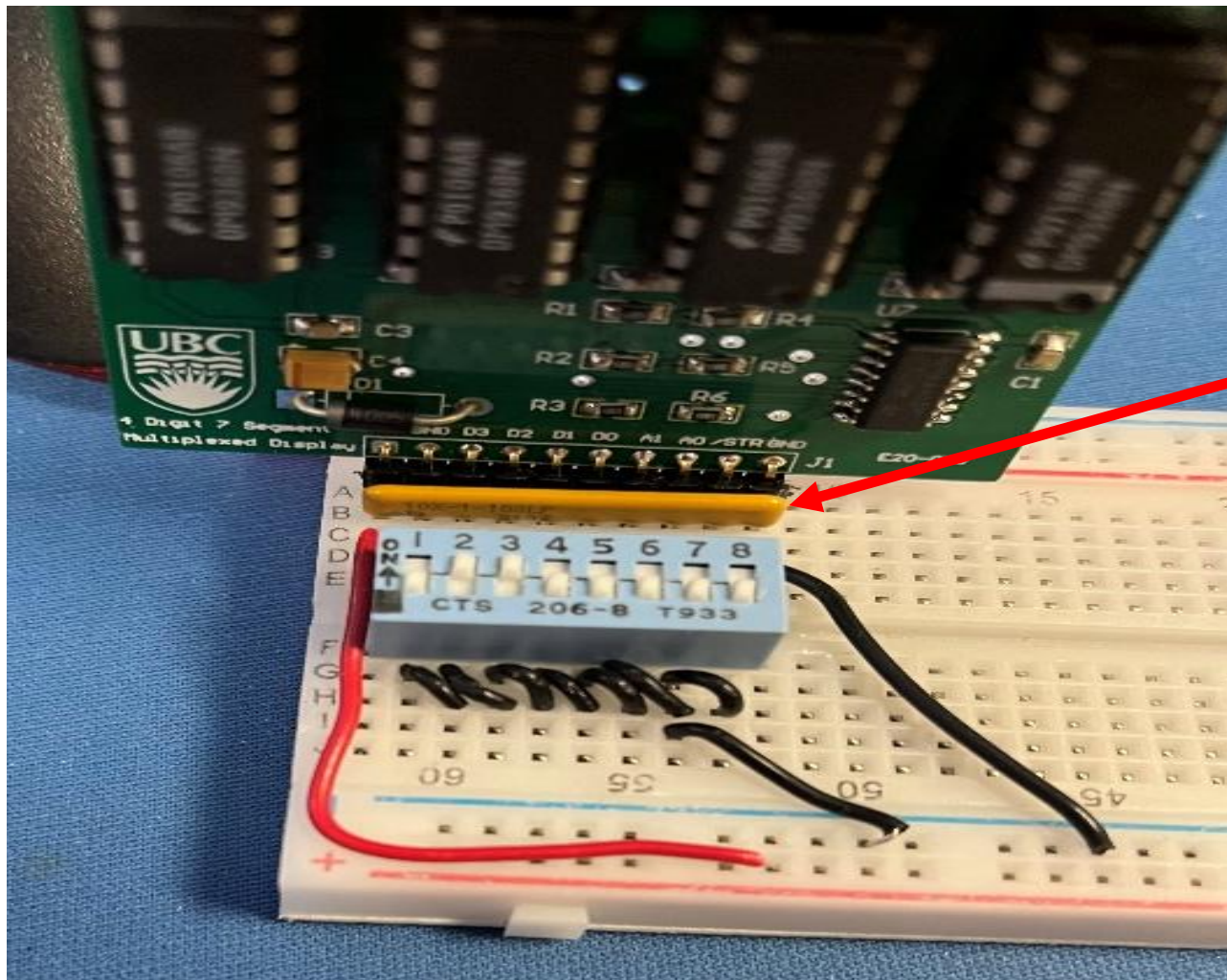
# Multiplexed 7-segment Display



# The actual display board



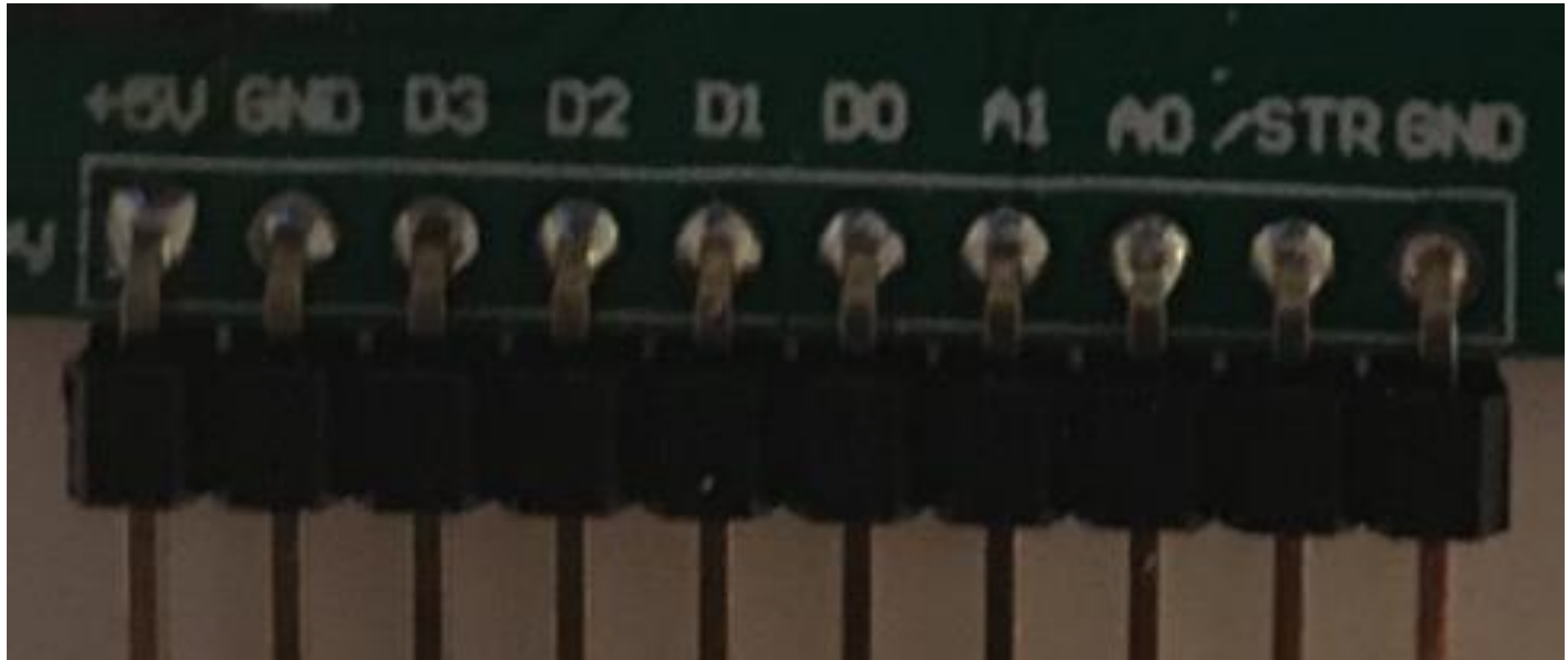
# Connections to resistor block and switches



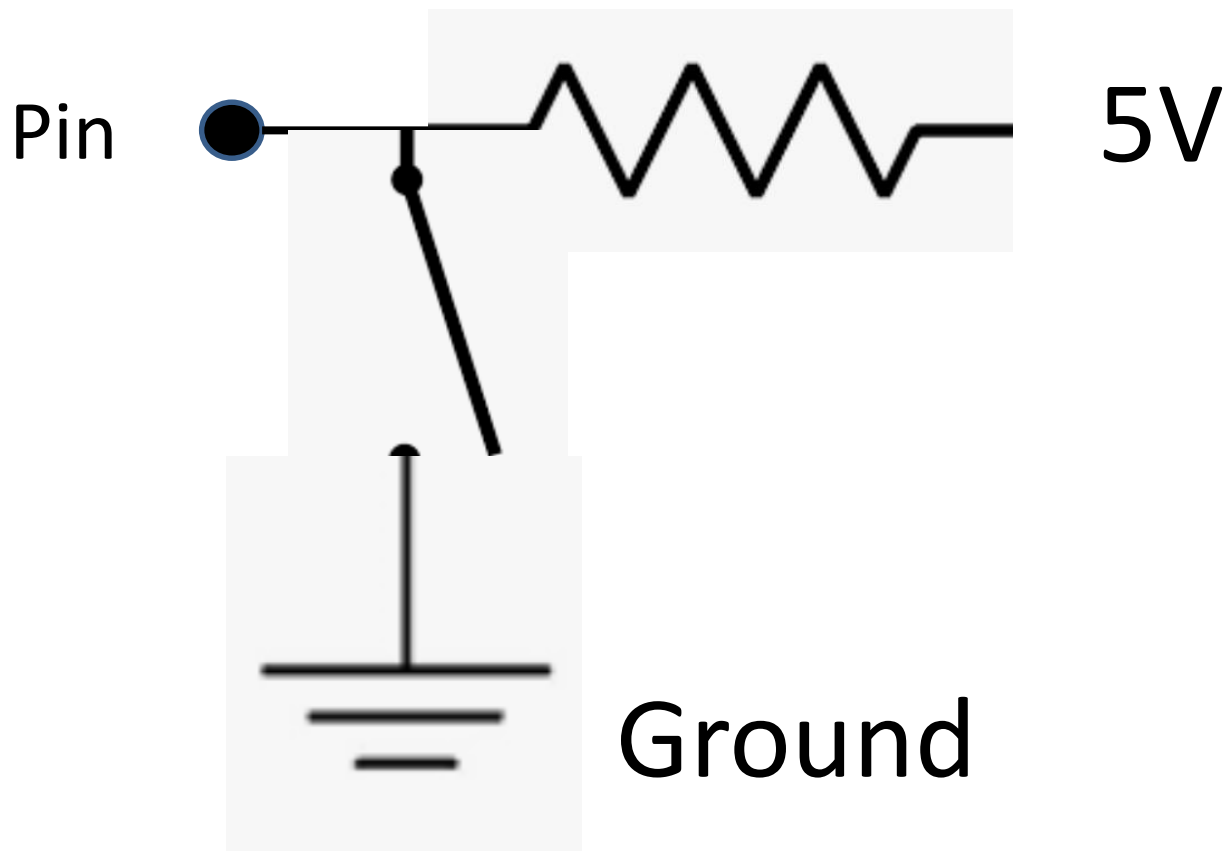
resistor block  
With pin 1 on  
the left  
connected to  
red wire



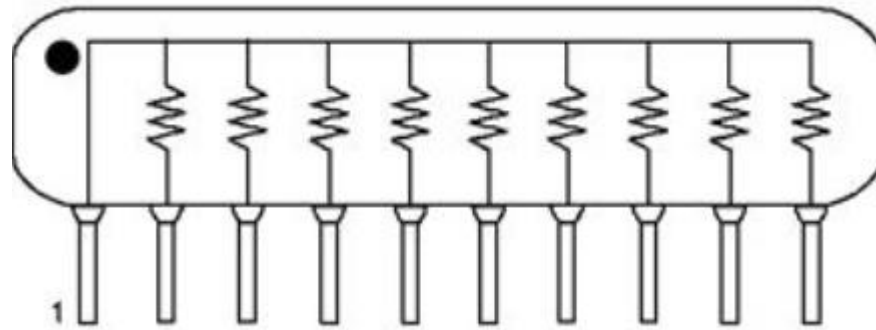
# Pins of the display board



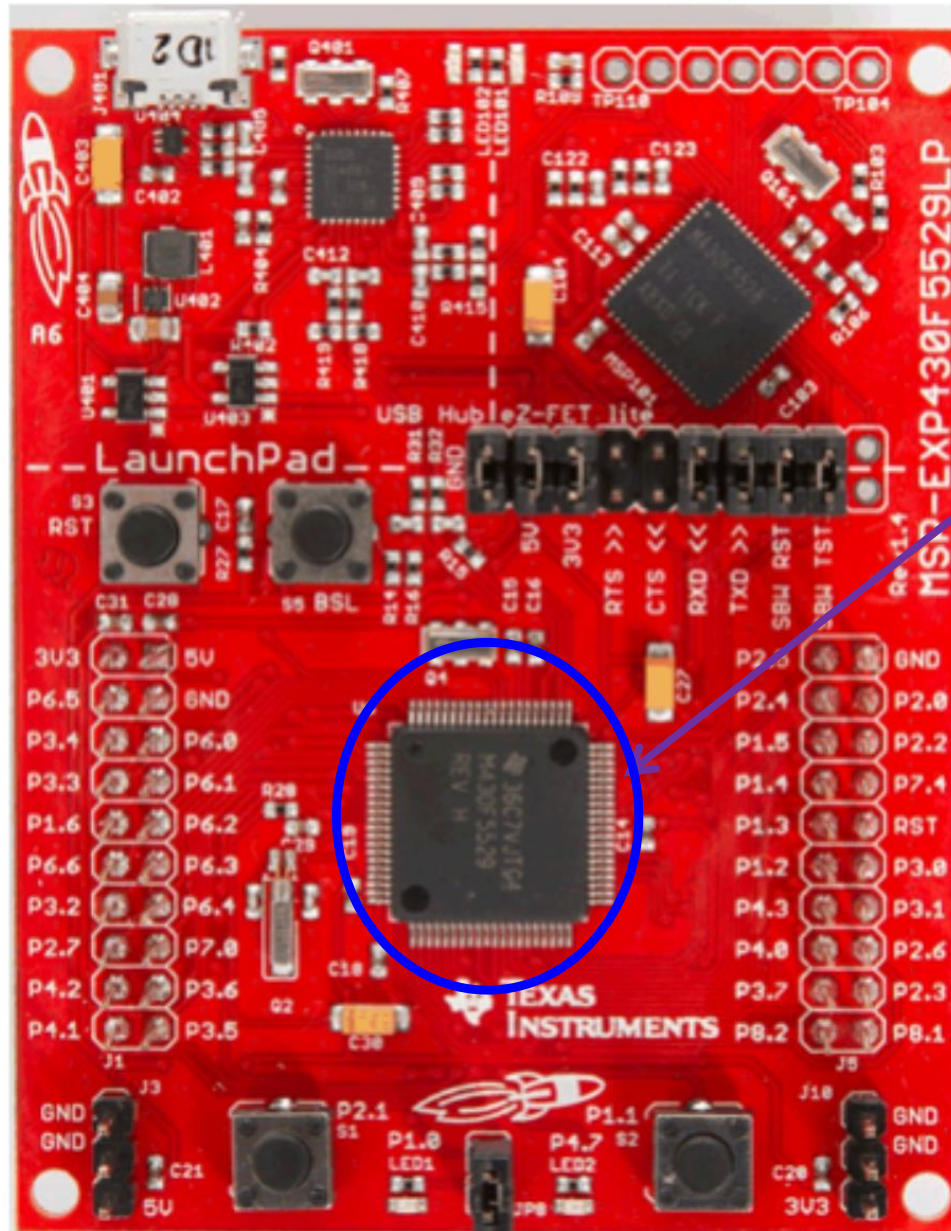
Each of the seven pins (/STR, A0, A1 and D0-D3) has to be connected like this for the first lab segment



You should use this set of 8 resistors



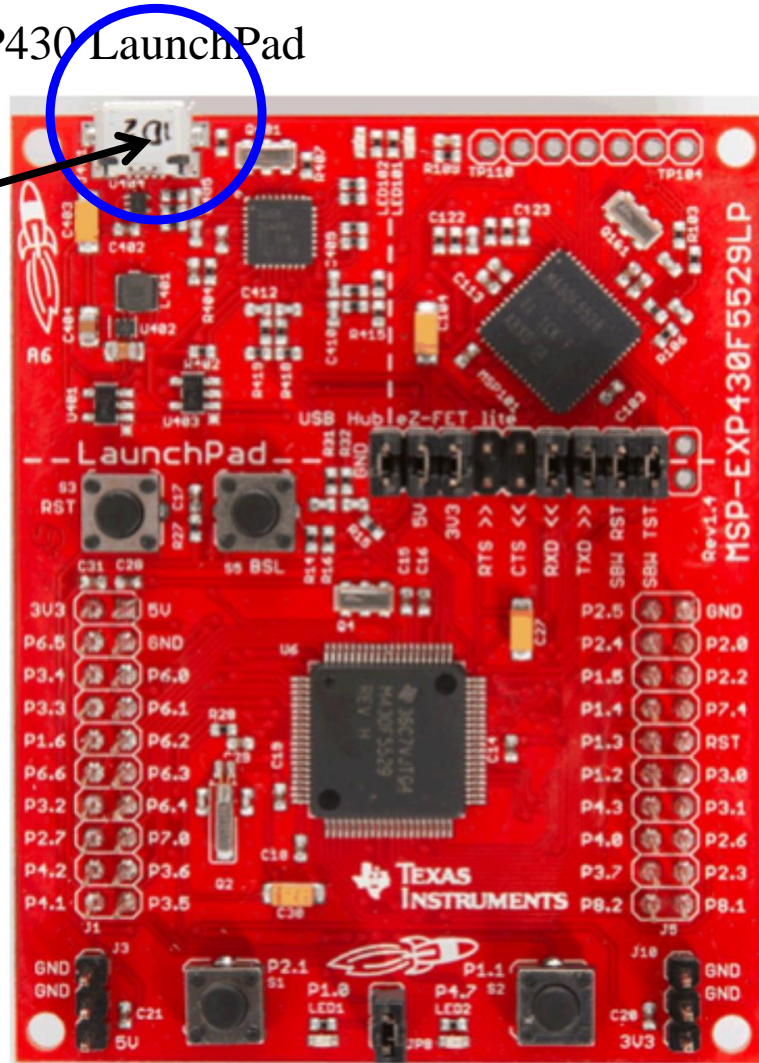
# Texas Instruments MSP430F5529 LaunchPad



MSP430F5529

One of more than 400 different part numbers in the MSP430 family.

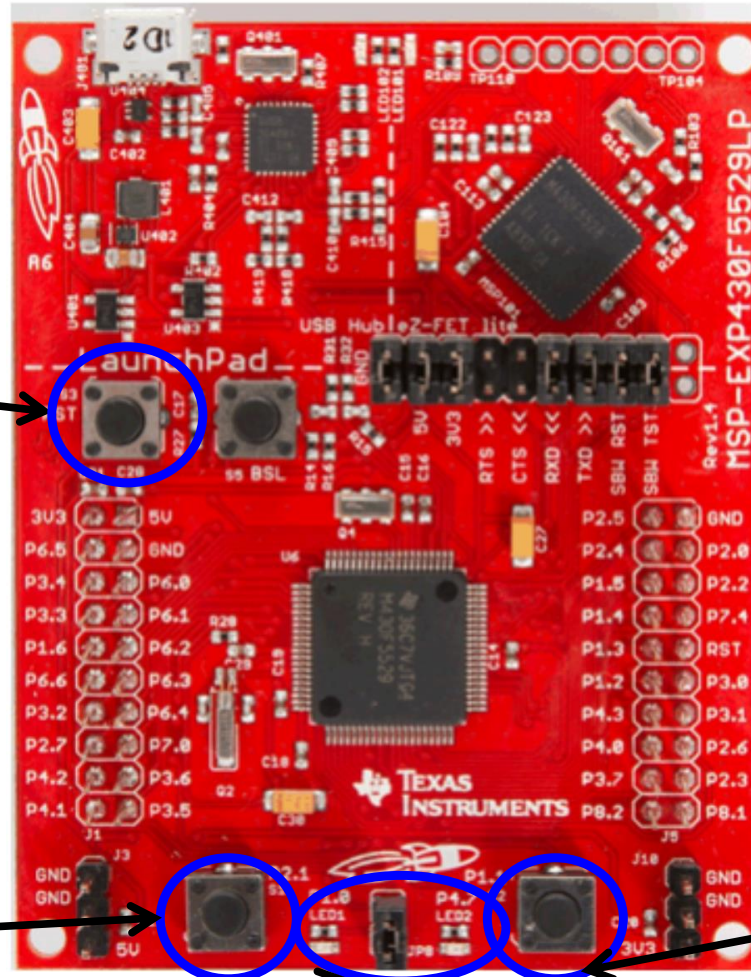
# Texas Instruments MSP430 LaunchPad



- USB connector:
- 1) loading programs into the MSP430
  - 2) power
  - 3) communications between program running on the MSP430 and a program on the host.

# Texas Instruments MSP430 LaunchPad

Push button to reset  
(debug has to be off  
for this to work)



Push button for program use

Push button for program use

Two LED's for program use

## Board Safety:

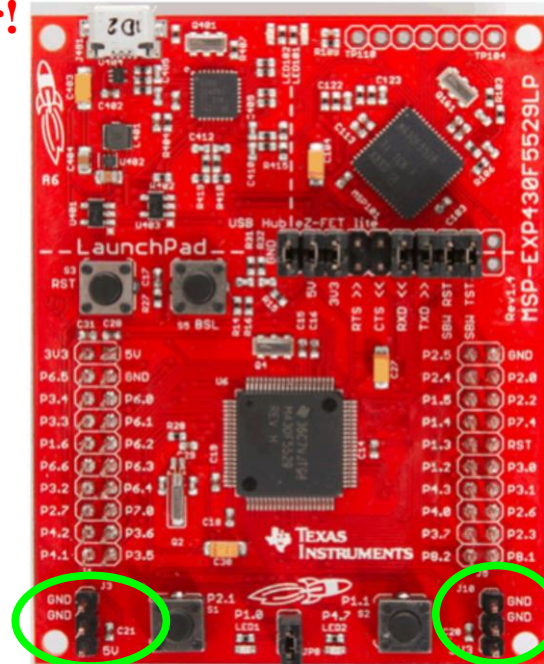
When Launchpad board is connected to USB:

- **don't** connect power sources to 5V or 3.3V pins (chip is USB powered). You can use these as sources to power low power sensors but not ultrasonic range sensor or motors
- **do** connect ground (GND) pin to your circuit (no floating ground)
- **do** always connect Px.x pins to your circuit through resistors

**Any voltage higher than 3.6V or any negative voltage has the potential to damage the Launchpad board.**

**Any voltage higher than 5V or any negative voltage has the potential to damage the USB controller in your computer.** Resistors in series with the 5V power supply make the likelihood of damage **much smaller!**

5V and  
ground



3.3 V and  
Gnd

# How does the Microprocessor work?

- Performs simple operations:

- add
- subtract
- compare
- fetch/store bytes/words

(For the MSP430, a word is 16 bits = 2 bytes).



# Binary numbers

- Decimal numbers
- There are 10 digits 0-9
- Each digit represents consecutive power of 10
- $256 = 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$
- Binary numbers
- There are only 2 digits 0 and 1
- Each digit represents consecutive power of 2
- Binary  $101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5$  in decimal

# Input – output port

- Connection of 8 pins which can receive or output voltages.
- When we use it as digital port the only possible inputs or outputs are about 0 V or about 3.3 Volts.
- The way each pin works depends on the state of number of 8 bits in various memory locations.
- To set the state of each of these locations we send to it an 8 digit binary number.
- The binary number is indicated with prefix 0b

# General Purpose Input/Output (GPIO) Ports

Several registers control the configuration and operation of sets of pins. In these registers, the different bits in the register control different pins. For our microprocessor x can be any integer between 1 and 8 as we have 8 GPIOs. Not all the pins are accessible. x below can be an integer between 1 and 8.

PxDIR – sets the pin directions. Bit = 0 = input, Bit = 1 = output.

PxIN – input register. When configured for input, this register contains the digital input values

PxOUT – output register. When configured for output, writing to this register sets the outputs. When configured as input sets the pullup (1) or down (0)

PxREN – pullup/pulldown enable. Bit = 1, enable resistor (P1OUT sets whether pullup or down).

PxSEL – alternate function enable – 0 means GPIO.

PxIE – Enable interrupt on some input port pins

PxIES – chooses if interrupt occurs on raising (0) or falling (1) edge

eg setting P1DIR = 0b00000011 configures pins P1.0 and P1.1 as outputs, P1.2-P1.7 as inputs

## Before the first lab:

- read the lab manual parts 1, 2, 3 and
- Install Code Composer on your PC or Mac.
- We will test it during the first lab period.
- You are expected to do parts 1, 2 and 3 described in the manual in the first 2 weeks of the labs and submit your notes for marking at the beginning of third week's lab