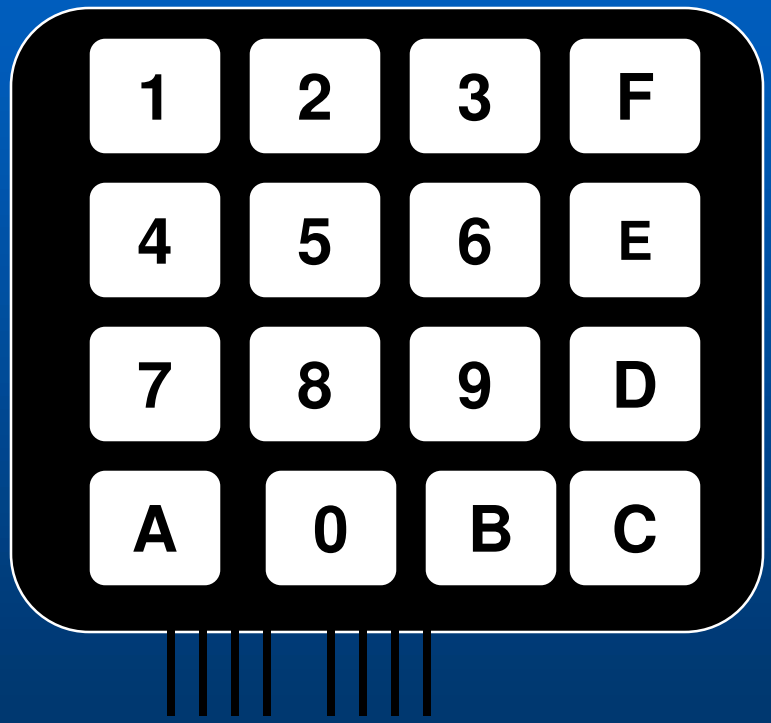
Decoding and Using a 4x4 Keyboard

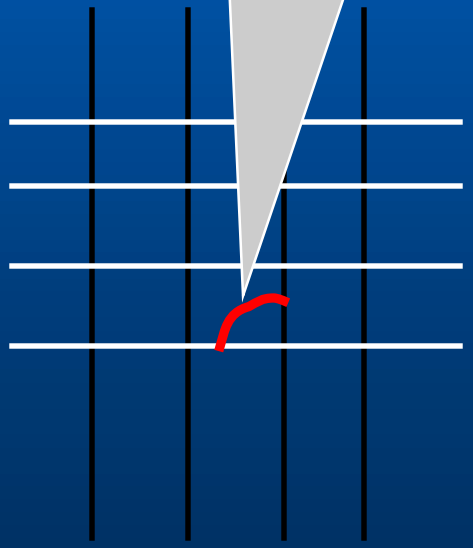
Costas Foudas, Imperial College,
Rm: 508, x47590

The Keyboard

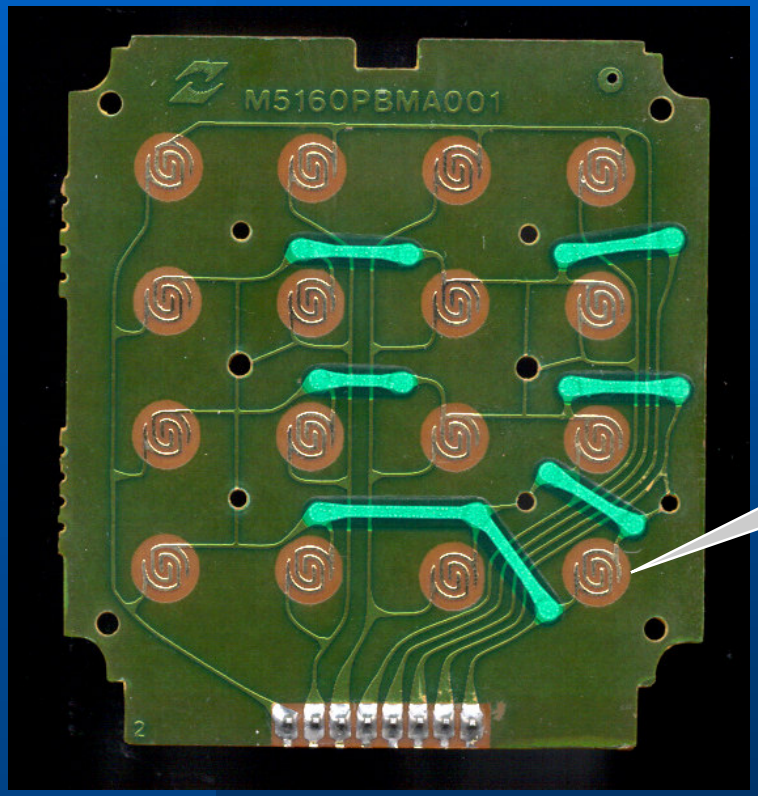


8 lines corresponding to 4 rows and 4 columns

By Pressing a button you connect one of the white lines to one of the black lines. *But you don't know which.*



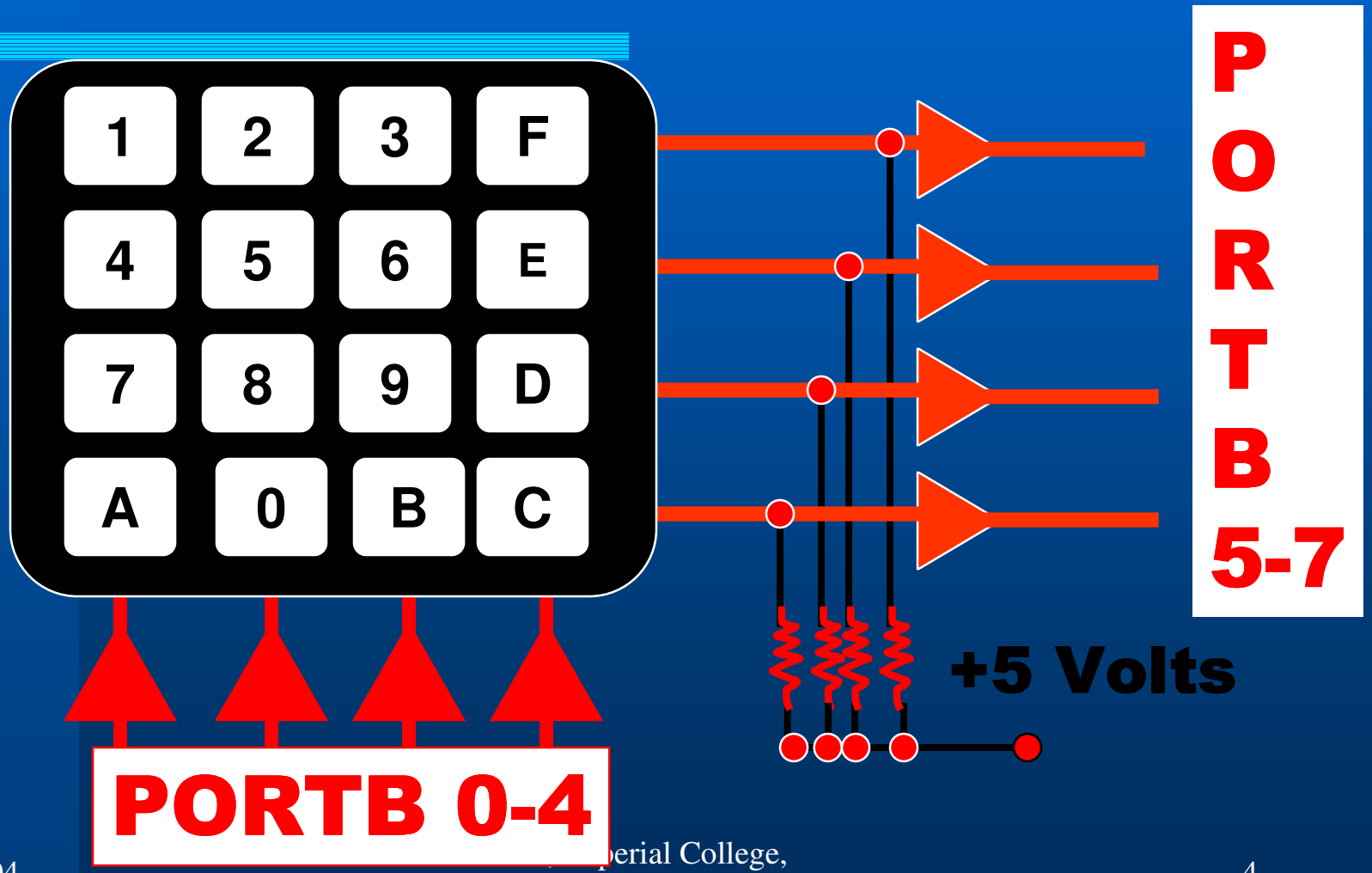
The PCB of the Key-Board



From this picture you get an idea of what exactly happens in the keyboard.
IT IS ALL PASSIVE: No ICs no Transistors

The buttons as you may guess connect one row with one column !!!!

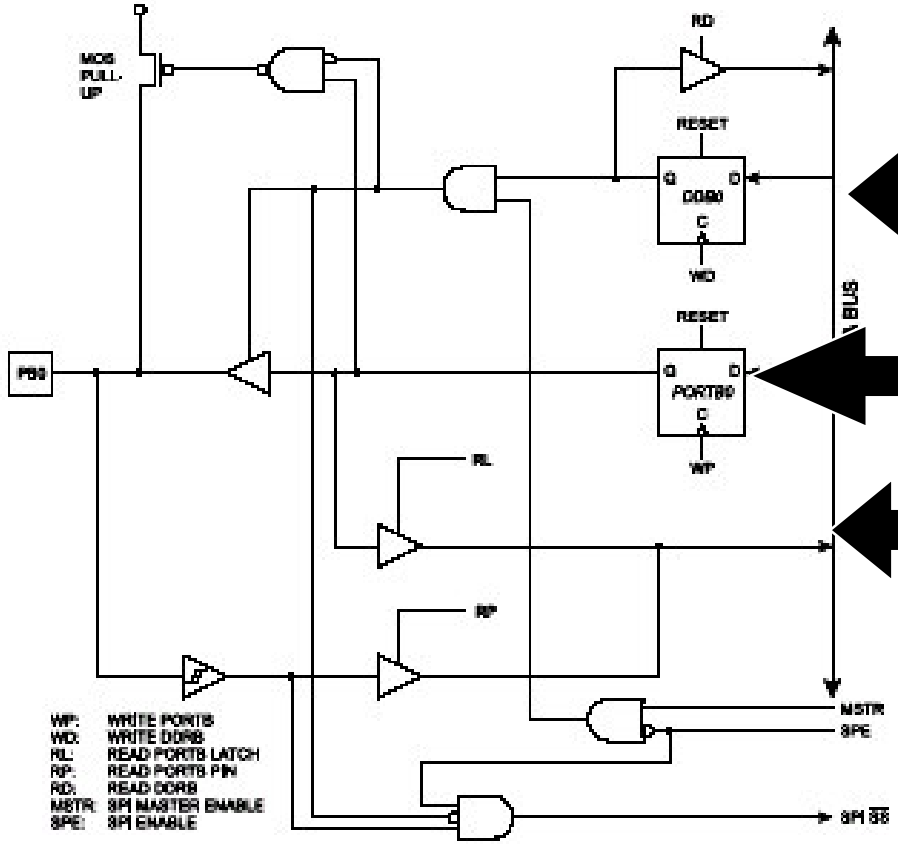
Decoding the Key-Board



erial College,
Rm: 508, x47590

Pull-up Resistors on ATmeag103 Ports

Figure 54. Port B Schematic Diagram (Pin PB0)



Port Example:

DDRA : Direction Reg.

PORTA : OUTPUT Reg.

PINA : INPUT (no Reg.)

Notice that each pin in PORTB is different !!

How to configure PORTB

(1) Configure PORTB 5-7 as <u>inputs</u> and PORTB 0-4 as <u>outputs</u> (DDRB: $0F)

(2) Set the PORT REGISTERS in such a way so that the inputs have pull up resistors and the outputs drive low (PORTB: $F0)

Task Plan I

Write a routine that decodes the key on the 4x4 Keyboard and tells you what key was pressed on the LCD display

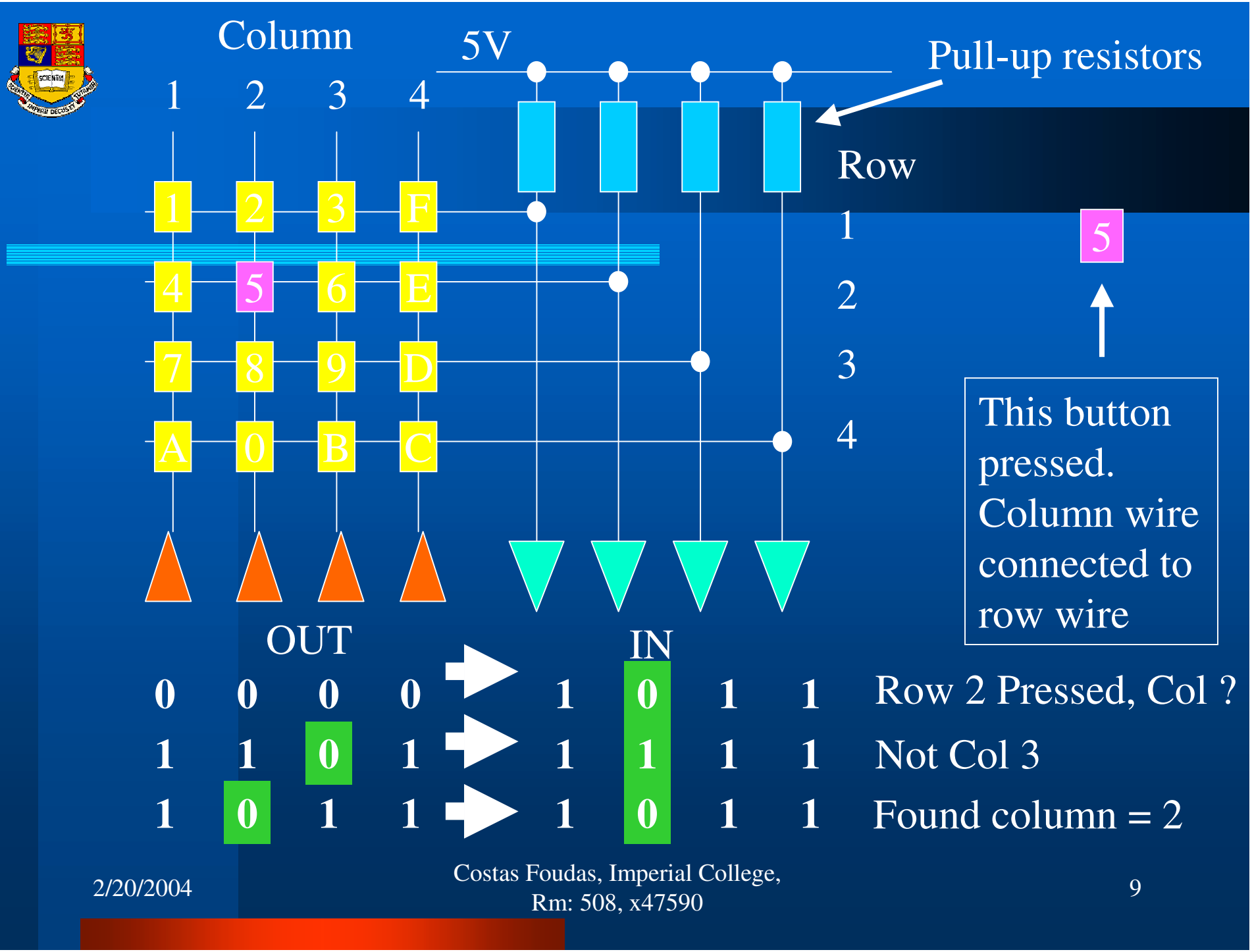
But first you need to know exactly what does the keyboard do internally

Task Plan II

STEP1: Set the pullup resistors on the 4 bits of port B and connect the 4 pins of the keyboard (through resistor packs). At this point the 4 lines of the keyboard are held high by the pullup resistors (your LEDs should be dark).

STEP2: Connect the rest of keyboard pins to 4 PORTB pins (resistor packs).

STEP3: Push a keyboard button and have a program that drives the PORTB outputs low one at a time (1110,1101,1011,0111). **This should tell you which Pin does is connected with which when you press that button.**



Costas Foudas, Imperial College,
Rm: 508, x47590

Keyboard ⇔ ATmega103 Port

We need a reliable setup where the ATmega103 interprets relatively fast any key pressed on the keyboard .

Noice that the microprocessor is faster than Your finger and can 'see' the bouncing up And down that result when you press the a button of the keyboard. So you need to check if what you read is stable

Two questions:

(1) What type of hardware setup do we need ?

(2) What kind of driver software do we need ?