

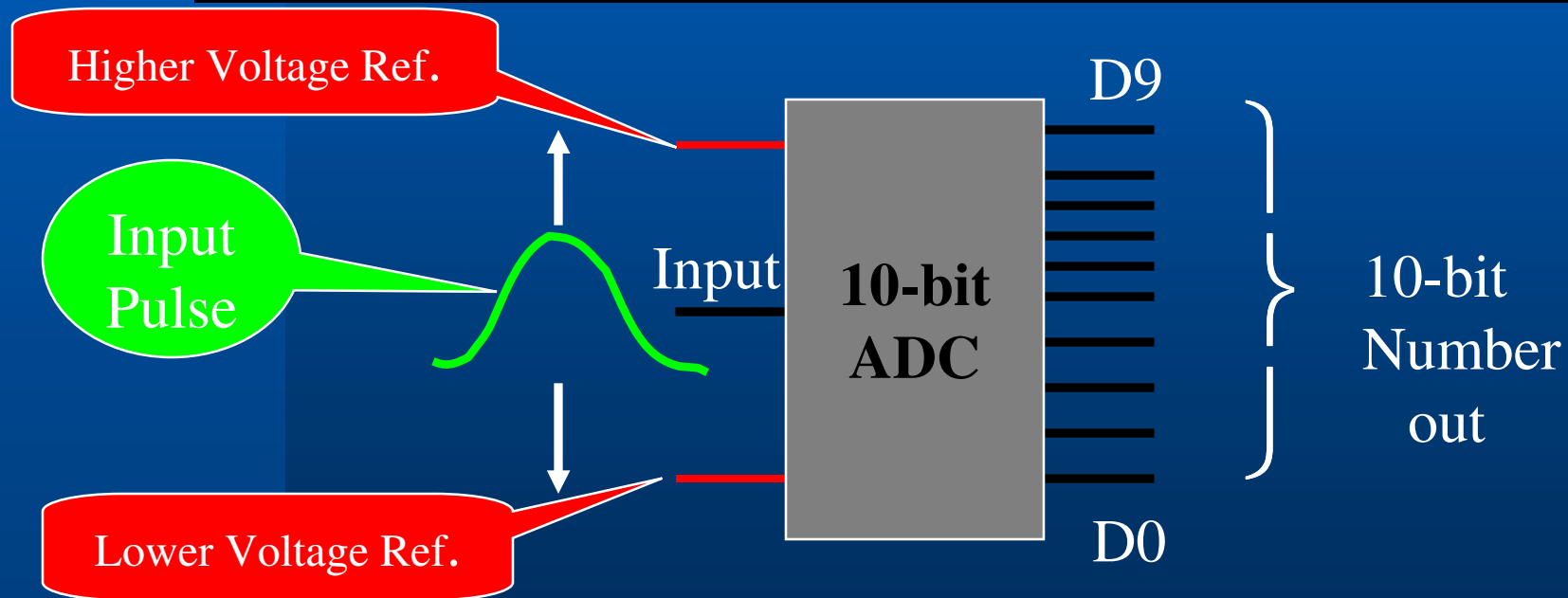
# Device Drivers – Digital Voltmeter

---



# Analog to Digital Converters

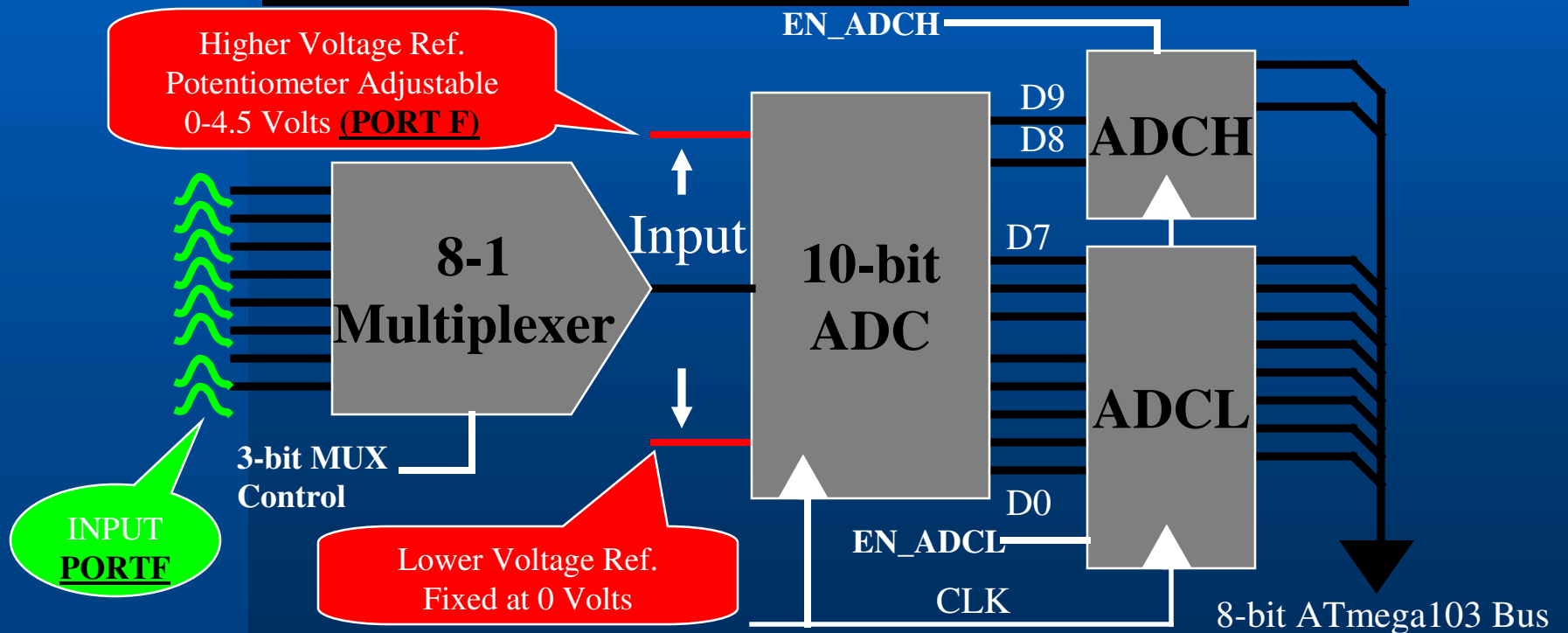
The ANALOG to DIGITAL CONVERTERS (ADC) are devices that measure analog signals and convert them to numbers:





# The ATmega103 ADC

The Atmega103 has one 10-Bit ADC (on-chip) connected to an 8 to 1 analog multiplexer:



2/20/2004

Costas Foudas, Imperial College,  
Rm: 508, x47590



# The ADC Registers I

The 10-bit ADC on Atmega103 receives its inputs from PORTF.

The Reference voltage is controlled by a potentiometer.

The device can be controlled by 4 on board registers:

- (1) **ADCSR** : ADC Status Register
- (2) **ADMUX** : Multiplexer Control
- (3) **ADCH** : Data Register for D9,D8
- (4) **ADCL** : Data Register for D0-D7



# The ADC Registers II

## The ADC Status Register DCSR :

D7	D6	D5	D4	D3	D2	D1	D0
ADEN	ADSC	0	ADIF	ADIE	ADPS2	ADPS1	ADPS0

**ADEN** : It turns on the ADC clock (You must do it first)

**ADSC** : Start conversion

**0** : The data book says it should always be zero

**ADIF** : Interrupt flag (*gets set during a conversion and gets cleared if you write '1' to it AFTER you have read BOTH data registers*)

**ADIE** : Interrupt enable (*in the software I have not used interrupts so I have kept it '0'*)

**ADPS0-2** : Three bits that determine the ADC clock prescale  
(e.g. 3 = 1/8)



# The ADC Registers III

The ADC Multiplexer Register ADMUX is used to select a channel to digitize (3 bits for Channel 0-7) :

D7	D6	D5	D4	D3	D2	D1	D0
					MUX2	MUX1	MUX0



# The ADC Registers IV

The ADC Data Register ADCL is used for the 8 lower data bits (D7 - D0) :

D7	D6	D5	D4	D3	D2	D1	D0
D7	D6	D5	D4	D3	D2	D1	D0

The ADC Data Register ADCH is used for the two highest data bits (D9,D8) :

D7	D6	D5	D4	D3	D2	D1	D0
						D9	D8



# Task Plan

Design and construct a Digital Voltmeter:

- The Atmega103 on-chip ADCs should be used to digitize the input analog voltages.
- The LCD should be used to display the voltmeter readings.
- Switches or a keyboard can be used to tell the Voltmeter to capture data.

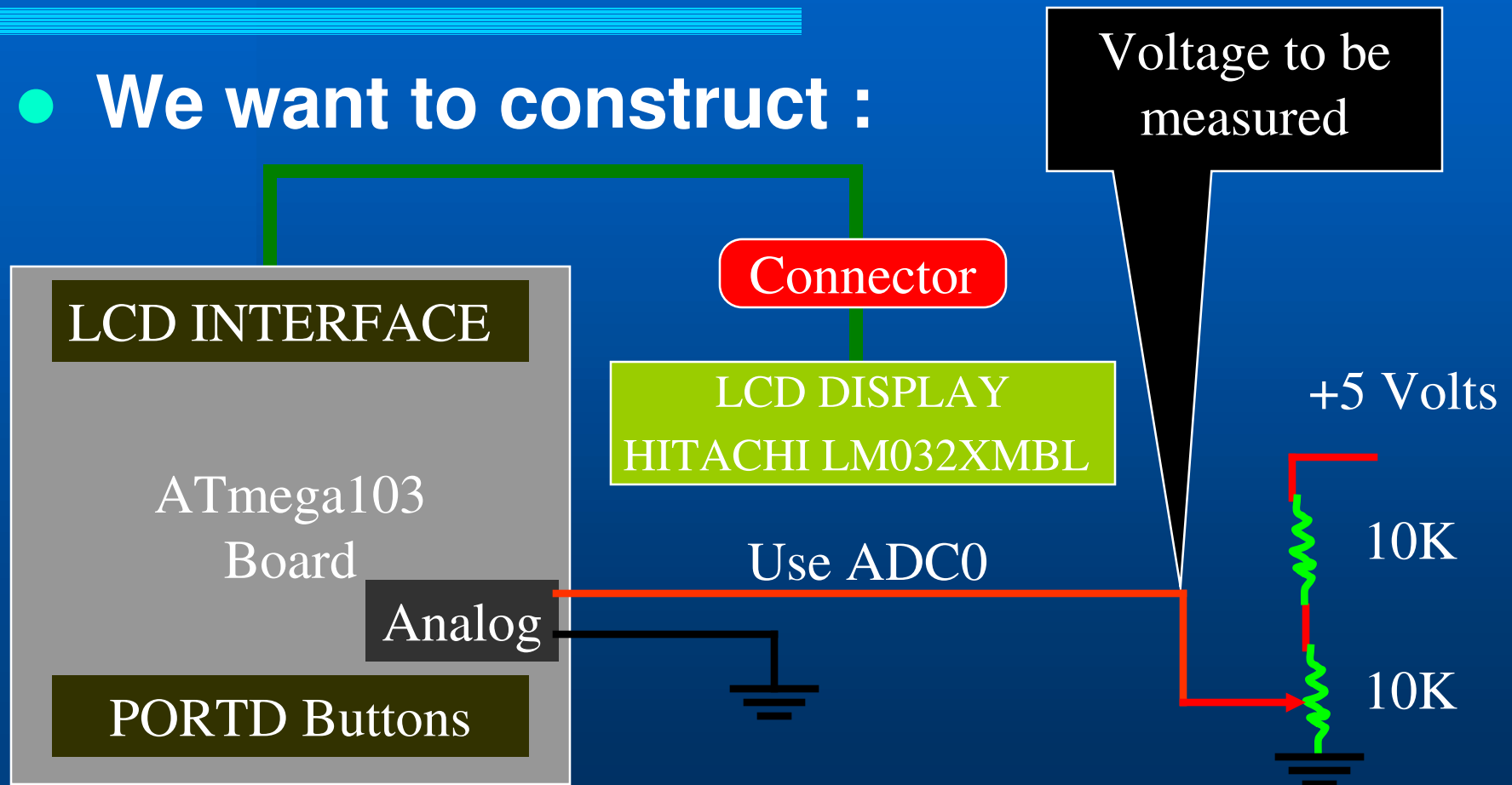
*Connect your Voltmeter to the potentiometer and measure the voltage. Calibrate it against a Voltmeter in the Lab. Demonstrate that your device works !!!!*





# Conceptual Design

- We want to construct :





# Some Important points :

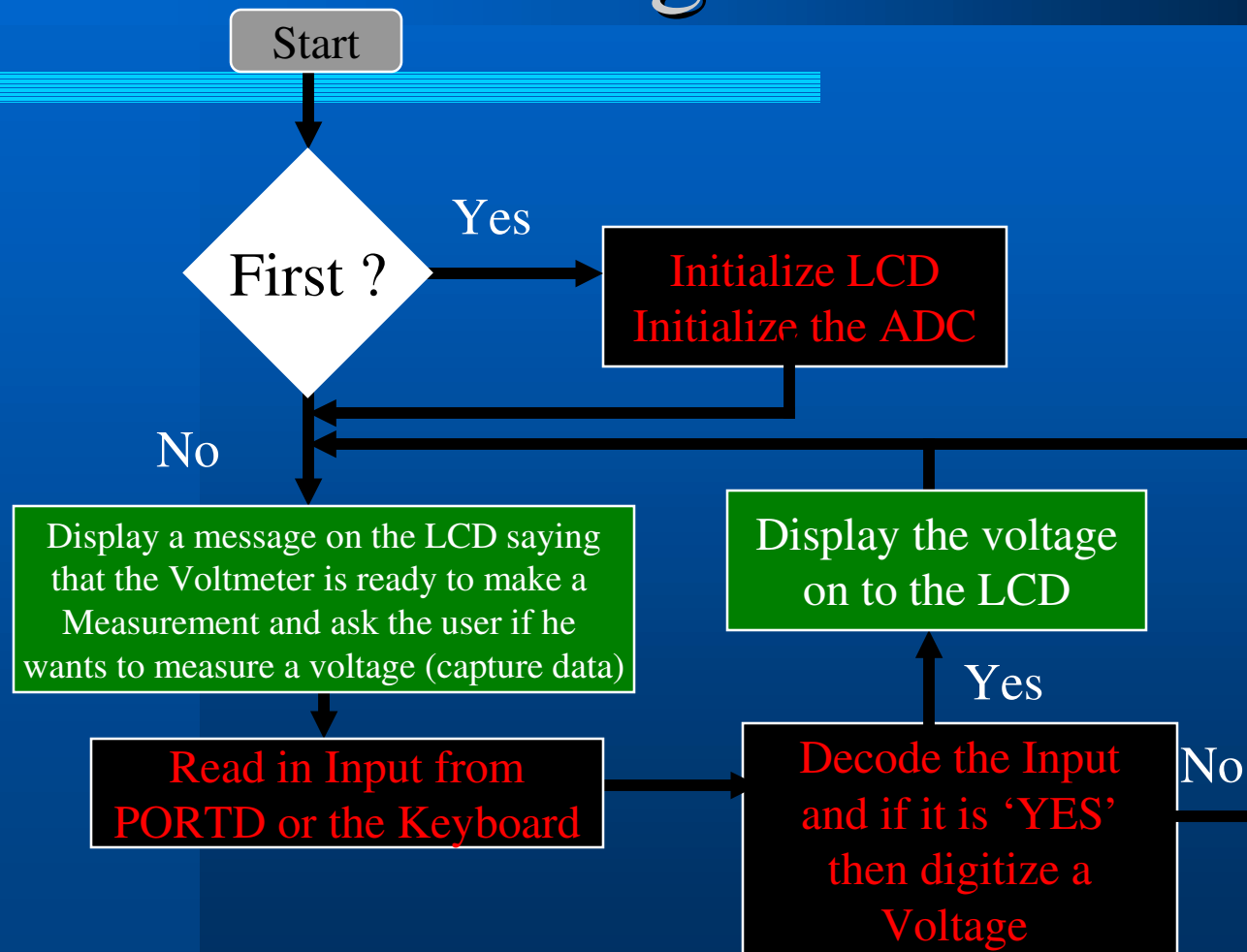
The Voltage you are trying to measure is obviously between 0 – 2.5 Volts

The reference voltage of the ADC should not be lower than the voltage you are trying to measure!

First you should set the reference voltage at its highest value (4.5 V) using the potentiometer and by measuring it at the Ref. pin of PORTF



# Software Design of the Voltmeter





# Getting Started

- (1) Use the user interface you created with the LCD display and the keyboard or the PORTD switches.
- (2) Read about using the on-board ADC in the ATmeag103 manual.
- (3) Use the routines you will find in the **ADC1.inc** program in the **web page**.
- (4) ADC1.inc is a standalone program that works you may want to start from that.



# ADC Driver routines

```
.def TempReg =r16
```

```
ADCInit:
```

```
; ***** ADC Setup Code *****
```

```
ldi TempReg, $83
```

```
; ADC Interrupt Disabled, ADC Enabled
```

```
out ADCSR, TempReg
```

```
; ADC Free Run Mode, Prescaler:CK/8
```

```
ret
```

```
ADCsel:
```

```
out ADMUX, ADCChannel
```

```
; Channel Selection
```

```
ret
```

```
DCATrig:
```

```
SBI ADCSR, 6
```

```
; Cause a Conversion
```

```
rcall DEL600mus
```

```
IN ADCDL, ADCL
```

```
; Read in Low Byte
```

```
IN ADCDH, ADCH
```

```
; Read in High Byte
```

```
ADCCLR:
```

```
SBI ADCSR, 4
```

```
; Reset ADIF
```

```
IN TempReg, ADCSR
```

```
; Read in Status
```

```
SBIC ADCSR, 4
```

```
; Wait till the Interrupt flag
```

```
RJMP ADCCLR
```

```
; is cleared
```

```
RET
```