## **Contents**

Control-Lab-in-a-Box (	CLB	-3.3
------------------------	-----	------

<u>3.1 Hardware2</u>	
3.2 Algorithm Design3	
3.2.1 Low Pass Filter (Digital) <u>5</u>	
3.2.2 Exercise	
3.3 Summary8	<b>)</b>
Ot Sollies Av.	
MADE IN	

# Temperature Sensor TMP36 and Low Pass Filter

# **Key Learning Points**

After this Lecture, you will be able to:

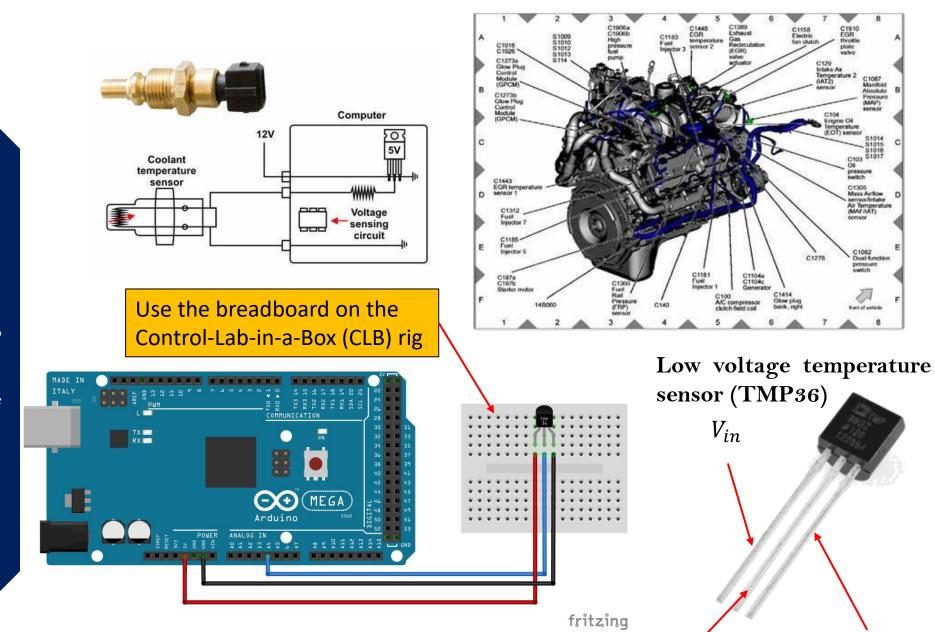
Obtain measured temperature data from the TMP36 sensor, and undertake the process of filtering noisy measured data using a low pass filter

- 1. Introduction

- 2. Actuation 3. Measurement 4. System Identification and Control

### 3.1 Hardware

- Required hardware for the exercise:
  - i. Supported Arduino Uno board
  - ii. USB cable
  - iii. Low voltage temperature sensor (TMP36)
  - iv. 3 x male-male wires
- Voltage input: 2.7*V* to 5*V*
- Operating temperature: -40° C to +125° C



Analogue  $V_{out}$ 

GND

# 3.2 Algorithm Design

- Output from Pin 4 is a 10-bit analogue value ranging from 0 to 1023
- E.g., 150 output from Pin 4, i.e.

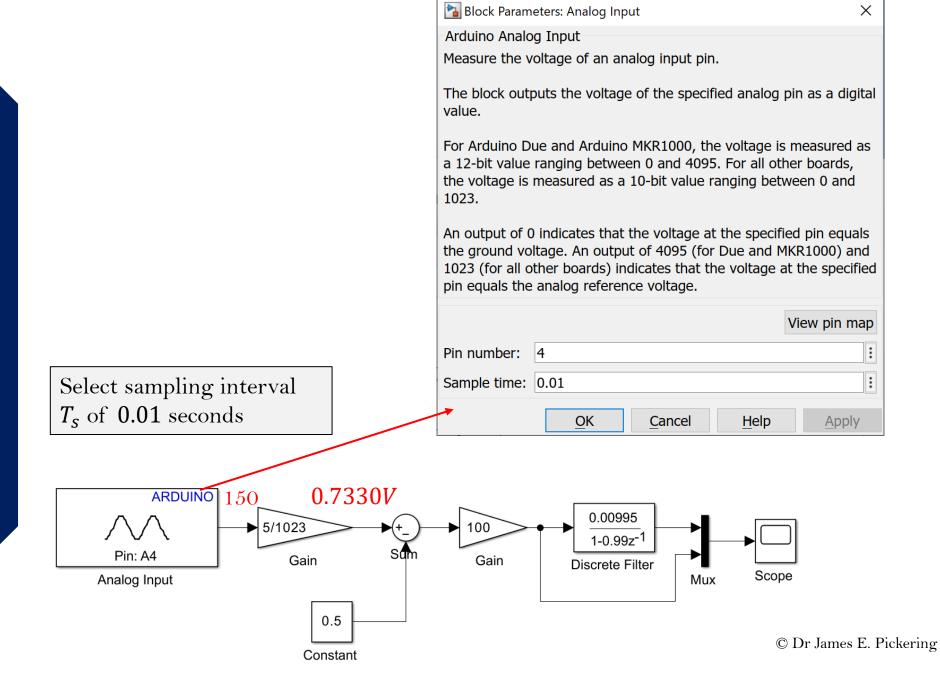
$$\frac{150}{1023} = 0.1466$$

which is basically a ratio between 0 and 1

Multiplied by 5V (Arduino supply) gives:

$$0.1466 \times 5 = 0.7330V$$

Temperature Sensor TMP36 and Low Pass Filter

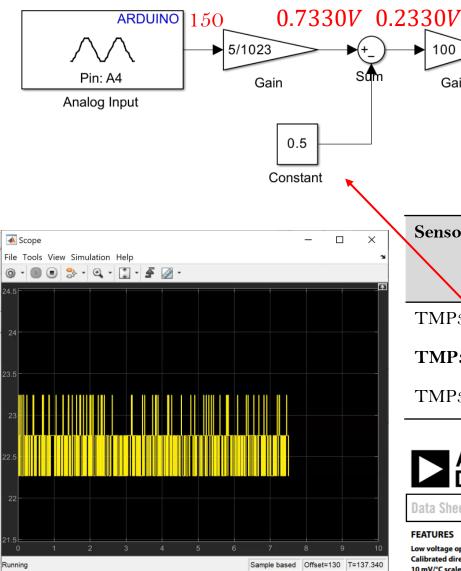


## 3.2 Algorithm Design

TMP36 has a voltage offset of 0.5V (so that negative values can be accounted for), e.g.,

$$0.7330 - 0.5 = 0.2330V$$

The above value is then multiplied by 100, as the TMP36 sensor has an output scale factor of 10mV/ °C, e.g.,  $0.2330 \left( \frac{1^{\circ}C}{0.01V} \right) = 23.3^{\circ} C$ 



Sensor	Offset Voltage (V)	Output Voltage Scaling $(\frac{mV}{{}^{\circ}C})$	Output Voltage @25°C(mV)
TMP35	0	10	250
TMP36	0.5	10	750
TMP37	O	20	500

Mux

0.00995

1-0.99z<sup>-1</sup>

Discrete Filter



Gain

#### **Low Voltage Temperature Sensors**

23.3°C

Scope

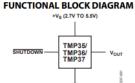
TMP35/TMP36/TMP37 **Data Sheet** 

#### FEATURES

Low voltage operation (2.7 V to 5.5 V) Calibrated directly in °C 10 mV/°C scale factor (20 mV/°C on TMP37) ±2°C accuracy over temperature (typ)

±0.5°C linearity (typ)

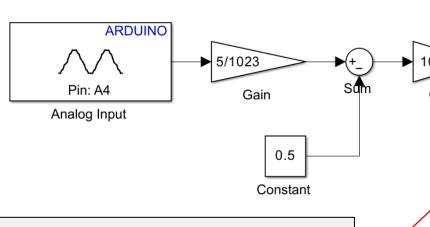
Stable with large capacitive loads Specified -40°C to +125°C, operation to +150°C



**Temperature Sensor** TMP36 and Low Pass Filter

# 3.2.1 Low Pass Filter (Digital)

- Low pass filter applied (stops high frequencies passing through)
- A continuous-time transfer function of an RC (time constant τ) circuit has been discretised using MATLAB
- Ensure the sampling interval  $T_s$  used throughout is consistent



- The discrete-time transfer function will appear in the command window (insert the values into the 'Discrete Filter'
- Recall:  $G(s) = \frac{1}{RCs+1} = \frac{1}{\tau s+1}$

clear; clc; close all;

%% Continuous and Discrete Transfer Function
R=1000; % resistor [Ohms]
C=1000\*10^(-6); % capacitor [F]
Tau= R\*C; % time constant
G = tf([1],[Tau 1]);

T\_s = 0.01; % sample interval
Gd = c2d(G,T\_s) %ZOH method by default

'	Discre	te Filter	Mux	Scope	
				1	
Block Para	ameters: Discrete Filter				×
denominato		ding order of powers	of 1/z.	IR filter. Specify the nu	merator and
	ata Types State At		i sa decare outer are	an birect form 11.	
		anduces			
	ire: Direct form II			-	
Data	Source	Value			
Numerator:	Dialog -	[0.00995]			
Denominato	or: Dialog -	[1 -0.99]			:
Initial states	S: Dialog -	0			:
External res	set: None				
Input proce	essing: Elements as ch	nannels (sample base	ed)		
☐ Optimize	by skipping divide by	leading denominator	r coefficient (a0)		
Sample time	(-1 for inherited):				
0.01				1	
0			<u>O</u> K	<u>C</u> ancel	<u>H</u> elp <u>Apply</u>

Low Pass Filter

0.00995

1-0.99z<sup>-1</sup>

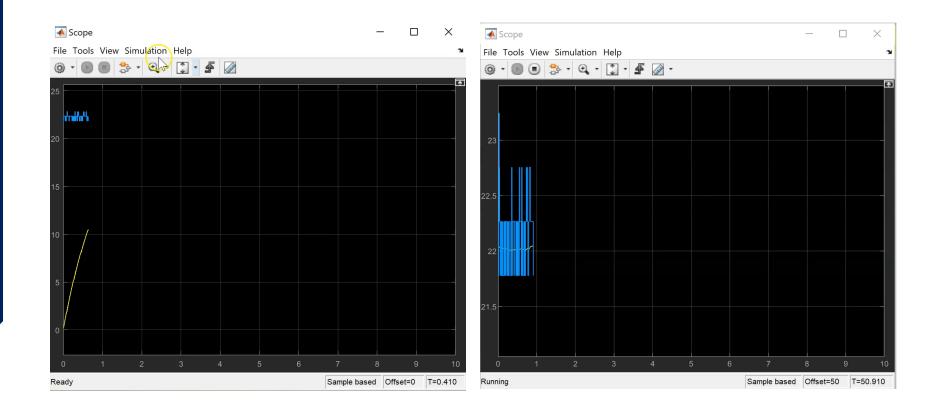
# 3.2.1 Low Pass Filter (Digital)

- Unfiltered signal is blue, and the filtered signal is white
- By adding the filter, the temperature estimate is much less noisy
- A drawback of the filtering is that it adds a delay/lag (this is visible when inspecting the unfiltered and filtered signal), hence no information is provided

• Recall:

$$G(s) = \frac{1}{RCs + 1} = \frac{1}{\tau s + 1}$$

• Reducing the time constant  $\tau$  (i.e., RC) will reduce the lag, however the trade-off is that the noise won't be filtered as well



## 3.2.2 Exercise

- Further task: investigate different sampling intervals  $T_s$  and time constants  $\tau$
- Fill in the table

Sample interval, $T_s$ [seconds]	Time constant, $ au$ (i.e., $RC$ value)	Filter lag [seconds]	Filtering performance [worse/better]
0.01	1	5	N/A as baseline

## **Contents**

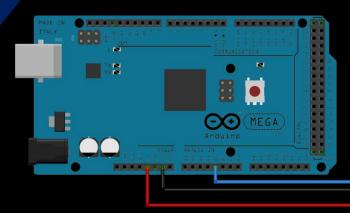
Control-Lab-in-a-Box (CLB	) - 3.3
---------------------------	---------

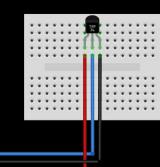
<u>3.1 Hardware2</u>	
3.2 Algorithm Design3	
3.2.1 Low Pass Filter (Digital)5	
3 0 0 Evercise	
<u>3.3 Summary</u> 8	
3.3 Summary	
MADE IN MANAGEMENT	

# Temperature Sensor TMP36 and Low Pass Filter

# 3.3 Summary

- The low voltage TMP36 temperature sensor has been used with an Arduino Uno to capture data (hardware)
  - Simulink has then been used to develop a signal processing algorithm to convert the data outputted from the temperature sensor into a physical value (i.e., degree Celsius)





- 1. Introduction 2. Actua
- 2. Actuation 3. Measurement
  - 4. System Identification and Control