02-1	02-1	02-2 02-	-2
Process State, Sensors, and Interfacing		Process State and Process Variables	
Function of sensors: convert physical quantity to information.		The <i>process state</i> is the current condition of the process, down to in- finitesimal detail.	
Information will (usually) be read by computer via an interface.		The process variable is a part or characterization of a process state,	
Ultimately, the information, in the desired form, will be stored in a memory location.	a	usually in terms of a common measure.	
		For example, consider a coffee maker.	
The following steps are typical: 1: A transducer converts process state to a raw electrical quantity.		<u>Process state</u> : amount of water in carafe, water temperature, chemical description of water in carafe, type of coffee beans, etc.	
<ol> <li>A conditioning circuit converts the raw electrical quantity.</li> </ol>	1	<u>Process variable:</u> temperature of water.	
electrical quantity.		<u>Process variable value:</u> 70 °C.	
3: An analog-to-digital converter (ADC) converts the useful electrical quan- tity to information.	-	Characteristics	
4: A buffer and interface store, format, and present the information to a computer.	<b>a</b>	It is impossible to know the complete process state (because of infinite detail).	
5: An <i>interface routine</i> reads the information, converts it to the desired	1	It is impossible to know the exact value of a process variable.	
form, and stores it in the desired place.		A process variable value, however, can be determined to a high degree of precision.	
02-1 EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from isli02.	02-1	02-2 EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isil02. 02-	-2
02-3	02-3	02-4 02-	-4
Dimensions		Algebraic Manipulation of Dimensions	
Basics		In expressions, dimensions are manipulated in the same way as numbers and variables.	
A process variable's value is usually expressed as the product of a number and a dimension.	r	For example:	
For example, let process variable $T$ be the temperature of water in a coffee maker carafe.	a	$\frac{3 \mathrm{km}}{5 \mathrm{MPH}} = \frac{3 \mathrm{km} \mathrm{hr}}{5 \mathrm{Mi}} = \frac{3 \mathrm{km} \mathrm{hr}}{5 \mathrm{Mi}} \frac{\mathrm{Mi}}{1.6 \mathrm{km}} = \frac{3}{8} \mathrm{hr}.$	
Then a value for $T$ might be 60 °C.		Graphs of Values	
An equivalent value might be $T = 333.15 \mathrm{K}.$		Axes will be labeled with a symbol divided by a dimension.	
Notation		For example, $x/V$ or $R/k\Omega$ .	
Dimensions will be written in Roman (upright) type. For example, mA, V, and m.		The numbers on the axis are then dimensionless.	
Symbols representing values (variables) will be written in italic type: T, x, and $R$ .			
Thus, $3V\mathrm{V}$ means "three vee volts."			

02-4

Transtations			hempto	
Transducer:			A variable resistor can be used as a transducer.	
Device which converts a physical quantity from one form to an- other.			Consider a variable resistor which consists of a slider which can move over a distance of $15\mathrm{mm}$ while resistance varies linearly from 0 to	
Usually from a physical quantity which is a process variable to			$10 \mathrm{k}\Omega$ .	
some useful electrical quantity.			Process variable: position of slider, $x$ . 10 kO	
For example, a transducer might convert temperature to resistance.			Mapping: $H_t(x) = x \frac{10 \mathrm{k}\Omega}{15 \mathrm{mm}}.$	
Transducer Modeling			Process variable value approximated from transducer output.	
Mapping (function) from process variable to electrical quantity.			Let $y = H_t(x)$ where $H_t$ and $x$ are as above.	
Symbol $H_{\rm t}$ denotes the function.			Quantity $y$ is a resistance.	
Let $x$ be a process variable.			The position $x$ is found by inverting $H_t$ :	
Then $H_t(x)$ is the output of the transducer with function $H_t$ .			$H_t^{-1}(y) = y \frac{15\mathrm{mm}}{10\mathrm{k}\Omega}$	
			(In this case the process variable is not approximated.)	
			The process of finding the inverse is equivalent to solving for x in the equation $y = x \frac{10 \text{ k}\Omega}{15 \text{ mm}}$ .	
02-5 EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isli02.	02-5	02-6	EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isli02.	0
02-7	02-7	02-8		c
	02-7	02-0	Anglanta Divital Commission	U
Conditioning Circuits			Analog to Digital Conversion	
Purpose			Conditioning-circuit output is usually fed to an <i>ADC</i> .	
The output of a transducer is a raw electrical quantity.			An analog-to-digital converter converts electrical quantities to informa- tion quantities.	
It might have to amplified or otherwise processed.			Input is usually a voltage, output is usually a binary number.	
This is done by conditioning circuits.			Symbol $H_{ADC}(v)$ will be used for an ADC function.	
Conditioning circuits might have to do one or more of the following: • Amplify a tiny voltage.		St	andard ADC Function	
<ul><li>Ampiny a tiny voltage.</li><li>Convert resistance to voltage.</li></ul>			Since most ADCs will convert voltage to integers a standard function	
<ul> <li>Detect tiny changes in resistance (e.g., 100.1 to 100.2Ω).</li> </ul>			will be used.:	
<ul> <li>Add an offset to the transducer output.</li> </ul>			$H_{\text{ADC}(h,b)}(v) = \left\lfloor \frac{v}{h} (2^b - 1)  ight floor,$	
• Correct for nonlinearities in the transducer function.			where $h$ is a voltage and $b$ is an integer.	
• Other functions.			This ADC would convert voltages in the range 0 to $h$ (inclusive) to a binary number from 0 to $2^{b} - 1$ .	
Notation			For example, $H_{ADC(10V,8)}(5V) = 127$	
The symbol $H_c$ will be used for the conditioning circuit's function.			and $H_{\text{ADC}(17\text{ V},16)}(1.3\text{ V}) = 5011$ .	
An amplifier is a simple conditioning circuit: $H_{c}(v) = Av$ , where A, the gain, is a dimensionless number.				
For example, if x is a process variable, then $H_t(x)$ is the transducer output and $H_c(H_t(x))$ is the conditioning-circuit output.				
Sensors				
The combination of transducer and conditioning circuit is referred to as a <i>sensor</i> .				
02-7 EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from bil02.	02-7	02-8	EE 4770 Lecture Transparency. Formatted 17.36, 23 January 1998 from bil02.	c

02-5

02-6

Example

02-5

Transducers

02-8

02-6

02-8

	Two reasons for buffering the value of a process variable:			value t
	The value of the variable <i>at a particular time</i> is needed. (The value is buffered at that time.)		Т	The follo memor
	The value of a variable is only valid at certain times. (The value is buffered when it is valid.)		1:	: The int raw=
	The buffer itself can be a simple flip-flop, a register, a RAM, etc.		2:	The int value
	Usually, the contents of the buffer will be read by a computer through an <i>interface</i> .		3:	: The res
	The interface presents the buffered data to the computer in some stan- dard form.			theM The fur
	The computer is running some RT program. The RT program has one or more <i>interface routines</i> .			or m • Conv
	The interface could tell the RT program that data is available by making an <i>interrupt request</i> .			usua • Corr
	07			• Conv
	An interface routine could read the buffer without being alerted by			crons
	an external signal.			In town
	For example, it might read the buffer every millisecond.			In term
	Sampling is the process of reading a process variable at regular intervals.			
				Once w out v
02-9	EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isli02.	02-9	02-10	
02-11		02-11	02-12	
	The Conditioning Problem			I
	Archetypical Problem:			
	Design a system to write variable <b>procVar</b> with $H(x)$ , the value of in, where process variable x is, x can take values in the range $[x_{\min}, x_{\max}]$ and where $H(x) = \cdots$ .			Jse poter Construc
	Example Problem:		Т	Three lea
	Design a system to write variable waterLevel with $H(x)$ , an integer giving the water level in meters, where process variable x is the water level in room 2161 CEBA, x can take on values in the range		R	Resistance where

02-9

## Sampling, Buffering, and Interfacing

Buffering is the short-term storing information.

02-9

02-10

Interface Routine

- Here we will consider the ultimate destination of a process-variable value to be a memory location. wing is done to transfer a value from the interface to the ry location: terface routine makes a system call to read the interface. =readInterface();
  - terface routine, or some other code, applies a function,  $H_{\rm f}$  to the e read.
  - sult is written into the memory location.

femoryLocation=HsubF(raw);

nction  $H_{\rm f}$  puts the value into the final form. It may perform one nore of the following operations:

- vert the raw value to a floating-point quantity. (ADC output is ally an integer.)
- ect for any nonlinearities in the transducer or conversion circuit.
- vert the quantity to the desired dimensions. (E.g., meters, mis.)

is of the process variable, the final value written is:

## $H_{\rm f}(H_{\rm ADC}(H_{\rm c}(H_{\rm t}(x))))$

ritten, the value is read by the parts of the RT system that figure what's going on.

02-9	02-10	EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isli02.	02-10
------	-------	---	-------

## Example Problem Selection of The Transducer

ntiometer:

tion: shaft that can rotate 6 radians.

ads.

Resistance between center and lower lead,  $\frac{\theta}{6} 100 \,\mathrm{k\Omega}$ , where  $\theta \in [0, 6]$  is the shaft angle.

Resistance between center and upper lead,  $100 \text{ k}\Omega - \frac{\theta}{6} 100 \text{ k}\Omega$ .

Transfer function for center and lower lead,  $H_{\rm vr}(\theta) = \frac{\theta}{6} 100 \,\mathrm{k}\Omega$ .

Use of Potentiometer

Floats, guides, and cables will convert water level to shaft rotation.

These constructed so that  $\theta = x \frac{6}{1m}$ 

	This will convert resistance to voltage.
4:	Design the buffer and interface.
	Details for this part will be skipped here.

[0 m, 1 m] and where  $H(x) = \frac{5x}{\text{m}}$ 

Solution Overview

 ${\tt 2:}$  Choose an ADC.

1: Choose a transducer.

3: Design a conversion circuit.

5: Write the function for computing the final value. Easy, but tedious.

A variable resistor connected to a float with cables.

Suppose an ADC with function  $H_{ADC(5V,8)}$  is available.

02-11

 $H_{\rm t}(x) = \frac{x}{\rm m} 100 \,\rm k\Omega.$ 

02-12

02-12

0.110.110.210.240.240.24The experime former of events at well at 101 at 100		
The corput of the measurements we fix and the fixed of the corpus of 0 and 0	02-13 02-	3 02-14 02-14
<ul> <li>(q) (q) = (q) (q) (q) (q) (q) (q) (q) (q) (q) (q)</li></ul>	The Conversion Circuit	ADC, Buffering, and Final Processing
Reconserve quenchange, the input to the ADC mass lie in the mage of 0 SV. Reconserve quenchange, the input to the ADC mass lie in the mage of 0 SV. A subject of provention contract and the input to the ADC mass lie in the mage of 0 SV. A subject of provention contract and the input to the ADC mass lie in the mage of 0 SV. A subject of provention contract and the input to the ADC mass lie in the mage of 0 SV. A subject of provention contract and the input to the ADC mass lie in the mage of 0 SV. A subject of provention contract from residence is ordered.	The output of the conversion circuit will be:	The ADC function is fixed at
$1 = \frac{1}{10^{10}} = \frac{1}{10^$	$H_c(H_{\rm t}(x))=H_c(\frac{x}{\rm m}100{\rm k}\Omega).$	$H_{\text{ADC}(5\text{V},8)}(v) = \left\lfloor \frac{v}{5\text{V}}(2^8 - 1) \right\rfloor = \left\lfloor \frac{v}{5\text{V}}255 \right\rfloor.$
computer fixing a minimize. Consider the second later in the seco		$H_{\text{ADC}(5\text{ V},8)}(H_{\text{c}}(H_{\text{t}}(x))) = \left\lfloor \frac{(x/\text{ m})5\text{ V}}{5\text{ V}}255 \right\rfloor = \left\lfloor \frac{x}{\text{m}}255 \right\rfloor.$
A variety of conversion drame under leased. The singular is a linear conversion from existance to voltage. $\mu(H_{2}) = \frac{H_{1}}{(1000000000000000000000000000000000$		
The simplest is a linear conversion from resistance to voltage. $ \begin{aligned} \mu_{c}(t_{0}) &= \int_{0.0147}^{\infty} 3\sqrt{.} \\ \mu_{c}(t_{1}(x)) &= (x', m) 1005 \\ \hline 22.13 & 202.14 & 202.14 \\ \hline 22.14 & 202.14 & 202.14 \\ \hline 22.15 & 202.15 \\ \hline Flualty, the interface routine converts the raw form into the desired form: H(x) = \frac{5x}{m}.H_{c}(t_{1}(x_{c})) \approx y_{c}(t_{1}(t_{c}(x_{c}))) = H(x) \\ H_{c}(t_{1}(x_{c})) \approx y_{c}(t_{c}(t_{c}(x_{c}))) = H(x) \\ H_{c}(t_{1}(x_{c})) = H(y) $		Details of these parts will be covered later in the semester.
$H_{n}(R) = \frac{H_{n}}{10 \text{ MMS}} \text{ SV.}$ $H_{n}(R_{1}(z)) = \frac{(x' - x)(20M}{10(0.01)} \text{ SV.}$ $H_{n}(R_{1}(z)) = (x' - x)(3N)$ $H_{n}(R_{1}(z)) = (x' - x)(3N)$ $(22.13  \text{ contribute the sense terms (10); it is an extent to the desired form: H(x) = \frac{5\pi}{10}. H_{n}(H_{n}(x) + x)(R_{n}(R_{n}(x))) = H(x) H_{n}((x' - x))(R_{n}(R_{n}(x))) = H(x) H_{n}((x' - x))(R_{n}(R_{n}(x))) = H(x) H_{n}((x' - x))(R_{n}(R_{n}(x))) = H(x) H_{n}((x' - x))(R_{n}(x) + H(x)) H_{n}((x' - x))(R_{n}(x) + H(x))$	A variety of conversion circuits could be used.	
$ \begin{aligned} & f(t)(t) = \frac{f(t)(1000011)}{1000011} S_V, \\ & f(t)(t) = (t/10)SV, \\ \end{aligned} $	The simplest is a linear conversion from resistance to voltage.	
$ \begin{aligned} & f(t)(t) = \frac{f(t)(1000011)}{1000011} S_V, \\ & f(t)(t) = (t/10)SV, \\ \end{aligned} $	$H_{\rm c}(R) = \frac{R}{1001R} 5 \mathrm{V}.$	
$H_{1}(H_{1}(x)) = (x/m)^{3} \overline{Y}.$ 22.13 2420 Lobert Transmisson (20.2 (Moreover, Neuron 10.2 (Moreover, Neuro		
22.13 2.2 (2.14) 2.2		
<b>02.15 02.15</b> Finally, the historic routine converts the raw form into the desired form: $H(x) = \frac{5x}{m}$ . $H_{\ell}(H_{ADC(5V,8)}(H_{\ell}(H_{\ell}(x)))) = H(x)$ $H_{\ell}(  x  m)255] = H(x)$ Define $y = g(x) = [(x/m)255]$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0, n, 1, m]$ . Then $H_{\ell}(g(x)) = H(x)$ $H_{\ell}(y) = H(y)^{-1}(y)$ $g = \frac{1}{2} \frac{m}{255}$ $g = \frac{1}{5} \frac{m}{7}$ . The code fragment in the RT program is then: int raw; double statestare1; raw=readInterface(); waterLevel=raw/51.0; and we're doue!	$H_{\rm c}(H_{\rm t}(x)) = (x/{{ m m}})5{{ m V}}.$	
<b>02.15 02.15</b> Finally, the historic routine converts the raw form into the desired form: $H(x) = \frac{5x}{m}$ . $H_{\ell}(H_{ADC(5V,8)}(H_{\ell}(H_{\ell}(x)))) = H(x)$ $H_{\ell}(  x  m)255] = H(x)$ Define $y = g(x) = [(x/m)255]$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0, n, 1, m]$ . Then $H_{\ell}(g(x)) = H(x)$ $H_{\ell}(y) = H(y)^{-1}(y)$ $g = \frac{1}{2} \frac{m}{255}$ $g = \frac{1}{5} \frac{m}{7}$ . The code fragment in the RT program is then: int raw; double statestare1; raw=readInterface(); waterLevel=raw/51.0; and we're doue!		
<b>02.15 02.15</b> Finally, the historic routine converts the raw form into the desired form: $H(x) = \frac{5x}{m}$ . $H_{\ell}(H_{ADC(5V,8)}(H_{\ell}(H_{\ell}(x)))) = H(x)$ $H_{\ell}(  x  m)255] = H(x)$ Define $y = g(x) = [(x/m)255]$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0, n, 1, m]$ . Then $H_{\ell}(g(x)) = H(x)$ $H_{\ell}(y) = H(y)^{-1}(y)$ $g = \frac{1}{2} \frac{m}{255}$ $g = \frac{1}{5} \frac{m}{7}$ . The code fragment in the RT program is then: int raw; double statestare1; raw=readInterface(); waterLevel=raw/51.0; and we're doue!		
<b>02.15 02.15</b> Finally, the historic routine converts the raw form into the desired form: $H(x) = \frac{5x}{m}$ . $H_{\ell}(H_{ADC(5V,8)}(H_{\ell}(H_{\ell}(x)))) = H(x)$ $H_{\ell}(  x  m)255] = H(x)$ Define $y = g(x) = [(x/m)255]$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0, n, 1, m]$ . Then $H_{\ell}(g(x)) = H(x)$ $H_{\ell}(y) = H(y)^{-1}(y)$ $g = \frac{1}{2} \frac{m}{255}$ $g = \frac{1}{5} \frac{m}{7}$ . The code fragment in the RT program is then: int raw; double statestare1; raw=readInterface(); waterLevel=raw/51.0; and we're doue!		
<b>02.15 02.15</b> Finally, the historic routine converts the raw form into the desired form: $H(x) = \frac{5x}{m}$ . $H_{\ell}(H_{ADC(5V,8)}(H_{\ell}(H_{\ell}(x)))) = H(x)$ $H_{\ell}(  x  m)255] = H(x)$ Define $y = g(x) = [(x/m)255]$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0, n, 1, m]$ . Then $H_{\ell}(g(x)) = H(x)$ $H_{\ell}(y) = H(y)^{-1}(y)$ $g = \frac{1}{2} \frac{m}{255}$ $g = \frac{1}{5} \frac{m}{7}$ . The code fragment in the RT program is then: int raw; double statestare1; raw=readInterface(); waterLevel=raw/51.0; and we're doue!		
Finally, the interface routine converts the raw form into the desired form: $H(x) = \frac{5x}{m}$ , $H_1(H_{ADC(3V,8)}(H_1(H_1(x)))) = H(x)$ $H_2(\lfloor (x/m)255 \rfloor) = H(x)$ Define $y = g(x) = \lfloor (x/m)255 \rfloor$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0 \text{ m}, 1 \text{ m}]$ . Then $H_1(y) = H(y^{-1}(y))$ $= H(y \frac{m}{255})$ $= \frac{5}{m} y \frac{m}{255}$ $= \frac{y}{10}$ . The code fragment in the RT program is then: int raw; double waterLevel; raw;eadInterface(); waterLevel. and we're done!	02-13 EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isil02. 02-	3 02-14 EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isli02. 02-14
form: $H(x) = \frac{5x}{m}$ . $H_{t}(H_{ADC(3V,8)}(H_{c}(H_{t}(x)))) = H(x)$ $H_{t}(\lfloor (x/m)255 \rfloor) = H(x)$ Define $y = g(x) = \lfloor (x/m)255 \rfloor$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0, n, 1m]$ . Then $H_{t}(g(x)) = H(x)$ $H_{t}(y) = H(g^{-1}(y))$ $= H(y \frac{m}{255})$ $= \frac{5}{m} y \frac{m}{255}$ $= \frac{5}{m} \cdot \frac{1}{2}$ The code fragment in the RT program is then: int raw; double waterlevel; raw=readInterface(); waterlevel=raw/51.0; and we're done!	02-15 02-	5
$\begin{split} & H_{t}(H_{ADC(S \setminus S)}(H_{c}(H_{t}(x)))) = H(x) \\ & H_{t}(\lfloor (x/m)255 \rfloor) = H(x) \end{split}$ Define $y = g(x) = \lfloor (x/m)255 \rfloor$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0, m, 1, m]$ . Then $\begin{aligned} & H_{t}(g(x)) = H(x) \\ & H_{t}(y) = H(g^{-1}(x)) \\ & H_{t}(y) \frac{m}{255} \\ & = \frac{5}{m} y \frac$		
$H_{t}([x/m)255]) = H(x)$ Define $y = g(x) = [(x/m)255]$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0 m, 1 m]$ . Then $H_{t}(g(x)) = H(x)$ $H_{t}(y) = H(g^{-1}(y))$ $= H(y \frac{m}{255})$ $= \frac{5}{m} y \frac{m}{255}$ The code fragment in the RT program is then: int raw; double waterlevel; raw=readInterface(); waterlevel=raw/51.0; and we're done!		
Define $y = g(x) = \lfloor (x/m)255 \rfloor$ . Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0 \text{ m}, 1 \text{ m}]$ . Then $\begin{aligned} H_t(g(x)) = H(x) \\ H_t(y) = H(g^{-1}(y)) \\ &= H(y \frac{m}{255}) \\ &= \frac{5}{m} y \frac{m}{255} \\ &= \frac{y}{51}. \end{aligned}$ The code fragment in the RT program is then: int raw; double waterlevel; int raw; double waterlevel; double waterlevel; int raw; double waterlev		
Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0 \text{ m}, 1 \text{ m}]$ . Then $H_t(g(x)) = H(x)$ $H_t(y) = H(g^{-1}(y))$ $= H(y \frac{m}{255})$ $= \frac{5}{m} y \frac{m}{255}$ $= \frac{y}{51}.$ The code fragment in the RT program is then: int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0; and we're done!		
Then $\begin{aligned} H_{t}(g(x)) &= H(x) \\ H_{t}(y) &= H(g^{-1}(y)) \\ &= H(y\frac{m}{255}) \\ &= \frac{y}{51} \\ \end{aligned} The code fragment in the RT program is then: int raw; double vaterLevel; raw=readInterface(); vaterLevel=raw/51.0; and we're done!$	Define $y = g(x) = \lfloor (x/m)255 \rfloor$ .	
Then $\begin{aligned} & H_{t}(g(x)) = H(x) \\ & H_{t}(y) = H(g^{-1}(y)) \\ & = H(y\frac{m}{255}) \\ & = \frac{y}{51}. \end{aligned}$ The code fragment in the RT program is then: int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0; and we're done!	Then $x = g^{-1}(y) \approx y \frac{m}{255}$ for $x \in [0 \text{ m}, 1 \text{ m}].$	
$H_{f}(y) = H(g^{-1}(y))$ $= H(y\frac{m}{255})$ $= \frac{5}{m}y\frac{m}{255}$ $= \frac{y}{51}$ The code fragment in the RT program is then: int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0; and we're done!		
$H_{f}(y) = H(g^{-1}(y))$ $= H(y\frac{m}{255})$ $= \frac{5}{m}y\frac{m}{255}$ $= \frac{y}{51}$ The code fragment in the RT program is then: int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0; and we're done!	$\Pi(-(-))$ $\Pi(-)$	
$= H(y\frac{m}{255})$ $= \frac{5}{m}y\frac{m}{255}$ $= \frac{y}{51}.$ The code fragment in the RT program is then: int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0; and we're done!	$H_{f}(y) = H(a^{-1}(y))$	
The code fragment in the RT program is then: int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0; and we're done!	$H_{\mathrm{f}}(y) = H(y - \frac{\mathrm{m}}{\mathrm{m}})$ $= H(y - \frac{\mathrm{m}}{\mathrm{m}})$	
The code fragment in the RT program is then: int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0; and we're done!	$= \frac{5}{255} \frac{m}{m}$	
The code fragment in the RT program is then: int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0; and we're done!	$=\frac{y^{9}255}{y}$	
<pre>int raw; double waterLevel; raw=readInterface(); waterLevel=raw/51.0;  and we're done!</pre>	51	
<pre>double waterLevel; raw=readInterface(); waterLevel=raw/51.0;  and we're done!</pre>	The code fragment in the RT program is then:	
raw=readInterface(); waterLevel=raw/51.0; and we're done!	int raw;	
waterLevel=raw/51.0; and we're done!	double waterLevel;	
and we're done!	<pre>raw=readInterface();</pre>	
	<pre>waterLevel=raw/51.0;</pre>	
02-15 EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isli02. 02-15	and we're done!	
	02-15 EE 4770 Lecture Transparency. Formatted 17:36, 23 January 1998 from Isli02. 02-7	5