Hands-on Lab 3:

LabVIEW – Virtual Touch Sensor and Voltmeter

Previously LabVIEW was shown to generate voltages. By connecting the NXT Brick's motor output ports, one could generate voltages. Here, the reverse is also possible. One connects to the Brick's sensor ports. This lab demonstrates that the LEGO NXT can detect events (like a pressed button) and voltages.

Concept 1: Virtual Touch Sensor (Analog Voltage Output)

Step 1: Create Front Panel

Open LabVIEW, File - Save All with nxtLabviewVirtualTouchSensor1.0.vi. In your front panel, add an LED and STOP button as seen in Figure 1A (left). Create the associated block diagram: add a while-loop structure and wire the STOP button to terminate the loop. It's also good practice to comment your code as seen in Figure 1A (right).



Figure 1A: LED and STOP Boolean controls in Front panel (left). While-loop structure in block diagram

Step 2: Add Touch Sensor to Block Diagram

In the block diagram, go to the LEGO sensors control window as shown in Figure 2A (left). Recall, this is done by selecting View - Functions Palette - MINDSTORMS Robotics - I/O. Click and drag the Sensor control into the block diagram as shown in Figure 2A (right).



Figure 2A: MINDSTORMS Robotics – I/O dialog box shows the Sensor control (left). Click and drag the Sensor control into the block diagram (left).

Next, add a Numeric Constant to the block diagram. Recall, this is done by from the Functions palette where one selects MINDSTORMS Robotics - Programming - Numeric. Wire this numeric constant to the Sensor control's port input. One can view a control's terminals by selecting your wire tool and hovering over areas of the control. By default this numeric control's value is 0. Lastly, wire the Sensor control's Yes/No output port to the LED control's input. Your block diagram should look like **Figure 2B**.



Figure 2B: Final Block Diagram – a numeric constant (default value is 0) is added and wired to the Touch sensor block.

Step 3: Breadboard the Touch Sensor

Recall in the previous lab, an NXT breadboard adapter facilitates access to ports (see figure below for reference).

Pin	Name/Color	Input Role	Output Role
1	ANA (WHITE)	Analog interface	Motor Power1
2	GND (BLACK)	Ground	Motor Power2
3	GND (RED)	Ground	Ground
4	PWR (GREEN)	+4.3V supply	+4.3V supply
5	DIGI0 (YELLOW)	12C clock	Encoder Signal 1
6	DIGI1 (BLUE)	I2c data	Encoder Signal 2
	A CONTRACTOR CONTRACTOR AND A CONTRACTOR A		



A standard NXT cable consists of 6 colored wires. When this cable is plugged into the brick's output port, and then each wire has the function given in the right-most column.

Plug a NXT cable to (sensor) Port 1 (see **Figure 3 left**). Next, add hookup wires to the NXT breadboard adapter's ANA and GND pins as in **Figure 3 right**.



Figure 1N: Breadboard connections (left) and NXT adapter close up (right)

From the Front Panel, run your program. When the hookup wires are shorted then the LED should glow green. When the wires are disconnected, then the LED should stop glowing. Click the STOP button to terminate. Essentially, you've created a virtual touch sensor.

Exercise 1: In LabVIEW create programs for the following:

- 1-1.Connect the NXT cable to Port 2.Change nxtLabview-virtualTouchSensor1_0.vi so that the LED glows green when the ANA and GND wires are shorted
- 1-2. Write a new program which displays a count of how many times the ANA and GND wires are shorted

Concept 2: Virtual Voltmeter (analog voltage input)

Step 1: Create the front panel seen in Figure 1A left

The Meter control can be found from View - Controls Palette - MINDSTORMS Robotics - Numeric. The Meter control resembles an analog display. Right click on the meter and select Visible - Digital Display. This will allow one to view the meter's numeric value. Next add a numeric indicator control and label it as "Digital Voltmeter". This will just provide another display of any voltages read by the NXT.



Figure 1A: Front panel has analog and digital indicators (left). While loop surrounds these two controls (right)

Step 2: Create the Block Diagram

Create a while-loop around the controls in the block diagram (Figure 1A right) and wire the STOP button to terminate the loop.

Next, from MINSTORMS Robotics - I/O - Advanced, find the Setup Sensor and Read Generic Sensor controls (Figure 2A). Click and drag the Setup Sensor control to the left of the while loop. Click and drag the Generic Sensor control into the while loop. These two controls are used to configure and initialize the sensor. Wire these two controls via their NXT/EV3 terminals as seen in Figure 2B.





Figure 2A: Setup Sensor and Read Generic Sensor controls

Figure 2B: Setup Sensor and Read Generic Sensor wired thru the while-loop

Next, create 2 numeric constant controls. Wire each constant control to the Port terminal of the Setup Sensor and Read Generic Sensor controls (see Figure 2C).



Figure 2C: Numeric constant controls each wired into the top of the Setup Sensor and Read Generic Sensor controls. 0 represents Port 1 on the NXT Brick

LEGO NXT LabVIEW

To configure the sensor hover your mouse of the Setup Sensor control. Right click and select Create - Constant. This will result in the block diagram shown in Figure 2D left. By default, the constant has 2 components: NO_SENSOR and RAWMODE. Hover your mouse over the NO_SENSOR, right click and choose Properties. A pop up box will display. Select the Edit Items tab. Here, one sees various possible sensor types. Figure 2D right below shows that SWITCH has the numeric value 1.

📴 nxtLabviewVirtualVoltmeter1_0.ni.vi Block Diag	ram*	🗍 🚺 🖪 R	ling Constant	Properties: Se	nsorType		×
File Edit View Operate Tools Window	Help		ppearance	Dala Type	Display Format	Edit Items	Documentation
	15pt Application Font 🛛 🚛 🖓 🖓 🖓 🎾	2	Sequential v	values		1	
// FIE: mtl.abview/irtual/oltmete // DATE: 07.18/18.16:54 // AUTH: P.Oh // DESC: Uses NXT adapter. Displa Must use constants to de [0, 1, 2, 3] = Ports 1, 2, 3, 4	r1_0.vi - WORKSI y voltage across ANA and GND pins ine ports as follows: . Currently set at Port 3 i.e. constant = 2		Ilems NO_SENSOR SWITCH TEMPERATUF REFLECTION ANGLE LIGHT_ACTIV	RE /E	Values 0 1 2 3 4 5		Insert Delete Move Up Move Down
	Analog Voltmeter [V] Digital Voltmeter [V] Digital Voltmeter [V] Quit Store P Trr Digital Voltmeter [V] Quit Store P Trr Digital Voltmeter [V] Digital V		Allow unde	fined values at	o t run time		Disable Item
Main Application Instance <						ОК	Cancel Help

Figure 2D: Right clicking the lower left input of the Setup Sensor control allows one to create a constant (left) with default values for NO_SWITCH and RAWMODE. Hovering over NO_SWITCH, right-clicking and selecting Properties and Edit reveals that SWITCH has a value of 1 (right).

Hover your mouse over RAWMODE, right click and choose Properties. The pop up box will display and select the Edit Items tab. Here, one sees that RAWMODE has a value of 0 (Figure 2E).

Sequential values Items Values RAWMODE 0 BOOLEANMODE 32 TRANSITIONCNTIMODE 64 PERIOD COUNTERMODE 96 PCTFULL SCALEMO DE 128 CELSIUSMODE 160 FAHRENHEITMODE 192 Allow undefined values at run time	Appearance	Data Type	Display Format	Edit Items		ocumentation
Items Values A RAWMODE 0 BOOLEANMODE 32 TRANSITIONCNTMODE 64 PERIOD COUNTERMODE 96 PCTFULLSCALEMODE 128 CELSIUSMODE 160 FAHRENHEITMODE 192 Allow undefined values at run time	Sequential v	alues				
RAWMODE 0 BOOLEANMODE 32 TRANSITIONCNTIMODE 64 PERIODC OUNTERMODE 96 PCTFULLSCALEMODE 128 CELSIUSMODE 160 FAHRENHEITMODE 192 Allow undefined values at run time	Items		Values		*	Insert
BOOLEANMODE 32 TRANSITIONCNTMODE 64 PERIODCOUNTERMODE 96 PCTFULSCALEMODE 128 CELSIUSMODE 160 FAHRENHEITMODE 192 Allow undefined values at run time	RAWMODE		0			
TRANSITIONCNTMODE 64 F PERIODCOUNTERMODE 96 P PCTFULISCALEMODE 128 Move Up CELSIUSMODE 160 P FAHRENHEITMODE 192 T	BOOLEANMO	DE	32			Delete
PERIODCOUNTERMODE 96 PCTFULISCALEMODE 128 CELSUSMODE 160 FAHRENHEITMODE 192 Allow undefined values at run time	TRANSITION	ONTMODE	64		=	
PCTFULLSCALEMODE 128 CELSIUSMODE 160 FAHRENHEITMODE 192 Allow undefined values at run time	PERIODCOUN	ITERMODE	96			Move Up
CELSIUSMODE 160 FAHRENHEITMODE 192 Control Disable Item Allow undefined values at run time	DOTELLU SOA	EMODE	128			Move Down
FAHRENHEITMODE 192	FUTIOLESCA	CCINIO D C				
Allow undefined values at run time	CELSIUSMOD	E	160		-	INIOVE DOWN
	CELSIUSMOD FAHRENHEIT	E MODE fined values a	160 192 at run time		~	Disable Item

Figure 2E: Similarly to Figure 2D, hovering over RAWMODE in the block diagram and right clicking Properties and selecting the Edit Items tab shows RAWMODE has a value of 0.

The NXT's input Ports contain a 10-bit analog-to-digital converter (ADC). Thus, when the port receives a voltage input, it is converted into a digital value from 0 to 1023. Furthermore, each input port is configured as a voltage divider with a 5 Volt source. Thus, when a voltage enters the port, it is scaled by 5 Volts. Add a Multiply and Divide controls to your block diagram and wire with numeric constants as seen in **Figure 2F**.



Figure 2F: Shows the Multiply and Divide controls within the While-loop. Constants have been added to these controls to properly convert the value from Read Generic Sensor into a voltage value.

To simplify programming, create a Cluster constant. Here choose MINDSTORMS Programming - Cluster - Cluster Constant (Figure 2H right) and drag into the block diagram (Figure 2H left)



Figure 2H: Click and drag a cluster constant (right) into the block diagram (left)

Drag 2 numeric constants in the Cluster constant. Assign 1 to the first numeric constant and 0 to the second one. These are respectively for SWITCH and RAWMODE (as determined from Figures 2D left and 2E previously). Now one can delete the original Cluster constant and replace it with the one you just created (see Figure 2I).

🖪 nxtLabviewVoltmeter1_0.vi Block Diagra	im *	
File Edit View Operate Tools Win	idow Help * □ 15pt Application Font ▼ 🚛 ▼ 🚋 ♥ 缺 🔩 🔍 🄊	2
// FILE: nxtLabviewVoltmeterI_0.vi - W // DATE: 07/21/1817:47 // AUTH: P.Oh // DESC: Uses NXT adapter. Display vc	ORKS!	
// Must use constants to define ports a // [0, 1, 2, 3] = Ports 1, 2, 3, 4. Currenti // REFS: based on notes in nxtLabview // NOTES: nxtLabviewVoltmeter1_0.vi // constants. SWITCH should be1 and	as follows: y set at Port 3 i.e. constant = 2 VirtualVoltmeter1_5.vi creates a Cluster constant with 2 numeric I RAWMODE should be 0	
	Analog Voltmeter [V]	
I Switch = 1 I Rawmode -= 0		
Main Application Instance 4	III. F	

Figure 2I: Completed block diagram with newly created cluster constant (brown box). The cluster contains two numeric constants. The values of these constants represent values for setting the sensor as a switch with raw mode data output.

Step 3: Breadboard the Voltmeter Sensor

Here, one can use the Touch sensor breadboard (Concept 1 and Figure 1N). Here, connect the ANA and GND wires, respectively, to a battery's +'ve and -'ve ends. The front panel's analog meter's needle should move the measured voltage and the numeric indicator should display the digital value.

Exercise 2: In Labview create programs to:

- 3-1.By default the analog meter gives a range of 0 to 10. Change this to 0 to 5V and rerun your program.
- 3-2. Replace the analog meter with a gauge control
- 3-3. Make Port 3 the Brick act as the voltmeter