NI myRIO

Project Essentials Guide

By Ed Doering

POWERED BY NATIONAL INSTRUMENTS LabVIEW

NATIONAL INSTRUMENTS™

NI myRIO Project Essentials Guide

Ed Doering
Electrical and Computer Engineering Department
Rose-Hulman Institute of Technology

PRELIMINARY DRAFT
September 20, 2013

This is a **preliminary draft version** printed on September 20, 2013. Download the up-to-date version at `http://www.ni.com/myrio/project-guide`.

Contents

1 Introduction

Welcome to the *NI myRIO Project Essentials Guide*, and welcome to the exciting world of NI myRIO! This book will serve as your guide to interfacing NI myRIO to the wide variety of sensors, actuators, and displays that you will need for your projects. Each chapter concentrates on a specific component or device, using a mixture of text and video to guide you through the learning process necessary to successfully integrate the component or device into your system.

For example, consider the photocell of Chapter 9 on page 41, one of the many parts included with the NI myRIO Starter Kit. You'll get immediate hands-on experience with the photocell as you build a simple demonstration circuit, connect it to one of the NI myRIO expansion ports, and run a provided LabVIEW program to confirm that the photocell works properly. After that, study a video tutorial that explains photocell operating principles and learn how to design a suitable interface circuit. Next, try your hand with the suggested modifications to the demonstration LabVIEW code to deepen your understanding of the photocell behavior and LabVIEW programming techniques. At last you will be ready to tackle the suggested systems integration project in which you combine the photocell with other devices.

This book is intended for students at the junior or senior year, especially those students engaged in capstone projects or research. A background in electrical and computer engineering is ideal, but students pursuing other disciplines will find the level of tutorial detail to be more than adequate.

This document is fully hyperlinked for section and figure references, and all video links are live hyperlinks. Open the PDF version of this document for the most efficient way to access all of the links; click a video hyperlink to automatically launch the video in your browser. Within the PDF, use `ALT+leftarrow` to navigate back to a starting point.

> **NOTE**: *This book is a preliminary draft edition. Items indicated by "[TBD]" indicate "to be developed." Visit* `http://www.ni.com/myrio/project-guide` *to obtain the latest edition.*

Part I

myRIO Starter Kit

2 Discrete LED

LEDs, or light-emitting diodes, provide simple yet essential visual indicators for system status and error conditions. Figure 2.1 shows the four types of LEDs included in the NI myRIO Starter Kit.

Figure 2.1: NI myRIO Starter Kit discrete LED s; from left to right: standard red and green, high-efficiency in various colors, and RGB.

Learning Objectives: You will understand these core concepts related to the discrete LED after completing the activities in this chapter:

1. LED permits only one-way current,
2. Forward-bias voltage drop varies depending on color (wavelength),
3. Interface circuit depends on knowledge of DIO output resistance and source voltage, and
4. LEDs may be direct-connected to the DIO under some circumstances.

2.1 Component Demonstration

Follow these steps to demonstrate correct operation of the discrete LED component.

Select these parts from the NI myRIO Starter Kit:

- Resistor, 220 Ω `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- "Basic Red" LED
- Breadboard
- Jumper wires, M-F ($2\times$)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 2.2 on the following page. The discrete LED interface circuit requires two connections to NI myRIO MXP Connector B:

1. Anode $\rightarrow$ B/+3.3V (pin 33)
2. LED control $\rightarrow$ B/DIO0 (pin 11)

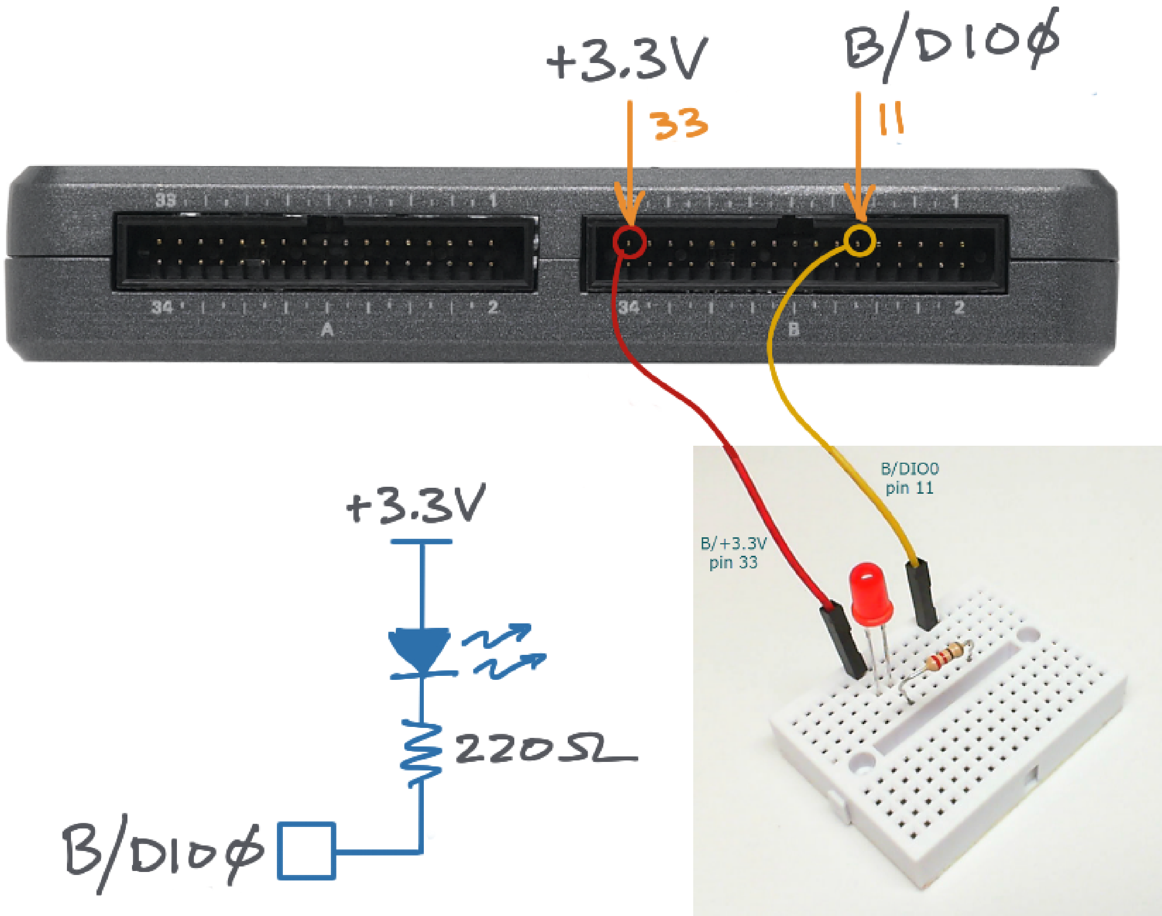


Figure 2.2: Demonstration circuit for discrete LED: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Discrete LED demo.lvproj` contained in the subfolder `Discrete LED demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: Your discrete LED should be blinking on and off in synchronism with the front-panel indicator digital output state. Click the enable blinker front-panel button to disable blinking and to enable the digital level button; click this button to manually set the digital output state either high or low. Because this interface circuit is the *sinking current* form (explained in the next section), the LED is active when the digital output is in the *low* state, i.e., this is an active-low LED interface circuit.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct LED orientation — the diode conducts current in one direction only; remove the LED an reinsert in the opposite direction, and
- Correct resistor value — use an ohmmeter to verify that the resistance is 220 ohms.

2.2 Interface Theory

Interface circuit: Study the video "Discrete LED Interfacing Theory" at `http://youtu.be/9-RlGPVgFW0` to learn the basics of LEDs and the two types of interface circuits (current-sinking and current-sourcing). Also study "LED Current Management" at `http://youtu.be/JW-19uXrWNU` to learn about the voltage-current characteristics of the various types of diodes included in the NI myRIO Starter Kit (standard, superbright, and RGB), learn principles of operation of the LED interface circuit including the current-sinking and current-sourcing forms, and learn how to choose the size of the current-limiting resistor.

LabVIEW programming: Study the video "Digital Output Express VI" at `http://youtu.be/Y8mKdsMAqrU` to learn how to access all of the available digital outputs with the NI myRIO Digital Output Express VI, including single output, multiple outputs, and choice of connector.

2.3 Basic Modifications

Study the video "LED Demo Walk-Through" at `http://youtu.be/SHJ-vu4jorU` to learn the design principles of `Discrete LED demo`, and then try making these modifications to the interface circuit and to `Main.vi`:

- Add a front-panel control to adjust the blink frequency specified in Hertz; at what frequency does the blinking become imperceptible?
- Blink two adjacent LEDs to simulate a railroad crossing signal.
- Blink the green and blue LEDs of the RGB LED using the same LabVIEW code as the railroad crossing signal; refer to Figure 2.3 on the next page for the RGB LED pinout diagram. Use the current-sourcing interface circuit.
- Create an LED variable-intensity dimmer with the |PWM Express VI to create a pulse-width modulated digital output. Set the Frequency to the default constant value of 1000 Hz and create a pointer slide control to adjust the duty cycle; select the logarithmic mapping option for the control, too. Add some code to account for the active-low current-sinking LED interface (bonus

points with a Boolean control to select between current-sinking and current-sourcing interfaces).

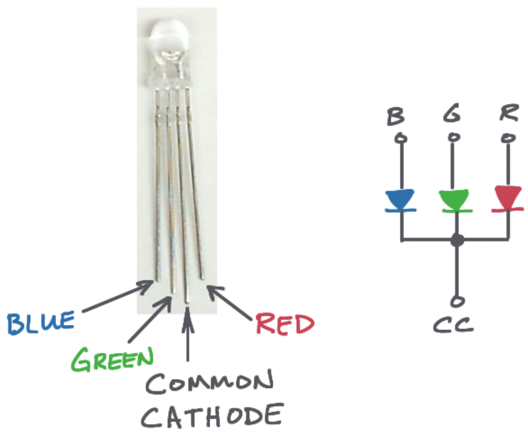


Figure 2.3: RGB LED pins and schematic diagram.

2.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the discrete LED with other components and devices.

2.5 For More Information

- *LED Mixed Bag (5mm)* by SparkFun – need more LEDs? The LED Mixed Bag from SparkFun offers the same type as those in the NI myRIO Starter Kit; also search SparkFun for many other sizes and types of LEDs:
  `http://www.sparkfun.com/products/9881`
- *Engineering Thursday: LED Light Boxes* by SparkFun – glowing multi-color boxes as household art:
  `http://www.sparkfun.com/news/1210`
- *Use LEDs as photodiodes* by EDN – LEDs can *detect* light, too, making them an interesting type of photosensor:
  `http://www.edn.com/design/led/4363842/Use-LEDs-as-photodiodes`
- *LED Lighting Applications* by OSRAM Opto Semiconductors – LEDs are everywhere these days, including street and outdoor lighting, architectural illumination, downlights (i.e., ceiling lights), flashlights, and greenhouses:
  `http://ledlight.osram-os.com/applications/`

3 Seven-Segment LED Display

Displays based on seven LED segments arranged in an "8" pattern provide a simple means to display numbers 0 to 9 and some letters of the alphabet. Figure 3.1 shows the NI myRIO Starter Kit seven-segment display.

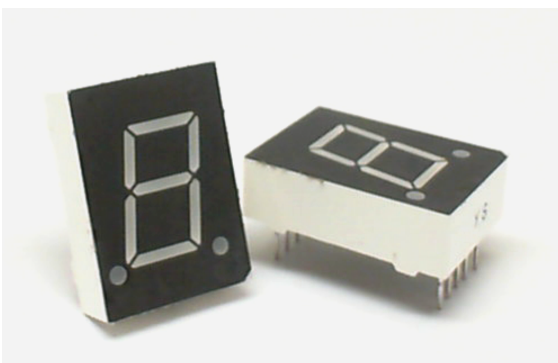


Figure 3.1: NI myRIO Starter Kit seven-segment display.

Learning Objectives: In this module you will create a standard interface circuit to verify correct operation of the seven-segment LED display, learn LabVIEW programming techniques for multiple digital outputs, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the seven-segment LED with additional components or devices.

3.1 Component Demonstration

Follow these steps to demonstrate correct operation of the seven-segment LED component.

Select these parts:

- Seven-segment LED display `[TBD]`
- Breadboard
- Jumper wires, M-F (9×)

Build the interface circuit: Refer to the pin diagram and recommended breadboard layout shown in Figure 3.2 on the next page. The interface circuit requires nine connections to NI myRIO MXP Connector B:

1. Common anode (CA) → B/+3.3V (pin 33)
2. Segment *a* → B/DIO0 (pin 11)
3. Segment *b* → B/DIO1 (pin 13)
4. Segment *c* → B/DIO2 (pin 15)
5. Segment *d* → B/DIO3 (pin 17)
6. Segment *e* → B/DIO4 (pin 19)
7. Segment *f* → B/DIO5 (pin 21)
8. Segment *g* → B/DIO6 (pin 23)
9. Decimal point → B/DIO7 (pin 25)

> **TIP**: *Use the resistor color code for the DIO wire colors, e.g., black (0) for B/DIO0, brown (1) for B/DIO1, and so on.*

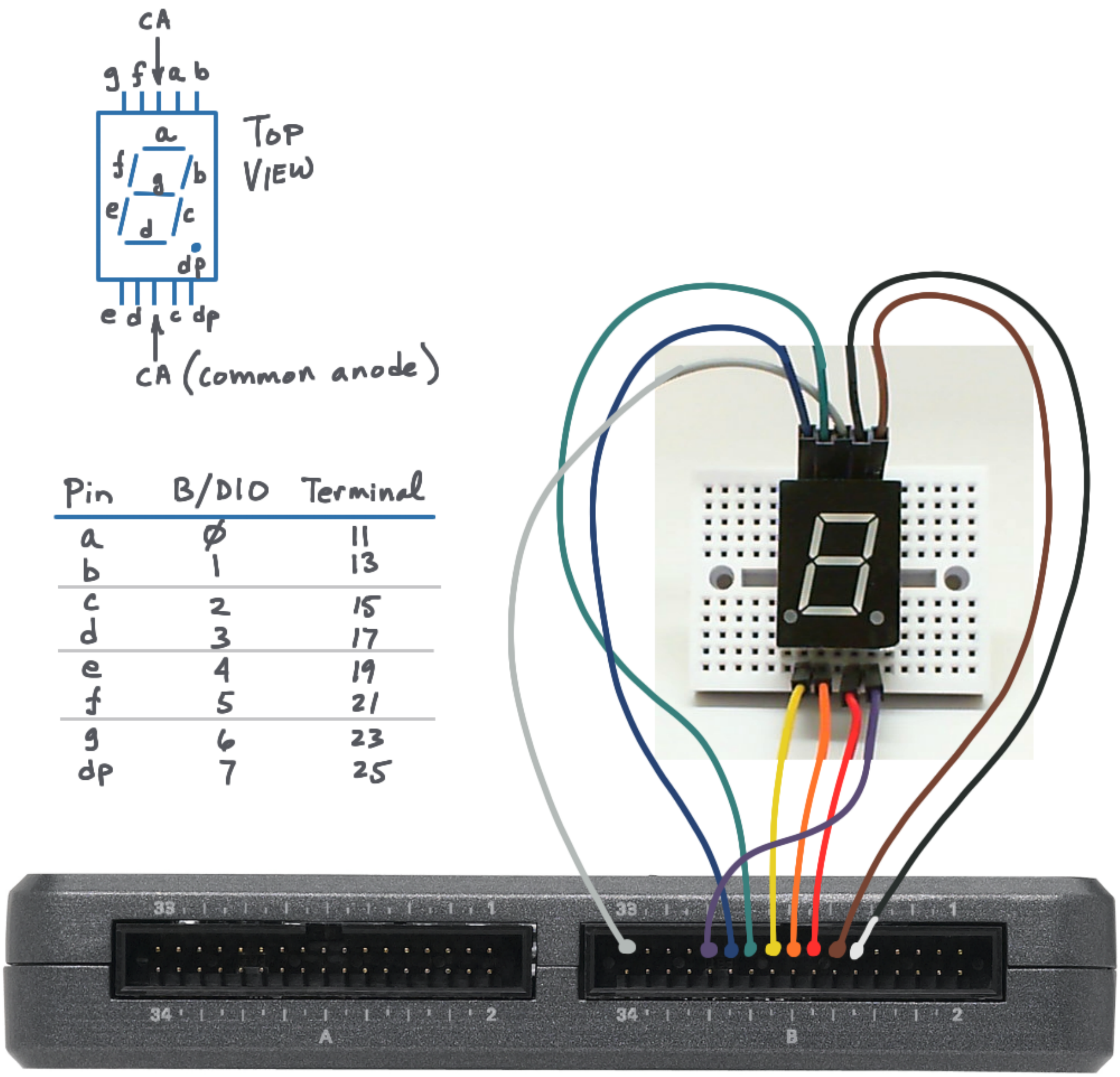


Figure 3.2: Demonstration circuit for seven-segment display: pin diagram, recommended breadboard layout, and connection to eight digital I/O terminals on NI myRIO MXP Connector B.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Seven-Segment LED demo.lvproj` contained in the subfolder `Seven-Segment LED demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically..

Expected results: Toggle the eight front-panel switches to activate and deactivate each segment *a* through *g* as well as the decimal point; refer again to Figure 3.2 on the facing page to see the standard labeling scheme for the segments. Activating the front-panel switch should cause the corresponding segment to light.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black "Run" button on the toolbar signifying that the VI is in run mode, and
- Correct wiring — ensure that your wiring order is correct and that you have started at terminal 11.

3.2 Interface Theory

Interface circuit: Each of the seven line segments as well as the decimal point is an individual LED, each with its own anode and cathode. To conserve the number of electrical connections on the display all of the eight anodes are tied together an brought out as the "common anode" pin. Study the video "Seven-Segment LED Interfacing Theory" at `http://youtu.be/9D60cn7OFoc` to learn about the voltage-current characteristics of the individual LED, why the segment controls are active-low, and why no current-limiting resistors are required for this particular device.

LabVIEW programming: Study the video "Digital Output Express VI" at `http://youtu.be/Y8mKdsMAqrU` to learn how to access all of the available digital outputs with the NI myRIO Digital Output Express VI, including single output, multiple outputs, and choice of connector. In addition, study the video "Digital Output Low-Level subVIs" at `http://youtu.be/[TBD]:doVI` to learn how to access and use the lower-level code created by the Express VI to connect Boolean arrays directly to the digital outputs, i.e., as a bus.

3.3 Basic Modifications

Study the video "Seven-Segment LED Demo Walk-Through" at `http://youtu.be/X1v_NjLxVqM` to learn the design principles of `7-segment LED demo.lvproj`, and then try making these modifications to `Main.vi`:

- Maintain the same behavior, but convert the Digital Output Express VI to its underlying code, and then connect the array-style front-panel control directly to the Write subVI from the NI myRIO Advanced I/O → Digital I/O subpalette. Also create a front-panel control to make user-selectable digital I/O channels.

- Display a numerical (integer) front-panel control value as its corresponding pattern on the seven-segment display; a case structure with Boolean array constants works nicely here (see `http://cnx.org/content/m14766/latest/?collection=col10440`). Display the values 0 to 9 and a dash for values greater than 9 (bonus points for a hexadecimal display to include values A to F).
- Create a rotating chase sequence in which a single active segment appears to move around the periphery of the display. Make the speed adjustable, and also include a control to reverse the direction of rotation. Consider a single Boolean array constant connected to Rotate 1D Array in the Programming → Array subpalette. TIP: Use Quick Drop (Ctrl+Space) to search for a programming element by name.

3.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the seven-segment display with other components and devices.

3.5 For More Information

- *7-Segment Display - LED (Blue)* by SparkFun – need more display digits? This is the same (or very similar) product:
  `http://www.sparkfun.com/products/9191`
- *World's Simplest Font* by Twyman Enterprises – TrueType font for seven-segment displays; use this font and your favorite wordprocessor to quickly translate your text phrases into suitable segment patterns:
  `http://www.twyman.org.uk/Fonts`
- *Nixie Tubes and Projects* by Neonixie – before seven segment displays you would have used *nixie tubes*; see `http://en.m.wikipedia.org/wiki/Nixie_tube` for the origin of "nixie." You can buy nixie tubes and related projects here:
  `http://neonixie.com/`

4 Pushbutton Switch

Pushbutton switches – also called momentary contact switches – serve as basic user-interface devices as well as simple sensors, e.g., bump sensors. Figure 4.1 pictures the pushbutton integrated with the rotary encoder covered in a later chapter.

Figure 4.1: NI myRIO Starter Kit pushbutton switch integrated with rotary encoder.

Learning Objectives: You will understand these core concepts related to the pushbutton switch after completing the activities in this chapter:

1. Pushbutton switch appears as a short circuit when pressed, otherwise as an open circuit,
2. Interface circuit to digital input relies on the DIO internal pull resistors to eliminate the need for additional components (pull-up on MXP Connectors A and B, pull-down on MSP Connector C),
3. Software views the switch as a Boolean (two-level) signal that is either active-high or active-low depending on the type of pull resistor, and
4. Software edge detection converts a pushbutton press to a trigger event.

4.1 Component Demonstration

Follow these steps to demonstrate correct operation of the pushbutton switch.

Select these parts from the NI myRIO Starter Kit:

- Pushbutton switch `[TBD]`
- Breadboard
- Jumper wires, M-F (2×)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 4.2 on page 19. The pushbutton switch interface circuit requires two connections to NI myRIO MXP Connector B:

1. Pushbutton Terminal 1 → B/DIO0 (pin 11), and
2. Pushbutton Terminal 2 → B/GND (pin 12).

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Pushbutton demo.lvproj` contained in the subfolder `Pushbutton demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the input state of three DIOs, one on each connector. The states of Connector A and B DIOs should be high due to the internal pull-up resistors, and the Connector C DIO should be low because of the internal pull-down resistor. Press the pushbutton and you should see the B/DIO0 state indicator change to low; release the pushbutton and the state should go high again.

The demo VI also counts pushbutton presses detected on B/DIO0. Stop and restart the VI to clear the counter value.

Disconnect Pushbutton Terminal 1 and reconnect to A/DIO0 (pin 11). Confirm that the A/DIO0 state indicator changes in response to pushbutton presses.

Disconnect Pushbutton Terminal 1 and reconnect to C/DIO0 (pin 11). Does the C/DIO0 state indicator change in response to a pushbutton press? Can you explain the behavior you observe?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

4.2 Interface Theory

Interface circuit: The pushbutton switch normally appears as an open circuit, and becomes a short circuit when pressed. The pushbutton may be connected directly to the digital input without any additional components because of the internal pull resistors on the NI myRIO DIO lines.

Study the video "Pushbutton Interfacing Theory" at `http://youtu.be/e7UcL5Ycpho` to learn about the DIO pull resistors and how to properly connect the pushbutton switch for pull-up resistors (MXP Connectors A and B) and pull-down resistors (MSP Connector C). Study "Detect a Switch Signal Transition" at `http://youtu.be/GYBmRJ_qMrE` to learn how to detect a switching signal transition inside a software while-loop structure.

LabVIEW programming: Study the video "Digital Input Express VI" at `http://youtu.be/[TBD]:diExVI` to learn how to use Digital Input Express VI to sense the state of the pushbutton.

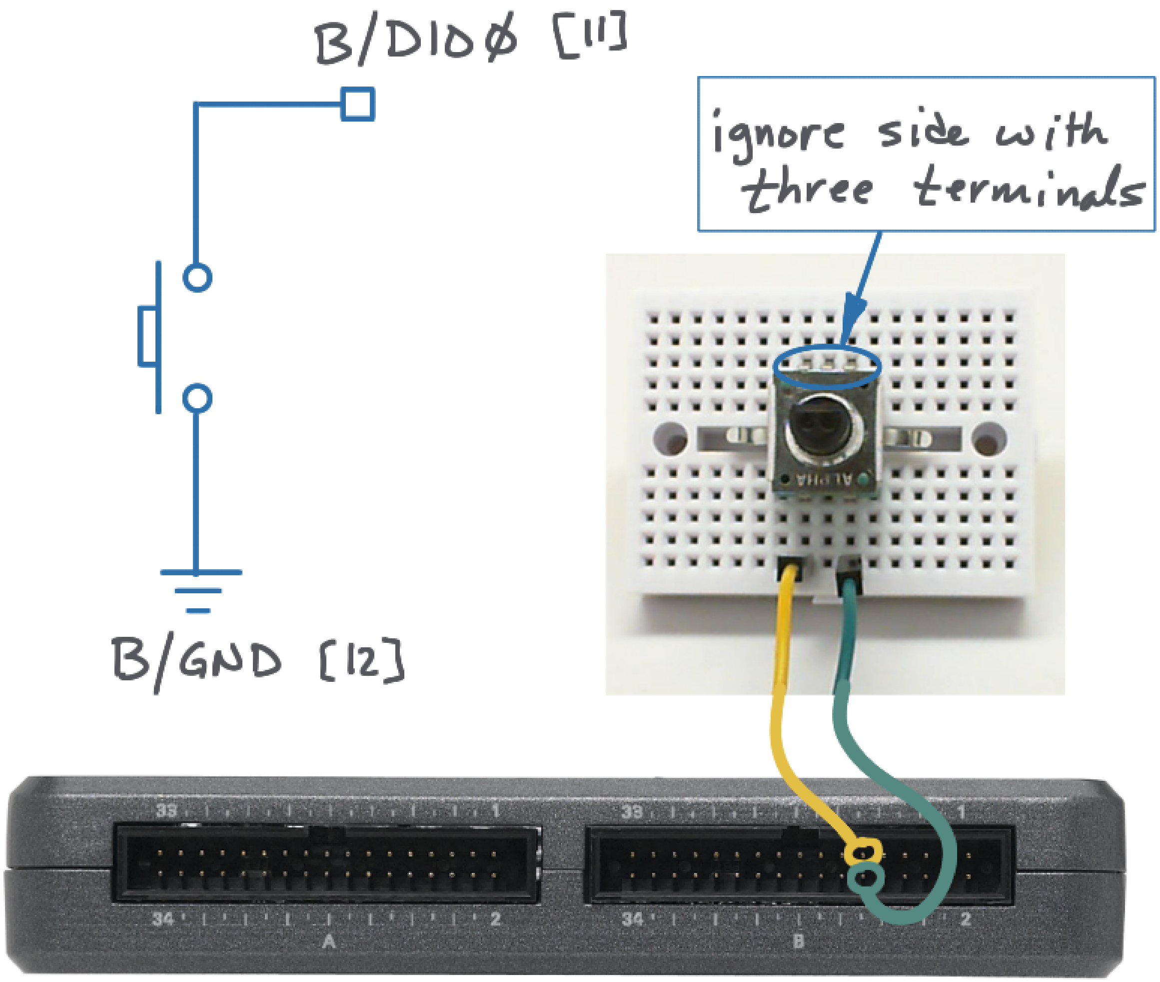


Figure 4.2: Demonstration circuit for pushbutton switch: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

4.3 Basic Modifications

Study the video "Pushbutton Demo Walk-Through" at `http://youtu.be/Xm1A4Cw2POU` to learn the design principles of `Pushbutton demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Add the onboard LED Express VI (myRIO | Onboard subpalette) as an indicator on the output of the edge detector (the AND gate). Confirm that the LED flashes briefly when you press the pushbutton.

- Experiment with different values of loop speed by adjusting the value of Wait (ms); you may find it more convenient to change the constant to a front-panel control. At what value does the VI introduce noticeable delay responding to the pushbutton press?
- Adjust the presses counter behavior to count pushbutton *releases* instead of presses.
- Adjust the presses counter behavior to count both pushbutton presses and releases. HINT: Try a single exclusive-OR gate from the Programming | Boolean subpalette.
- Modify the loop termination condition so that the VI runs only while the pushbutton is pressed.

NOTE: The SPDT (single-pole single-throw) slide switch included with the NI myRIO Starter Kit (see Figure ???) can connect to myRIO in the same way as the pushbutton switch. Simply use the middle terminal and either end terminal in place of the pushbutton terminals. Use the slide switch anytime you need to maintain the DIO at a specific level, for example, as a mode setting for your NI myRIO program.

4.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the pushbutton switch with other components and devices.

4.5 For More Information

- *Mini Push Button Switch* by SparkFun – a handy switch for circuit boards:
  `http://www.sparkfun.com/products/97`
- *Applications* by Knitter-Switch – learn about the myriad practical applications for switches as well as the wide variety of switch types:
  `http://www.knitter-switch.com/p_applications.php`

5 DIP Switches

DIP switches bundle multiple SPST switches together into a single component; "DIP" stands for "dual in-line package," the standard IC package style that is breadboard compatible, and "SPST" means "single pole, single throw," the simplest possible switch type. Figure 5.1 pictures two popular DIP switch styles: a standard DIP switch containing eight SPST switches and a 16-position rotary DIP switch that manipulates the open-and-closed states of four SPST switches in a binary sequence.

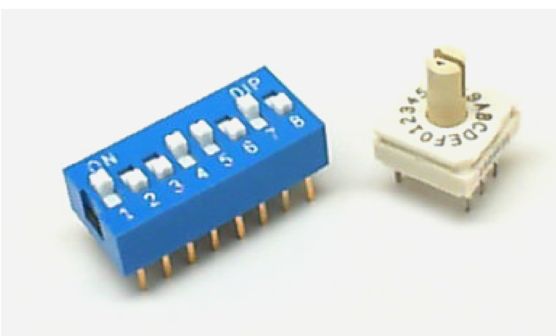


Figure 5.1: NI myRIO Starter Kit DIP switches: DIP switch (blue) and 16-position rotary DIP switch.

Learning Objectives: You will understand these core concepts related to DIP switches after completing the activities in this chapter:

1. DIP switch bundles N SPST switches into a single component with each switch appearing as a short circuit in one position and as an open circuit in the other,
2. 2^N-position rotary switch bundles N SPST switches into a single component; rotating the dial create a binary sequence of open-closed switch states,
3. Interface circuit to digital input relies on the DIO internal pull resistors to eliminate the need for additional components (pull-up on MXP Connectors A and B, pull-down on MSP Connector C),
4. Software views each switch as a Boolean (two-level) signal that is either active-high or active-low depending on the type of pull resistor, and
5. Software interprets the combined switch open-closed patterns in several ways including: integer numerical value, binary array, and individual bit fields.

5.1 Component Demonstration

Follow these steps to demonstrate correct operation of the DIP switches.

Select these parts from the NI myRIO Starter Kit:

- DIP switch `[TBD]`
- Rotary DIP switch `[TBD]`
- Breadboard
- Jumper wires, M-F (14×)
- Small screwdriver

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 5.2 on the facing page. The DIP switches interface circuit requires five connections to NI myRIO MXP Connector A and nine connections to Connector B:

1. DIP Switch 8 → B/DIO0 (pin 11)
2. DIP Switch 7 → B/DIO1 (pin 13)
3. DIP Switch 7 → B/DIO2 (pin 15)
4. DIP Switch 5 → B/DIO3 (pin 17)
5. DIP Switch 4 → B/DIO4 (pin 19)
6. DIP Switch 3 → B/DIO5 (pin 21)
7. DIP Switch 2 → B/DIO6 (pin 23)
8. DIP Switch 1 → B/DIO7 (pin 25)
9. DIP Switch common → B/GND (pin 8)
10. Rotary DIP 1 → A/DIO0 (pin 11)
11. Rotary DIP 2 → A/DIO1 (pin 13)
12. Rotary DIP 4 → A/DIO2 (pin 15)
13. Rotary DIP 8 → A/DIO3 (pin 17)
14. Rotary DIP C (common) → A/GND (pin 20)

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `DIP Switches demo.lvproj` contained in the subfolder `DIP Switches demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the individual switch states of the two DIP switches. Because both MXP connectors A and B include pull-up resistors on each DIO, an open switch appears as a high state.

With the rotary DIP switch dial at Position 0 all A/DIO state indicators should be active; turn the dial one click to the left to Position F and all indicator should be dark. Try clicking through the remaining positions and observe the binary sequence, remembering that the switches appear active-low.

Try each of the eight switches on the DIP switch and confirm that you can individually activate the indicators for A/DIO state. Is the switch open or closed in the "up" position?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using both Connectors A and B and that you have the correct pin connections, and
- Connecting wires link all eight of the lower pins of the DIP switch.

5.2 Interface Theory

Interface circuit: Each SPST switch in the standard DIP switch appears either as an open circuit or a short circuit depending on the switch position "up" or "down." The 16-position rotary switch opens all

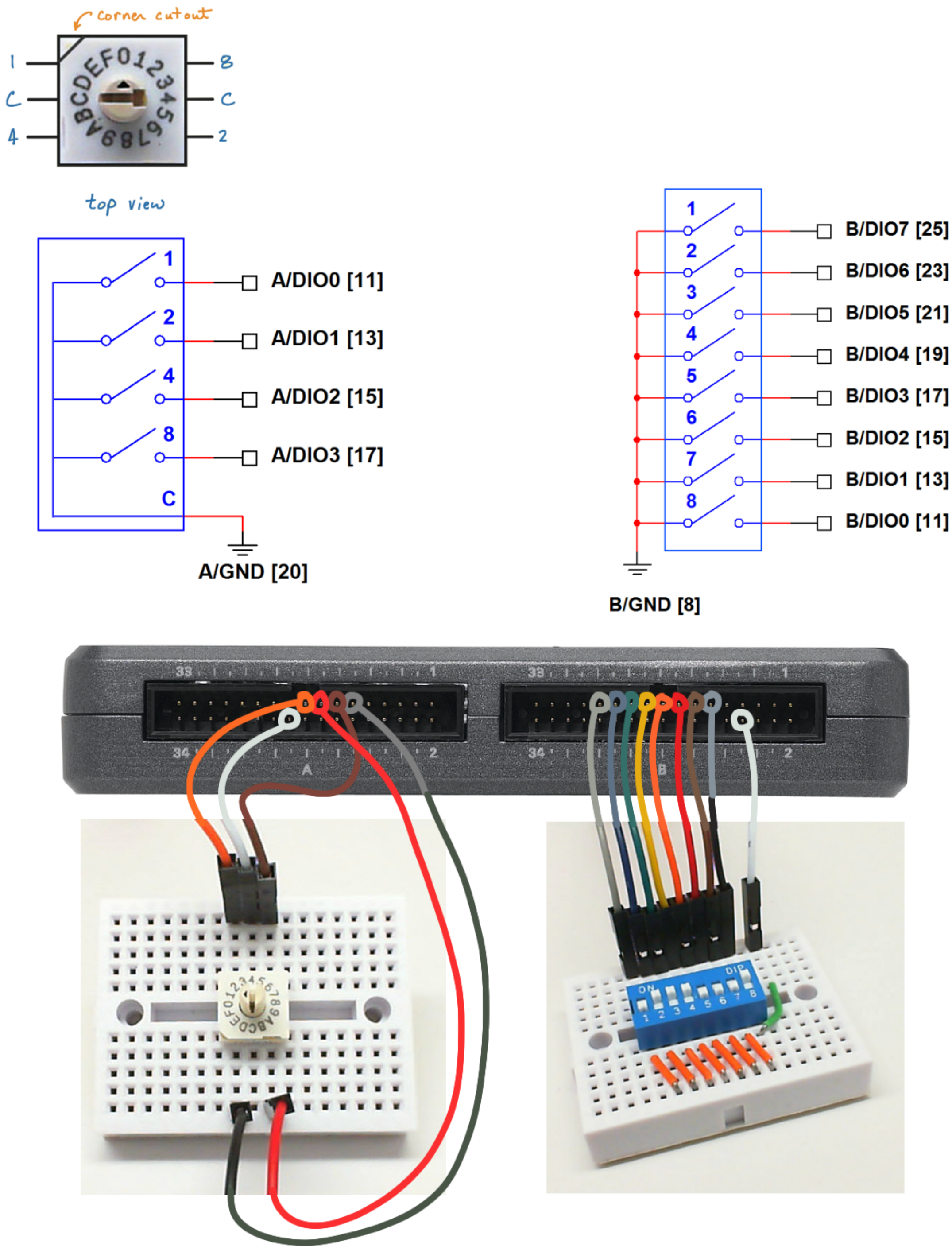


Figure 5.2: Demonstration circuit for DIP switches: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connectors A and B.

four SPST switches at Position 0 and then applies an ascending binary sequence to switch closings as the dial rotates clockwise. The DIP switches may be connected directly to the digital input without any additional components because of the internal pull resistors on the NI myRIO DIO lines.

Each DIP switch can be interpreted by software in a number of different ways, including: single integer numerical value, single Boolean array pattern, and groups of binary patterns or numerical values called bitfields.

Study the video "Pushbutton Interfacing Theory" at `http://youtu.be/e7UcL5Ycpho` to learn about the DIO pull resistors and how to properly connect a single SPST switch for pull-up resistors (MXP Connectors A and B) and pull-down resistors (MSP Connector C). Each of the SPST switches on the standard DIP switch must have one terminal tied low to ground; it is customary to ground all of the terminals on a given side, but you could use any pattern of grounded terminals that you like.

Study the video "DIP Switch Interfacing Theory" at `http://youtu.be/KNzEyRwcPIg` to learn more about the DIP switch and the rotary DIP switch, especially various ways that you can interpret the switch patterns as meaningful information in software.

LabVIEW programming: Study the video "Digital Input Low-Level subVIs" at `http://youtu.be/[TBD]:diVI` to learn how to use the low-level Digital Input subVIs to select the DIP switch connector pins from the front panel instead of editing the VI itself.

5.3 Basic Modifications

Study the video "DIP Switch Demo Walk-Through" at `http://youtu.be/ZMyYRSsQCac` to learn the design principles of `DIP Switches demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Display the DIP switch pattern as an 8-bit unsigned integer (UINT8 data type) using the right-most switch as the least-significant bit (LSB) and the "down" position as logical 0.
- Display the DIP switch pattern as three distinct fields as follows: Field 1 (bits 2:0) = 3-bit integer, Field 2 (bits 6:3) = 4-bit integer, and Field 3 = single-bit Boolean.
- Display the 16-position rotary DIP switch pattern as a 4-bit integer displayed in both decimal and in hexadecimal.
- Move either or both of the DIP switches to the MSP Connector C (remember, these have pull-down resistors) and repeat some of the previous exercises. Use a strategically-placed single "NOT" gate to avoid changing other parts of the block diagram. Also remember to connect the DIP switch common terminal to C/+5V (pin 20).
- Experiment with different values of loop speed by adjusting the value of Wait (ms); you may find it more convenient to change the constant to a front-panel control. At what value does the VI introduce noticeable delay responding to the pushbutton press?
- Adjust the presses counter behavior to count pushbutton *releases* instead of presses.
- Adjust the presses counter behavior to count both pushbutton presses and releases. HINT: Try a single exclusive-OR gate from the Programming | Boolean subpalette.
- Modify the loop termination condition so that the VI runs only while the pushbutton is pressed.

5.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the DIP switches with other components and devices.

5.5 For More Information

- *2-Wire Controlled Digital DIP Switch* by Maxim Integrated – an electronic replacement for mechanical DIP switches, the DS3904 contains microcontroller-controlled *nonvolatile variable resistors* that offer smaller footprint, higher reliability, and lower cost:

  `http://www.maximintegrated.com/app-notes/index.mvp/id/238`

6 Relay

Low-power digital outputs lack the necessary current drive to operate motors, lights, and other high-current appliances. Relays bridge the power gap using a relatively low-power magnetic coil to control a switch designed to carry large currents. Figure 6.1 shows the NI myRIO Starter Kit SPDT relay.

Figure 6.1: NI myRIO Starter Kit relay.

Learning Objectives: In this module you will create a standard interface circuit to verify correct operation of the relay, learn interface circuit design principles and related LabVIEW programming techniques, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the relay with additional components or devices.

6.1 Component Demonstration

Follow these steps to demonstrate correct operation of the relay.

Select these parts from the NI myRIO Starter Kit:

- Relay `http://www.cndongya.com/pdf/relayjzc-11f.pdf`
- General-purpose rectifier `http://www.vishay.com/docs/88503/1n4001.pdf`
- ZVP2110A p-channel enhancement-mode MOSFET `http://www.diodes.com/datasheets/ZVP2110A.pdf`
- Breadboard
- Jumper wires, M-F (3×)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 6.2 on page 29. Note that the three relay switch contacts do not sit on tenth-inch centers, therefore this end of the relay must hang off the side of the breadboard. The interface circuit requires three connections to NI myRIO MXP Connector B:

1. 5-volt power supply → B/+5V (pin 1)
2. Ground → B/GND (pin 6)

3. Relay control → B/DIO0 (pin 11)

Run the demonstration VI:
- Download http://www.ni.com/academic/myrio/project-guide-vis.zip if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Relay demo.lvproj` contained in the subfolder `Relay demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically..

Expected results: Your relay should be clicking in synchronism with the front-panel indicator DIO state. Click the cycle front-panel button to disable automatic mode and to enable the manual button; click this button to manually set the digital output state either high or low. The relay control signal is active-low, therefore the coil is energized when the DIO is low.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:
- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct transistor orientation — the transistor has a rounded shape on one side, and
- Correct rectifier orientation — when the rectifier is backwards the relay coil will never reach the voltage level necessary to turn on.

6.2 Interface Theory

Interface circuit: The relay contains an electromagnet coil that operates a spring-loaded switch. The coil current is approximately 100 mA, well beyond the current drive limits of the NI myRIO digital output. The interface circuit uses a p-channel enhancement FET as a switch to turn the coil current on and off and a rectifier to protect the transistor from large back-emf voltage when the transistor shuts off the coil current.

Study the video "Relay Interfacing Theory" at http://youtu.be/jLFL9_EWlwI to learn more about the relay principles of operation and interface circuit design principles including: sizing the transistor for relay coil current, importance of the rectifier to deal with back-emf voltage spiking, and circuit topologies for DIOs with internal pull-up resistors (MXP connector) and internal pull-down resistors (MSP connector).

Interface circuit design considerations:
1. Relay coil turn-on voltage
2. Relay coil current
3. Transistor current capacity
4. Transistor threshold voltage
5. Transistor remains off on NI myRIO reset
6. Rectifier can handle current spike
7. Rectifier can handle reverse voltage
8. Pull-up vs pull-down resistors on DIO

LabVIEW programming: Study the video "Run-Time Selectable DIO Channels" at http://youtu.be/[TBD]:diochans to learn how to use the low-level Digital I/O subVIs Open, Write, and Close to create a VI with run-time selectable

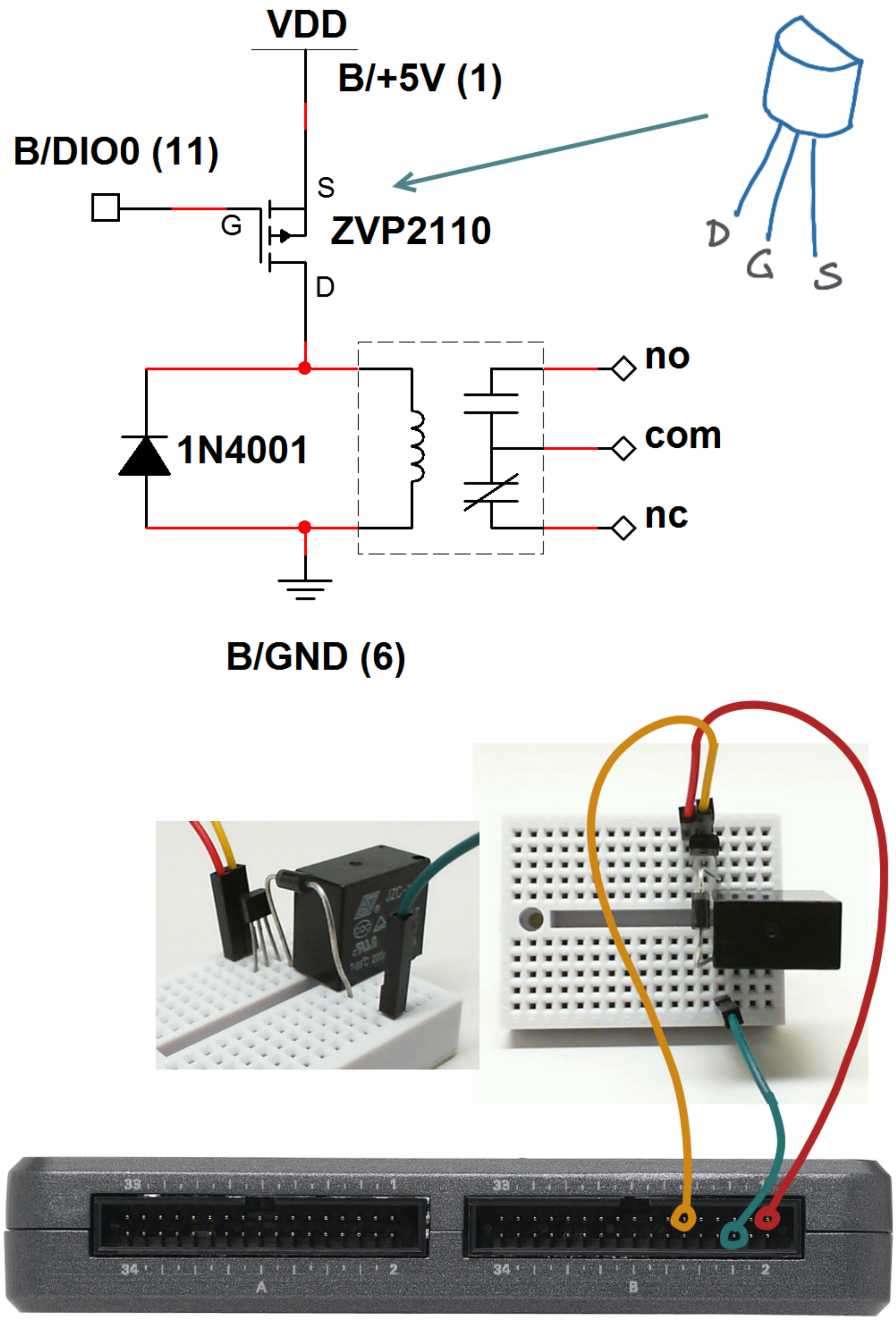


Figure 6.2: Demonstration circuit for relay: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

DIO channels, i.e., select the DIO channel directly on the front panel rather than editing the VI itself.

6.3 Basic Modifications

Study the video "Relay Demo Walk-Through" at `http://youtu.be/W2iukd8WVIA` to learn the design principles of `Relay demo.lvproj`, and then try making these modifications:

- Add a front-panel control to adjust the cycle frequency specified in Hertz; at what frequency is the relay unable to keep up?
- Blink two LEDs to simulate a railroad crossing signal; use the three relay contacts as shown in Figure [FIGURE].
- Build and test the interface circuit for a relay controlled from the MSP connector; see Figure 6.3 on the next page. Select C/DIO7 (pin 18) as the relay control line on the myRIO DIO channel front-panel control. Power the circuit from the MSP connector with C/+5V (pin 20) and the digital ground C/DGND (pin 19).

6.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the relay with other components and devices.

6.5 For More Information

- *Using Relays (Tips & Tricks)* by Jumper One – learn how to reduce relay switching time and minimize relay current for battery-powered applications:

  `http://jumperone.com/2011/10/using-relays`

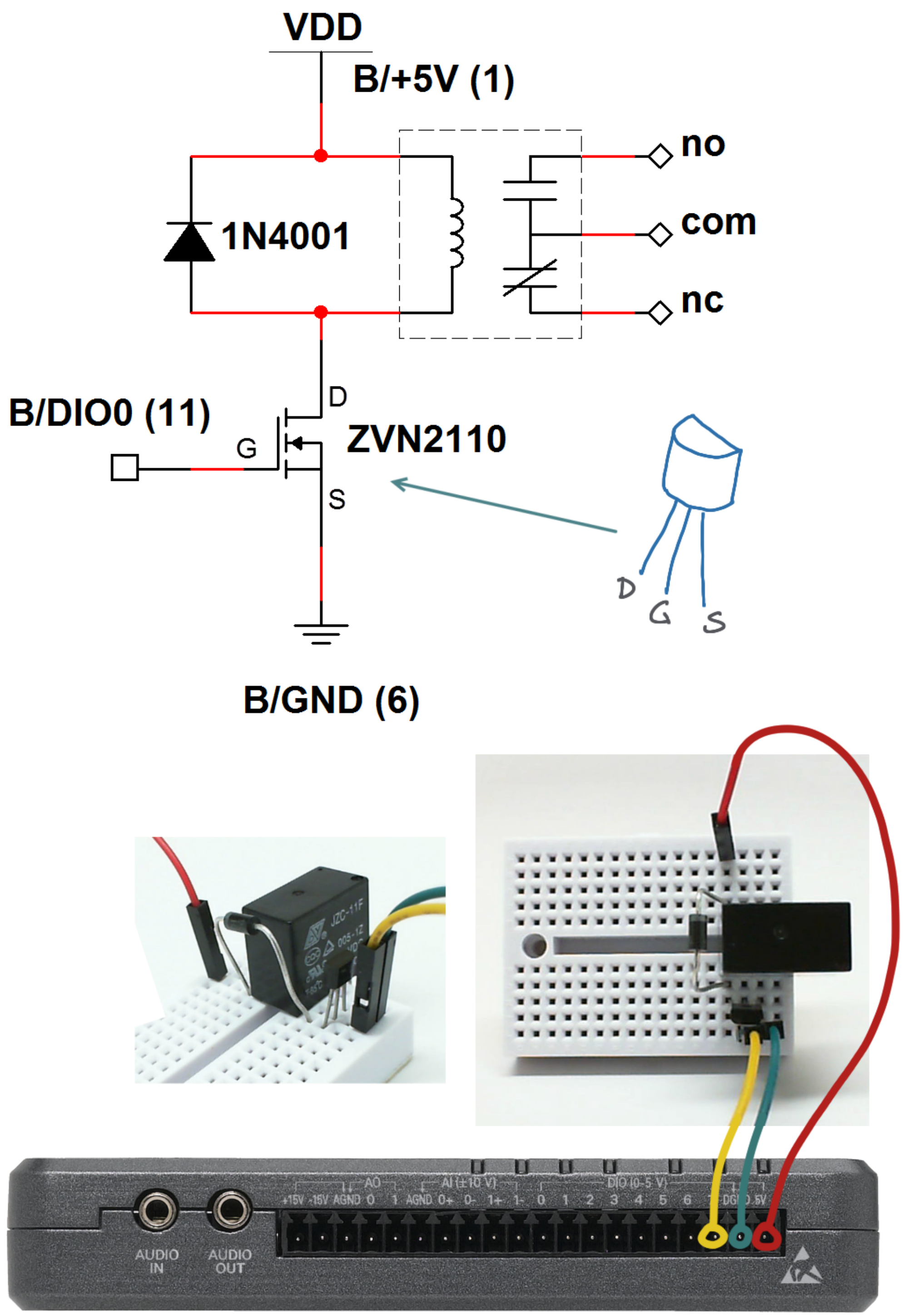


Figure 6.3: Relay interface circuit suitable for internal pull-down resistors on NI myRIO MSP Connector C.

7 Potentiometer

A *potentiometer* is a three-terminal variable resistor. When connected to a power supply to form a voltage divider a potentiometer acts as a proportional rotation sensor. Figure 7.1 pictures the NI myRIO Starter Kit potentiometer.

Figure 7.1: NI myRIO Starter Kit potentiometer.

Learning Objectives: You will understand these core concepts related to the potentiometer after completing the activities in this chapter:

1. Potentiometer can be used as either one or two variable resistors,
2. Connecting the potentiometer as a voltage divider produces a voltage proportional to rotation angle, and
3. Select the potentiometer resistance to minimize power consumption and to minimize loading effects.

7.1 Component Demonstration

Follow these steps to demonstrate correct operation of the potentiometer.

Select these parts from the NI myRIO Starter Kit:

- 10 kΩ potentiometer `[TBD]`
- Breadboard
- Jumper wires, M-F (3×)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 7.2 on the next page. The potentiometer interface circuit requires three connections to NI myRIO MXP Connector B:

1. Pot Terminal 1 → B/GND (pin 16).
2. Pot Terminal 2 → B/AI0 (pin 3), and
3. Pot Terminal 3 → B/+5V (pin 1), and

NOTE: Flatten the two tabs on either side of the potentiometer so that it fits flush on the breadboard surface.

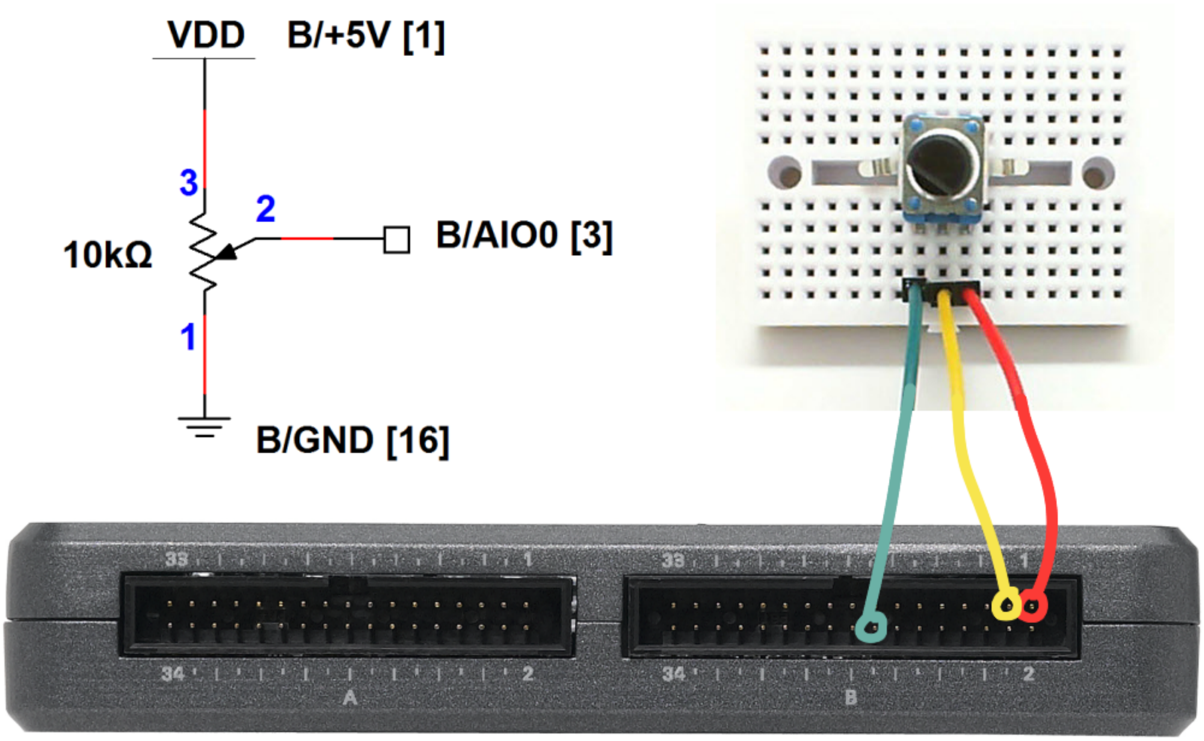


Figure 7.2: Demonstration circuit for potentiometer: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Potentiometer demo.lvproj` contained in the subfolder `Potentiometer demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the voltage on the analog input, B/AI0. Turn the potentiometer dial and you should observe a corresponding change in the voltage sensed on the analog input.

Because the potentiometer acts as an ajustable voltage divider between ground and the +5-volt supply, you should observe that a full rotation of the potentiometer dial from one extreme to the other causes the voltage to change from 0 to 5 volts.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

7.2 Interface Theory

Interface circuit: The potentiometer provides a fixed resistance between the two out terminals, while the middle terminal connects to a movable contact point that effectively makes the potentiometer appear as two variable resistors. As one resistor increases in value, the other resistor decrease by the same amount. Wiring the potentiometer between ground and the power supply produces a voltage divider with voltage output proportional to the position of the contact. Connecting this variable voltage to the NI myRIO analog input provides a convenient sensing technique for angular position.

Study the video "Potentiometer Characteristics" at `http://youtu.be/3gwwF9rF_zU` to learn about the potentiometer as a variable voltage source, and also to learn about proper sizing of the potentiometer to minimize power required and also to minimize loading effects that could distort the measurement.

LabVIEW programming: Study the video "Analog Input Express VI" at `http://youtu.be/[TBD]:aiExVI` to learn how to use Analog Input Express VI to measure the voltage divider output voltage.

7.3 Basic Modifications

Study the video "Potentiometer Demo Walk-Through" at `http://youtu.be/RYeKIuU6DX8` to learn the design principles of `Potentiometer demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Make front-panel control selectable
- Connect to DIO and monitor state; use pot to locate hysteresis edges
- Replace dial indicator with another types
- Make a bargraph indicator of pot wiper position with onboard LEDs

EDIT: Add details

7.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the potentiometer with other components and devices.

7.5 For More Information

- *Potentiometer* by Resistorguide – describes a variety of potentiometer types and characteristics: `http://www.resistorguide.com/potentiometer`

8 Thermistor

The *thermistor* – a contraction of "thermal" and "resistor" – is a two-terminal semiconductor device whose resistance varies with temperature. Most thermistors are of the *negative temperature coefficient* (NTC) type, meaning their resistance varies inversely with temperature. Figure 8.1 pictures the NI myRIO Starter Kit thermistor.

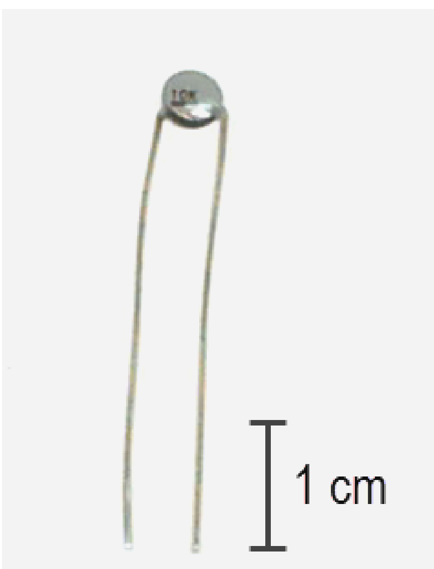


Figure 8.1: NI myRIO Starter Kit thermistor.

Learning Objectives: In this module you will create a standard interface circuit to verify correct operation of the thermistor, learn interface circuit design principles and related LabVIEW programming techniques, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the thermistor with additional components or devices.

8.1 Component Demonstration

Follow these steps to demonstrate correct operation of the thermistor.

Select these parts from the NI myRIO Starter Kit:

- 10 kΩ thermistor, EPCOS B57164K103J `http://www.epcos.com/inf/50/db/ntc_09/LeadedDisks__B57164__K164.pdf`
- Resistor, 10 kΩ `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- 0.1 μF ceramic disk capacitor, marking "104" `http://www.avx.com/docs/Catalogs/class3-sc.pdf`
- Breadboard
- Jumper wires, M-F (4$\times$)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 8.2 on page 39. The interface circuit requires four connections to NI myRIO MXP Connector B:

1. 5-volt power supply $\rightarrow$ B/+5V (pin 1),
2. Ground $\rightarrow$ B/GND (pin 6)

3. Temperature measurement → B/AI0 (pin 3), and
4. Supply voltage measurement → B/AI1 (pin 5).

Measure the resistance of the 10 kΩ resistor with an ohmmeter, as this value is required for the LabVIEW VI.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Thermistor demo.lvproj` contained in the subfolder `Thermistor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Enter the measured resistance of the 10 kΩ resistor as R [ohms].

Expected results: The demo VI displays the measured resistance of your thermistor; expect to see a value close to 10 kΩ at room temperature. Try heating the thermistor by gently pinching the thermistor body with your finger tips; you may also use a straw to blow warm air on the thermistor. You should observe the resistance going down. How low can you make the resistance?

Use a plastic sandwich bag filled with two ice cubes or crushed ice. Surround the thermistor with ice and you should observe the resistance going up. How high can you make the resistance?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

8.2 Interface Theory

Interface circuit: Constructing a voltage divider from a thermistor and a fixed-value resistor offers an easy-to-build yet effective interface circuit. Placing the thermistor in the top branch of the divider makes the measured voltage increase along with higher temperature.

Study the video "Thermistor Characteristics" at `http://youtu.be/US406sjBUxY` to learn more about thermistor characteristics and the *Steinhart-Hart thermistor equation* that converts measured thermistor resistance to temperature in degrees Kelvin. Study "Thermistor Resistance Measurement" at `http://youtu.be/PhZ2QlCrwuQ` to learn how to measure the thermistor resistance with a voltage divider, and also how to size the resistor R for best measurement sensitivity and range. Also take a look at "Measure Resistance with a Voltage Divider" at `http://youtu.be/9KUVD7RkxNI` for a more complete treatment of voltage dividers as a measurement technique.

LabVIEW programming: Study the video "Analog Input Express VI" at `http://youtu.be/[TBD]:aiExVI` to learn how to use Analog Input Express VI to measure the voltage divider's primary output as well as the voltage divider supply voltage.

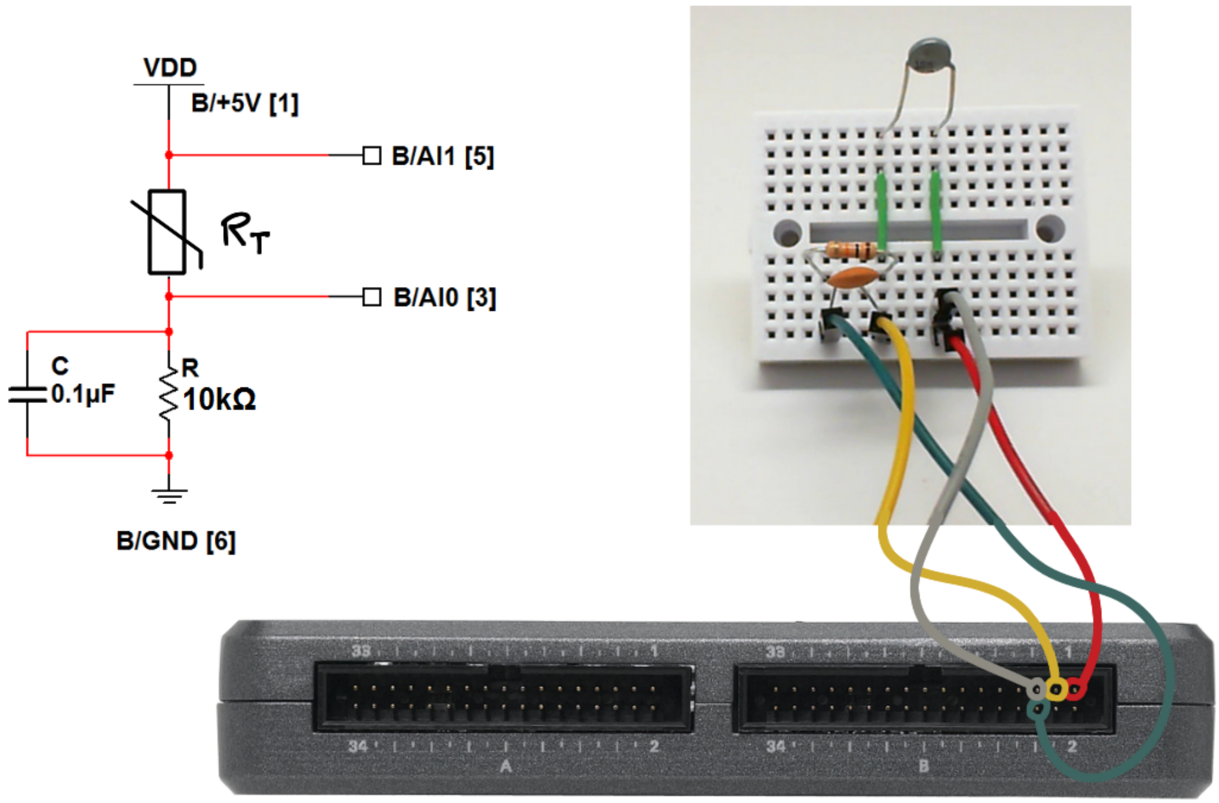


Figure 8.2: Demonstration circuit for thermistor: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

8.3 Basic Modifications

Study the video "Thermistor Demo Walk-Through" at `http://youtu.be/xi0VIpGpf4w` to learn the design principles of `Thermistor demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Add the necessary computation to convert the measured resistance to temperature in degrees Celsius; display the temperature on the large front-panel dial indicator. Use the built-in subVIs Mathematics | Elementary | Natural Logarithm and Mathematics | Polynomial | Polynomial Evaluation. Use the polynomial coefficient values presented in the earlier video.
- Modify your temperature display to display in degrees Fahrenheit.
- Create a Boolean indicator to indicate when the measured temperature exceeds (or falls below) a preset threshold.

8.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the thermistor with other components and devices.

8.5 For More Information

- *Thermistors* by National Instruments – thermistor characteristics and the Steinhart-Hart thermistor equation:
  `http://zone.ni.com/reference/en-XX/help/370466V-01/measfunds/thermistors`
- *NTC Thermistors* by Vishay – learn about thermistor principles of operation, selection criteria, design equations, and example circuits and applications:
  `http://www.vishay.com/docs/29053/ntcintro.pdf`

9 Photocell

A *photocell* is a two-terminal device constructed from cadmium sulfide (CdS) whose resistance varies with illumination in the visible spectrum of 400 to 700 nm. The photocell pictured in Figure 9.1 has a resistance that varies over many orders of magnitude: 10 kΩ at moderate illumination, less than 100 Ω at high illumination, and more than 10 MΩ in darkness.

Figure 9.1: NI myRIO Starter Kit photocell.

Learning Objectives: In this module you will create a standard interface circuit to verify correct operation of the photocell, learn interface circuit design principles and related LabVIEW programming techniques, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the photocell with additional components or devices.

9.1 Component Demonstration

Follow these steps to demonstrate correct operation of the photocell.

Select these parts from the NI myRIO Starter Kit:

- Photocell, API PDV-P9203 `http://www.advancedphotonix.com/ap_products/pdfs/PDV-P9203.pdf`
- Resistor, 10 kΩ `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- Breadboard
- Jumper wires, M-F (3$\times$)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 9.2 on the next page. The interface circuit requires three connections to NI myRIO MXP Connector B:

1. 5-volt power supply $\rightarrow$ B/+5V (pin 1),
2. Ground $\rightarrow$ B/GND (pin 6), and
3. Photocell measurement $\rightarrow$ B/AI0 (pin 3).

Measure the resistance of the 10 kΩ resistor with an ohmmeter, as this value is required for the LabVIEW VI.

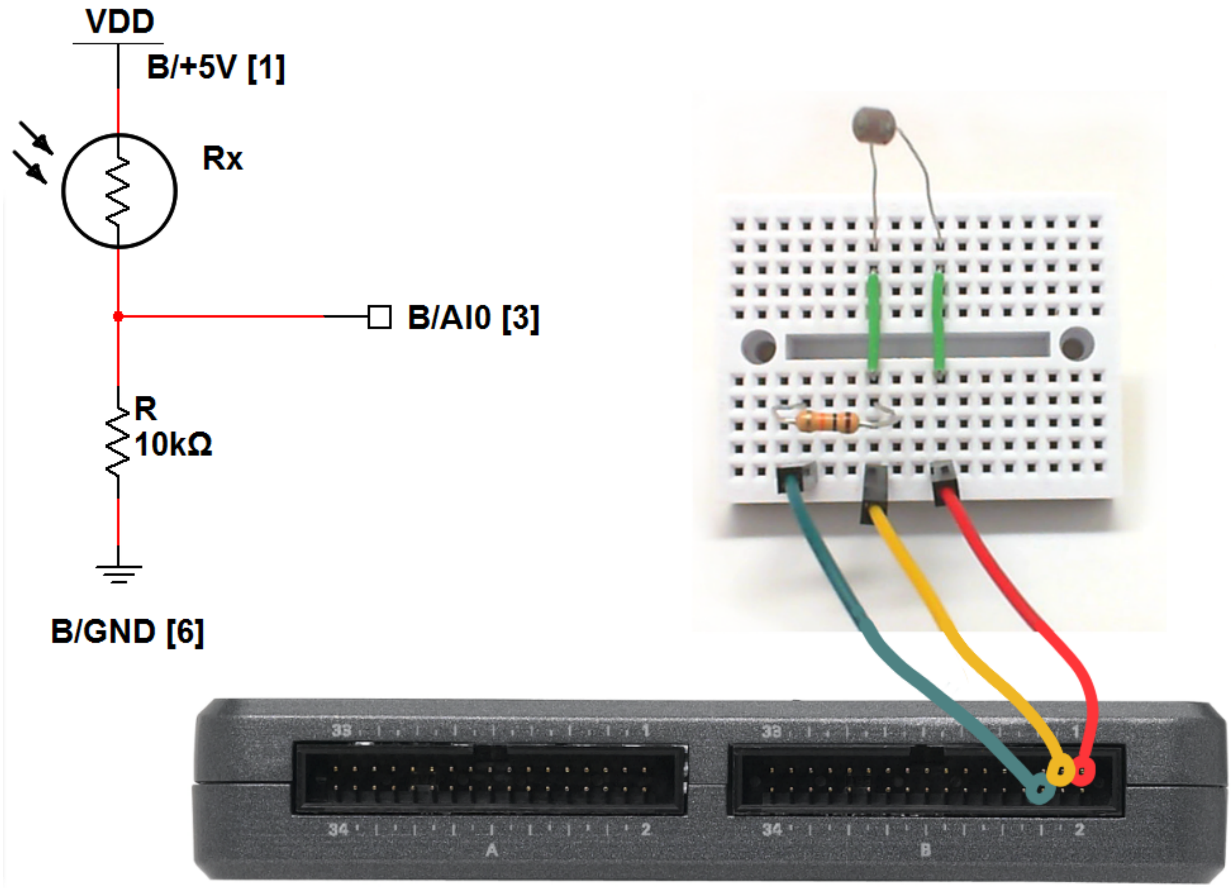


Figure 9.2: Demonstration circuit for photocell: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Photocell demo.lvproj` contained in the subfolder `Photocell demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the measured resistance of your photocell; expect to see a values in the range 1 kΩ to 10 kΩ at moderate illumination. Try blocking the light with a cover or cylindrical shroud such as a black straw. You should observe the resistance going up. How high can you make the resistance?

Use a flashlight or bright LED as illumination; you should observe the resistance going down. How low can you make the resistance?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

9.2 Interface Theory

Interface circuit: Constructing a voltage divider from a photocell and a fixed-value resistor offers an easy-to-build yet effective interface circuit. Placing the photocell in the top branch of the divider makes the measured voltage increase with more illumination.

Study the video "Photocell Characteristics" at `http://youtu.be/geNeoFUjMjQ` to learn about photocell characteristics, and then study "Measure Resistance with a Voltage Divider" at `http://youtu.be/9KUVD7RkxNI` to learn how to measure the photocell resistance with a voltage divider, and also how to properly choose the resistance R to maximize measurement sensitivity and range.

LabVIEW programming: Study the video "Analog Input Express VI" at `http://youtu.be/[TBD]:aiExVI` to learn how to use Analog Input Express VI to measure the voltage divider's primary output.

9.3 Basic Modifications

Study the video "Photocell Demo Walk-Through" at `http://youtu.be/jZQqsc5GmoY` to learn the design principles of `Photocell demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Add a Boolean front-panel control to make the voltage divider configuration user-selectable, i.e., one state of the control corresponds to the photocell in the lower branch while the other state selects the upper branch. Confirm that your modification works properly by swapping positions of the photocell and resistor.
- Create a "room lights ON" detector with a suitable node from the Programming | Comparison subpalette and a Boolean front-panel indicator. Include a user-selectable threshold resistance as a front-panel numerical control.
- Because analog inputs are not as plentiful as digital inputs, create the same "room lights ON" detector behavior, but do the comparison directly at a digital input; study "Resistive-Sensor Threshold Detector" at `http://youtu.be/TqLXJroefTA` to learn the design procedure.

9.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the photocell with other components and devices.

9.5 For More Information

- *Photocell Tutorial* at Digital DIY:
  `http://digital-diy.com/general-electronics/269-photocell-tutorial.html`
- *Photocells* by Adafruit – a good overview of CdS photocells with interesting applications such as light-based motor control, line-following robots, and laserpointer breakbeam sensor:
  `http://learn.adafruit.com/photocells/overview`
- *Photocell Tutorial* by Digital DIY – several different light detector circuits, including a dual-photocell version that acts as a bi-stable latch:
  `http://digital-diy.com/general-electronics/269-photocell-tutorial.html`

10 Microphone

A microphone serves as an acoustic sensor to record audio signals and monitor acoustic level. The Analog Devices ADMP504 microphone pictured in Figure 10.1, a MEMS (micro electro-mechanical sensor) device with an on-board amplifier, finds application in smartphones. The device is packaged on a DIP carrier for use on a breadboard.

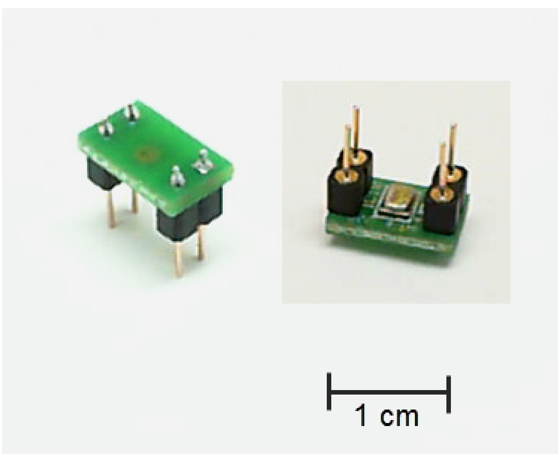


Figure 10.1: NI myRIO Starter Kit microphone. The photo on the right shows the microphone mounted to the underside of the DIP carrier.

Learning Objectives: In this module you will create a standard interface circuit to verify correct operation of the microphone, learn interface circuit design principles and related LabVIEW programming techniques, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the microphone with additional components or devices.

10.1 Component Demonstration

Follow these steps to demonstrate correct operation of the microphone.

Select these parts from the NI myRIO Starter Kit:

- ADMP504 ultra-low-noise microphone `http://www.analog.com/ADMP504`
- OP37 low-noise precision high-speed op amp `http://www.analog.com/OP37`
- AD8541 rail-to-rail single-supply op amp `http://www.analog.com/AD8541`
- 0.1 μF ceramic disk capacitor, marking "104" `http://www.avx.com/docs/Catalogs/class3-sc.pdf`
- 1.0 μF electrolytic capacitor `http://industrial.panasonic.com/www-data/pdf/ABA0000/ABA0000CE12.pdf`
- Resistor, 10 kΩ (3$\times$) `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- Resistor, 100 kΩ `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- Breadboard
- Jumper wires, M-F (5$\times$)
- 3.5 mm stereo audio cable `[TBD]`

- Test clips (2×)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 10.3 on the next page. The interface circuit requires four connections to NI myRIO Connectors B and C:
1. +15-volt power supply → C/+15V (pin 1),
2. −15-volt power supply → C/-15V (pin 2),
3. +3.3-volt power supply → B/+3.3V (pin 33),
4. Ground → C/AGND (pin 3), and
5. Mic output → AUDIO IN.

Connect the 3.5 mm stereo audio cable to the AUDIO IN. Use test clips to connect the other plug tip (left channel) to the mic output and the plug sleeve to ground; refer to Figure 10.2.

Alternatively you may build the interface circuit designed for direct connection to the MXP analog input; refer to Figure 10.4 on page 48. This version requires five connections to NI myRIO MXP Connector B:
1. +5-volt power supply → B/+5V (pin 1),
2. Ground → B/GND (pin 6),
3. +3.3-volt power supply → B/+3.3V (pin 33),
4. Ground → B/GND (pin 30), and
5. Mic amplifier output → B/AI0 (pin 3).

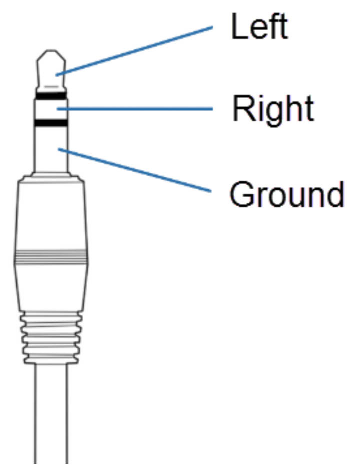


Figure 10.2: 3.5 mm stereo audio cable plug connections for ground, left channel, and right channel.

Run the demonstration VI:
- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Microphone demo.lvproj` contained in the subfolder `Microphone demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the audio signal detected by the microphone as an oscilloscope display. Select the appropriate mic input depending on the interface circuit you built: AudioIn/Left or B/AI0. Also, double-click the upper and lower limits of the waveform chart and set to -2.5 and 2.5 for AudioIn/Left and to 0 and 5 for B/AI0.

Try whistling, speaking, singing, or any other sound, and you should see the corresponding waveform. Note that the waveform is centered about zero when using the audio input and centered about approximately 2.5 volts when using the analog input.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

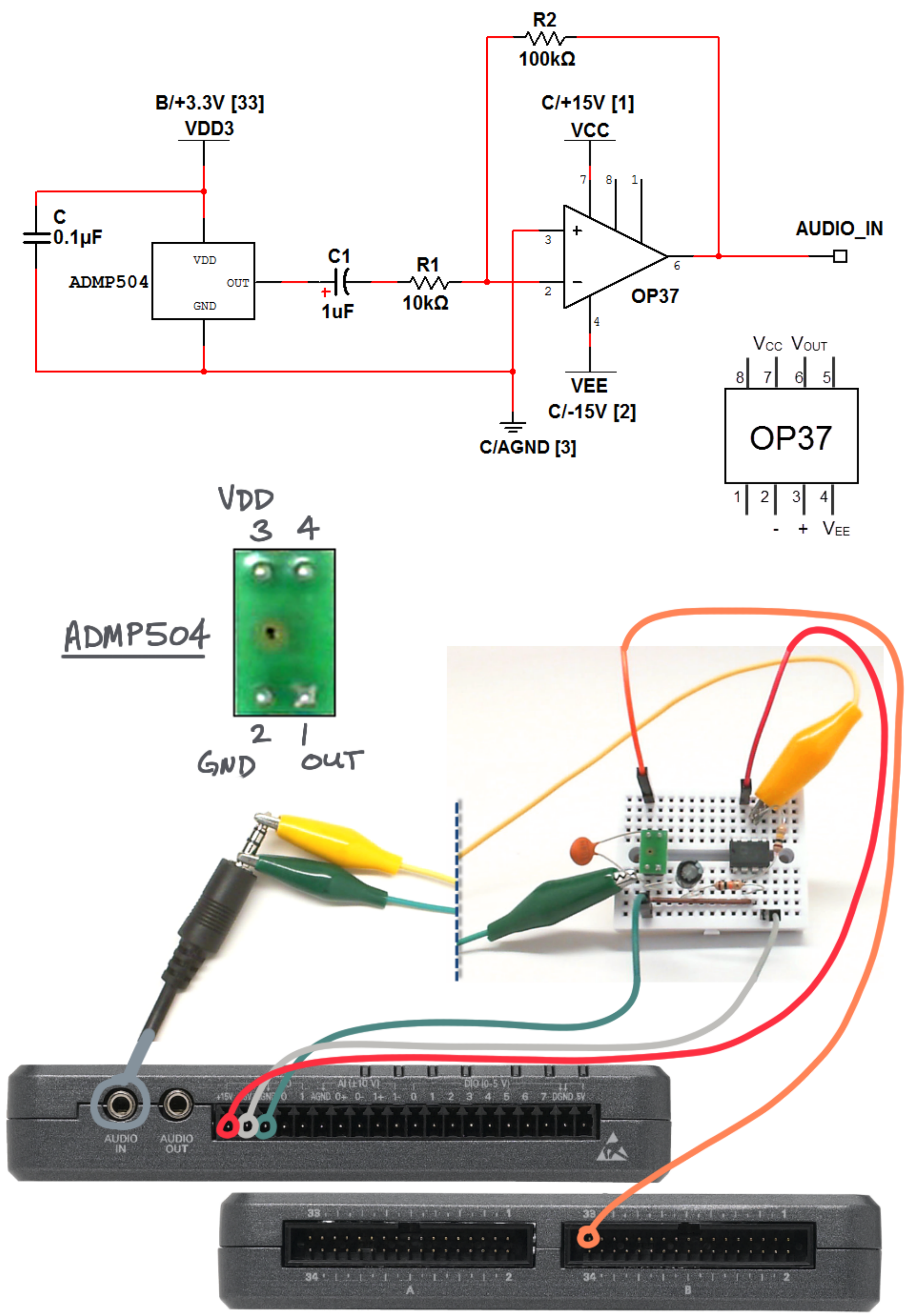


Figure 10.3: Demonstration circuit for microphone with AUDIO IN: schematic diagram, recommended breadboard layout, and connection to NI myRIO Connectors B and C.

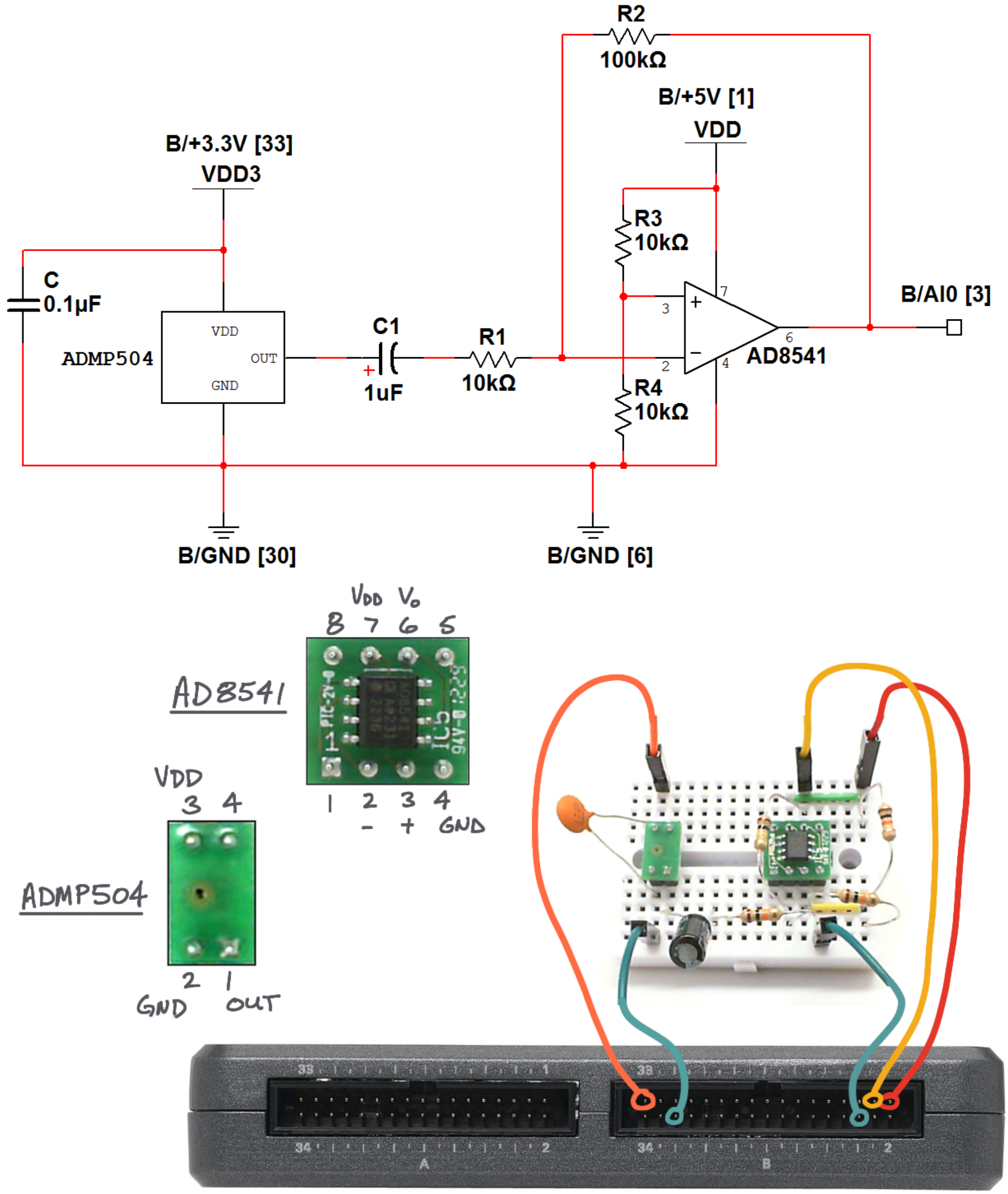


Figure 10.4: Demonstration circuit for microphone with analog input (AI): schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B (and possibly Connector C) and that you have the correct pin connections, and
- Use the B/AI0 (pin 3) jumper wire as probe to check the following signal points:
  - ADMP504 output: 0.8 volt DC offset with signal at up to ± 0.25 volts
  - Negative terminal of 1 μF capacitor: same as ADMP504 output but with zero DC offset (for OP37) or 2.5 V offset (AD8541)
  - Noninverting (+) op amp terminal: zero (for OP37) or 2.5 V offset (AD8541)
  - Inverting ($-$) op amp terminal: zero (for OP37) or 2.5 V offset (AD8541); if some other level, double-check all of your op amp connections or try another op amp

10.2 Interface Theory

Interface circuit: The ADMP504 produces a maximum voltage of 0.25 volts with a DC offset of 0.8 V. Study the video "ADMP504 Microphone Interfacing Theory" at `http://youtu.be/99lpj7yUmuY` to learn about the microphone characteristics, the need for a power-supply bypass capacitor, the DC blocking capacitor, and the inverting amplifier that boosts microphone output to level matched to the ± 2.5 V input range of the audio input and analog input.

LabVIEW programming: Study the video "Analog Input Low-Level subVI" at `http://youtu.be/[TBD]:aiVI` to learn how to use the low-level Analog Input subVIs to access the individual channels of the audio input as run-time selectable front-panel controls.

10.3 Basic Modifications

Open the block diagram of `Main.vi` in `Mic demo.lvproj`, and then try making these modifications:

- Spectrum plotter [DETAILS NEEDED], use point-by-point signal processing subpalette.
- VU meter [DETAILS NEEDED], use point-by-point lowpass filter and absolute value.
- Anti-aliasing filter by adding a parallel cap with the feedback resistance.
- Use B AI, range is 0 to 5V, has the same resolution as the audio connector, need to offset the output by 2.5 V, do this with a equal-resistance voltage divider on 5 V supply.

10.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the microphone with other components and devices.

10.5 For More Information

- *Microphone Array Beamforming with the ADMP504* by Analog Devices – video demonstration of two ADMP504 microphones combined with DSP (digital signal processing) to create a virtual directional microphone:
  `http://videos.analog.com/video/products/MEMS-sensors/1979997938001/Microphone-Array-Beamforming-with-the-ADMP504`
- *ADMP504 Flex Eval Board* by Analog Devices – the ADMP504 packaged with a bypass capacitor and extension wires:
  `http://www.analog.com/en/evaluation/EVAL-ADMP504Z-FLEX/eb.html`

11 Buzzer/Speaker

The buzzer/speaker pictured in Figure 11.1 (also called a magnetic transducer) generates tones over much of the audible frequency spectrum. The speaker coil is designed for on-off operation 5 volts at 80 mA when on), therefore cycling the speaker with a square wave created from a digital output is the most straightforward way to operate the speaker. A transistor-based interface circuit is required to drive the coil.

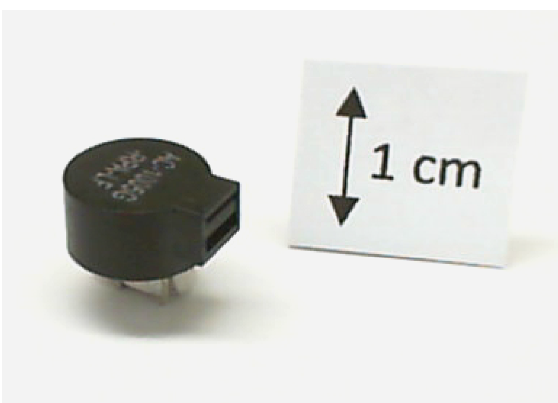


Figure 11.1: NI myRIO Starter Kit buzzer/speaker.

Learning Objectives: In this module you will create a standard interface circuit to verify correct operation of the buzzer/speaker, learn interface circuit design principles and related LabVIEW programming techniques, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the buzzer/speaker with additional components or devices.

11.1 Component Demonstration

Follow these steps to demonstrate correct operation of the buzzer/speaker.

Select these parts from the NI myRIO Starter Kit:

- Buzzer/speaker, Soberton GT-0950RP3 `http://www.soberton.com/product/gt-0950rp3`
- 1N3064 small-signal diode `http://www.fairchildsemi.com/ds/1N/1N3064.pdf`
- 2N3904 npn transistor `http://www.fairchildsemi.com/ds/MM/MMBT3904.pdf`
- Resistor, 1.0 kΩ `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- Breadboard
- Jumper wires, M-F (3×)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 11.2 on page 53. Note that the two buzzer/speaker terminals do not sit on tenth-inch centers, however, they fit just fine in two diagonally-adjacent breadboard holes. The interface circuit requires three connections to NI myRIO MXP Connector B:

1. 5-volt power supply → B/+5V (pin 1),
2. ground → B/GND (pin 6)
3. buzzer/speaker control → B/PWM0 (pin 27).

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Buzzer-Speaker demo.lvproj` contained in the subfolder `Buzzer-Speaker demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: Your buzzer/speaker should be making a low-frequency tone at 40 Hz. Move the front-panel control frequency [Hz] to increase the frequency. At what frequency is the sound output level the highest? What is the highest frequency that you can hear? Try blocking the port (rectangular hole) – how does this affect the sound at different frequencies?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

Troubleshooting tips: Not hearing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct transistor orientation — the transistor has a rounded shape on one side, and
- Correct diode orientation — when the diode is backwards the buzzer/speaker coil will never reach the voltage level necessary to turn on.

11.2 Interface Theory

Interface circuit: The buzzer/speaker contains an electromagnetic coil that vibrates a small diaphragm. The coil current is approximately 80 mA, well beyond the current drive limits of the NI myRIO digital output. The interface circuit uses an NPN transistor as a switch to turn the coil current on and off and a diode to protect the transistor from large back-emf voltage when the transistor abruptly shuts off the coil current.

Study the video "Buzzer/speaker Characteristics" at `http://youtu.be/8IbTWH9MpV0` to learn more about the buzzer/speaker characteristics and interface circuit design principles including: sizing the transistor for relay coil current, importance of the diode to deal with back-emf voltage spiking, and circuit topologies for DIOs with internal pull-up resistors (MXP connector) and internal pull-down resistors (MSP connector).

LabVIEW programming: Study the video "PWM Express VI" at `http://youtu.be/[TBD]:pwmExVI` to learn how to use PWM Express VI (PWM = pulse-width modulation) to create a squarewave in the frequency range 40 Hz to 40 kHz with adjustable pulse width.

11.3 Basic Modifications

Open the block diagram of `Main.vi` in `Buzzer-speaker demo.lvproj`, and then try making

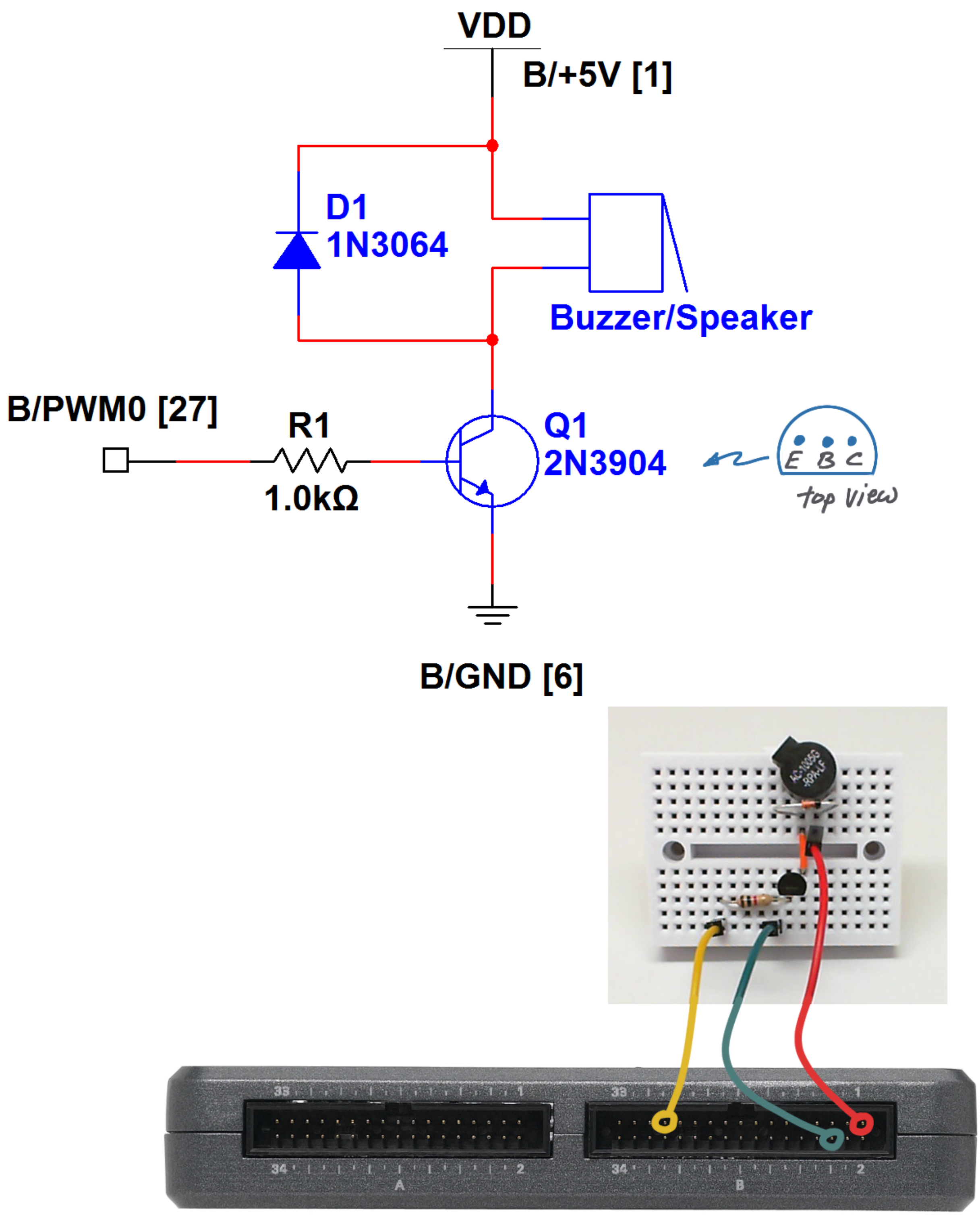


Figure 11.2: Demonstration circuit for buzzer/speaker: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

these modifications:

- Add a front-panel control to adjust the waveform pulse width (called duty cycle) between 0 % and 100 %. How does a narrow pulse (low duty cycle) affect the tone quality at various frequencies?
- Create a two-tone alarm signal with a Boolean front-panel control as an enable. Review `Discrete LED demo.lvproj` from Section 2.3 on page 11 to learn how to make a two-state oscillator.

11.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the buzzer/speaker with other components and devices.

11.5 For More Information

- *Products* by Soberton Inc. – discover related products to the buzzer/speaker in the NI myRIO Starter Kit:
  `http://www.soberton.com/products`

12 Motor

The low-voltage DC motor pictured in Figure 12.1 provides sufficient mechanical power to drive small fans or to spin lightweight objects. While the voltage is relatively low at 1.5 to 4.5 volts, the current can get as high as several hundred milliamps or even several amps in stall (blocked rotor) conditions. For this reason a power MOSFET is used as a motor driver. Figure 12.1 shows the NI myRIO Starter Kit DC motor.

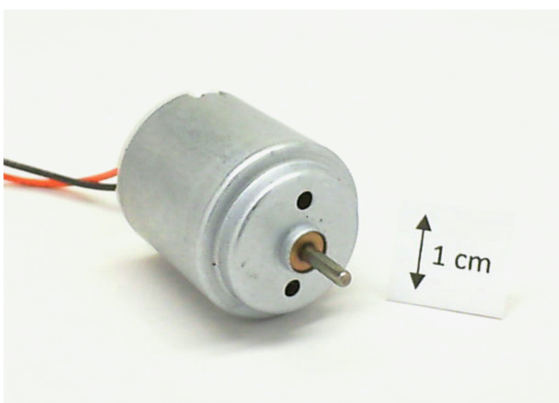


Figure 12.1: NI myRIO Starter Kit DC motor.

Learning Objectives: In this module you will create a standard interface circuit to verify correct operation of the motor, learn interface circuit design principles and related LabVIEW programming techniques, make some basic modifications to extend your understanding of the interface, and then challenge yourself to design a system that integrates the motor with additional components or devices.

12.1 Component Demonstration

Follow these steps to demonstrate correct operation of the motor.

Select these parts from the NI myRIO Starter Kit:

- DC motor `http://www.mabuchi-motor.co.jp/cgi-bin/catalog/e_catalog.cgi?CAT_ID=ff_180phsh`
- General-purpose rectifier `http://www.vishay.com/docs/88503/1n4001.pdf`
- ZVN2110A n-channel enhancement-mode MOSFET `http://www.diodes.com/datasheets/ZVN2110A.pdf`
- ZVP2110A p-channel enhancement-mode MOSFET `http://www.diodes.com/datasheets/ZVP2110A.pdf`
- IRF510 n-channel enhancement mode power MOSFET `http://www.vishay.com/docs/91015/sihf510.pdf`
- Breadboard
- Jumper wires, M-F (4×)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 12.2 on page 57. The interface circuit requires four connections to NI myRIO MXP Connector B:

1. 5-volt power supply → B/+5V (pin 1)
2. 3.3-volt power supply → B/+3.3V (pin 33)
3. Ground → B/GND (pin 30)
4. Motor control → B/DIO8 (pin 27)

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Motor demo.lvproj` contained in the subfolder `Motor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: Click the DIO state button to set the digital output to its low state and your motor should spin at high speed, then click the button again to stop the motor. Note that the motor driver interface circuit is *active low*.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO; a myRIO reset causes all of the digital I/O pins to revert to input mode.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct transistor orientation — carefully follow the pin diagrams for each transistor, especially note that the IRF510 has the gate pin on the side rather than in the middle as do the lower-power MOSFETS, and
- Correct rectifier orientation — when the rectifier is backwards the motor will never reach the voltage level necessary to turn on.

12.2 Interface Theory

Interface circuit: The motor requires approximately 180 mA (at 3.3 V) when unloaded and over 1000 mA when running at maximum efficiency, three times higher than the maximum available current from all three connectors combined. Stalling the motor due to excessive loading or blocking the rotor demands even higher current because effective resistance of the motor is less than 1 Ω. For these reasons the IRF510 n-channel enhancement power MOSFET serves as a high-current solid-state switch to operate the motor. Because the IRF510 gate-to-source threshold voltage $V_{\mathrm{GS(th)}}$ ranges from 2 to 4 V the NI myRIO DIO output voltage of 3.3 V is not sufficient to turn on the IRF510. The two low-power MOSFETS arranged as a standard CMOS logic inverter supplied by the 5 V supply act as a 3.3-to-5 V *level shifter* to ensure that the IRF510 gate voltage is either 0 V (off) or 5 V (on).

Study the video "Motor Interfacing Theory" at `http://youtu.be/C_22XZaL5TM` to learn more about the motor principles of operation and interface circuit design principles including: sizing the power transistor for motor current under various load conditions, importance of the rectifier to deal with back-emf voltage spiking, level-shifting circuit for 3.3-to-5 V, and required modification to connect the interface to the MSP connector with integral pull-down resistors.

Interface circuit design considerations:

1. Power MOSFET current capacity
2. Motor current under various loads

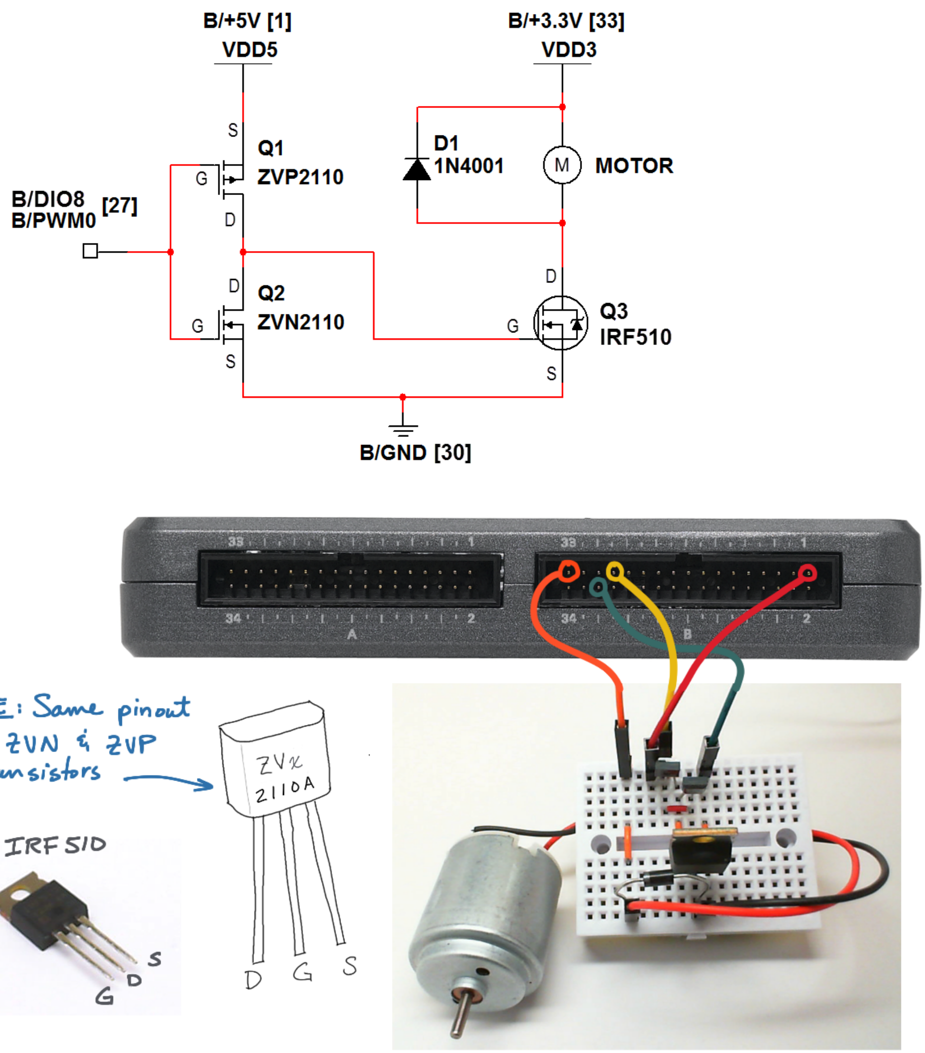


Figure 12.2: Demonstration circuit for DC motor: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

3. Power MOSFET threshold voltage
4. Power MOSFET RDS when fully turned on (vs when using a 3.3 V direct connection)
5. CMOS inverter as 3.3-to-5 V level shifter
6. Rectifier can handle current spike
7. Rectifier can handle reverse voltage
8. Pull-up vs pull-down resistors on DIO; use a 4.7K external pull-up to ensure the motor remains off on power-up or after NI myRIO reset

LabVIEW programming: Study the video "PWM Express VI" at `http://youtu.be/[TBD]:pwmExVI` to learn how to use the PWM Express VI to create a pulse-width modulated square wave to provide variable-speed motor operation.

12.3 Basic Modifications

Study the video "Motor Demo Walk-Through" `http://youtu.be/UCqFck0CLpc` to learn the design principles of `Motor demo.lvproj`, and then try making these modifications:

- Create variable-speed motor operation as follows:
  - Replace the existing Digital Output Express VI witht the PWM Express VI. Choose the PWM channel as B/PWM0, the same connector pin as B/DIO8 (pin 27). Choose the remaining dialog box options so that both frequency and duty cycle are available as inputs,
  - Create pointer slide front-panel controls for each; right-click on each control and choose "Visible items" and the "Digital display," and
  - Right-click on the frequency control, select "Scale" and then "Mapping," and choose "Logarithmic." Also, double-click the upper limit of your frequency control and enter "40000" and then similarly set the lower limit to "40."

  Experiment with both the duty cycle and frequency. What frequency minimizes audible PWM noise and maximizes your ability to create very slow motor speeds? What do you notice about restarting the motor after it stops? If you have a DMM ammeter handy, you may wish to observe the motor current under various conditions including mechanical loading vs. free running and starting current.
- Insert additional code to deal with the fact that the motor control interface is active-low. That is, you want 0 % duty cycle to turn the motor off rather than causing maximum speed as it does now.
- Add a Boolean front-panel control as a motor enable. Trying using a Select node under the Programming | Comparison subpalette to set the duty cycle either to 0 or to the value of the front-panel duty cycle control.
- Disconnect the motor control line and re-connect to C/PWM0 (pin 14) on MSP Connector C; adjust your VI to refer to this channel, too. You should observe that the motor is on due to the internal pull-down resistor. Now counter the effect of the internal pull-down with a 4.7 kΩ external pull-up resistor connected between the motor control line and the +5-volt supply. The motor should now remain off when the NI myRIO first powers on or after executing a myRIO reset.

12.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the DC motor with other components and devices.

12.5 For More Information

- *Brushed DC Motor Fundamentals* by Microchip – learn about DC motor principles of operation, drive circuits, direction control with an H-bridge, and speed control with Hall-effect sensors as feedback:

`http://ww1.microchip.com/downloads/en/AppNotes/00905B.pdf`

- *Brushless DC Motors Used in Industrial Applications* by Ohio Electric Motors – the biggest problem with brushed DC motors is the brushes (they wear out). Learn about *brushless* DC motors as the modern replacement:
`http://www.ohioelectricmotors.com/brushless-dc-motors-used-in-industrial-applications-1617`

13 Rotary Encoder

A *rotary encoder*, also known as a *quadrature encoder*, combines a rotary knob and two switches that open and close in a staggered fashion as the knob turns. With suitable decoding of the switching waveforms the knob angle and rotation direction can be sensed. Figure 13.1 pictures the NI myRIO Starter Kit rotary encoder.

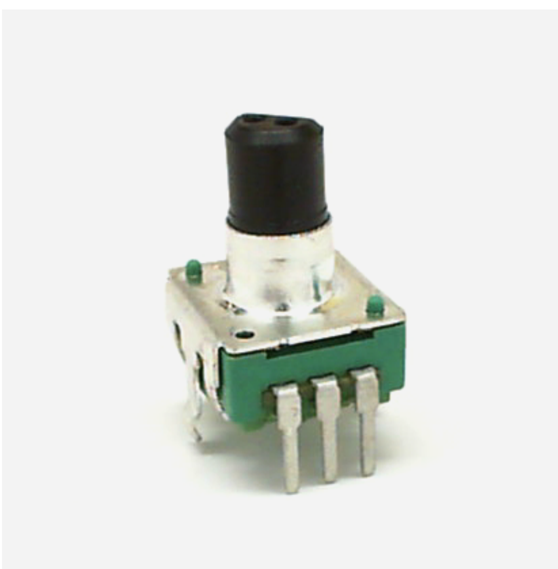


Figure 13.1: NI myRIO Starter Kit rotary encoder.

Learning Objectives: You will understand these core concepts related to the rotary encoder after completing the activities in this chapter:

1. Produces a pair of quadrature waveforms A and B,
2. NI myRIO provides built-in encoder inputs and associated subVI to indicate counts and direction,
3. Encoder waveforms have switch bounce and require debouncing circuitry, and
4. Connect encoder common to ground for internal pull-up resistors (MXP) or to 5V for internal pull-down resistors (MSP).

13.1 Component Demonstration

Follow these steps to demonstrate correct operation of the rotary encoder.

Select these parts from the NI myRIO Starter Kit:

- Rotary encoder `[TBD]`
- Resistor, 10 kΩ (2×) `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- 0.01 μF ceramic disk capacitor, marking "103" (2×) `http://www.avx.com/docs/Catalogs/class3-sc.pdf`
- Breadboard
- Jumper wires, M-F (5×)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 13.2 on page 63. The rotary encoder interface circuit requires three connections to NI myRIO MXP Connector B:

1. Encoder A → B/ENC.A (pin 18),
2. Encoder A → B/DIO0 (pin 11),
3. Encoder B → B/ENC.B (pin 22),
4. Encoder B → B/DIO1 (pin 13), and
5. Encoder COM → B/GND (pin 20).

NOTE: Flatten the two tabs on either side of the rotary encoder so that it fits flush on the breadboard surface.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Rotary Encoder demo.lvproj` contained in the subfolder `Rotary Encoder demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the encoder A and B switch states as either "open" or "closed." Slowly turn the encoder shaft clockwise and you should observe the following sequence: 'A' switch state closed with 'B' switch state open, then both closed, then A open with B closed, and finally both open again. You should also observe that both switches are open when the encoder shaft is at rest in one of its twelve *detente* positions. Rotate the shaft in the counter-clockwise direction and you should see a similar sequence, but with switch B closing first.

The demo VI also maintains a counter of A/B switch transitions, and should increment by four counts for each click of the encoder in the clockwise direction and decrement by found counts for each click in the counter-clockwise direction. The counter direction is also indicated with a front-panel indicator. Click the Reset Counter control to clear the counter to zero; click again to continue counting.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

13.2 Interface Theory

Interface circuit: The rotary encoder translates shaft rotation into a pair of switch opening/closing patterns known as *quadrature encoding*. The pair of switches called A and B connect directly to one of four NI myRIO encoder inputs shared with the standard digital input/output (DIO) terminals. The switching patterns are decoded by the Encoder Express VI which outputs a counter value as well as the counter direction. These outputs indicate the relative position of the rotary encoder shaft since the last time the counter was initialized.

NOTE: The additional connections to B/DIO0 and B/DIO1 simply provide another way to observe the switching activity; they are not required decoding.

Study the video "Rotary Encoder Interfacing Theory" at `http://youtu.be/CpwGXZX-5Ug` to learn about the rotary encoder principles of operation,

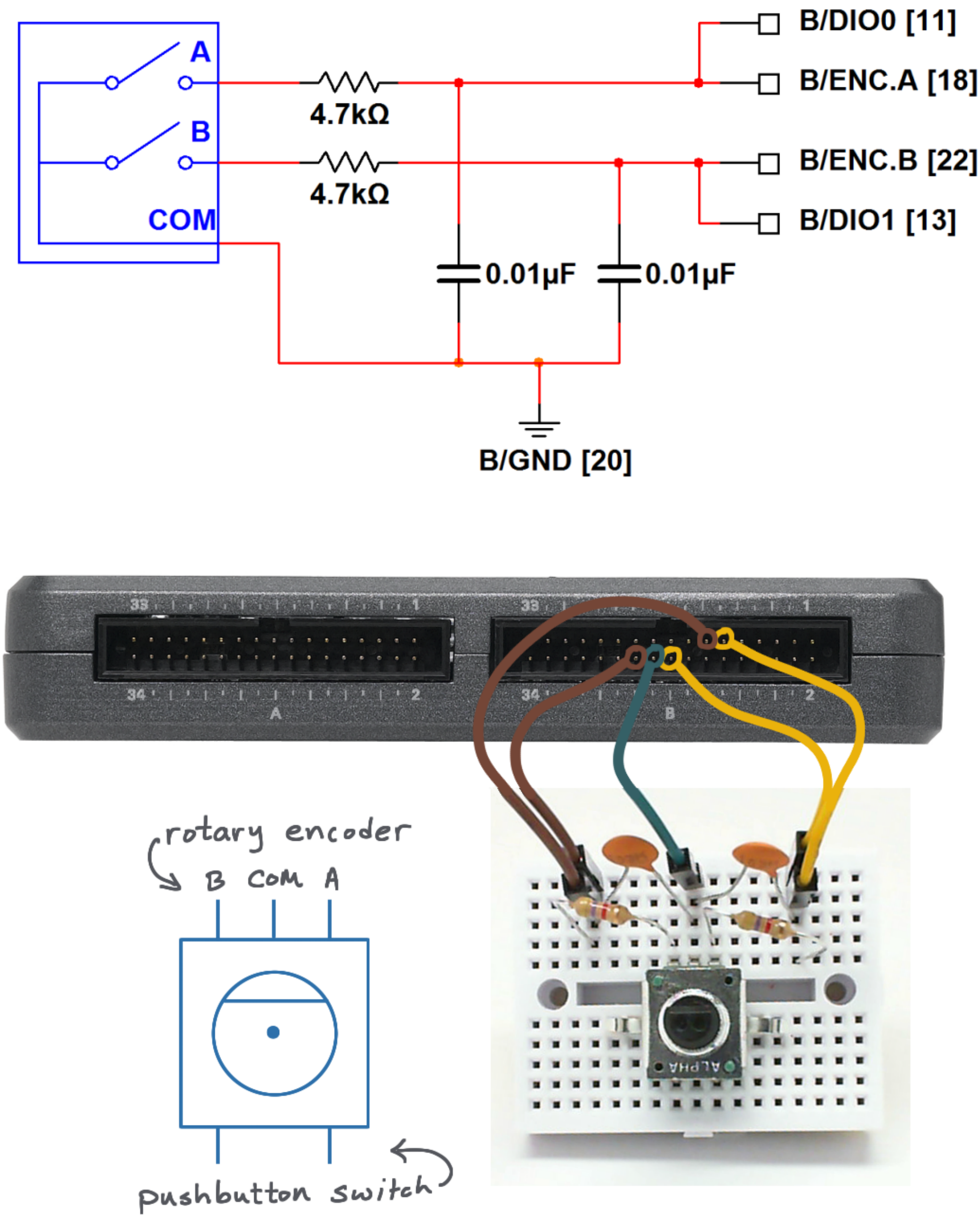


Figure 13.2: Demonstration circuit for rotary encoder: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

the quadrature waveforms produced by Switches A and B, interfacing techniques for the NI myRIO MXP and MSP connectors, and dealing with *switch bounce*, which when ignored will cause erroneous decoding of the switching waveforms.

13.3 Basic Modifications

Study the video "Rotary Encoder Demo Walk-Through" at `http://youtu.be/nmGlRqhQ6Rw` to learn the design principles of `Rotary Encoder demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Temporarily disconnect the two capacitors from the circuit, thereby removing the switch debouncing circuit. Experiment with various shaft rotation speeds and see if you can observe any relationship between rotation speed and counting error. Replace the capacitors and see if you can cause any count errors to occur – remember that each detente click corresponds to four counts.
- Create a front-panel indicator to display the number of full encoder shaft revolutions.
- Add the onboard button (myRIO | Onboard | Button Express VI as another way to reset the counter value.
- Add two wires to use pushbutton on the decoder

EDIT: Add details

13.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the rotary encoder with other components and devices.

13.5 For More Information

- *Quadrature Encoder Velocity and Acceleration Estimation with CompactRIO and LabVIEW FPGA* by National Instruments – a good review of quadrature encoders:
  `http://www.ni.com/white-paper/3921/en`
- *Quadrature Encoding in a Rotary Encoder* by Robot Room – look at the insides of a rotary encoder to better understand how it works; see the adjacent pages in this article, too:
  `http://www.robotroom.com/Counter5.html`
- *Rotary Encoder: H/W, S/W or No Debounce?* by HiFiDUINO – nice discussion of the rotary encoder switch bounce problem, with hardware and software solutions:
  `http://hifiduino.wordpress.com/2010/10/20/rotaryencoder-hw-sw-no-debounce`

14 Photointerrupter

A *photointerrupter*, also called a *photogate*, combines an infrared LED *emitter* to produce an optical light path and an infrared *detector* to sense when the light path is broken. With timers and suitable targets of alternating opacity and transparency, a photointerrupter servers as the basis for position and speed measurement. Figure 14.1 pictures the NI myRIO Starter Kit photointerrupter.

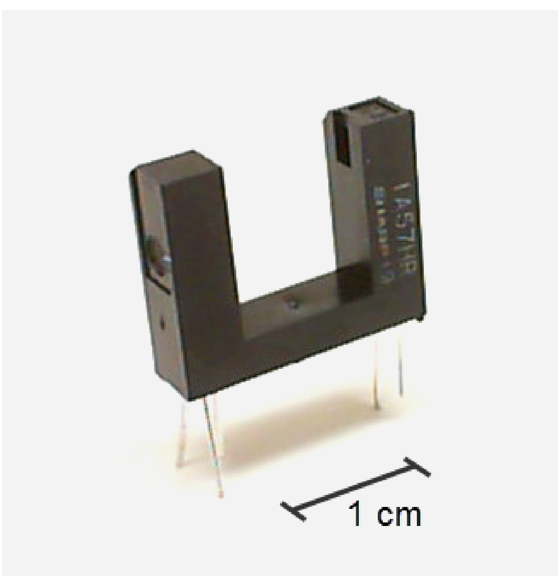


Figure 14.1: NI myRIO Starter Kit photointerrupter.

Learning Objectives: You will understand these core concepts related to the photointerrupter after completing the activities in this chapter:
1. Sizing the emitter current-limiting resistor,
2. Active-low output and compatibility with both MXP and MSP connectors,
3. Software edge detection counts photointerrupter events.

14.1 Component Demonstration

Follow these steps to demonstrate correct operation of the photointerrupter.

Select these parts from the NI myRIO Starter Kit:
- Photointerrupter `http://sharpmicroelectronics.com/download/gp1a57hr-epdf`
- Resistor, 470 Ω `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- Breadboard
- Jumper wires, M-F (3×)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 14.2 on the next page. The photointerrupter interface circuit requires three connections to NI myRIO MXP Connector B:
1. +5-volt supply → B/+5V (pin 1),
2. Ground → B/GND (pin 6), and
3. Photointerrupter output → B/DIO0 (pin 11).

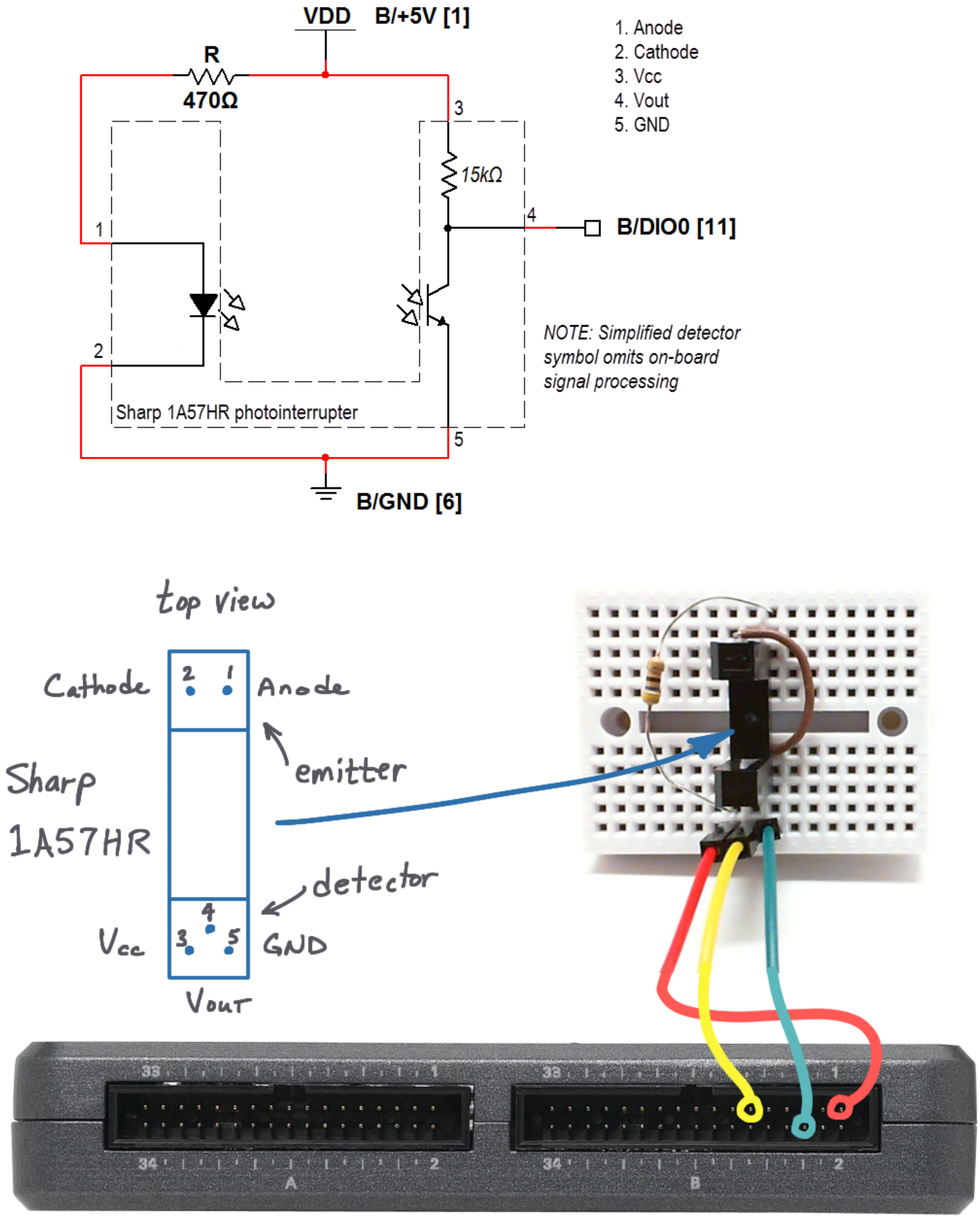


Figure 14.2: Demonstration circuit for photointerrupter: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Photointerrupter demo.lvproj` contained in the subfolder `Photointerrupter demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the state of the photointerrupter output and an event count. The state should normally be high. Block the photointerrupter optical path with an opaque card, your finger tip, and a sheet of paper. How much opacity does the photointerrupter require to trigger the event counter?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct resistor value – ensure that you have a 470 Ω resistor (yellow-violet-brown) and not a 470 kΩ resistor (yellow-violet-red).

14.2 Interface Theory

Interface circuit: The photointerrupter places an infrared LED as the photoemitter on one side of the gap and photodetector with signal conditioning circuitry on the other. The output is normally 5 volts and then zero volts when the optical path is blocked. The photointerrupter output may be connected directly to any NI myRIO digital input (MXP and MSP connectors).

Study the video "Photointerrupter Characteristics" at `http://youtu.be/u1FVfEvSdkg` to learn more about the photointerrupter characteristics including the voltage output details and requirements for sizing the current-limiting resistor for the infrared LED emitter.

LabVIEW programming: Study the video "Digital Input Low-Level subVIs" at `http://youtu.be/[TBD]:diVI` to learn how to use the low-level Digital Input subVIs to sense the state of the photointerrupter.

14.3 Basic Modifications

Study the video "Photointerrupter Demo Walk-Through" at `http://youtu.be/yuzNb1ZDbv4` to learn the design principles of `Photointerrupter demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Add the onboard LED Express VI (myRIO | Onboard subpalette) as an indicator on the output of the edge detector (the AND gate). Confirm that the LED flashes briefly when you block the photointerrupter optical path.
- Measure and display the elapsed time between photointerrupter events – use the Elapsed Time

Express VI (Programming | Timing subpalette) inside a case structure with the selector terminal connected to the edge detector output.

14.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at http://www.ni.com/myrio/project-guide for interesting systems integration project ideas that combine the photointerrupter with other components and devices.

14.5 For More Information

- *How do I set up photogates for data collection?* by Vernier – review of various ways to use a photointerrupter to take time-based measurements such as elapsed time and speed:
  http://vernier.com/til/1623
- *Switch Tips: Photointerrupter switches* by Machine Design – replace mechanical pushbuttons with a photointerrupter-based pushbutton to improve reliability and maximize equipment life:
  http://machinedesign.com/archive/switch-tips-photointerrupter-switches

15 Hall-Effect Sensor

The *Hall effect* provides an elegant way to sense magnetic fields as a change in voltage. Hall-effect sensors provide either digital or analog outputs, with the former finding numerous applications in detecting proximity and sensing position and speed, while the latter can map the field strength pattern of a magnet. Figure 15.1 shows a photo of a typical Hall-effect sensor.

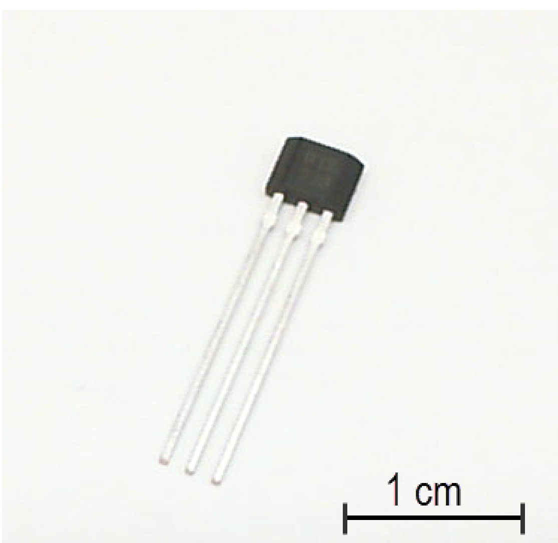


Figure 15.1: NI myRIO Starter Kit Hall effect sensor.

Learning Objectives: You will understand these core concepts related to the Hall effect sensor after completing the activities in this chapter:

1. Hall effect principle,
2. Latch vs. switch,
3. Open-collector output and compatibility with both MXP and MSP connectors,

15.1 Component Demonstration

Follow these steps to demonstrate correct operation of the Hall effect sensor.

Select these parts from the NI myRIO Starter Kit:

- US1881 Hall-effect latch `http://www.melexis.com/Hall-Effect-Sensor-ICs/Hall-Effect-Latches/US1881-140.aspx`
- 0.1 μF ceramic disk capacitor, marking "104" `http://www.avx.com/docs/Catalogs/class3-sc.pdf`
- Breadboard
- Jumper wires, M-F (3$\times$)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 15.2 on the next page. The Hall effect sensor interface circuit requires three connections to NI myRIO MXP Connector B:

1. +5-volt supply $\rightarrow$ B/+5V (pin 1),
2. Ground $\rightarrow$ B/GND (pin 6), and
3. Hall-effect sensor output $\rightarrow$ B/DIO0 (pin 11).

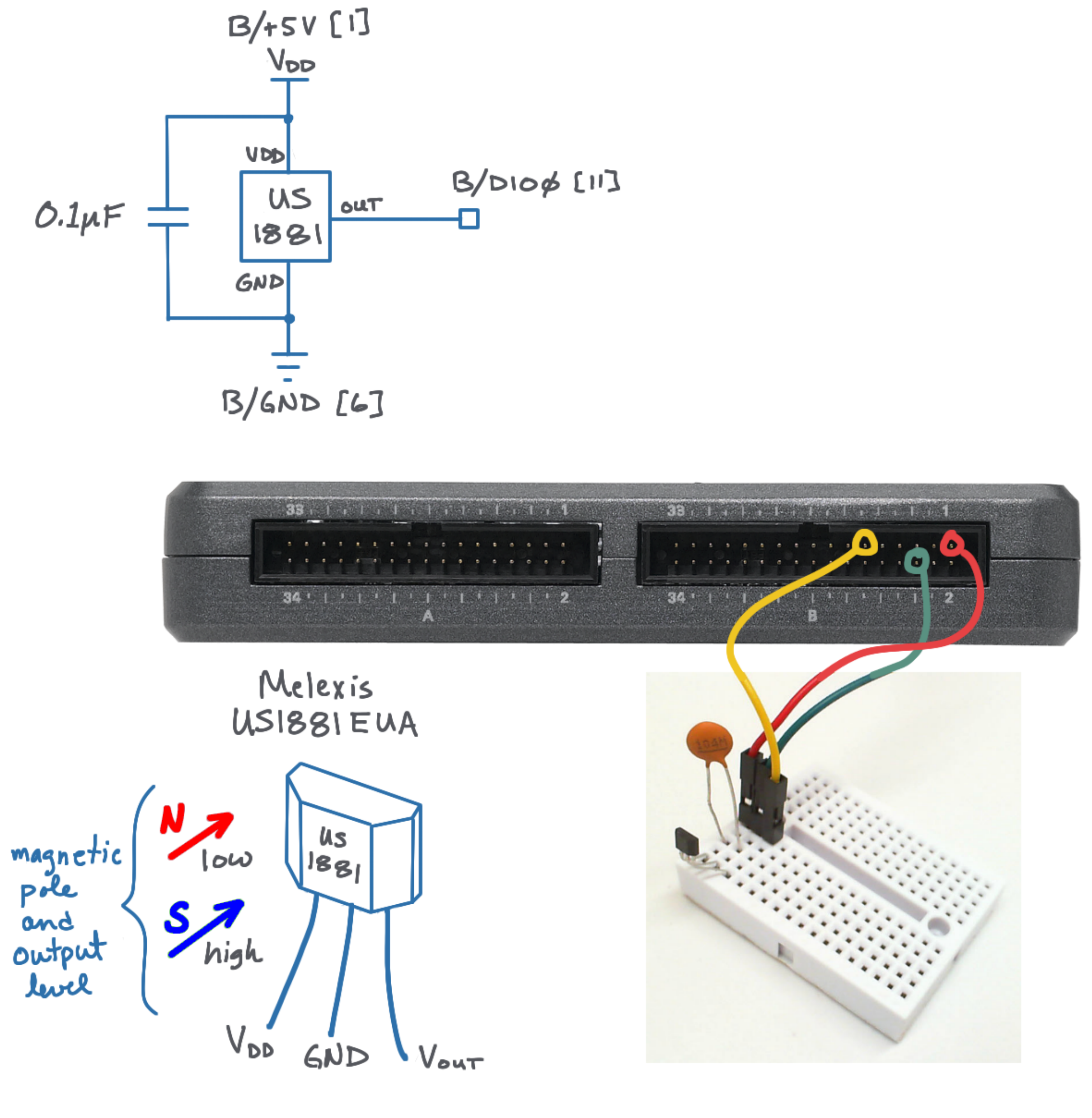


Figure 15.2: Demonstration circuit for Hall effect sensor: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Hall-Effect Sensor demo.lvproj` contained in the subfolder `Hall-Effect Sensor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the state of the Hall-effect sensor output and the most-recent magnetic pole (north or south) applied to the marked side. The US1881 behaves as a *latch*, meaning that a magnetic field of the opposite pole must be applied to flip the latch to its opposite state. Find as many magnets as you can — refrigerator magnets work nicely — and experiment with the sensitivity of the sensor (how close does the magnet need to be to flip the state) as well as the location of the north and south poles on the magnet. How many distinct pole locations can you detect on the magnet? (the answer may surprise you!)

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections, and
- Correct orientation of the US1881 — as you face the labeled side of the sensor you have V_{DD} on the left, ground in the middle, and the output on the right.

15.2 Interface Theory

Interface circuit: The *Hall effect* relies on the fact that electrons moving at right angles to a magnetic field experience a force. A current source establishes a current in a small semiconductor known as a *Hall plate*, and when subjected to magnetic field the electrons tend to deflect to one side of the plate, leaving behind positive charges on the other side of the plate. The complete Hall-effect sensor on the US1881 senses the charge displacement as a voltage, applies amplification and other signal conditioning operations, and provides the sensed magnetic pole type as the state of an open-drain output suitable for digital inputs.

Study the video "Hall-Effect Sensor Interfacing Theory" at `http://youtu.be/T9GP_cnz7rQ` to learn more about the Hall effect, the various types of Hall effect sensor output behaviors (latch, switch, and linear), and interface circuit techniques for both the MXP and MSP connectors.

LabVIEW programming: Study the video "Digital Input Low-Level subVIs" at `http://youtu.be/[TBD]:diVI` to learn how to use the low-level Digital Input subVIs to sense the state of the Hall-effect sensor.

15.3 Basic Modifications

Study the video "Hall-Effect Sensor Demo Walk-Through" at `http://youtu.be/BCJLg-WbIK4` to learn the design principles of `Hall-Effect Sensor demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Add the onboard LED Express VI (myRIO | Onboard subpalette) as an indicator of the Hall-effect sensor output state.

15.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at http://www.ni.com/myrio/project-guide for interesting systems integration project ideas that combine the Hall effect sensor with other components and devices.

15.5 For More Information

- *What is the Hall Effect?* by Melexis.com – includes an excellent animation of the charge displacement phenomenon due to a magnetic field:
  http://www.melexis.com/Assets/What-is-the-Hall-Effect-3720.aspx
- *Hall Applications Guide* by Melexis.com – slide-by switch, proximity sensor, rotary interrupt switch, pushbutton, flowmeter, liquid-level sensing, brushless DC motor commutation, and many more; includes detailed discussion of magnet properties:
  http://www.melexis.com/Assets/Hall-Applications-Guide--3715.aspx
- *A Strange Attraction: Various Hall Effect Sensors* by bildr.org – video demonstration of latch, switch, and linear-mode behavior:
  http://bildr.org/2011/04/various-hall-effect-sensors

16 Piezoelectric-Effect Sensor

A *piezoelectric sensor* sandwiches a piezoelectric material between two plate electrodes. Deforming the sensor causes a charge displacement that appears as a measurable voltage between the plates, much like a capacitor. Combining the piezoelectric sensor with a charge-to-voltage converter provides a useful sensor to detect shock and vibration. Figure 16.1 pictures the NI myRIO Starter Kit piezo sensor.

Figure 16.1: NI myRIO Starter Kit piezo sensor.

Learning Objectives: You will understand these core concepts related to the piezo sensor after completing the activities in this chapter:

1. Piezo sensor produces a voltage when deformed,
2. Interface circuit is a charge-to-voltage converter,
3. Feedback capacitor sets gain (deflection sensor or a sensitive vibration/shock sensor).

16.1 Component Demonstration

Follow these steps to demonstrate correct operation of the piezo sensor.

Select these parts from the NI myRIO Starter Kit:

- DT-series piezo film sensor `http://meas-spec.com/product/t_product.aspx?id=2478`
- AD8541 rail-to-rail single-supply op amp `http://www.analog.com/AD8541`
- 100 pF ceramic disk capacitor, marking "101" `http://www.vishay.com/docs/45171/kseries.pdf`
- 0.001 μF ceramic disk capacitor, marking "102" `http://www.vishay.com/docs/45171/kseries.pdf`
- Resistor, 10 MΩ `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- Resistor, 10 kΩ (2$\times$) `http://industrial.panasonic.com/www-data/pdf/AOA0000/AOA0000CE28.pdf`
- Breadboard
- Jumper wires, M-F (3$\times$)

Build the interface circuit: Refer to the schematic diagram and recommended breadboard layout shown in Figure 16.2 on the next page. The piezo sensor interface circuit requires three connections to NI myRIO MXP Connector B:

1. +5-volt supply $\rightarrow$ B/+5V (pin 1),
2. Ground $\rightarrow$ B/GND (pin 6), and
3. Sensor output $\rightarrow$ B/AI0 (pin 3).

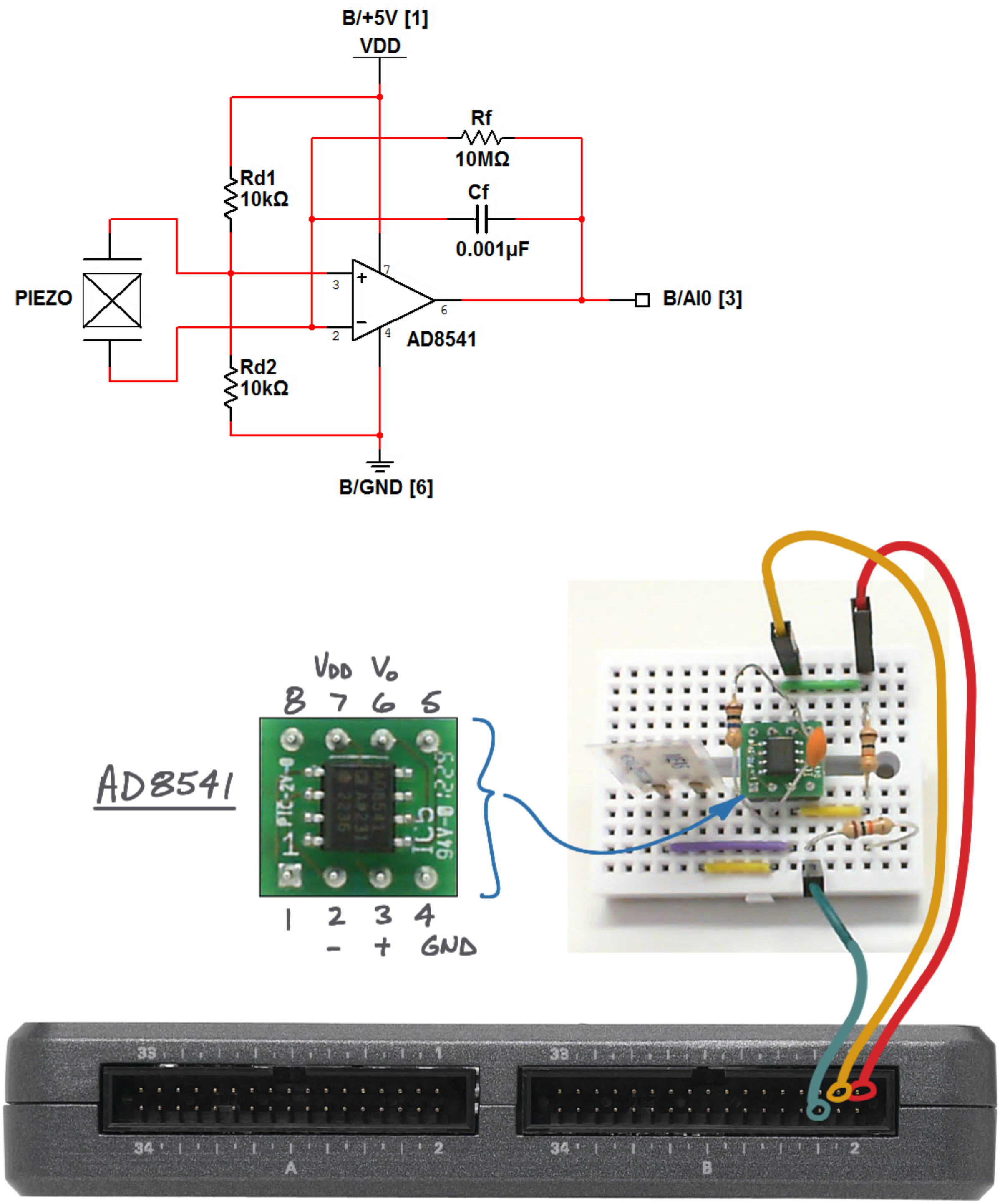


Figure 16.2: Demonstration circuit for piezo sensor: schematic diagram, recommended breadboard layout, and connection to NI myRIO MXP Connector B.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Piezoelectric-Effect Sensor demo.lvproj` contained in the subfolder `Piezoelectric-Effect Sensor demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the piezo sensor amplifier output much like an oscilloscope display. You should observe that the voltage is a constant close to 2.5 volts. Flex the piezo sensor and you should see a corresponding change in the voltage; observe the difference as you flex the sensor one way and then the other. Next, try flicking the sensor and jostling the breadboard. You should see a decaying sinusoid similar to an underdamped second-order system. Experiment to determine the sensitivity of the sensor, i.e., what is the lightest disturbance that still registers on the display?

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct op amp wiring – be certain that you understand the pinout of the AD8541, and
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections.

16.2 Interface Theory

Interface circuit: The piezo sensor acts like a charge pump (current source) when flexed. The interface circuit is a *charge-to-voltage converter* based on an op amp.

Study the video "Piezo Sensor Interfacing Theory" at `http://youtu.be/dHaPUJ7n-UI` to learn about the piezo-electric effect, charge-to-voltage converters, and choice of feedback capacitor to control the sensitivity of the sensor.

16.3 Basic Modifications

Study the video "Piezoelectric-Effect Sensor Demo Walk-Through" at `http://youtu.be/b1me4f-3iOE` to learn the design principles of `Piezoelectric-Effect Sensor demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Add the onboard LED Express VI (myRIO | Onboard subpalette) as a "bump" indicator; use the Programming | Comparison | In Range and Coerce subVI as the upper-limit and lower-limit range test for the analog input Read subVI. Confirm that the LED flashes briefly when you bump the sensor. Try one or more of the following methods to increase the sensitivity of your bump sensor:
  - Increase the charge-to-voltage converter gain by 10× by changing the feedback capacitance to 100 pF.

  - Right-click on the waveform chart Y-axis, select autoscaling mode, and observe the minimum and maximum values of the sensor waveform when the sensor is at rest. Use this information to more precisely set the threshold values for the In Range and Coerce subVI.
  - Add mass to the sensor tab (modeling clay works well).

  Try to make the bump indicator flash at the slightest disturbance such as tapping on the table.
- Microphone [need to explain how to export data and listen]
- Connect directly to DIO (2.5V bias is well into the high-level zone)

16.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the piezo sensor with other components and devices.

16.5 For More Information

- *Fundamentals of Piezoelectric Shock and Vibration Sensors* by Digi-Key TechZone – detect excessive shock and vibration that could damage equipment:
  `http://www.digikey.com/us/es/techzone/sensors/resources/articles/fundamentals-of-piezoelectric-sensors.html`
- *Bicycle Anti-Theft Alarm Circuit Diagram* by Electronic Circuit Diagrams & Schematics – alarm circuit based on the same piezo sensor in the NI myRIO Starter Kit:
  `http://circuitsstream.blogspot.com/2013/05/bicycle-anti-theft-alarm-circuit-diagram.html`
- *Signal Conditioning Piezoelectric Sensors* by Texas Instruments – piezoelectric sensor principle of operation and circuit model, voltage-mode amplifier and charge-mode amplifier circuits:
  `http://www.ti.com/lit/an/sloa033a/sloa033a.pdf`

Part II

myRIO Embedded Systems Kit

17 Keypad

A keypad provides an essential component for a human user interface. Figure 17.1 pictures the the NI myRIO Embedded Systems Kit keypad. The keypad pushbutton switches wired in a 4×4 matrix can be scanned to determine single and multiple key presses.

Figure 17.1: NI myRIO Embedded Systems Kit keypad.

Learning Objectives: You will understand these core concepts related to the keypad after completing the activities in this chapter:
1. Matrix connection
2. Pull-up vs pull-down
3. Single keypress vs multi keypress

17.1 Component Demonstration

Follow these steps to demonstrate correct operation of the keypad.

Select these parts from the NI myRIO Embedded Systems Kit:

- 4x4 matrix keypad `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,940&Prod=PMODKYPD`
- Jumper wires, F-F (9×)

Build the interface circuit: Refer to the schematic diagram shown in Figure 17.2 on page 81; the keypad requires three connections to NI myRIO MXP Connector B:
1. +3.3-volt supply → B/+3.3V (pin 33),
2. Column line 1 → B/DIO0 (pin 11),
3. Column line 2 → B/DIO1 (pin 13),
4. Column line 3 → B/DIO2 (pin 15),
5. Column line 4 → B/DIO3 (pin 17),
6. Row line 1 → B/DIO4 (pin 19),
7. Row line 2 → B/DIO4 (pin 21),

8. Row line 3 → B/DIO4 (pin 23), and
9. Row line 4 → B/DIO4 (pin 25).

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Keypad demo.lvproj` contained in the subfolder `Keypad demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI display the state of the keypad buttons as a 4×4 grid and as a 1-D array of scan results. Try a variety of single-key combinations and confirm that only one LED indicator is activity. Also confirm that the position of the keypad button indicator exactly matches the position of the button you press. Next, try several two-key combinations. Do they all work properly?

Experiment with multi-key combinations involving three or more keys. You should find that some work as expected, i.e., the front-panel indicator properly corresponds to the actual keys pressed, while other combinations produce incorrect display. Try to identify a pattern that explains when a given multi-key combination will or will not lead to an incorrect display.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections, and
- Correct keypad connector terminals — double-check your connections, and ensure that you have not accidentally interchanged the row and column lines.

17.2 Interface Theory

Interface circuit: The keypad contains sixteen single-pole single-throw (SPST) pushbuttons arranged in 4×4 grid; refer to Chapter ??? for a complete treatment of pushbutton interfacing to both the MXP and MSP connectors. Because the keypad normally receives a sequence of single-button presses and occasionally two- or even three-button combination presses, the matrix connection based on common row lines and column lines requires only eight connections to the NI myRIO DIO lines instead of the perhaps more expected sixteen connections that would be required to properly sense all $2^{16} = 65536$ possible switch open-closed patterns.

Study the video "Keypad Interfacing Theory" at `http://youtu.be/oj2-CYSnyo0` to learn more about the matrix approach to keypad switch wiring, the reasoning behind the pull-up resistors on the row lines, the keypad scanning process of driving column lines with a low-level voltages and reading the row lines, and the rule that explains when a multi-key button combination will produce "ghost" keypresses (false reads). Look for the "challenge" offered in the video, and see if you can determine

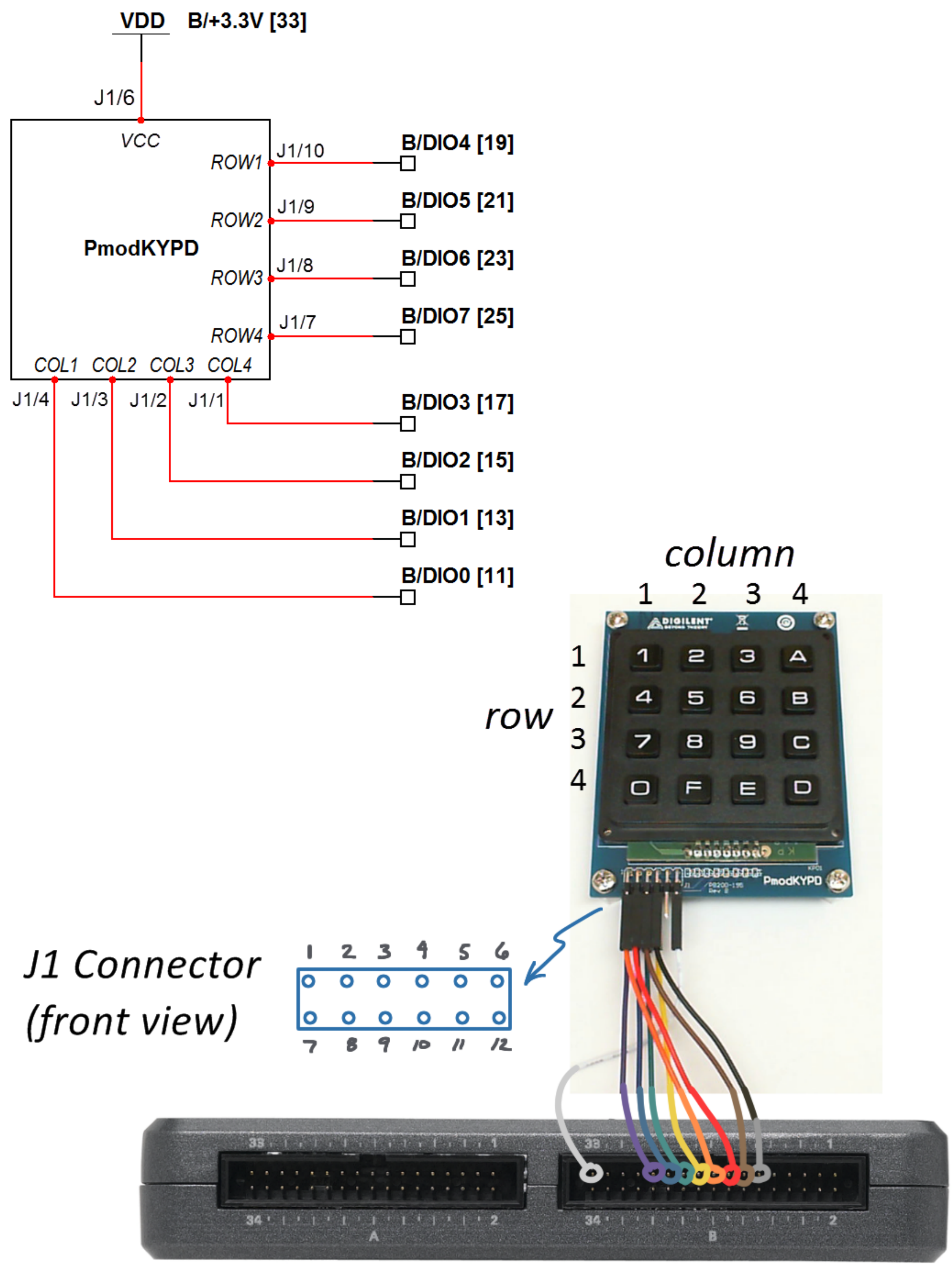


Figure 17.2: Demonstration setup for keypad connected to NI myRIO MXP Connector B.

the maximum possible multi-key combination that does not produce "ghosting."

LabVIEW programming: Study the video "Digital Output Low-Level subVIs" at `http://youtu.be/[TBD]:doVI` to learn how to use the low-level Digital Output subVIs set the DIOs to either high impedance or to a devined voltage level.

17.3 Basic Modifications

Study the video "Keypad Demo Walk-Through" at `http://youtu.be/7r_LwcDa2AM` to learn the design principles of `Keypad demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Temporarily remove the Transpose 2D Array node and observe the new behavior of the Keypad buttons front-panel indicator. Thinking about the Scan display, can you explain how the transpose operation corrects the Keypad buttons information display?
- Add the Boolean Array to Number node just before the Scan indicator connection, and then create a numerical indicator. Try various single-button and multi-button keypress combinations until you understand the relationship between the keypress and the indicated numerical value.
- Create a numerical indicator whose value matches the value printed on each button of the keypad, with the letters corresponding to hexadecimal values; the indicator should display -1 when no key is pressed. Consider using the Boolean Array to Number node and a case structure.
- Re-connect the display to the MSP connector and updated the DIO values accordingly. Note that the MSP connector includes internal pull-down resistors instead of pull-up resistors as on the MXP connectors. You can either rely on the 10 kΩ pull-up resistors on the PmodKYPD to override the approximately 40 kΩ MSP pull-downs (use the MSP 5-volt supply for the PmodKYPD), or you can connect the PmodKYPD "V" terminal to *ground* thereby converting the 10 kΩ resistors to pull-down resistors – this latter connection requires that you also change the column drives to *high* level with a Boolean "T" instead of the current "F" value.

17.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the keypad with other components and devices.

17.5 For More Information

- *PmodKYPD Reference Manual* by Digilent – reference manual for the 4×4 keypad; scroll to the bottom of the page:
  `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,940&Prod=PMODKYPD`
- *PmodKYPD Schematics* by Digilent – schematic diagram of the 4×4 keypad; scroll to the bottom of the page:
  `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,940&Prod=PMODKYPD`

18 LCD Character Display – UART Interface

An LCD character display provides an excellent means for your LabVIEW block diagram code to display measurements, status, and conditions with ASCII character strings. The LCD character display also provides visual feedback for a user interface. Figure 18.1 pictures the NI myRIO Embedded Systems Kit LCD character display which supports three different serial communications standards; this chapter focuses on the UART interface and the next two chapters cover the SPI and I^2C-businterfaces to the display.

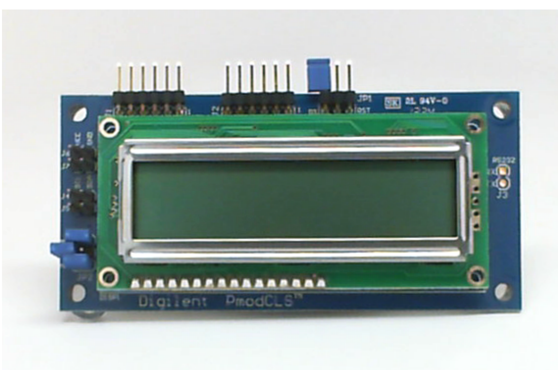


Figure 18.1: NI myRIO Embedded Systems Kit LCD character display.

Learning Objectives: You will understand these core concepts related to the LCD character display after completing the activities in this chapter:

1. Configure the display for UART serial communications at a desired baud rate,
2. Send characters directly to appear on display, and
3. Escape sequences to adjust display modes.

18.1 Component Demonstration

Follow these steps to demonstrate correct operation of the LCD character display.

Select these parts from the NI myRIO Embedded Systems Kit:

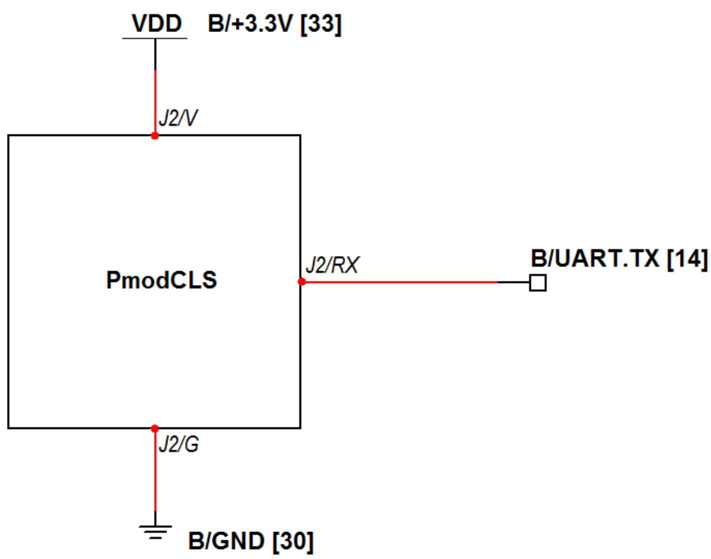
- LCD character display with serial interface `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS`
- Jumper wires, F-F (3×)

Build the interface circuit: Refer to the schematic diagram shown in Figure 18.2 on the next page; the LCD character display requires three connections to NI myRIO MXP Connector B:

1. +3.3-volt supply → B/+3.3V (pin 33),
2. Ground → B/GND (pin 30), and
3. UART receiver → B/UART.TX (pin 14).

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,



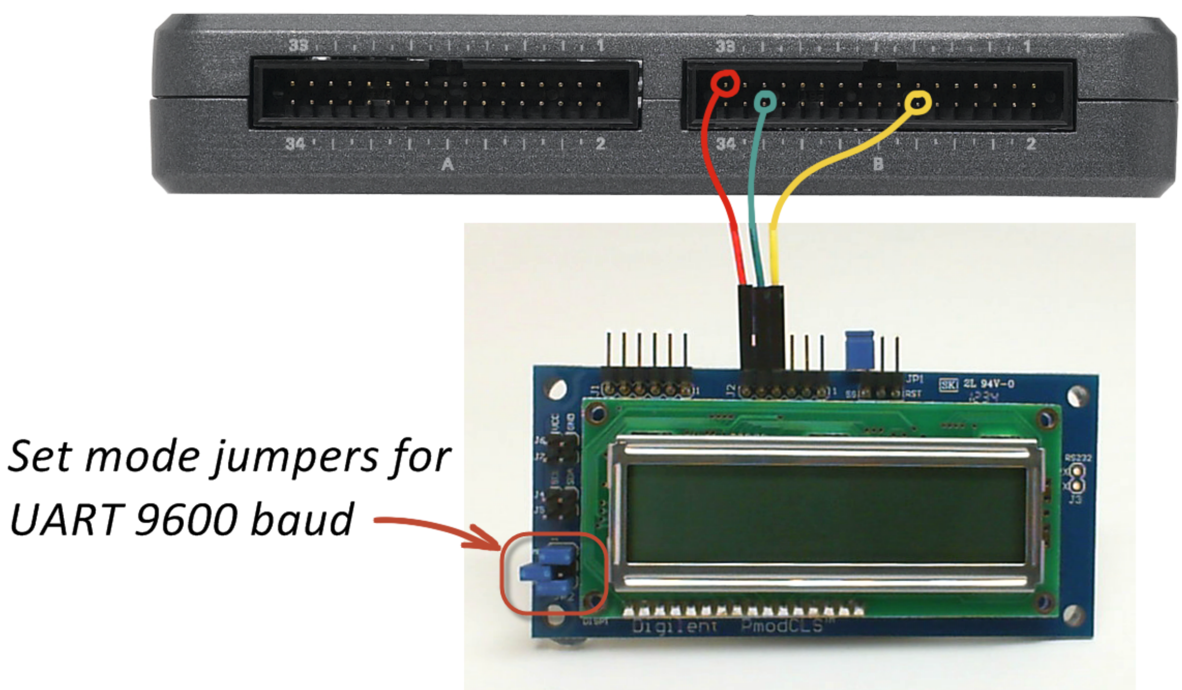


Figure 18.2: Demonstration setup for LCD character display connected to NI myRIO MXP Connector B. Remember to set the mode jumpers as shown for UART serial communications.

- Open the project `LCD (UART) demo.lvproj` contained in the subfolder `LCD (UART) demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the state of the NI myRIO on-board 3-axis accelerometer as three values (X, Y, and Z directions) and the state of the built-in pushbutton on the bottom side of myRIO; press the button to see the state change from 0 to 1. Change the orientation of myRIO to see the accelerometer values change, and shake myRIO to see larger acceleration values. You may wish to hold or tape the LCD character display to the top of myRIO, too.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct LCD character display connector terminals — double-check your connections, and ensure that you have connected the NI myRIO UART "transmit" output to the LCD character display "receive" input; also check that you have not accidentally crossed the power supply connections, and
- Correct mode jumper settings – refer again to the circled region of Figure 18.2 on the facing page for the proper settings.

18.2 Interface Theory

Interface circuit: The LCD character display supports three serial communication standards: UART (univeral asynchronous receiver-transmitter), SPI (serial peripheral interface), and I^2C-bus (inter-IC). This chapter concentrates on the UART interface while the next two chapters cover the other interfaces; the functionality of the LCD character display itself remains independent of the selected communication standard. Simply send ASCII characters to the display's UART receive line to make them appear on the display screen. Use "escape sequences" to configure other aspects of the display such as homing the cursor, displaying and blinking the cursor, scrolling the display, and so on.

Study the video "LCD Character Display Interfacing Theory" at `http://youtu.be/m0Td7KbhvdI` to learn more about the LCD character display operation with UART interface including setting the baud rate, sending display information and escape sequences, and available configuration instructions. Study "Serial Communication: UART" at `http://youtu.be/p7CPyYRS8I8` to understand how the UART Express VI configuration options relate to the signaling waveforms between UART transmitters and receivers.

LabVIEW programming: Study the video "UART Express VI" at `http://youtu.be/0FMnkFDsGQs` to learn how to use the UART Express VI to read and write character strings, including strings with special characters.

18.3 Basic Modifications

Study the video "LCD (UART) Demo Walk-Through" at `http://youtu.be/JsEMMnIWg4k` to learn the design principles of `LCD (UART) demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Interchange the two display rows when the NI myRIO on-board button is pressed; revert to the original display when the button is released.
- Display the full character set – Cycle through all possible 256 8-bit patterns... see if you can spot the ASCII code for the "degrees" symbol
- Create an instruction tester – Make two string controls, one for the instruction and a second for the display; use Boolean controls to disable sending either one

18.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the LCD character display with other components and devices.

18.5 For More Information

- *PmodCLS Reference Manual* by Digilent – reference manual for the LCD character display: `http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS_rm_RevD-E.pdf`
- *PmodCLS Schematics* by Digilent – schematic diagram of the LCD character display: `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS`

19 LCD Character Display – SPI Interface

An LCD character display provides an excellent means for your LabVIEW block diagram code to display measurements, status, and conditions with ASCII character strings. The LCD character display also provides visual feedback for a user interface. Figure 19.1 pictures the NI myRIO Embedded Systems Kit LCD character display which supports three different serial communications standards. This chapter focuses on the SPI interface to the display, while the previous chapter covers the UART interface and the next chapter covers the I^2C-bus interface.

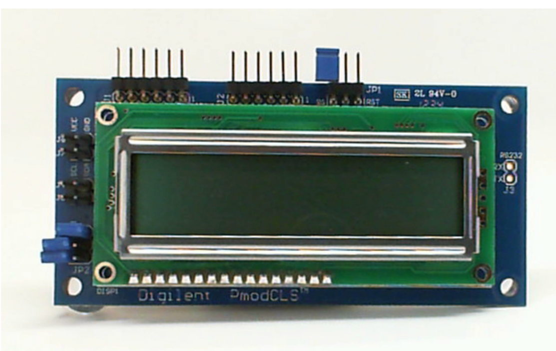


Figure 19.1: NI myRIO Embedded Systems Kit LCD character display.

Learning Objectives: You will understand these core concepts related to the LCD character display after completing the activities in this chapter:

1. Configure the display for SPI serial communications at a desired clock frequency,
2. Send characters directly to appear on display, and
3. Escape sequences to adjust display modes.

19.1 Component Demonstration

Follow these steps to demonstrate correct operation of the LCD character display.

Select these parts from the NI myRIO Embedded Systems Kit:

- LCD character display with serial interface `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS`
- Jumper wires, F-F (4×)

Build the interface circuit: Refer to the schematic diagram shown in Figure 19.2 on the following page; the LCD character display requires four connections to NI myRIO MXP Connector B:

1. +3.3-volt supply → B/+3.3V (pin 33),
2. Ground → B/GND (pin 30),
3. SPI receiver → B/SPI.MOSI (pin 25), and
4. SPI clock → B/SPI.CLK (pin 21).

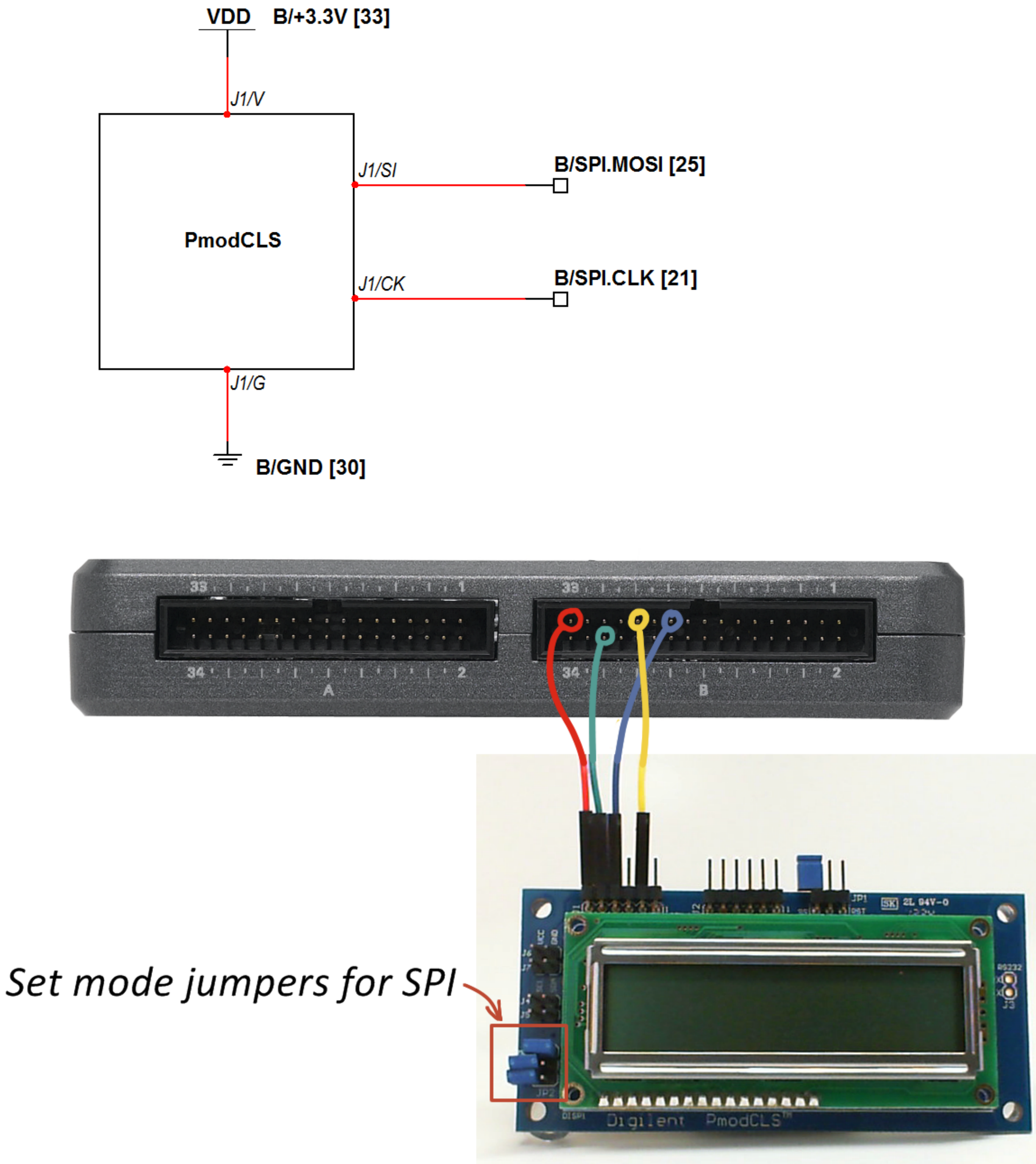


Figure 19.2: Demonstration setup for LCD character display connected to NI myRIO MXP Connector B. Remember to set the mode jumpers as shown for SPI serial communications.

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `LCD (SPI) demo.lvproj` contained in the subfolder `LCD (SPI) demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the state of the NI myRIO on-board 3-axis accelerometer as three values (X, Y, and Z directions) and the state of the built-in pushbutton on the bottom side of myRIO; press the button to see the state change from 0 to 1. Change the orientation of myRIO to see the accelerometer values change, and shake myRIO to see larger acceleration values. You may wish to hold or tape the LCD character display to the top of myRIO, too.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct LCD character display connector terminals — double-check your connections, and ensure that you have connected the NI myRIO SPI "MOSI" output to the LCD character display "MOSI" input; also check that you have not accidentally crossed the power supply connections, and
- Correct mode jumper settings – refer again to the circled region of Figure 19.2 on the preceding page for the proper settings.

19.2 Interface Theory

Interface circuit: The LCD character display supports three serial communication standards: UART (univeral asynchronous receiver-transmitter), SPI (serial peripheral interface), and I^{2}C-bus (inter-IC). This chapter concentrates on the SPI interface while the adjacent chapters cover the other interfaces; the functionality of the LCD character display itself remains independent of the selected communication standard. Simply send ASCII characters to the display's SPI receive line to make them appear on the display screen. Use "escape sequences" to configure other aspects of the display such as homing the cursor, displaying and blinking the cursor, scrolling the display, and so on.

Study the video "LCD Character Display Interfacing Theory" at `http://youtu.be/m0Td7KbhvdI` to learn more about the LCD character display operation including setting the baud rate, sending display information and escape sequences, and available configuration instructions. Study "Serial Communication: SPI" at `http://youtu.be/GaXtDamw5As` to understand how the SPI Express VI configuration options relate to the signaling waveforms between SPI transmitters and receivers.

LabVIEW programming: Study the video "SPI Express VI" at `http://youtu.be/S7KkTeMfmc8` to learn how to use the SPI Express VI to read and write character strings, including strings with special characters; the video "UART Express VI" at `http://youtu.be/0FMnkFDsGQs` provides additional useful string formatting techniques.

19.3 Basic Modifications

Study the video "LCD (SPI) Demo Walk-Through" at `http://youtu.be/oOXYryu4Y-c` to learn the design principles of `LCD (SPI) demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Interchange the two display rows when the NI myRIO on-board button is pressed; revert to the original display when the button is released.
- Display the full character set – Cycle through all possible 256 8-bit patterns... see if you can spot the ASCII code for the "degrees" symbol
- Create an instruction tester – Make two string controls, one for the instruction and a second for the display; use Boolean controls to disable sending either one
- Experiment with the serial clock frequency: what is the highest possible frequency that still provides reliable communication with the LCD? What is the lowest frequency that still updates the LCD display without introducing noticeable lag? Remember to change the speed of *both* I2C Express VIs. NOTE: You must power cycle the LCD display each time you change the serial clock frequency because while the display automatically adapts to the SPI clock frequency, it does so only one time.

19.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the LCD character display with other components and devices.

19.5 For More Information

- *PmodCLS Reference Manual* by Digilent – reference manual for the LCD character display: `http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS_rm_RevD-E.pdf`
- *PmodCLS Schematics* by Digilent – schematic diagram of the LCD character display: `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS`
- *M68HC11 Reference Manual* by Freescale Semiconductor – refer to Section 8 for a complete treatment of the SPI serial bus standard, including timing diagrams and multi-master systems: `http://www.freescale.com/files/microcontrollers/doc/ref_manual/M68HC11RM.pdf`

20 LCD Character Display – I^{2}C-bus Interface

An LCD character display provides an excellent means for your LabVIEW block diagram code to display measurements, status, and conditions with ASCII character strings. The LCD character display also provides visual feedback for a user interface. Figure 20.1 pictures the NI myRIO Embedded Systems Kit LCD character display which supports three different serial communications standards. This chapter focuses on the I^{2}C-bus interface to the display, while the previous two chapters cover the UART and SPI interfaces.

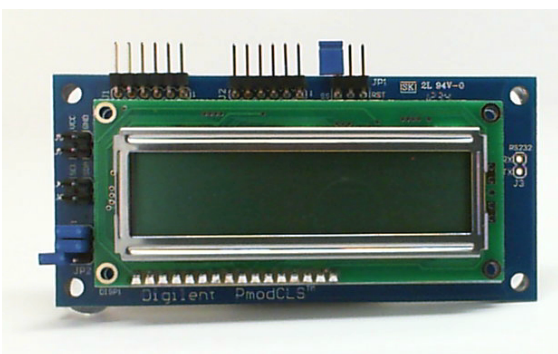


Figure 20.1: NI myRIO Embedded Systems Kit LCD character display.

Learning Objectives: You will understand these core concepts related to the LCD character display after completing the activities in this chapter:

1. Configure the display for I^{2}C-bus serial communications,
2. Send characters directly to appear on display, and
3. Escape sequences to adjust display modes.

20.1 Component Demonstration

Follow these steps to demonstrate correct operation of the LCD character display.

Select these parts from the NI myRIO Embedded Systems Kit:

- LCD character display with serial interface `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS`
- Jumper wires, F-F (4×)

Build the interface circuit: Refer to the schematic diagram shown in Figure 20.2 on the next page; the LCD character display requires four connections to NI myRIO MXP Connector B:

1. +3.3-volt supply → B/+3.3V (pin 33),
2. Ground → B/GND (pin 30),
3. Serial data (SD) → B/I2C.SDA (pin 34), and
4. Serial clock (SC) → B/I2C.SCL (pin 32).

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done

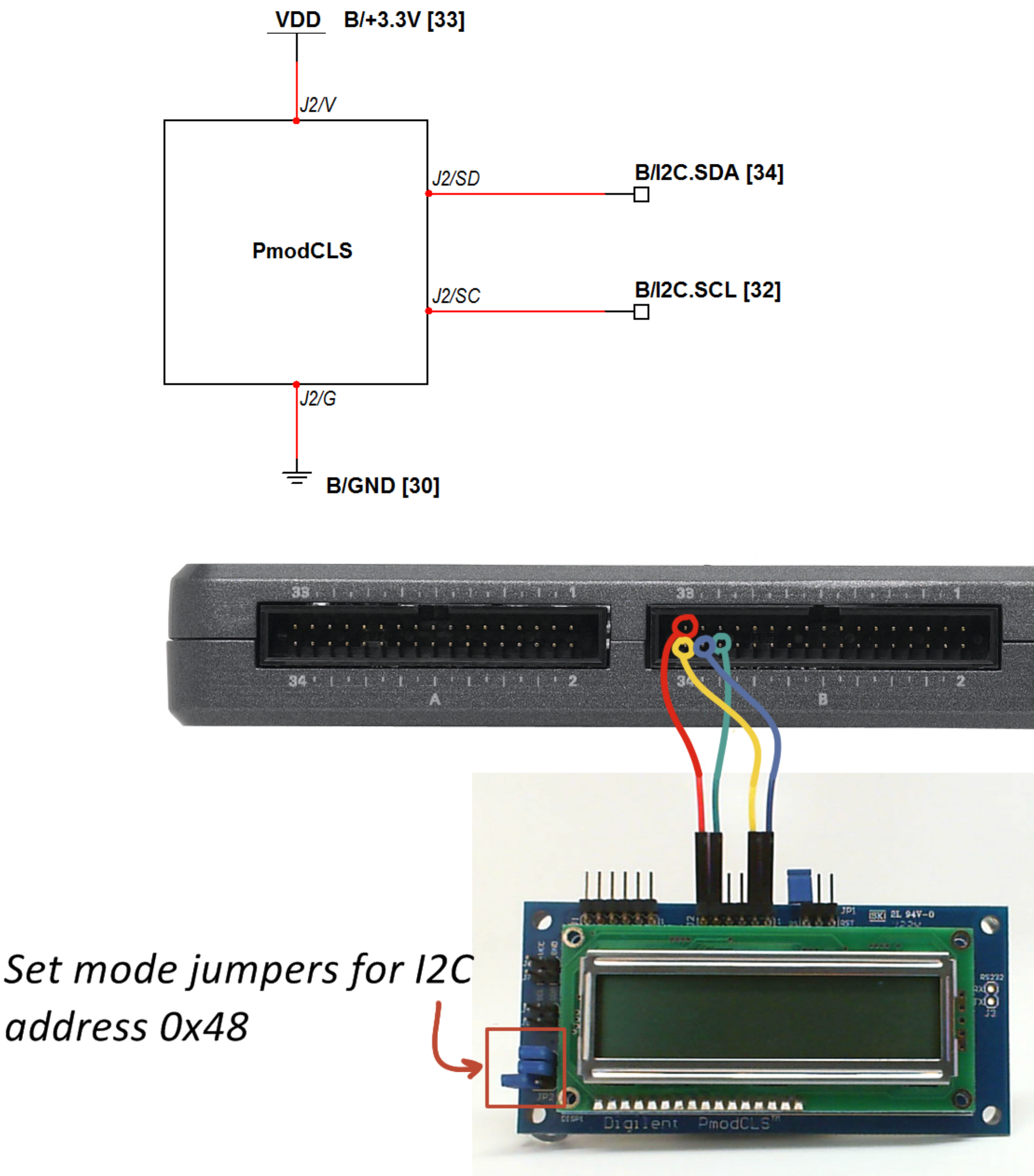


Figure 20.2: Demonstration setup for LCD character display connected to NI myRIO MXP Connector B. Remember to set the mode jumpers as shown for I^{2}C-bus serial communications.

so previously and unpack the contents to a convenient location,

- Open the project `LCD (I2C) demo.lvproj` contained in the subfolder `LCD (I2C) demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI displays the state of the NI myRIO on-board 3-axis accelerometer as three values (X, Y, and Z directions) and the state of the built-in pushbutton on the bottom side of myRIO; press the button to see the state change from 0 to 1. Change the orientation of myRIO to see the accelerometer values change, and shake myRIO to see larger acceleration values. You may wish to hold or tape the LCD character display to the top of myRIO, too.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct LCD character display connector terminals — double-check your connections, and ensure that you have connected the NI myRIO I²C-bus "SDA" line to the LCD character display "SD" terminal on connector J2 and the "SCL" line to the "SC" terminal; also check that you have not accidentally crossed the power supply connections, and
- Correct mode jumper settings – refer again to the circled region of Figure 20.2 on the facing page for the proper settings.

> **NOTE**: *Double-check the "SDA" and "SCL" connections should you see the message "Error -36011 occurred at myRIO Write I2C.vi" or similar; this message indicates that NI myRIO did not receiving an expected acknowledgement from the LCD character display I²C-bus interface.*

20.2 Interface Theory

Interface circuit: The LCD character display supports three serial communication standards: UART (univeral asynchronous receiver-transmitter), SPI (serial peripheral interface), and I²C-bus (inter-IC). This chapter concentrates on the I²C-bus interface (commonly denoted "I2C") while the previous two chapters cover the other interfaces; the functionality of the LCD character display itself remains independent of the selected communication standard. Simply send ASCII characters to the display's I²C-bus receive line to make them appear on the display screen. Use "escape sequences" to configure other aspects of the display such as homing the cursor, displaying and blinking the cursor, scrolling the display, and so on.

Study the video "LCD Character Display Interfacing Theory" at `http://youtu.be/m0Td7KbhvdI` to learn how to send display information, escape sequences, and configuration instructions to the LCD character display. Study "Serial Communication: I2C" at `http://youtu.be/7CgNF78pYQM` to understand how the I2C Express VI configuration options relate to the signaling waveforms between I^2C-bus transmitters and receivers.

LabVIEW programming: Study the video "I2C Express VI" at `http://youtu.be/[TBD]:iicExVI` to learn how to use the I2C Express VI to read and write character strings, including strings with special characters; the video "UART Express VI" at `http://youtu.be/0FMnkFDsGQs` provides additional useful string formatting techniques.

EDIT: wait on this video until i2c eeprom is available for demo

20.3 Basic Modifications

Study the video "LCD (I2C) Demo Walk-Through" at `http://youtu.be/qbD31AeqOMk` to learn the design principles of `LCD (I2C) demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Interchange the two display rows when the NI myRIO on-board button is pressed; revert to the original display when the button is released.
- Display the full character set – Cycle through all possible 256 8-bit patterns... see if you can spot the ASCII code for the "degrees" symbol
- Create an instruction tester – Make two string controls, one for the instruction and a second for the display; use Boolean controls to disable sending either one
- Experiment with the serial clock frequency; does the LCD character display appear to support both "Standard mode" and "Fast mode" frequencies? Remember to change the speed of *both* I2C Express VIs.
- Try disconnecting either I^2C-bus line from the LCD character display. What LabVIEW error message do you see?
- Change the "slave" address to some other value. What LabVIEW error message do you see?

20.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the LCD character display with other components and devices.

20.5 For More Information

- *PmodCLS Reference Manual* by Digilent – reference manual for the LCD character display:
  `http://digilentinc.com/Data/Products/PMOD-CLS/PmodCLS_rm_RevD-E.pdf`
- *PmodCLS Schematics* by Digilent – schematic diagram of the LCD character display:
  `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,473&Prod=PMOD-CLS`
- *UM10204 I^2C-bus Specification and User Manual* by NXP Semiconductors – a complete treatment of the I^2C-bus standard, including timing diagrams and multi-master systems:
  `http://www.nxp.com/documents/user_manual/UM10204.pdf`

21 Digital Potentiometer

A digital potentiometer is a three-terminal variable resistor that behaves just like its mechanical analog potentiometer counterpart (see Chapter 7 on page 33) but with a digital numerical value to set the virtual knob position. The digital potentiometer provides a convenient replacement for mechanical potentiometers, offers computer-adjustable gain control for amplifiers, and software adjustment of trim potentiometers used to null resistive sensor circuit offsets. Figure 21.1 pictures the NI myRIO Embedded Systems Kit digital potentiometer with 10 kΩ end-to-end resistance and 8-bit resolution; SPI serial bus conveys the digital wiper position.

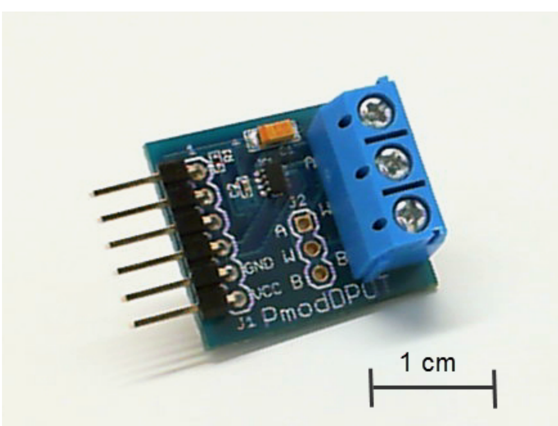


Figure 21.1: NI myRIO Embedded Systems Kit digital potentiometer.

Learning Objectives: You will understand these core concepts related to the digital potentiometer after completing the activities in this chapter:

1. Adjust the digital potentiometer wiper position with SPI serial communications,
2. Properly connect and operate the digital potentiometer as either a rheostat or as a potentiometer (voltage divider), and
3. Principles of the "virtual wiper" as implemented by an array of solid-state switches.

21.1 Component Demonstration

Follow these steps to demonstrate correct operation of the digital potentiometer.

Select these parts from the NI myRIO Embedded Systems Kit:

- Digital potentiometer `http://digilentinc.com/Products/Detail.cfm?NavPath=2,401,1075&Prod=PMOD-DPOT`
- Jumper wires, F-F (5×)
- Jumper wires, M-F (3×)
- Small screwdriver

Build the interface circuit: Refer to the schematic diagram shown in Figure 21.2 on page 97; the digital potentiometer requires four connections to NI myRIO MXP Connector B:

1. +5-volt supply → A/+5V (pin 1),

2. Ground → A/GND (pin 6),
3. SPI receiver → A/SPI.MOSI (pin 25),
4. SPI clock → A/SPI.CLK (pin 21),
5. Chip select → A/DIO0 (pin 11),
6. "A" → B/+5V (pin 1),
7. "B" → B/GND (pin 6),
8. "W" → B/AI0 (pin 3),

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `Dpot demo.lvproj` contained in the subfolder `Dpot demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI provides a front-panel slide control to set the virtual wiper position as an 8-bit value. The five-volt power supply connected across the end terminals of the potentiometer creates a proportional variable voltage at the wiper terminal "W" which is read by an anlog input and displayed on the indicator dial. Move the slide and you should see a corresponding change on the dial position. Press the page-up and page-down keys to make single-bit changes to the digital value.

If you have an ohmmeter handy, disconnect all three potentiometer terminals from NI myRIO and then measure the resistance between the "W" and "B" terminals as you vary the digital wiper position; repeat for the "W" and "A" terminals. If the measurement does not seem sufficiently stable, try connecting either "A" or "B" to one of the NI myRIO ground terminals.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector A to power the digital potentiometer board and Connector B to establish the potentiometer variable voltage, and
- Correct SPI connector terminals — double-check your connections, and ensure that you have connected the NI myRIO SPI "MOSI" output to the digital potentiometer "SDI" input and digital output DIO0 to the chip select input.

21.2 Interface Theory

Interface circuit: The Digilent PmodDPOT board provides a convenient set of interface connectors for the Analog Devices AD5160 digital potentiometer. The digital potentiometer provides the conventional trio of terminals like a mechanical potentiometer, and an 8-bit value between 0 and 255 transmitted via SPI (serial peripheral interface) sets the virtual wiper position by closing exactly one of 256 solid-state switches that establish the connection point to a string of 256 equal-valued resistors between terminals "A" and "B."

Study the video "Digital Potentiometer Interfacing Theory" at `http://youtu.be/C4iBQjWn7OI` to learn more about the digital potentiometer theory of operation including SPI bus and chip select timing,

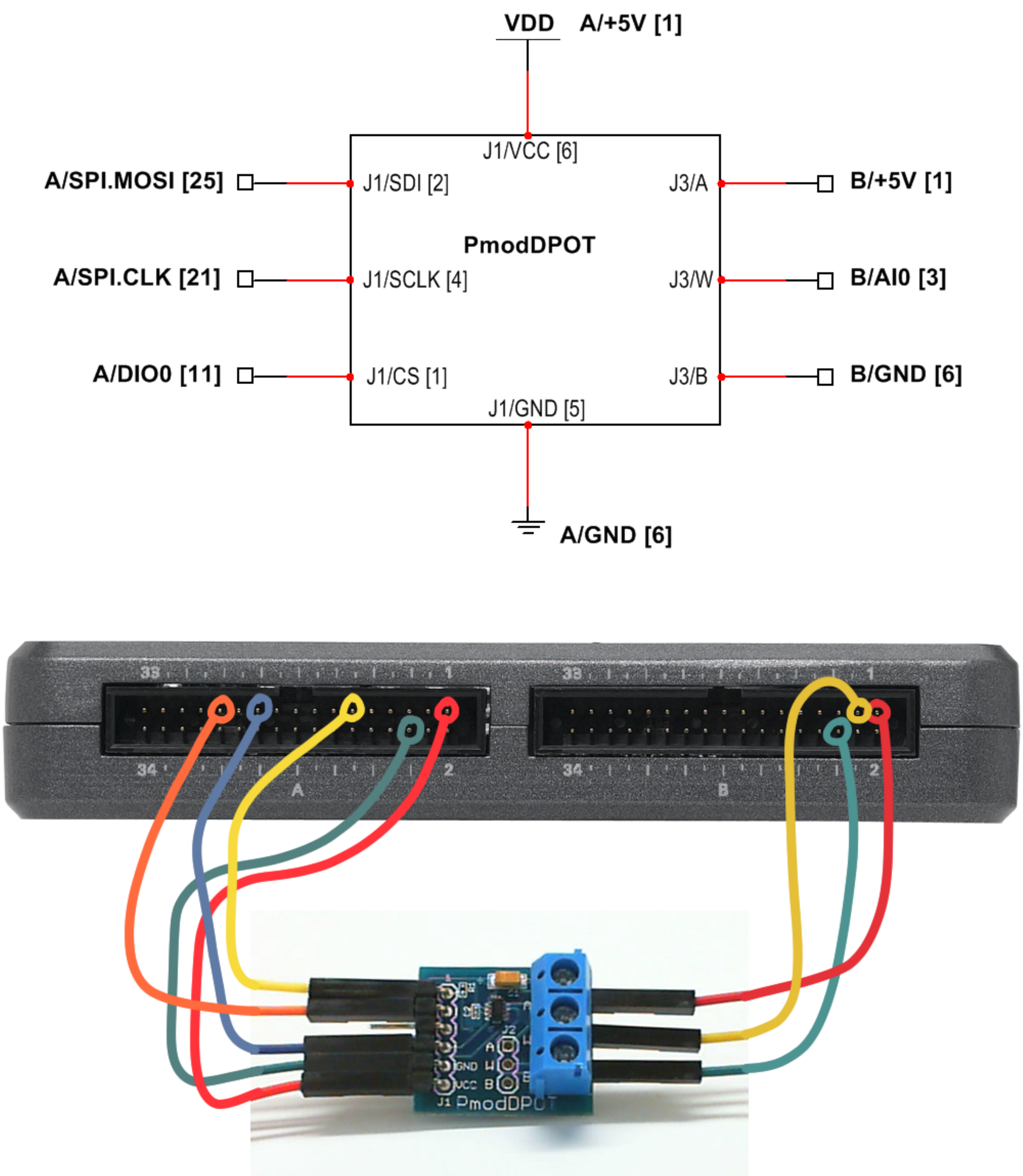


Figure 21.2: Demonstration setup for digital potentiometer connected to NI myRIO MXP Connector B. Use the M-F jumper wires and a screwdriver for three potentiometer terminals.

internal switch array circuit, and design equations for using the digital potentiometer in rheostate mode (a single variable resistance) and in potentiometer mode (a voltage divider providing an adjustable voltage). Study "Serial Communication: SPI" at `http://youtu.be/GaXtDamw5As` to understand how the SPI Express VI configuration options relate to the signaling waveforms between SPI transmitters and receivers.

LabVIEW programming: Study the video "SPI Express VI" at `http://youtu.be/S7KkTeMfmc8` to learn how to use the SPI Express VI.

21.3 Basic Modifications

Study the video "Dpot Demo Walk-Through" at `http://youtu.be/dtwXOj5vvy4` to learn the design principles of `Dpot demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Interchange the "A" and "B" connections, and confirm that the analog voltage decreases as you increase the digital value.
- Evaluate the linearity of the digital potentiometer: Change the while-loop structure to a for-loop, create an array of the analog voltage at each digital value, and then plot the analog voltage as a function of digital voltage.
- Continue the linearity evaluation of the previous step by plotting the difference of the measured analog voltage and the ideal analog voltage. This difference plot makes it much easier to identify any trends of nonlinearity.

21.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at `http://www.ni.com/myrio/project-guide` for interesting systems integration project ideas that combine the digital potentiometer with other components and devices.

21.5 For More Information

- *PmodDPOT Reference Manual* by Digilent – reference manual for the digital potentiometer board: `http://digilentinc.com/Data/Products/PMOD-DPOT/PmodDPOT_rm.pdf`
- *PmodCLS Schematics* by Digilent – schematic diagram of the digital potentiometer board: `http://digilentinc.com/Data/Products/PMOD-DPOT/PmodDPOT_sch.pdf`
- *AD5160 Data Sheet* by Analog Devices – complete information on the AD5160 that serves as the heart of the digital potentiometer board: `http://www.analog.com/ad5160`
- *M68HC11 Reference Manual* by Freescale Semiconductor – refer to Section 8 for a complete treatment of the SPI serial bus standard, including timing diagrams and multi-master systems: `http://www.freescale.com/files/microcontrollers/doc/ref_manual/M68HC11RM.pdf`

22 RFID Reader

RFID (radio-frequency identification) offers a non-contact scanning method for access control and inventory management. An RFID reader queries an RFID tag to determine the unique bit pattern encoded in the tag. Passive RFID tags power their on-board electronics from the magnetic field generated by the reader. Figure 22.1 shows the NI myRIO Embedded Systems Kit RFID reader along with the breakout board that converts the 2 mm pin spacing to standard 0.1 inch pin spacing for breadboards.

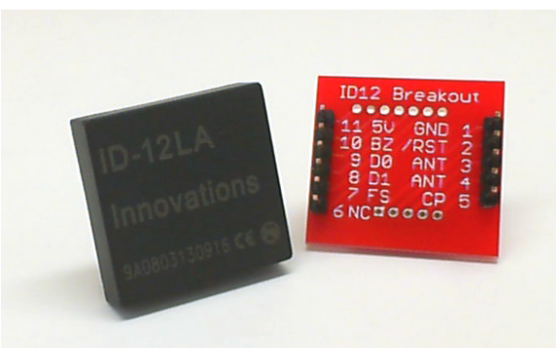


Figure 22.1: NI myRIO Embedded Systems Kit RFID reader with breakout board.

Learning Objectives: You will understand these core concepts related to the RFID reader after completing the activities in this chapter:

1. EM4001 RFID tag standard,
2. ID-12LA UART output format, and
3. Checksum calculation.

22.1 Component Demonstration

Follow these steps to demonstrate correct operation of the RFID reader.

Select these parts from the NI myRIO Embedded Systems Kit:

- ID-Innovations ID-12LA RFID reader `http://www.hobbytronics.co.uk/datasheets/sensors/ID-12LA-ID-20LA.pdf`
- Breakout board for ID-12LA RFID reader `http://www.sparkfun.com/products/8423`
- Breakaway header, straight pin `https://www.sparkfun.com/products/116`
- RFID tag, 125 kHz, EM4001 format (2×) `https://www.sparkfun.com/products/8310`
- LED [needs info]
- Jumper wires, M-F (4×)

You will also need access to a soldering station.

Build the interface circuit: The RFID reader breakout board requires soldering. Refer to Figure 22.2 on the next page to see how the reader and breakout board should look when finished. Begin by breaking off a 5-pin section and a 6-pin section of the breakaway header. Place the headers in a breadboard (longer pin down) to keep them properly aligned, and then fit the breakaway board over the headers *with the pin numbers facing down* and the SparkFun logo facing up. Solder the short pins in place. When cool, remove the breakout board and then fit the RFID reader onto the breakout board. Solder the RFID reader pins on the other side of the breakout board.

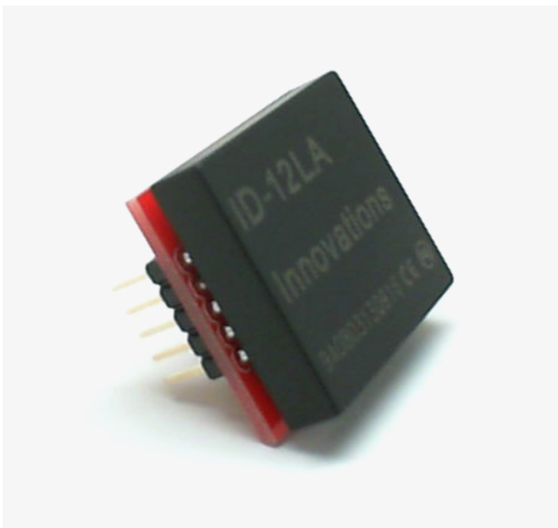


Figure 22.2: RFID reader, breakout board, and headers after soldering.

Refer to the schematic diagram and recommended breadboard layout shown in Figure 22.3 on the facing page. Note that the compact layout requires several breadboard jumper connections *under* the RFID reader. The RFID reader requires four connections to NI myRIO MXP Connector B:

1. +3.3-volt supply → B/+3.3V (pin 33),
2. Ground → B/GND (pin 30),
3. UART output D0 → B/UART.RX (pin 10), and
4. Tag-in-range → B/DIO0 (pin 11), and

Run the demonstration VI:

- Download `http://www.ni.com/academic/myrio/project-guide-vis.zip` if you have not done so previously and unpack the contents to a convenient location,
- Open the project `RFID demo.lvproj` contained in the subfolder `RFID demo`,
- Expand the hierarchy button (a plus sign) for the myRIO item and then open `Main.vi` by double-clicking,
- Confirm that NI myRIO is connected to your computer, and
- Run the VI either by clicking the Run button on the toolbar or by pressing Ctrl+R.

Expect to see a "Deployment Process" window showing how the project compiles and deploys (downloads) to NI myRIO before the VI starts running. NOTE: You may wish to select the "Close on successful completion" option to make the VI start automatically.

Expected results: The demo VI includes three main indicators:

1. tag detected lights anytime a valid RFID tag is within range of the RFID reader,
2. RFID tag bytes displays the sixteen bytes transmitted by the RFID reader corresponding to a valid RFID tag, and
3. data string extracts the data section of the message and displays it as a hexadecimal numerical value.

Hold one RFID tag card (Figure 22.4 on page 102)near the RFID reader and you should see the tag detected indicator light up for as long as you hold the card sufficiently close. The data string indicator should show a five-digit hexadecimal number (contains digits 0 to 9 and A to F), and the RFID tag bytes indicator should should a sequence beginning with 0x02 (ASCII "start of text" character) and end with 0x03 (ASCII "end of text" character).

Try the other RFID tag card and confirm that you see a different value for the data string. Also, experiment with the minimum required distance for the RFID reader to scan the tag. Does waving the card or moving the card quickly make any difference?

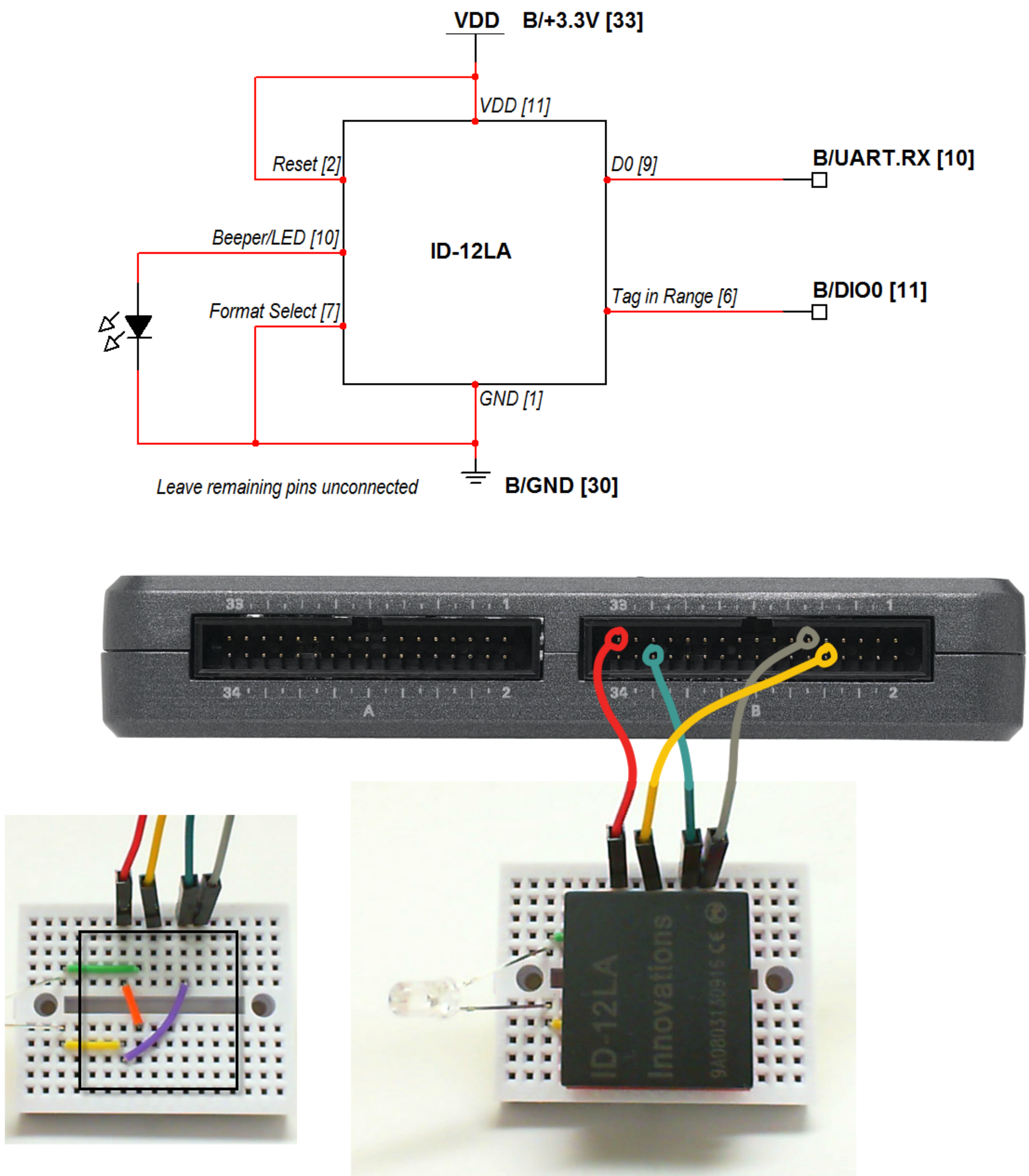


Figure 22.3: Demonstration setup for RFID reader connected to NI myRIO MXP Connector B. Note that the compact layout requires several breadboard jumper connections *under* the RFID reader.

Click the Stop button or press the escape key to stop the VI and to reset NI myRIO.

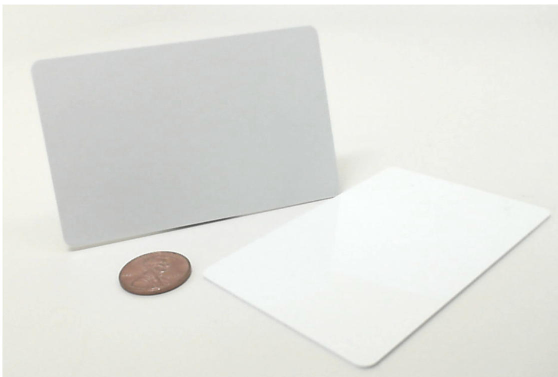


Figure 22.4: NI myRIO Embedded Systems Kit RFID card transponders.

Troubleshooting tips: Not seeing the expected results? Confirm the following points:

- Glowing power indicator LED on NI myRIO,
- Black Run button on the toolbar signifying that the VI is in run mode,
- Correct MXP connector terminals — ensure that you are using Connector B and that you have the correct pin connections,
- Correct RFID reader terminals — double-check your connections, and ensure that have connected the NI myRIO UART "receive" input to the RFID reader "D0" output; also check that you have not accidently crossed the power supply connections, and
- Hidden jumpers — be certain that you have addeded the hidden jumper wires as shown in Figure 22.3 on the preceding page.

22.2 Interface Theory

Interface circuit: The RFID reader uses an electromagnetic field to supply power to the RFID tag *and* to receive a transmitted digital signal from the tag. This noncontact method is popular for ID badges and inventory control. The ID-Innovations ID-12LA RFID reader reads RFID tags encoded with the EM4001 standard and formats the tag information into one of three formats, one of which is compatible with the NI myRIO UART serial communications port.

Study the video "RFID Reader Interfacing Theory" at `http://youtu.be/z1v0vCue83c` to learn more about the RFID reader principles of operation, including the EM4001 tag standard, ID-12LA RFID reader configuration and UART data output format, and checksum calculation.

LabVIEW programming: Study the video "UART Low-Level SubVIs" at `http://youtu.be/[TBD]:uartVI` to learn how to use the UART low-level subVIs to read and write serial communications data.

22.3 Basic Modifications

Study the video "LCD (UART) Demo Walk-Through" at `http://youtu.be/JsEMMnIWg4k` to learn the design principles of `RFID demo`, and then try making these modifications to the block diagram of `Main.vi`:

- Calculate the checksum for the data segment and compare to the checksum field of the RFID tag message; use a Boolean indicator to show whether or not a valid message was received from the RFID reader.

22.4 Project Ideas

COMING SOON: Download the latest edition of the *NI myRIO Project Essentials Guide* at http://www.ni.com/myrio/project-guide for interesting systems integration project ideas that combine the RFID reader with other components and devices.

22.5 For More Information

- *EM4001 Protocal Description* by Priority 1 Design – detailed tutorial on the EM4001 protocol used by the RFID tag cards included with the NI Embedded Systems Kit:
  http://www.priority1design.com.au/em4100_protocol.html
- *RFID Made Easy (AppNote 411)* by EM Microelectronic – everything you need to know about RFID: system principles, electromagnetic theory, antenna design, and data coding techniques:
  http://www.emmicroelectronic.com/webfiles/Product/RFID/AN/AN411.pdf

A MXP and MSP Connector Diagrams

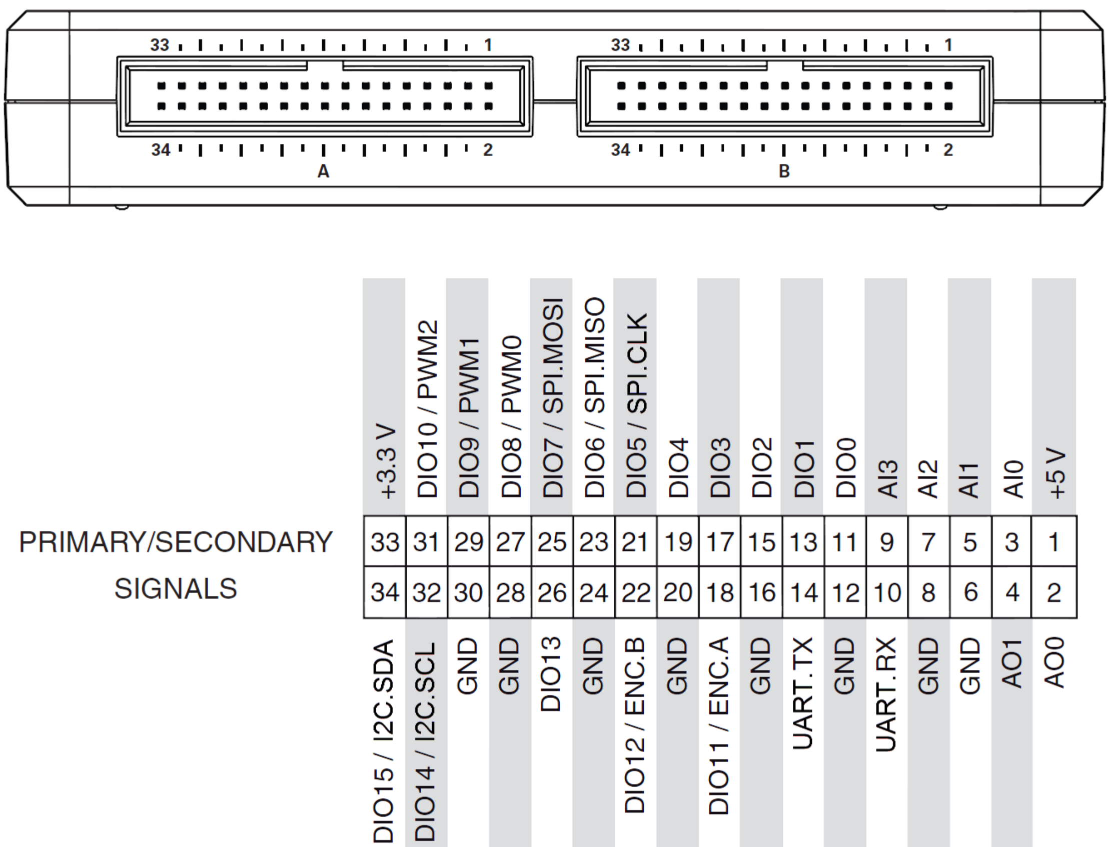


Figure A.1: MXP (**m**yRIO e**X**pansion **P**ort) connector diagram.

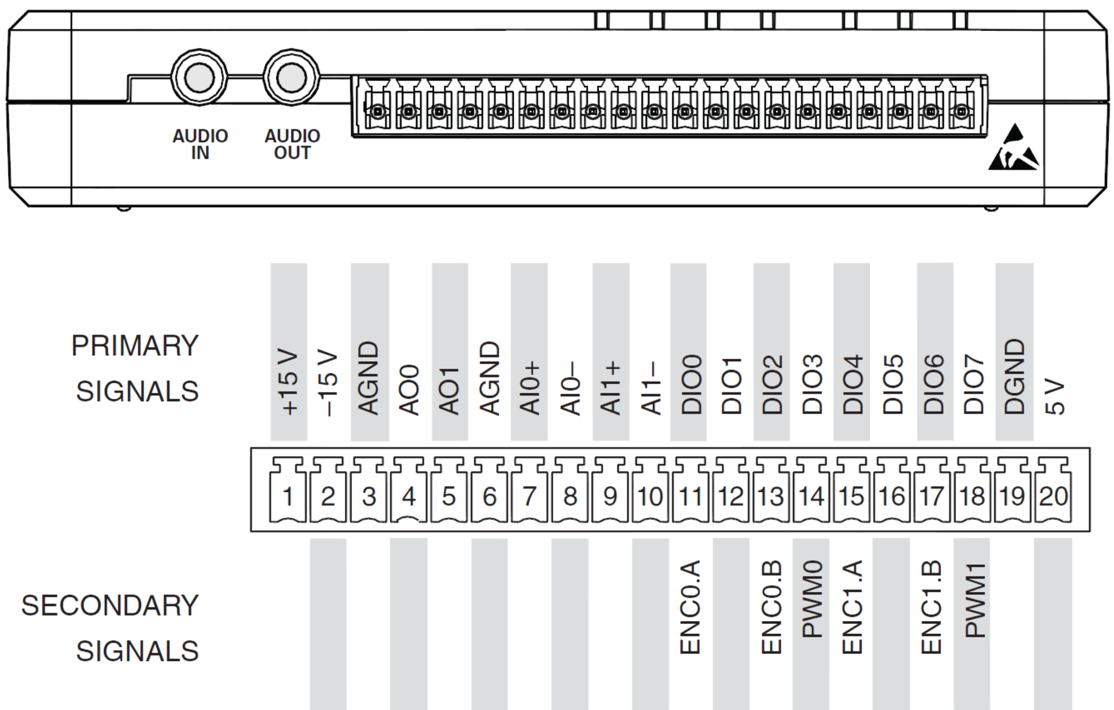


Figure A.2: MSP (**m**ini**S**ystem **P**ort) connector diagram.

B NI myRIO Starter Kit Data Sheets

- US1881 Hall-effect latch http://www.melexis.com/Hall-Effect-Sensor-ICs/Hall-Effect-Latches/US1881-140.aspx
- DT-series piezo film sensor http://meas-spec.com/product/t_product.aspx?id=2478
- Rotary encoder [TBD]
- SPDT switch [TBD]
- DIP switch [TBD]
- Rotary DIP switch [TBD]
- Pushbutton switch [TBD]
- Photointerrupter http://sharpmicroelectronics.com/download/gp1a57hr-epdf
- 2N3904 npn transistor http://www.fairchildsemi.com/ds/MM/MMBT3904.pdf
- 2N3906 pnp transistor http://www.fairchildsemi.com/ds/2N/2N3906.pdf
- ZVN2110A n-channel enhancement-mode MOSFET http://www.diodes.com/datasheets/ZVN2110A.pdf
- ZVP2110A p-channel enhancement-mode MOSFET http://www.diodes.com/datasheets/ZVP2110A.pdf
- IRF510 n-channel enhancement mode power MOSFET http://www.vishay.com/docs/91015/sihf510.pdf
- 1N3064 small-signal diode http://www.fairchildsemi.com/ds/1N/1N3064.pdf
- ADMP504 ultra-low-noise microphone http://www.analog.com/ADMP504
- OP37 low-noise precision high-speed op amp http://www.analog.com/OP37
- AD8541 rail-to-rail single-supply op amp http://www.analog.com/AD8541
- AD22100 temperature sensor http://www.analog.com/AD22100
- 10 kΩ thermistor, EPCOS B57164K103J http://www.epcos.com/inf/50/db/ntc_09/LeadedDisks__B57164__K164.pdf
- 10 kΩ potentiometer [TBD]
- Photocell, API PDV-P9203 http://www.advancedphotonix.com/ap_products/pdfs/PDV-P9203.pdf
- General-purpose rectifier http://www.vishay.com/docs/88503/1n4001.pdf
- Buzzer/speaker, Soberton GT-0950RP3 http://www.soberton.com/product/gt-0950rp3
- 100 pF ceramic disk capacitor, marking "101" http://www.vishay.com/docs/45171/kseries.pdf
- 0.001 μF ceramic disk capacitor, marking "102" http://www.vishay.com/docs/45171/kseries.pdf
- 0.01 μF ceramic disk capacitor, marking "103" http://www.avx.com/docs/Catalogs/class3-sc.pdf
- 0.1 μF ceramic disk capacitor, marking "104" http://www.avx.com/docs/Catalogs/class3-sc.pdf
- 1.0 μF electrolytic capacitor http://industrial.panasonic.com/www-data/pdf/ABA0000/ABA0000CE12.pdf

- AD22151G linear-output magnetic field sensor `http://www.analog.com/AD22151`
- Jumper wires, M-F (×)
- DC motor `http://www.mabuchi-motor.co.jp/cgi-bin/catalog/e_catalog.cgi?CAT_ID=ff_180phsh`
- Relay `http://www.cndongya.com/pdf/relayjzc-11f.pdf`
- Seven-segment LED display `[TBD]`
- 3.5 mm stereo audio cable `[TBD]`

C Video Tutorial Links

Component Principles of Operation and Interfacing Techniques

- “Discrete LED Interfacing Theory” at `http://youtu.be/9-RlGPVgFW0`
- “Seven-Segment LED Interfacing Theory” at `http://youtu.be/9D60cn7OFoc`
- “Pushbutton Interfacing Theory” at `http://youtu.be/e7UcL5Ycpho`
- “DIP Switch Interfacing Theory” at `http://youtu.be/KNzEyRwcPIg`
- “Relay Interfacing Theory” at `http://youtu.be/jLFL9_EWlwI`
- “Potentiometer Characteristics” at `http://youtu.be/3gwwF9rF_zU`
- “Thermistor Characteristics” at `http://youtu.be/US406sjBUxY`
- “Photocell Characteristics” at `http://youtu.be/geNeoFUjMjQ`
- “ADMP504 Microphone Interfacing Theory” at `http://youtu.be/99lpj7yUmuY`
- “Buzzer/speaker Characteristics” at `http://youtu.be/8IbTWH9MpV0`
- “Motor Interfacing Theory” at `http://youtu.be/C_22XZaL5TM`
- “Rotary Encoder Interfacing Theory” at `http://youtu.be/CpwGXZX-5Ug`
- “Photointerrupter Characteristics” at `http://youtu.be/u1FVfEvSdkg`
- “Hall-Effect Sensor Interfacing Theory” at `http://youtu.be/T9GP_cnz7rQ`
- “Piezo Sensor Interfacing Theory” at `http://youtu.be/dHaPUJ7n-UI`
- “LCD Character Display Interfacing Theory” at `http://youtu.be/m0Td7KbhvdI`
- “RFID Reader Interfacing Theory” at `http://youtu.be/z1v0vCue83c`
- “Keypad Interfacing Theory” at `http://youtu.be/oj2-CYSnyo0`
- “Digital Potentiometer Interfacing Theory” at `http://youtu.be/C4iBQjWn7OI`

Tutorials

- “Measure Resistance with a Voltage Divider” at `http://youtu.be/9KUVD7RkxNI`
- “Resistive-Sensor Threshold Detector” at `http://youtu.be/TqLXJroefTA`
- “Detect a Switch Signal Transition” at `http://youtu.be/GYBmRJ_qMrE`
- “Serial Communication: UART” at `http://youtu.be/p7CPyYRS8I8`
- “Serial Communication: SPI” at `http://youtu.be/GaXtDamw5As`
- “Serial Communication: I2C” at `http://youtu.be/7CgNF78pYQM`

LabVIEW Techniques for myRIO

- "Digital Output Express VI" at `http://youtu.be/Y8mKdsMAqrU`
- "Digital Input Express VI" at `http://youtu.be/[TBD]:diExVI`
- "Analog Input Express VI" at `http://youtu.be/[TBD]:aiExVI`
- "Run-Time Selectable DIO Channels" at `http://youtu.be/[TBD]:diochans`
- "Digital Output Low-Level subVIs" at `http://youtu.be/[TBD]:doVI`
- "Digital Input Low-Level subVIs" at `http://youtu.be/[TBD]:diVI`
- "Analog Input Low-Level subVI" at `http://youtu.be/[TBD]:aiVI`
- "PWM Express VI" at `http://youtu.be/[TBD]:pwmExVI`
- "UART Express VI" at `http://youtu.be/0FMnkFDsGQs`
- "SPI Express VI" at `http://youtu.be/S7KkTeMfmc8`
- "I2C Express VI" at `http://youtu.be/[TBD]:iicExVI`

LabVIEW Demo Project Walk-Throughs

- "LED Demo Walk-Through" at `http://youtu.be/SHJ-vu4jorU`
- "Seven-Segment LED Demo Walk-Through" at `http://youtu.be/X1v_NjLxVqM`
- "Pushbutton Demo Walk-Through" at `http://youtu.be/Xm1A4Cw2POU`
- "DIP Switch Demo Walk-Through" at `http://youtu.be/ZMyYRSsQCac`
- "Relay Demo Walk-Through" at `http://youtu.be/W2iukd8WVIA`
- "Potentiometer Demo Walk-Through" at `http://youtu.be/RYeKIuU6DX8`
- "Thermistor Demo Walk-Through" at `http://youtu.be/xi0VIpGpf4w`
- "Photocell Demo Walk-Through" at `http://youtu.be/jZQqsc5GmoY`
- "Microphone Demo Walk-Through" at `http://youtu.be/2ZpI_uDwOg4`
- "Buzzer/speaker Demo Walk-Through" at `http://youtu.be/kW4v16GuAFE`
- "Motor Demo Walk-Through" `http://youtu.be/UCqFck0CLpc`
- "Rotary Encoder Demo Walk-Through" at `http://youtu.be/nmGlRqhQ6Rw`
- "Photointerrupter Demo Walk-Through" at `http://youtu.be/yuzNb1ZDbv4`
- "Hall-Effect Sensor Demo Walk-Through" at `http://youtu.be/BCJLg-WbIK4`
- "Piezoelectric-Effect Sensor Demo Walk-Through" at `http://youtu.be/b1me4f-3iOE`
- "LCD (UART) Demo Walk-Through" at `http://youtu.be/JsEMMnIWg4k`
- "LCD (SPI) Demo Walk-Through" at `http://youtu.be/oOXYryu4Y-c`
- "LCD (I2C) Demo Walk-Through" at `http://youtu.be/qbD31AeqOMk`
- "RFID Demo Walk-Through" at `http://youtu.be/Jovn0kPJOKs`
- "Keypad Demo Walk-Through" at `http://youtu.be/7r_LwcDa2AM`
- "Dpot Demo Walk-Through" at `http://youtu.be/dtwXOj5vvy4`