Hands-on Lab

Lego Interfacing

Digital Input and Output (DIO) expanders, like the PCF8574, allow the NXT to interface to a wide range of devices. The DIO lines can be used to actuate binary devices like switches, relays and transistors and 8-bit peripherals like external LCD screens.

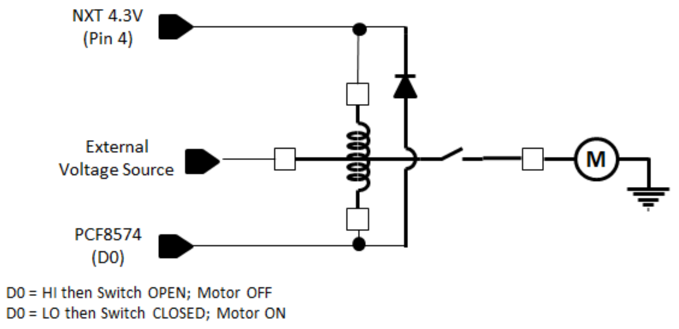
Review – NXT and the PCF8574

The PCF8574 is an I2C chip that when connected to an NXT, gives the Brick 8 digital lines. In previous concepts, these lines could turn on/off LEDs (output) or read a switch's on/off position (input). These concepts are the "Hello World" version of embedded micros. The usefulness of such on/off output or input becomes clear when LEDs are replaced with relays or transistors.

Concept 1 – NXT and Reed Relays

A relay is an electromechanical switch. A reed relay has a small form factor and is capable of handling low voltage, low current devices. Energizing its coil (often labeled COM) will throw the relay's switch from a normally open (often labeled N.O.) state to a closed one.

Step 1: With the PCF8574 circuit constructed from before, wire up the schematic (**Fig. 1A**). In this circuit, only one digital (output) line (D0) is needed. Typically, a SPST (single pole, single throw) relay has 4 pins. Two of them represent the coils' ends. The remaining two refer to the ends of switch.


NXT 4.3V
(Pin 4)
External
Voltage Source
PCF8574
(D0)
M
D0 = HI then Switch OPEN; Motor OFF
D0 = LO then Switch CLOSED; Motor ON

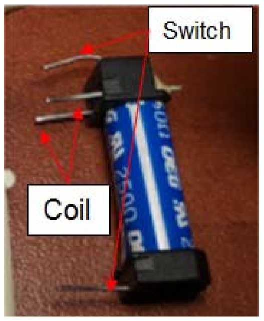


Fig. 1A: Interfacing a reed relay to the NXT using one PCF8574 digital output line.

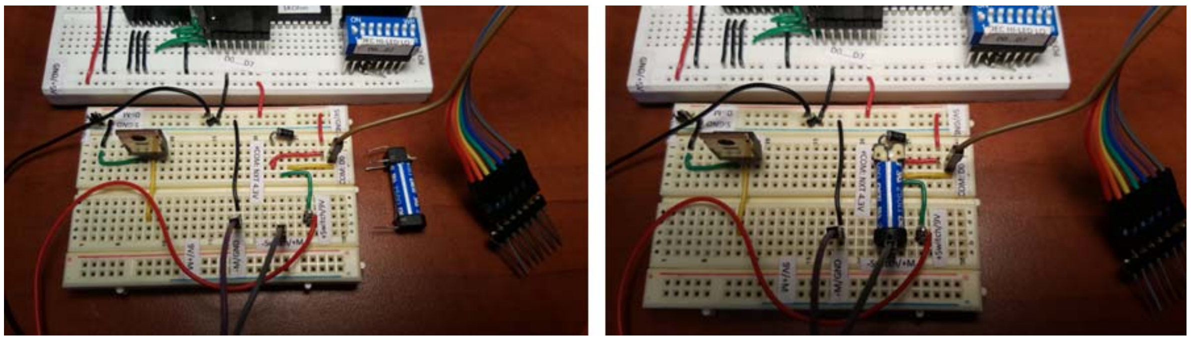


Fig. 1B: Part placement on a solderless breadboard (left). The reed relay inserted into place (right)

The toy DC motor used in **Fig. 1A** can easily draw over 1 A. Since the PCF8574 can only source about 20 mA, an external power supply is needed. A 9V battery's +'ve terminal connected to one end of the relay's switch, and the –'ve one connects to NXT's GND pin. This ties all ground lines together.

Step 2: Compose an NXC program called `dioRelay1_0.nxc`.

```
// FILE: dioRelay1_0.nxc - Works
// AUTH: P.Oh
// DATE: 09/19/16 11:26; 09/23/16 08:19
// VERS: 1.0 - PCF8574A digital output.  Reed relay (+5V coil) connected to D0
//             When D0 is LO, then relay COM should close
// NOTE: Uses PCF8574A chip (hence address A2-A1-A0 set to 0-0-0 hence 0x70
//       9V battery (or power supply) drives toy DC motor.

#define I2Cport S1 // Port number
#define I2CAddr8574 0x70 // I2C address x040 8574 or 0x70 for 8574A

task main() {

 // array variables (since NXC's I2C functions take array variables
 byte WriteBuf[2]; // data written to PCF8574A.  Declares a two one-byte variables
 byte ReadBuf[]; // data received from PCF8574A.  We won't be reading any data but we need this for I2CBytes
 int RdCnt = 1; // number of bytes to read

 // button variables
 bool orangeButtonPushed, rightArrowButtonPushed, leftArrowButtonPushed, greyButtonPushed, overflowFlag;
 // Counting variables
 int decimalNumber; // values from 0 to 255

 SetSensorLowspeed (I2Cport); // PCF8574A connect to NXT on S1
 // Prompt user to begin
 // First, set address with first I2CWrite.  Recall, WriteBuf[1] has address 0xF0 0x00
 WriteBuf[1] = 0x00; // i.e. write zeros to port sets up PCF8574A for writing
 WriteBuf[0] = I2CAddr8574; // i.e. address is 0x70
 I2CBytes(S1, WriteBuf, RdCnt, ReadBuf);

 // Lets start with D0...D7 set to HI
 WriteBuf[1] = 0xFF; // Port lines are HI; so, reed relay should be open and Line_Out should be LO
 WriteBuf[0] = I2CAddr8574; // i.e. address is 0x70
 I2CBytes(S1, WriteBuf, RdCnt, ReadBuf);

 TextOut (0, LCD_LINE1, "Orange Btn starts");
 do {
    orangeButtonPushed = ButtonPressed(BTNCENTER, FALSE);
 } while(!orangeButtonPushed);

 ClearScreen();
 TextOut(0, LCD_LINE1, "Orange BTN quits");
 TextOut(0, LCD_LINE3, "Reed Relay");
 TextOut(0, LCD_LINE4, "<-OFF/ON->");
 decimalNumber = 0;

do {
   greyButtonPushed = ButtonPressed(BTNEXIT, FALSE);
   // If pressed, then grey button becomes TRUE.  If not pressed, then grey button is FALSE
   rightArrowButtonPushed = ButtonPressed(BTNRIGHT, FALSE);
   leftArrowButtonPushed = ButtonPressed(BTNLEFT, FALSE);
```

```
    if(rightArrowButtonPushed) {
     // right button pushed, so turn on relay's coils.
     // set digital line D0 LO
     decimalNumber = 0; // D0 is now LO; NXT's 5V can now flow thru coil, thus switch ON and motor ON
     WriteBuf[1] = decimalNumber;
     WriteBuf[0] = I2CAddr8574;
     I2CBytes(S1, WriteBuf, RdCnt, ReadBuf);
     TextOut (0, LCD_LINE6, FormatNum("%3d - Relay ON " , decimalNumber));
     // now coils activated; switch is closed and resistance between COM and N.O. should be almost zero
    };

    if(leftArrowButtonPushed) {
     // left button pushed, so turn off relay's coils
     // set digital line D0 HI
     decimalNumber = 1; // DO is now HI; NXT's 5V can't flow thru coil, thus switch remains closed and motor OFF
     WriteBuf[1] = decimalNumber;
     WriteBuf[0] = I2CAddr8574;
     I2CBytes(S1, WriteBuf, RdCnt, ReadBuf);
     TextOut (0, LCD_LINE6, FormatNum("%3d - Relay OFF" , decimalNumber));
     // now coils are de-activated; switch is open and resistance between COM and N.O. should be infinite
    };

  Wait(250); // wait 250 millsec
  } while(!greyButtonPushed && !overflowFlag);

  TextOut(0, LCD_LINE5, "Finished!");
  PlaySound(SOUND_DOUBLE_BEEP);

} // end main
```

Code Explanation: Much like `dioOutput2_0.nxc` from a previous lab, `dioRelay1_0.nxc` configures the PCF8574's eight digital lines for output. A do-while loop continuous reads button presses. Pushing the right arrow button (`rightArrowButtonPushed`) closes the switch and turns the motor on and vice-versa with the left arrow button. The program aborts when the grey button is pressed.

Exercise 1: In NxC create programs for the following:

1-1 Replace the DC motor with a buzzer so that closing the relay's switch results in a sound

1-2 Modify your code and wiring to turn the motor on using digital line D1.

Concept 2 – Concept 1 – NXT and Transistors

Like a relay, a transistor is a switch. Unlike a relay, transistors do not have any mechanical parts. As such, they are capable of switching much quicker than relays, and have longer mean-time-between-failures (MTBF). Popular NPN transistors include the TIP31 and IRF510 which both come in TO-220 packages. The TIP31 is a current-driven transistor, whereas the IRF510 is a voltage-driven one. Since the PCF8574's source current is low, turning on or off a current-hungry device like a motor is best done using the IRF510. The IRF510 is called a MOSFET and allows current to flow when its gate pin is above a certain voltage (about 5 Volts). One could employ a TIP31, but then a current-to-voltage converter would be needed.

Step 1: Breadboard the schematic in **Fig. 2A**, noting the IRF510's pin labels (**Fig. 2B**)

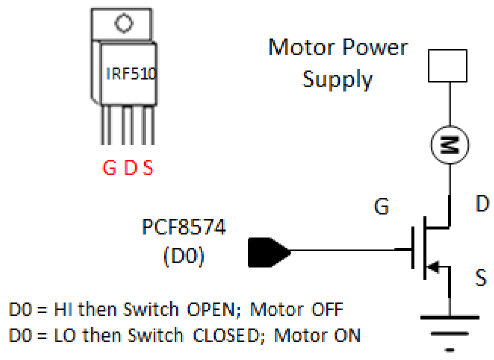

IRF510
G D S
Motor Power Supply
M
G
D
S
PCF8574 (D0)
D0 = HI then Switch OPEN; Motor OFF
D0 = LO then Switch CLOSED; Motor ON

Fig. 2A: Interfacing a IRF510 MOSFET to NXT using one of the PCF8574's digital output lines.

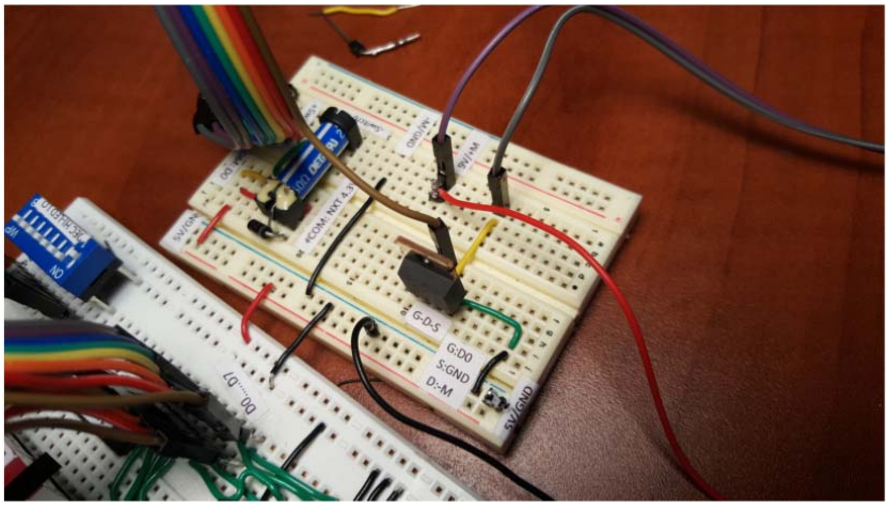


Fig. 2B: The IRF510 MOSFET can simply insert into the solderless breadboard.

Fig. 2A does not use any diodes (and capacitors) in contrast to **Fig. 1A**. While the circuit in **Fig. 2A** will work, the back-EMF from the motor could kick-back enough current to damage the digital line. The analogy is back-wash in hydraulics. A capacitor across the motor leads and a diode will prevent such back-wash to occur.

Step 2: Write and execute an NxC program called `dioMosfet1_0.nxc`

Much like `dioOutput2_0.nxc` and `dioRelay1_0.nxc`, one can set digital line D0 HI (+5V) or LO (GND) by setting decimalNumber to 1 or 0 respectively.

When the right arrow button is pressed, have the motor turn on and vice-versa with the left arrow button. Use the grey button to abort the program.

Exercise 2:

2-1: Refer to `dioInput2_0.nxc` and `dioDipLed2_0.nxc` from a previous lab. Write a program that reads DIP switch positions, and when the value is one, turn on the motor. For any other DIP switch value, the motor turns off. You can use either a reed relay or IRF510 circuit.