PE Kit Project Parts (#130-32000) Rev B Notice

RC Decay with Phototransistor

The PE Kit Project Parts Rev B replaces the cadmium sulfide photoresistor (#350-00009, left) with an RoHS compliant phototransistor (#350-00029, right).

If you have the Propeller Kit Labs: Fundamentals text version 1.1, please use the material below to replace the section "Two Concurrent RC Decay Measurements" on pages 128-129.

Two Concurrent RC Decay Measurements

Since a counter module keeps track of high time after the decay starts, and since each cog has two counter modules, it is possible to take two concurrent RC decay measurements on different pins with a single cog. Figure 7-6 shows an example of a second circuit connected to P25 to test concurrent decay measurements. Instead of a potentiometer for measuring knob position, this circuit has a phototransistor for measuring ambient indoor light levels. The amount of current the phototransistor allows to pass into its collector (C) terminal and then back out of its emitter (E) terminal is controlled by the brightness of light shining on its base (B) terminal. If the light is brighter, the phototransistor allows more current to pass, which results in a faster capacitor decay times. If the light is dimmer, the phototransistor allows less current, resulting in longer decay times.

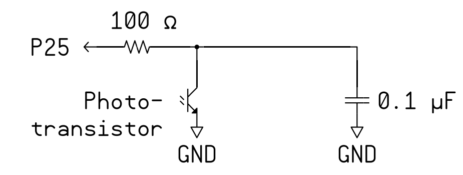
- ✓ Locate the phototransistor in your PE Kit Project Parts. There are three parts with clear plastic cases that resemble the infrared phototransistor on the right side of Figure 7-6. The two identical parts are infrared light emitting diodes. The third part will have a slightly shorter plastic case, and that's the infrared phototransistor.
- ✓ Build the circuit shown in Figure 7-6.

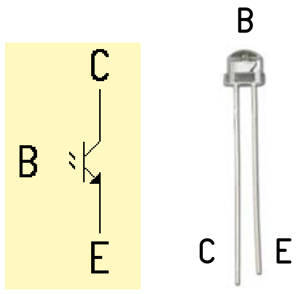
Figure 7-6: Second RC Decay Parts and Circuit

Parts List

(1) Resistor – 100 Ω
(1) Phototransistor
(1) Capacitor – 0.1 µF
(misc) Jumper wires

Schematic





ⓘ **The 100 Ω series resistor** in the Figure 7-6 phototransistor circuit prevents current surges into the capacitor when the I/O pin first switches from input to output-high. Keep in mind that the phototransistor conducts more current when the light is brighter. So under bright lighting conditions, the series resistor also reduces the load the phototransistor might otherwise place on the I/O pin as it charges the capacitor.

With a phototransistor in the circuit, the 100 Ω resistor does not prevent the capacitor from charging to 3.3 V before the decay measurement. If the phototransistor were instead replaced with a resistive light sensor, lower resistances would result in lower initial voltages before decay measurement starts.

TestRcDecay.spin can be modified so that it uses Counter B to measure light levels during a time period that overlaps with the Counter A potentiometer knob position decay measurement. Since a single cog manages both measurements, it initiates them sequentially—one after the other. However, since both counter modules can track the decay times independently, the cog's code does not have to wait for the first measurement to finish before starting the second one. It can start them both, one immediately after the other, move on to other tasks, and check the results in the phase registers later. One approach to modifying TestRcDecay.spin for the two measurements would be to start by converting the `time` variable to a two-element array so that each measurement can be stored in a different element:

```
PUB Main | time[2]
```

Each counter module can then be set to positive detector mode, with one monitoring P17 and the other monitoring P25, like this:

```
' Configure counter modules.

ctra[30..26] := %01000                    ' Set mode to "POS detector"
ctra[5..0] := 17                          ' Set APIN to 17 (P17)
frqa := 1                                 ' Increment phsa by 1 for each clock tick

ctrb[30..26] := %01000                    ' Set mode to "POS detector"
ctrb[5..0] := 25                          ' Set APIN to 25 (P25)
frqb := 1                                 ' Increment phsb by 1 for each clock tick
```

Both I/O pins can be set to output-high to charge both capacitors before starting the decay measurements. The capacitor in the phototransistor circuit is 10 times larger than the one in the potentiometer circuit, and there is also a resistor limiting the current charging the capacitor, so it might take longer to charge. The delay before the measurements start can be increased from 10 µs to 100 µs by changing `clkfeq/100_000` to `clkfreq/10_000`.

```
' Charge RC circuits.

dira[17] := outa[17] := 1              ' Set pin to output-high
dira[25] := outa[25] := 1              ' Set pin to output-high
waitcnt(clkfreq/10_000 + cnt)          ' Wait for circuit to charge
```

Since the counter modules are measuring the decay times, the cog can start each measurement in rapid succession without waiting for the first one to finish before starting the second. The potentiometer measurement is started by clearing `phsa` and `dira[17]`. When `dira[17]` is cleared, the I/O pin becomes an input. As an input, it no longer delivers charge to the capacitor, and the decay starts. Next, the phototransistor measurement is started by clearing `phsb` and `dira[25]`:

```
' Start RC decay measurements...

phsa~                                  ' Clear the phsa register
dira[17]~                              ' Pin to input stops charging circuit
phsb~                                  ' Clear the phsb register
dira[25]~                              ' Pin to input stops charging circuit
```

After enough time has passed, the contents of one phase register can be copied into the `time[0]` variable and the other into `time[1]`:

```
' Measurement has been ready for a while.  Adjust ticks between phsa~
' & dira[17]~.  Repeat for phsb.

time[0] := (phsa - 624) #> 0
time[1] := (phsb - 624) #> 0
```

Last, but certainly not least, display both results:

```
' Display Results

debug.Str(String(13, "time[0] = "))
debug.Dec(time[0])
debug.Str(String(13, "time[1] = "))
debug.Dec(time[1])
waitcnt(clkfreq/2 + cnt)
```

- ✓ Make a copy of TestRcDecay.spin.
- ✓ Use the approach just discussed to modify the copy so that it measures the circuits from Figure 7-1 and Figure 7-6 concurrently.

The code does not have to wait for a fixed period of time before checking the phase registers. It can instead find out if a given measurement is done by checking the state of the I/O pin. After the decay measurement has started, if `ina[17]` stores a 1, it means the decay is still in progress, so don't check `phsa` yet. If it stores 0 instead, the measurement is done. Likewise, if `ina[25]` stores a 1, the light measurement is still in progress, so don't check `phsb` yet. Here is a simple modification that makes the cog wait for both measurements to finish before copying the contents of the phase registers to the `time` variables:

```
' Wait for both measurements to finish.  Then, adjust ticks between phsa~
' & dira[17]~.  Repeat for phsb.

repeat until ina[17] == 0 and ina[25] == 0

time[0] := (phsa - 624) #> 0
time[1] := (phsb - 624) #> 0
```

With all the delays in the code, there isn't an appreciable difference in the display rate. It becomes more evident when you comment the code that causes the delays (except the `waitcnt` that gives the capacitors time to charge). It also helps to declare a variable for counting the repeat loop repetitions and display its value between each measurement. With this arrangement, you'll be able to see that many measurements per second are taken.

- ✓ Try it.